Adjoint Topology Optimisation of Polymer Melt Flow Channels Producible by Additive Manufacturing

Adjungierte Topologieoptimierung von additiv fertigbaren Fließkanälen für Polymerschmelze

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades einer Doktorin der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Jana Sasse

Berichter: Univ.-Prof. Dr.-Ing. Christian Hopmann
Univ.-Prof. Marek Behr, Ph. D.

Tag der mündlichen Prüfung: 04.02.2025
Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Abteilung Extrusion und Kautschuktechnologie am Institut für Kunststoffverarbeitung in Industrie und Handwerk (IKV) an der Rheinisch-Westfälischen Technischen Hochschule Aachen in den Jahren 2020 bis 2024.

Herrn Prof. Dr.-Ing. Christian Hopmann, dem Leiter des IKV, gilt mein herzlicher Dank für die Möglichkeit zur Durchführung dieser Arbeit, die Betreuung und die gewährten Freiräume sowie für das Vertrauen in meine Arbeit. Besonders danken möchte ich außerdem Prof. Marek Behr, Ph. D. für die Übernahme des Koreferats und Prof. Dr. sc. Julia Kowalski für die Übernahme des Prüfungsvorsitzes.

Weiterhin danke ich allen Mitarbeitenden des IKV für die fachlichen Diskussionen, die große Hilfsbereitschaft und die angenehme Arbeitsatmosphäre. Hervorzuheben sind insbesondere die Kolleginnen und Kollegen der Abteilung Extrusion und Kautschuktechnologie, die die gemeinsame Zeit am IKV wertvoll und unvergesslich gemacht haben. Danken möchte ich auch den Angestellten der Werkstatt und Verwaltung des IKV für ihre Unterstützung.

Ein besonderer Dank gilt Malte Schön für die hervorragende Ausgangssituation, in der er mir die Arbeitsgruppe hinterlassen hat, sowie dafür, wie er als Vorgesetzter und Mentor mir stets ermöglicht und mich dazu ermutigt hat, über mich hinauszuwachsen. Ihm sowie Christian Windeck möchte ich außerdem für die kritische Durchsicht meiner Arbeit und die vielen fachlichen Anmerkungen danken. Daniel Wolff und Sebastian Hube danke ich für ihre Unterstützung bei der Herleitung und Dokumentation des Optimierungsproblems.

Großer Dank gebührt auch allen studentischen Hilfskräften und Abschlussarbeitenden, die zu dem Erfolg der zugrunde liegenden Forschungsvorhaben beigetragen haben. Insbesondere möchte ich aber Constantin Arnold und Tom Watolla für ihre Unterstützung bei der Planung und Durchführung der Validierungsversuche für diese Arbeit danken.

Teile der in dieser Arbeit dargestellten Forschungsergebnisse wurden im IGF-Forschungsvorhaben 21298 N der Forschungsvereinigung Kunststoffverarbeitung im Rahmen des Programms zur Förderung der industriellen Gemeinschaftsforschung und -entwicklung (IGF) vom Bundesministerium für Wirtschaft und Klimaschutz aufgrund eines Beschlusses des Deutschen Bundestages gefördert. Andere Teile wurden wiederum von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen der Exzellenzstrategie des Bundes und der Länder – EXC-2023 Internet of Production – 390621612 gefördert. Allen Institutionen gilt mein ausdrücklicher Dank. Weiterhin danke ich der Firma LyondellBasell für die Bereitstellung von Versuchsmaterialien sowie den Firmen Protiq und Elkamet für ihre Unterstützung bei der Analyse und Fertigung der optimierten statischen Mischer.

Aachen, im Februar 2025 Jana Sasse

πάντα ῥεῖ.

- Heraklit

1 Introduction 1

2 Design of Polymer Melt Flow Channels 5

2.1 Extrusion Dies 5

2.1.1 Objectives for the Design of Extrusion Dies 6

2.1.2 Computational Design of Extrusion Dies 7

2.2 Mixing Elements 11

2.2.1 Performance Evaluation of Mixing Elements 14

2.2.2 Computational Design of Mixing Elements 15

2.3 Additive Manufacturing 18

2.3.1 Additively Manufactured Extrusion Dies 21

2.3.2 Additively Manufactured Mixing Elements 22

2.4 Research Questions and Hypotheses 23

3 Numerical Model of Polymer Melt Flows 25

3.1 Computational Fluid Dynamics Using the Immersed Boundary Method 26

3.2 Thermal and Material Transport Models 28

3.3 Material Model 29

3.4 Implementation 30

4 Adjoint Topology Optimisation Algorithm 32

4.1 Literature Review of Adjoint Optimisation in OpenFOAM 34

4.2 Optimisation Problem 36

4.2.1 Stabilised Adjoint Equations for Ducted Flows 38

4.2.2 Optimisation Objectives 41

5 Additive Manufacturing Constraints Algorithm 43

5.1 Manufacturing Restrictions for Self-Supporting Structures 43

5.2 Literature Review of Algorithms for Additive Manufacturing Restrictions 45

5.3 Geometric Constraints Algorithm 46

5.3.1 Overhang Angle Constraint 47

5.3.2 Horizontal Minimum Length Constraint 50

5.3.3 Connectivity Constraint 51

5.3.4 Application-Specific Geometric Constraint 52

5.3.5 Incorporation of Geometric Constraints into Optimisation Algorithm 52

6 Application to Static Mixers 56

6.1 Optimisation of an Additively Manufacturable Static Mixer 57

6.1.1 Simulation Setup 57

6.1.2 Optimisation Results 60

6.2 Simulative Investigation of the Operating Point Dependency of the Algorithm 68

6.2.1 Operating Points 68

6.2.2 Results 70

6.3 Manufacturing and Practical Evaluation 74

6.3.1 Lab Trial Setup 75

6.3.2 Lab Trial Results 76

6.4 Discussion 78

7 Application to Profile Extrusion Dies 84

7.1 Case Setup and Optimisation Objectives 85

7.2 Optimisation Results 87

7.3 Discussion 91

8 Conclusion and Outlook 96

9 Zusammenfassung and Summary 99

9.1 Zusammenfassung 99

9.2 Summary 100

Abbreviations 101

Symbols 102

Indices 106

References 108

A Appendix 122

A.1 Derivation of Adjoint Equations 122

A.2 Source Code Checking Geometric Constraints 128

A.2.1 initDataStructure.H 128

A.2.2 optimiseGeometry.H 136

A.3 Additional Evaluations from Use Case ”Static Mixer” 142

A.4 Lab Trial Data from Use Case ”Static Mixer” 146

1 Introduction

Products manufactured from plastics play a central role in today's society, impacting fields like health, mobility, building and construction, energy, leisure, or commodities [Aal22, NN23a]. The European plastics industry employs 1.5 million people as of 2022, with a 9.2 billion EUR trade balance and a 400 billion EUR turnover [NN23a]. The plastics processing industry operates on very small profit margins. As a result, the process chain must constantly be optimised, both in the planning and production phases. Process chain optimisation can impact not only the costs but also CO_2 emissions of the process [NN23b]. Optimisation can take many forms in the production phase, such as reducing energy usage, reducing material usage per part, or minimising production waste [Rau19]. The challenges lie in the technical complexity of plastics processing.

Extrusion is one of the two main processing methods for thermoplastics. Figure 1.1 depicts a typical setup of a profile extrusion line with a static mixer [HM16, Rau19]. Blue highlights the sections where the polymer is in a solid state, while green highlights the polymer melt conveying sections. The first state transition occurs in the extruder, which transports the plastic granulate in the *solids conveying zone* and melts it in the subsequent *plastification zone*. The polymer melt conveying sections include the *metering zone* of the extruder, which defines the throughput and pumps the melt towards the down-stream components, *mixing elements*, which homogenise the melt, and the *extrusion die*, which forms the melt stream into the extrudate shape. The second state transition occurs in the *calibration section*, where the extrudate is cooled and solidified into its final shape. A *haul-off* and *confectioning unit* prepare the extruded profiles for further processing.

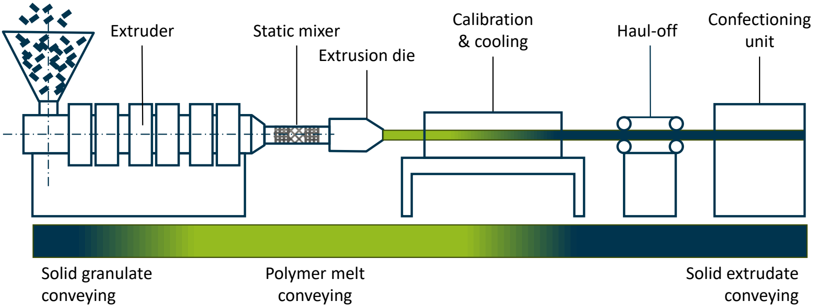


Figure 1.1: Components of a profile extrusion line with a static mixer (not to scale).

This thesis focuses on the melt conveying parts of extrusion lines, such as mixing elements or the extrusion die. Extrusion dies have the task of shaping the melt stream into the extrudate. Their design must balance many objectives and requirements, including the polymer melt's complex rheological and thermal phenomena. As a result, extrusion dies are often produced in very small lot sizes and are often unique due to their individual specification. The role of static mixers is the thermal and material homogenisation of the melt stream provided by the extruder, which is critical to achieving a high-quality product [PR19, Rau19]. This task may gain even more relevance in the future due to the inconsistent properties of recycled material resulting from unquantifiable contamination, which might change on a batch-by-batch basis. Here, the challenge lies in balancing the different performance traits like pressure drop and mixing capability.

In practice, the design of polymer melt conveying flow channels is very complex. While manual design is possible, it is an iterative and, thus, expensive process. User experience can reduce the number of design iterations. However, many of the typically employed design heuristics are not quantified or might be based on outdated information regarding the process. As a result, manual design is increasingly replaced by computational design and topology optimisation methods [GCS22, Pit11]. Topology optimisation aims to determine an optimal shape for a structure, which is usually subject to multiple constraints, such as maximising functionality, minimising material usage, or ensuring manufacturability [ADJ21, FY24].

Figure 1.2 depicts a selection of previously used optimisation strategies for the computational design of melt flow channels. Stochastic methods such as evolutionary, genetic and other types of black-box optimisation algorithms are among the most common topology optimisation methods in polymer melt flow channel design [GCS22]. However, they require many design point evaluations, leading to high computational complexity while limiting the degrees of freedom. Similar drawbacks are observed with direct search strategies. Gradient-based methods, such as adjoint methods, take a different approach. They provide a quick way to determine which changes to the geometry improve its performance, i.e., where material needs to be added or removed from the geometry.

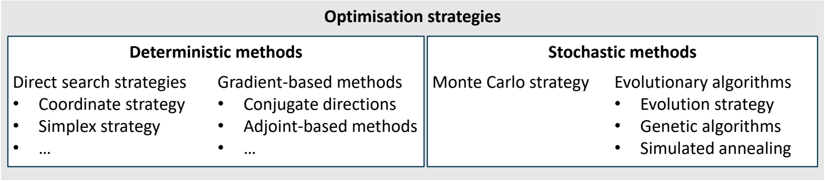


Figure 1.2: Overview over common optimisation strategies (adapted from [HM16]).

Adjoint topology optimisation algorithms start with an initial design, which can be obtained, for example, from existing flow channel designs. A simulation of the initial design determines the design's performance regarding the optimisation objectives. These optimisation objectives are integrated via so-called objective functions, which are mathematical expressions of the optimisation goal(s). At the core, the optimisation method is based on a sensitivity analysis. Sensitivities determine locations in the geometry where changes improve the objective function. Adjoint methods describe a collection of techniques for efficiently computing these sensitivities by expanding the original physics state equations used in the simulations. The adjoint equations are derived from the original state equations and need to be solved in a separate problem. Their solution provides the sensitivity information for the update of the design, which is computed from the gradient of the objective function with respect to the structure's shape. The design update steps are performed in an iterative process that does not require fully recalculating the structure's performance for every small change in the design. The algorithm terminates when a local optimum is found, or other predefined criteria are met. Adjoint optimisation algorithms come with several advantages [GP97, GP00]. They are commonly associated with a relatively low computational complexity since they avoid multiple design point evaluations. At the same time, they scale very well, enabling a high number of degrees of freedom. In addition, their embedded optimisation objectives deliver a high degree of explainability, unlike black-box optimisation algorithms. Existing implementations of the adjoint method in open source software like OpenFOAM facilitate its use in engineering applications [AG20b, GHP23a, Hel15, Oth14b]. The theory behind this optimisation algorithm and its implementation are discussed in more detail in Section 4.

Established manufacturing methods for extrusion dies and mixing elements are subtractive, such as milling, turning, or electrical discharge machining. Subtractive processes start from a steel block and remove material to obtain the final shape. Therefore, their design flexibility is limited by tool accessibility. Some mixing elements are manufactured in a multi-step casting process using a ceramic mould. This method has more design freedom but destroys the mould in the process, making it very expensive [NN14].

Additive Manufacturing (AM) is a manufacturing technology with a different approach. In laser powder bed fusion, also known under its commercial name selective laser melting, the part is constructed from layers of metal powder which are melted locally with a laser, minimising production waste. AM combines material efficiency with design freedom and functional integration in addition to fast path planning, making it particularly suited for individualised production like die manufacturing. The manufacturing method is discussed in more detail in Section 2.3. The AMPOWER Report 2024 lists the AM market at 10.5 billion EUR in 2023, with a projected compound annual growth rate of 13.9% until 2028 [NN24]. Metal powder bed fusion system sales comprise the largest share of the AM market, with around 40% of the sales revenue. Most applications of AM focus on rapid prototyping [Wen22] and individualised production [HYB+15, YHW+16], with the fastest growing sectors being aviation and defense [NN24]. Furthermore, an increasing number of applications exploit AM for its design flexibility [Neu21, PHL+24] and potential for lightweight applications. This includes lattice structures and low-density infills to reduce the part's mass [Lip18, ZSL+24]. However, AM is not free of manufacturing constraints. Therefore, the manufacturing method still must be considered during the design and optimisation process. The individual manufacturing constraints are discussed in Section 5.

This thesis aims to investigate the potential of the adjoint topology optimisation method to optimise the flow channel geometry in static mixing elements and extrusion dies, exploiting the algorithm’s low computational complexity and high degrees of freedom enabled by the optimisation algorithm and design flexibility from additive manufacturing.

The structure of the thesis is depicted in Figure 1.3. Section 2 discusses the state of the art regarding the simulation-based design and optimisation of extrusion dies and mixing elements and finishes by stating the research questions and hypotheses addressed in this thesis. In Sections 3 to 5, the newly developed optimisation algorithm under consideration of manufacturing restrictions is presented. This happens in three steps: The models necessary to numerically describe polymer melt flows are presented in Section 3, while Section 4 and Section 5 focus on the optimisation algorithm, its optimisation objectives and the manufacturing constraints from AM. The algorithm is demonstrated in Section 6 on the use case of an additively manufacturable static mixer. Finally, the algorithm’s potential is evaluated on a use case for optimising a simple profile extrusion die in Section 7. The thesis concludes with an outlook on possible next steps and future research questions in Section 8.

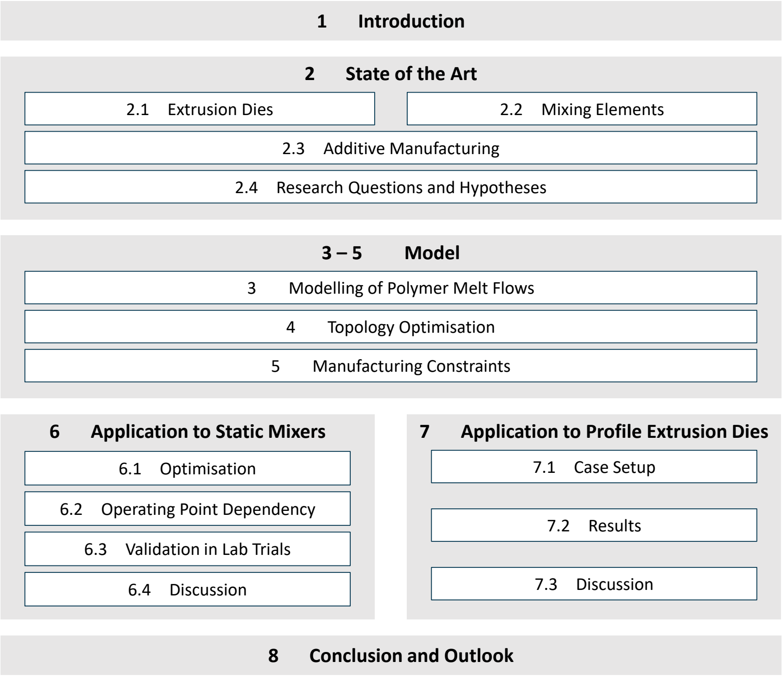

1 Introduction
2 State of the Art
2.1 Extrusion Dies
2.2 Mixing Elements
2.3 Additive Manufacturing
2.4 Research Questions and Hypotheses
3 – 5 Model
3 Modelling of Polymer Melt Flows
4 Topology Optimisation
5 Manufacturing Constraints
6 Application to Static Mixers
6.1 Optimisation
6.2 Operating Point Dependency
6.3 Validation in Lab Trials
6.4 Discussion
7 Application to Profile Extrusion Dies
7.1 Case Setup
7.2 Results
7.3 Discussion
8 Conclusion and Outlook

Figure 1.3: Thesis structure.

2 Design of Polymer Melt Flow Channels

This chapter provides background on the engineering aspects of the design of polymer melt flow channels. First, the design objectives of extrusion dies are discussed, followed by a literature review regarding the computational design of extrusion dies. Next, the design and performance evaluation of mixing elements are introduced. The subsequent section discusses the additive manufacturing of extrusion line components. The resulting research questions and hypotheses are stated at the end of the chapter.

2.1 Extrusion Dies

The extrusion die design process workflow is depicted in Figure 2.1.

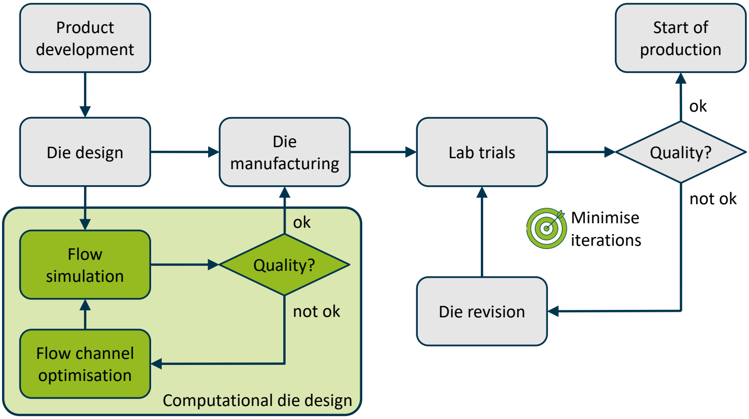


Figure 2.1: Workflow for conventional die design (grey); changes through computational die design and automatic optimisation are highlighted in green (adapted from [Sie18]).

For each new product, an initial die design is inferred from existing extrusion dies with similar properties. In the traditional workflow, this initial die design is manufactured, evaluated in lab trials and revised until it meets the predefined requirements. Traditional die design requires trial and error and experience, which results in a time-consuming and expensive process. This is particularly true for the rheological design of the flow channels. The goal of computational die design is to optimise the die geometry prior to manufacturing, thus reducing the number of die revisions in the lab trial stage until no further iterations are necessary [Sie18].

2.1.1 Objectives for the Design of Extrusion Dies

During the design of extrusion dies, multiple aspects need to be considered. The following guidelines are mainly reported for the design of extrusion dies but are also applicable to other types of melt flow channels, such as mixing elements, which are discussed later in this section.

First, there are general operational aspects [HM16, Pit11]. The die needs to fit into the limited available space of the extrusion line and be easy for the operators to handle during installation and dismantling. In addition, it should be readily detachable and have a leak-proof connection between extruder and die. Moreover, the die must be easy to clean and have as few as possible well-sealed surfaces. Secondly, the mechanical design needs to be considered since an adequate stiffness of the die is necessary to prevent deformations in the die due to the melt forces during processing [HM16, MBS07]. Both the operational and the mechanical aspects of die design will not be discussed in this thesis.

The third aspect concerns manufacturability and minimisation of manufacturing costs. The die, particularly the flow channel, is subject to the constraints of the chosen material and manufacturing method [HM16]. This covers the initial machining of the part as well as post-processing steps such as polishing or tempering the die.

Perhaps the most important aspect is the rheological design of the extrusion die [CN12, HM16]. For economic considerations, the pressure drop in the die should be as low as possible for the specified material throughput provided by the extruder. In addition, the flow channel design should have minimal operating point and material dependence so it does not have to be re-designed for each change in production. Furthermore, the flow channel should minimise damage to the extruded material. This can be achieved through short residence times so the material is not held at critically high temperatures for a long time. On the other hand, there are multiple constraints defining the acceptable range for shear rates and shear stresses in the melt [CN12, GMS11, Pit11]. In cases with insufficient shear stresses, macromolecules remain adsorbed to the die wall, increasing the residence time of the melt near the die wall and resulting in thermal degradation [Pit11]. Increased shear rates also increase the viscous dissipation and can result in disproportionate thermal loads. Eventually, high shear rates can lead to excessive mechanical loads on the polymer molecules, resulting in molecular scission [PR19]. In addition, there should be a narrow residence time spectrum so all parts of the extrudate have a similar history regarding thermal and rheological loads. Low shear rates and areas leading to high residence times can be avoided by ensuring a continual acceleration of the melt, which can be achieved through convergent flow channels. The flow channel should produce a velocity profile that does not prevent production of in-spec profiles on account of distortion. This can be achieved via a uniform velocity distribution at the outlet and uniform die swell [PBE13], which requires parallel zones at the die land where the melt can relax. Weld lines are to be avoided.

Finally, the thermal design of extrusion dies needs to be considered. This encloses two aspects, the first of which is minimising thermal mass. Extrusion dies can get very large and, as a result, have long heating times [HSB+20]. The other aspect is the thermal regime within the polymer melt, which is determined by the die wall temperature as well as shear heating from viscous dissipation in the melt. The temperature of the polymer melt needs to be closely controlled to provide a melt stream with homogeneous viscosity and to minimise the warpage of the extrudate [CN12, HM16].

This thesis will focus on the rheological design of flow channels and their suitability for additive manufacturing. In addition, the thermal design will be considered in the use case for static mixers.

2.1.2 Computational Design of Extrusion Dies

There are different methods to turn the objectives for rheological die design into objectives for an optimisation routine [Hee15, SSPH00]. In many cases, a multi-objective optimisation is preferred, where the optimisation attempts to balance multiple, sometimes contradictory objectives.

The Degrees of Freedom (DOF) of an optimisation routine are in part determined by the parameterisation of the initial flow channel geometry. Most design processes start with an initial Computer Aided Design (CAD) representation of the geometry, which was designed in a digital design software, e.g., using spline-based representations like Non-Uniform Rational B-Splines (NURBS). Some optimisation methods operate directly on this parameterisation, while others first transform the geometry into a different representation.

Gaspar-Cunha et al. list four basic methodologies for the simulation-based design of extrusion dies [GCS22]:

1. using the simulation during manual optimisation,
2. using modelling equations (e.g., from analytical models),
3. using a simulation-based optimisation algorithm,
4. performing data-driven optimisation using machine learning techniques.

The following paragraphs give an overview of recent research in each of these approaches to simulation-based die design.

Manual optimisation supported by simulation

Huang developed a manual optimisation strategy for the design of a spiral mandrel die supported by simulations [Hua98]. In a similar approach, *Rezaei Shahreza et al.* performed simulations of a complex profile extrusion die in Ansys Fluent (Ansys Inc., Canonsburg, PA, USA) to manually determine the optimum parameters for flow balance [RBBS10]. *Ryckebosh and Gupta* performed an optimisation of a complex profile co-extrusion die assisted by the software polyXtrue (Plastic Flow, LLC., Hancock, MI, USA) [RG15]. Besides flow balance, they considered pressure drop in the die and thermal homogeneity in the extrudate. They suggested that the simulation reduced the number of design iterations, thus saving cost and time in the die development process.

Optimisation using modelling equations

For simple geometries such as slit, pipe or annular gap, the relationship between throughput and pressure can be determined using analytical models [Chu19, HM16, Pit11]. The rheological design for more complex geometries is less straightforward. Sometimes, it can be performed by splitting the complex geometry into multiple simple geometry sections. The pressure-throughput relationship can be determined individually as a flow resistance for each simple geometry section. Exploiting the analogies between hydraulic and electric systems and using Kirchhoff's laws, the overall resistance of the complex geometry can be computed through network theory [HM16, Kau04]. However, this method only works up to a certain degree of complexity. *Wolff* combined network theory with the finite element method to enable a faster hybrid method of designing complex flow channels for profile extrusion dies [MKW01, Wol00]. *Kaul* built on this method and investigated different optimisation strategies, namely the coordinate strategy, the simplex-strategy and an evolutionary algorithm [Kau04]. He found that the evolutionary algorithm was the fastest but was also the least robust, while the coordinate strategy was comparatively slow but easy to handle and very robust.

Automatic optimisation using Computational Fluid Dynamics (CFD)

With increasing computing power availability, the rheological design of complex extrusion dies could rely more on CFD.

Saul developed an automated design methodology for spiral mandrel dies [Sau11]. His methodology required a parameterised CAD representation of the geometry, a CFD solver (in his case, Ansys Fluent), an automated evaluation method of different characteristics like pressure drop, weld lines, residence times and wall thickness distribution, and an optimisation algorithm that adjusts the geometry accordingly. A differential-evolution optimisation algorithm was chosen, and during optimisation, two different operating points were evaluated to avoid excessive operating point dependency. Building on this work, *te Heesen* developed a design and optimisation framework for spiral mandrel dies [Hee15]. He included a pre-dimensioning algorithm, formalised the quality criteria for single- and multi-objective optimisation, and performed optimisation using a combination of gradient-based and genetic algorithms in Ansys Fluent. He found that the multi-objective genetic optimisation algorithms were robust and showed the best potential as they also allowed for a weighting of the different objective functions.

Rajkumar et al. performed an optimisation for flow balance in a complex profile extrusion die. They divided the geometry into surrogate models based on the profile's elementary

L- and T-shaped geometries. They investigated two optimisation methods: the average method, based on averaging the lengths of each section and the minimum method, which is based on minimising the differences between the imposed and fitted length values [RFF+17]. Investigations into the operating point dependence showed that only the power law index from the shear-thinning material model significantly influenced the results. Ultimately, both optimisation approaches showed a similar performance [RFF+18].

Yilmaz et al. optimised the flow balance of an L-profile extrusion die [YGK14]. The flow channel geometry was parameterised so that the horizontal channel of the L-profile started with a thick channel followed by a transition area, resulting in a narrow channel. The resulting design variables for optimisation were the transition angle, the narrow channel's length, and the thick channel's thickness. Using latin hypercube sampling, a selection of die geometries across the design space was determined for which CFD simulations were performed in Ansys Polyflow (Ansys Inc., Canonsburg, PA, USA). Optimisation was performed using a simulated annealing-type algorithm in Matlab (The MathWorks, Inc., Natick, MA, USA). They investigated four different formulations of a flow balance objective function and identified two objective functions that were particularly suited to the problem. The limitations of their study were the limited DOF. In addition, no optimisation objectives outside of flow balance were considered.

Building on these results, *Zhang et al.* investigated the potential of additional DOF [ZHSD19]. Based on the optimised L-profile flow channel geometry obtained by *Yilmaz et al.* [YGK14], they performed an optimisation using NURBS-based modelling in Ansys Polyflow. In addition to flow balance, they also investigated objective functions for homogeneous die swell and the final geometry of the extrudate. Using this model, they were able to improve the extrudate geometry. *Zhang et al.* also presented an improved inverse design method for a thin-wall hollow profile extrusion die under consideration of deformation due to inhomogeneous flow balance using Ansys Polyflow [ZHL+20]. Recently, they extended their inverse design method to design co-extrusion dies [HZD23].

While the results obtained by *Zhang et al.* were promising, they were limited to isothermal and viscous flows. However, studies by *Aali et al.* have shown that using inelastic, purely viscous constitutive models can speed up the design process. Still, the viscoelastic model leads to more accurate results, requiring fewer design iterations [Aal22, ACN22a, ACN22b].

The influence of viscoelasticity becomes even more apparent when modelling extrusion dies for elastomers. *Sharma et al.* used an inverse design approach to optimise the flow channel of a keyhole profile die for rubber extrusion using Ansys Polyflow with a purely viscous constitutive model [SSG+20]. *Spanjaards et al.* investigated the effects of viscoelasticity and non-isothermal flows on the same keyhole extrusion die. They found that the extrudate swell increases with elasticity and opposes bending of the extrudate, while shear-thinning models reduce swell and have bending of the extrudate, thus implying that constitutive models have to be chosen carefully, especially for materials with high elasticity [SHA20, Spa22]. They also developed an inverse die design method using a real-time active control scheme instead of the conventional minimisation of an objective function, thus enabling local mesh refinement and the consideration of transient rheological phenomena like viscoelasticity [SHA21, Spa22]. However, the method's effectiveness was limited for high Weissenberg numbers, and they suggested that a spline-based parameterisation of the die shape might improve this method.

Profile extrusion die design and automatic optimisation were also investigated in a collaboration between the Institute for Plastics Processing (IKV) and the Chair for Computational Analysis of Technical Systems (CATS) within the Cluster of Excellence ”Integrative Production Technology for High-Wage Countries”. They parameterised the computational domains with NURBS and used an Elastic Mesh Update Method (EMUM) to compute the free surface flows of the extrudate behind the die exit [Elg11, EPW+12]. The same method could also be used to optimise the flow channel within the die. Optimisation for flow balance was approached by splitting the outlet of the die into multiple optimisation areas and aiming to balance the output in each of them [HWK+14, SEB+13, SYF+15, Win15]. Furthermore, they developed an optimisation framework for uniform die swell, which used EMUM of both the die and the extrudate and new complex objective functions. *Siegbert et al.* investigated different forms of objective functions for uniform die swell and identified a suitable objective function which projects the shadow of the profile onto its technical drawing [PBE13, SBE16, Sie18]. In addition, this collaboration project also investigated the challenges and opportunities for additively manufactured extrusion dies regarding surface quality and DOF, which will be discussed in Section 2.3.

While the flow balance and pressure drop are essential factors in the rheological design of extrusion dies, they are not sufficient. Instead, the thermal design must also be considered so the extrudate exits the die with an appropriate temperature and viscosity distribution, which is a prerequisite to the design of the cooling and calibration stage of the extrusion line [HM16]. *Yesildag et al.* investigated the influence of the temperature on the melt distribution in pre-distributors of spiral mandrel dies in Ansys Polyflow. They found that the local temperature inhomogeneities contributed to inhomogeneous melt distribution at the die outlet, and optimisation of the position of heat sources within the pre-distributor improved the melt distribution [HY15, HY17, HYS17, Yes17]. *Hopmann et al.* built on this work by optimising the melt distribution using heat pipes [HLSW22]. In their study, simulations were performed in the Open Source software OpenFOAM (OpenFOAM Foundation Ltd., London, UK) in a solver similar to the one used in this thesis, which will be discussed in detail in Section 3. *Zwar et al.* investigated a new approach for the thermal design of extrusion dies. They performed shape optimisation in a modular shape optimisation framework consisting of 1. a geometry kernel which translates the design variables into the geometry parameterisation, 2. a general heat equation solver, and 3. an optimisation driver. For the latter, they chose a gradient-free algorithm. Using this shape optimisation framework, they investigated the heating of extrusion die when introducing different microstructures into the die [ZEE22].

Data-driven optimisation

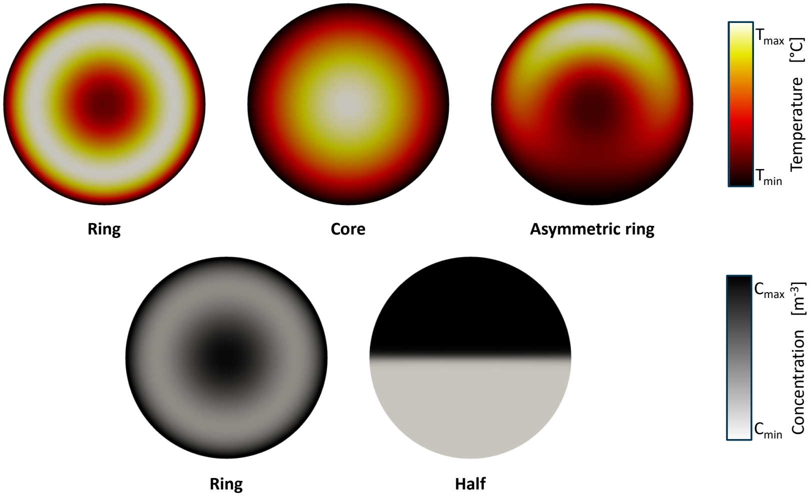
Recently, there has also been an increase in data-driven approaches to design extrusion dies using machine learning techniques. The platform Strömungsraum® (IANUS Simulation GmbH, Dortmund, DE) uses training data based on parametric representations of typical die geometries and existing CFD simulations to predict new die geometries [Wen22]. There is also research to optimise new die geometries from scratch using reinforcement learning [Wol23] and autoencoders [GGC+24]. However, the results in this field are still in the early stages and not ready for industrial applications yet since they are limited by the DOF in the geometry and the resulting computational complexity.

2.2 Mixing Elements

The design objectives presented in Section 2.1.1 imply that a well-designed extrusion die can only operate as intended if the incoming polymer melt stream is as homogeneous as possible. However, the processes upstream of the die can introduce multiple forms of inhomogeneity, which will be discussed below.

Generally speaking, polymer melts have low heat conduction properties. As a result, many *thermal inhomogeneities* introduced into the melt cannot be resolved without external intervention. Thermal inhomogeneities can lead to inhomogeneous thickness distributions in films and bottles [FM06, PR19]. While they can occur for various reasons, there are some commonly observed patterns. Long melt flow channels lead to excessive viscous dissipation, resulting in a so-called ring profile (Figure 2.2 top left), where the melt is coldest in the centre and at the wall, with a hot ring surrounding the centre [Cat12, Cat13]. In contrast, excessive shear heating at the screw can result in a so-called core profile (Figure 2.2 top middle), where the highest melt temperature is found near the centreline [BKC04, Cat12, Cat13]. Asymmetric temperature profiles (Figure 2.2 top right) result from bifurcations or bends of pre-existing ring profile inhomogeneities within the melt flow channel [Yes17] or from inhomogeneous heating of melt flow channel [Hee15, Yes17].

The other typical class are *material inhomogeneities*. High-quality extrudates require a homogeneous distribution of additives like colourants, stabilisers or blowing agents [MHMS11, MS16]. Commonly observed material inhomogeneities include ring structures (Figure 2.2 bottom left) stemming from screw rotation patterns [SKL+04, Sch23a]. Half profiles (Figure 2.2 bottom right) are often used in research since they represent maximal material inhomogeneity at the inlet [MSA12, Sin08].


Ring Core Asymmetric ring

Ring Half

Figure 2.2: Commonly observed thermal and material inhomogeneity patterns.

When it comes to melt homogeneity, two concepts have to be distinguished: Homogenising the overall distribution of a species throughout the domain is called *distributive mixing*, while the breakup of agglomerations of one species is called *dispersive mixing*. Distributive mixing of two phases can be achieved by maximising the interfacial surface area between those two phases. The most efficient method is the so-called baker's fold process, where the materials are stretched, cut and stacked [CS20, SKA+09]. However, this method is only applicable to batch mixing. For continuous mixing within an extrusion process, a different approach is necessary. Due to creeping flow conditions in polymer melt flows, any inhomogeneities can only be reduced through convective mixing, i.e., structures that actively redistribute the melt. This can be achieved through dynamic or static mixing elements. *Dynamic mixing elements* are attached to the screw and are thus moving with the screw rotation [Chu19]. In contrast, *static mixers* are positioned between the screw and the die and do not move. Their mixing capability stems purely from introducing secondary flows into the polymer melt [PM79], thus improving distributive mixing in radial direction [Chu19]. Application fields for static mixers include petrochemistry, processing of polymers and other chemicals, water treatment, food processing, pharmaceutical industry and construction [AM20]. In plastics processing, static mixers are particularly common in foam extrusion with physical blowing agents, where the melt has to be continuously moved to avoid local aggregation of gas [LP14]. In addition, the quality of foamed extrudate is highly dependent on the temperature; therefore, blowing agent-charged melt needs to be cooled [Heu18]. As a result, static mixers are often combined with cooling elements into heat exchanger systems for foam extrusion [Gal09].

Generally speaking, the design goals for extrusion dies (see Section 2.1.1) also apply to mixing elements. In addition, they should maximise mixing capability at minimal pressure drop and mixer volume, with no dead spots [Chu19, PR19].

A wide variety of static mixers is available commercially. The following presents some of the most common types. The Low Pressure Drop (LPD) mixer (Figure 2.3 left) is designed in a ”series of semielliptical plates positioned in a tubular housing” [SKA+09]. As the name suggests, it is known for its relatively low pressure drop, but many elements are necessary to achieve a good mixing result. The Kenics mixer (Figure 2.3 middle) consists of multiple twisted blades and achieves good distributive mixing at a relatively low pressure drop. However, this again comes at the price of a long mixer length. Static mixers in the third class are called X-mixers due to their cross bars within the flow channel. The most studied model here is the Static Mixers using cross bars X (SMX) mixer (Sulzer ChemTech Ltd., Winterthur, CH) (Figure 2.3 right) [Heu09, HKMS09]. Previous studies have shown that the SMX mixer is the most efficient compared to other mixer types regarding the trade-off between mixing performance and pressure drop [RTBB98], especially at limited available space [MSA12, RBBT00].

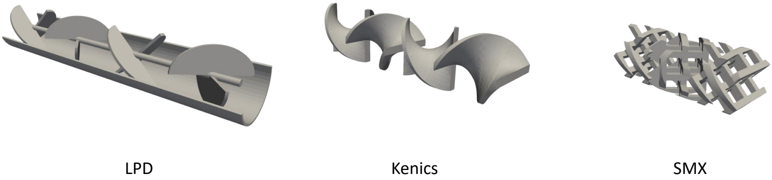

LPD Kenics SMX

Figure 2.3: Commercially available static mixers: LPD (left), Kenics (middle) and SMX (right).

In previous studies investigating the operating point dependency of the SMX mixer, it was demonstrated that up to a Reynolds number of Re = 1, the flow regime in a SMX mixer is independent of the flow rate, thus implying operating point independence from throughput and material for polymer melt flows [ZSMJ02]. However, an earlier investigation found a fully laminar and thus operating point independent flow regime up to a generalised Reynolds number of 15 [LFC96]. *Liu et al.* investigated the operating point dependence from the material by modelling fluid as Newtonian or shear-thinning. They found that a smaller power law index indicating more shear-thinning behaviour leads to a lower pressure drop and better mixing quality [LHW06]. The operating point can decide the mixer’s performance even more profoundly for thermal mixing. Since the mixing element can introduce new inhomogeneities through additional shear heating, positive mixing can only occur with sufficiently high inhomogeneities at the mixer inlet [Imh04].

2.2.1 Performance Evaluation of Mixing Elements

Different computational methods are available for the quantification of distributive mixing performance. Readers are referred to [Dom11, Rat15] for a more detailed discussion on the quantification of dispersive mixing.

According to *Erb*, three classes of computational methods can be identified [Erb18]. The first class are *particle tracking approaches*, where a discrete amount of tracer particles is tracked throughout the computational domain. While this approach is very popular [AMK04, AMK05, BS99, EB13, Erb18, Fra24, Kie10, RBBT00, RTBB98, XYT18], it can also be computationally expensive [Fra24]. The second class contains *mapping methods*, where a cross-sectional grid is deformed according to the calculated velocity field. The method was first proposed by *Spencer et al.* [SW51] and further developed to the extended mapping method by *Kruijt et al.* [KGPM01]. The third class contains *component mass balance and multiphase models*, such as the volume-of-fluid approach, level-set methods or mixing computations using a passive scalar [Hub23, Sch23a, SVJS23]. This class differs from the first two since it can analyse fields other than pure material mixing, such as temperature fields that are also subject to additional shear heating in the process [Sch23a].

Furthermore, these computational methods can be combined with different evaluation methods. A first visual impression is enabled by so-called *Poincaré maps*, which visualise the movement and intersection of two particle clouds [Erb18, Fra24, SVJS23]. A lot of mixing quantification methods utilise some form of statistical model. *Entropy-based methods* aim to maximise the probability that a particle is found within each discrete area [AMK04, AMK05, Dom11, WMK01, WMK03]. *Danckwerts* developed a method based on the concepts of the *scale of segregation* and the *intensity of segregation*, which quantify the distances between particles/concentrations of different species [Dan52]. *Erb* developed an evaluation model that performs the statistical analysis on *Voronoi diagrams* [EB13, Erb18]. The method was extended by *Frank*, who calculated a *Delauney triangulation*, which is the dual graph to the Voronoi diagram and has the advantage that it is intrinsically bounded in a convex hull [Fra24]. Finally, the mixing quantification can be performed using *kinematic* or *geometric* mixing models [BS99, Töb05].

The suitability of the different computational and evaluation methods is determined by the specific application. In addition, different data structures lend themselves to different methods. This thesis evaluates the distributive mixing performance using the mixing number e. This statistical method is performed on a component mass balance model based on the distribution of either the temperature or a passive scalar. The mixing number $e(\phi)$ quantifies the homogenisation for a scalar quantity ϕ as the relative change in the local cumulative inhomogeneity $\delta(\phi)$:

$$e\left(\phi\right) = \frac{\delta\left(\phi\right)_{\mathrm{before}} - \delta\left(\phi\right)_{\mathrm{after}}}{\delta\left(\phi\right)_{\mathrm{before}}}, \quad \text{with } \delta\left(\phi\right) = \sum_{i} \frac{|\mathbf{u}_{i}|}{|\mathbf{u}_{\mathrm{ave}}|} \cdot |\phi_{i} - \phi_{\mathrm{ave}}|. \tag{2.1}$$

The definition in Equation 2.1 includes a local velocity weighting [Sch23a]. This ensures comparability across different operating points and is also representative of the real process where inhomogeneities in areas with low flow velocity towards the edges of the flow channel have little influence on the overall process. $\mathbf{u}_{ave}$ and ϕ_{ave} denote the average velocity and scalar quantity across the evaluated cross-section. A mixing number of $e = 0$ means no change in the homogeneity, while $e = 1$ implies an ideal maximal homogenisation. For thermal mixing (Figure 2.4), the scalar quantity is the temperature ($\phi = T$), resulting in $e = e_{thermal}$ [Imh04, Sch23a]. Similarly, for material mixing, the scalar quantity is computed using the concentration of a passive scalar ($\phi = C$), and the resulting mixing number is called $e = e_{material}$ [Sch23a].

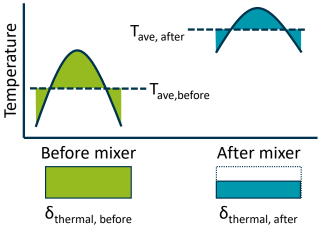


Figure 2.4: Visualisation of the mixing number $e_{thermal}$ for a core profile inhomogeneity.

In lab trials, mixing patterns can be evaluated using digital image processing methods [KEB19] alongside methods involving computer tomography [KMBK24], positron emission particle tracking or magnetic resonance imaging [MLM+15]. *Hopmann et al.* developed a mixer performance evaluation procedure that measures the pressure drop of the mixer alongside thermal and material mixing [HSRF20, SSH22]. The methodology is discussed in more detail in Section 6.3.

2.2.2 Computational Design of Mixing Elements

The most common objectives for optimising mixing elements are pressure drop and mixing capability. However, in many cases, these objectives conflict with each other. An additional challenge is that, in practice, convective mixing often conflicts with shear heating since the mixing element has the potential to introduce new inhomogeneities. Therefore, a multi-objective optimisation approach is often preferable. This section presents previous approaches to this challenge, starting with research on dynamic mixing elements and then the state of the art on optimising static mixers.

Erb et al. investigated the distributive mixing performance in static and dynamic mixing elements. They developed a new particle tracking approach to evaluate the mixing performance [EB13]. The CFD simulations and particle tracking were performed in OpenFOAM. In addition, they implemented an algorithm for reconstructing the interface between the two phases in Matlab as a new method to quantify mixing, where a high interfacial area between two phases in the fluid indicates good mixing performance. The algorithm was demonstrated and validated in lab trials on a Kenics- and a Saxton-type dynamic mixer. They optimised a Saxton mixer for different operating points using this algorithm and a design of experiments approach. The optimisation objective was maximising the interfacial area between the two phases. They found that while they were able to improve the mixer's performance, the results were highly dependent on the operating point [Erb18].

Janßen et al. used the same multi-objective genetic algorithm in Ansys Fluent as *te Heesen* [Hee15] to optimise a parameterised spiral shear mixer. They considered both distributive and dispersive mixing criteria for their optimisation and investigated different weighting strategies. They improved the mixing element's performance and identified multiple quality criteria for future investigations regarding the optimisation of mixing elements [JSVS22]. *Vorjohann et al.* used the same approach to optimise the number of flights, number of blocks, stagger angle and axial flight length of a block-head-mixer [VSJS22].

In a joint IKV and CATS research project, a free-form deformation algorithm was applied on a NURBS representation of a single element on a pineapple mixer. Combining a specially designed top surface, a unique cross-sectional profile and a twist of the outer contour, the new pineapple mixer geometry improved the pressure drop, shear heating and radial mixing capability [HBE+22, HS20, Hub23, NN21]. The optimised mixer was additively manufactured, and the results were validated in lab trials [HSRF20]. Building on this work, *Lee et al.* utilised a neural network to optimise the same base mixing element, resulting in novel geometries [Hub23, LHE24].

In contrast, the simulation-based optimisation of static mixers has a more extended history. This is due to the reduced computational complexity since static mixers can be modelled using steady flows and without moving domain boundaries.

In the early 2000s, *Kruijt et al.* developed the mapping method to quantify distributive mixing [KGPM01]. *van der Hoeven et al.* optimised a multiflux static mixer utilising lab trials alongside 3D numerical simulations [HWM01]. *Singh et al.* investigated the properties of an SMX mixer using a parameterisation in terms of the number of cross bars over the channel width, the number of parallel cross bars per element and the angle between cross bars. They performed CFD simulations, evaluated the mixing performance using the mapping method and found that an ideal configuration for an X-mixer follows universal design rules [Sin08, SAM09]. In another study, they investigated similarly on LPD mixers. They found that the crossing angle between the elliptical plates of the LPD was the most significant design variable for this geometry and that the optimal angle coincided with the value found in commercial Low Low Pressure Drop (LLPD) mixers [SKA+09]. *Meijer et al.* performed a quantitative comparison of different types of static mixers, where they investigated different configurations of SMX, Kenics and LPD-type mixers. For each mixer geometry, they performed CFD simulations and evaluated the mixing performance using the mapping method and the interface stretch. In their findings, they defined an optimum blade twist for Kenics mixer and confirmed the findings by *Singh et al.* that the LLPD mixer generally performs better than LPD mixers [SKA+09]. In addition, they found that the SMX-plus mixer (Sulzer ChemTech Ltd., Winterthur, CH) did not perform better than SMX and proposed a new X-mixer design configuration. They concluded that compact mixers are the ones that generally benefit most from complex structures [MSA12].

Soman investigated the influence of design modification on the mixing performance of a SMX mixer. Using a similar approach to *Singh et al.*, he found that the mixer's performance in polymerisation processes could be improved by perforating the mixing bars in several locations. Still, the same modification was not beneficial for distributive or dispersive mixing of polymer melt flows [Som16].

Tariq et al. investigated the pressure drop, distributive and dispersive mixing performance of different types of static mixers [TMU+22]. They looked at different parameterisations of LPD, Kenics and hybrid mixers. They varied different standard design variables, hoping to combine the radial and axial mixing capabilities of the LPD and Kenics mixer. After performing simulations in Ansys Fluent, they confirmed the optimal slope angle for LPD mixers found by *Singh et al*. Based on their findings, they proposed an optimal static mixer for flows with low Reynolds numbers: a Kenics mixer variation with a curved edge and a high blend radius.

Shahbazi et al. investigated different static mixer geometries. They found that the SMX static mixer had the highest heat transfer rate, while the twisted tape mixer had the lowest pressure drop. Based on these results, they employed a genetic algorithm to optimise a SMX mixer for minimal pressure drop while maximising the heat transfer rate [SAAJ21].

All of the approaches described above have one major drawback: The optimisation is performed either manually or using a genetic algorithm. However, both approaches are time-consuming and computationally expensive, requiring many design point evaluations. In addition, they operate on parameterisations of the mixer geometry that have only few DOF. Adjoint optimisation algorithms solve both of these problems since they only require two solver evaluations and can operate on a much larger scale of DOF (see Section 4).

Alexias et al. performed shape optimisation on a static mixer using an adjoint optimisation algorithm in OpenFOAM [AG20a, AG20b]. Their findings are discussed in more detail in Section 4.1. *Morales Ubal et al.* used a discrete adjoint algorithm to optimise a Kenics mixer for gas dynamics [MBKO24]. These investigations demonstrate the potential and flexibility of adjoint optimisation algorithms for static mixer applications.

2.3 Additive Manufacturing

Laser Powder Bed Fusion (LPBF), also known under the commercial name Selective Laser Melting (SLM), is an additive manufacturing method suitable for various types of metals. It was invented in the late 1990s by *Meiners et al.* [Mei99, NN01], and has since become the most commonly used additive manufacturing method [NN24].

Unlike conventional manufacturing methods, which start with a solid block and remove material in several processing steps by drilling or milling according to a technical drawing, additive manufacturing approaches the desired shape from a different perspective. Every additively manufactured part starts as a virtual representation of the geometry, usually as an STL file. However, not all parts can be directly processed for printing. For some geometries, support structures must be added to ensure the parts comply with the manufacturing restrictions of AM. These manufacturing restrictions are discussed more in Section 5.1. Once the geometry is cleared for manufacturability, it is processed by a slicer, which divides the three-dimensional geometry into thin layers.

Each layer's manufacturing process is divided into several steps (Figure 2.5). The part is printed on a build platform. Step 1 adds a thin layer of metal powder to the platform. In step 2, a laser locally melts the powder along a pre-programmed path, fusing it to the layer below (or the platform if it is the first layer). At the end of the cycle, the platform is lowered by the layer thickness (step 3), and a new powder layer is applied (step 4). Typical layer thicknesses are in the range of 10-100 µm [LL17], other process parameters include the powder particle size, laser power, beam thickness and traverse speed of the laser [FSR+18, GHZ14, HYB+15].

At the end of the printing process, the part is removed from the platform and cleared of the surrounding non-fused powder and support structures. For post-processing, the part is usually blasted and polished, and sometimes, it undergoes heat treatment. In some cases, additional components that cannot be printed directly on the part are manufactured separately and welded to the main part.

As a manufacturing method, AM and, in particular, LPBF comes with several advantages. While the manufacturing time for conventional methods depends on the complexity and the number of manufacturing steps, LPBF offers *complexity for free*, i.e., the manufacturing time depends primarily on the part's volume. Since there is no need to design and construct new workflows or tools for new part designs, it also offers *individualisation for free* [Klo15]. Both of these factors result in a relatively small *time to market*, and the *digital process chain* makes it easy to adapt the process, especially during prototyping [Lip18]. Finally, AM offers new freedoms during the design process of new parts [HGL+18, Lip18, Neu21].

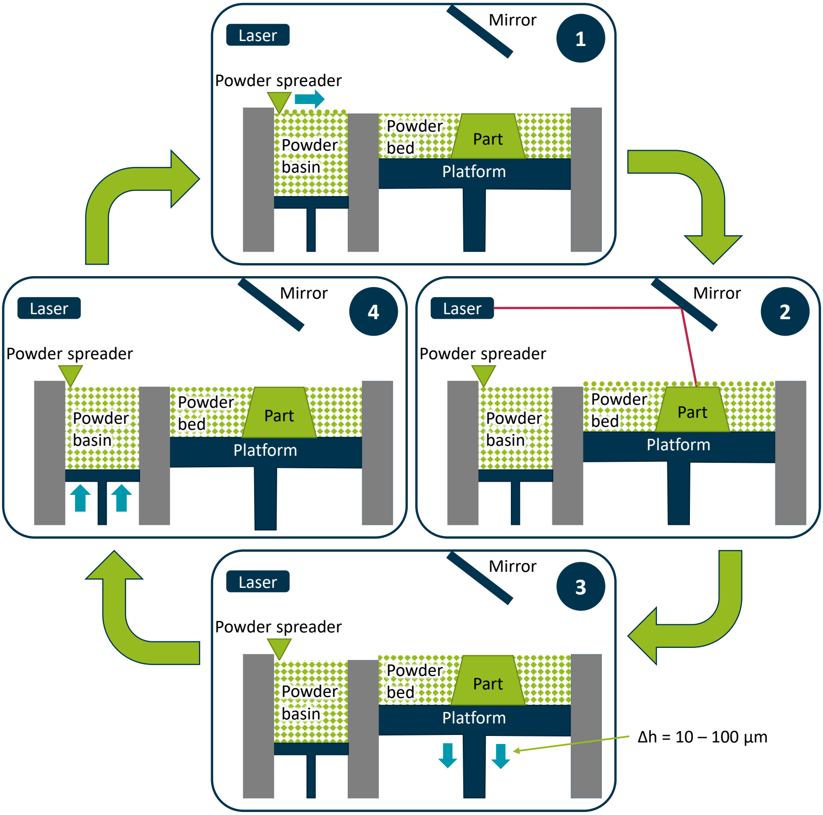


Figure 2.5: Steps in the LPBF manufacturing process.

However, LPBF also comes with some drawbacks. Since the part is manufactured layer by layer, its geometry has to be sliced before being sent to the machine for manufacturing. As a result, artefacts from these individual layers can be visible in the final part. This phenomenon is called the *staircase effect* (Figure 2.6 left) and can be partially controlled by the layer thickness and other process parameters [GHZ14, Lip18]. In addition, the surface quality of an additively manufactured part (Figure 2.6 right) is dependent on the metal powder particle size [SHL11]. Surface quality is an issue that also exists in conventional subtractive manufacturing processes. Therefore, the surface of melt conveying components of extrusion lines is usually polished or honed and sometimes chrome plated [HM16]. However, polishing is not always possible with additively manufactured parts since there is no guaranteed accessibility for polishing materials. Yet, there are methods available to improve the surface quality: Additional polishing of reachable surfaces can reduce the surface roughness [HYB+15], and methods like abrasive flow machining or chrome plating can be used to finish internal structures that are not easily accessible [HSRB20, HM16].

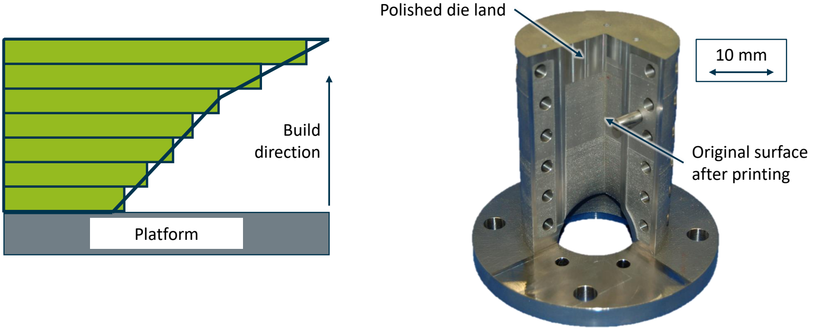


Figure 2.6: Left: visualisation of the staircase effect, right: surface quality of additively manufactured extrusion die part with polished die land.

Another common concern with additively manufactured components is their mechanical properties. To the author’s knowledge, no studies have been conducted on the long-term performance of additively manufactured parts in melt conveying equipment in extrusion lines, so results must be inferred from studies in related areas. Parts manufactured by LPBF have been reported to be capable of almost the same density as conventional parts and a similar or even improved tensile strength and corrosion resistance. However, they often have a reduced ductility and fatigue life [BHK+20]. In additively manufactured parts, the mechanical properties are generally anisotropic and influenced by the microstructures introduced intentionally or as a side effect during manufacturing [FSR+18, Lip18, Neu21]. Yet, additively manufactured parts exhibit similar behaviour under static loads as conventionally manufactured parts [Seh10]. In addition, the anisotropy can be reduced through heat treatment of the part during post-processing [BHHB12, Lip18], but can also affect the dimensions of the part [FSR+18].

While AM is very efficient for rapid prototyping and individualised production, the manufacturing time and price per part generally do not scale well for large-scale production. However, most extrusion dies are manufactured to the specifications for the extrusion line and are thus already part of individualised production [HM16, Win15].

However, many of the drawbacks of AM can be avoided with hybrid components. While most parts are printed on a traditional build platform, it is also possible to print the layers directly on existing parts [URL17]. This enables not only functional integration but also the repair of parts. Functional elements that are not well suited for AM can be added, e.g., by welding them to the additively manufactured part [Bre22]. This is also useful in cases where the desired part exceeds the build envelope of current commercially available LPBF machines.

2.3.1 Additively Manufactured Extrusion Dies

This section reviews the existing research into AM for extrusion dies.

Hopmann et al. investigated the influence of the extrusion die's surface roughness on the extrudate quality [HYB+15]. They used three L-profile die setups: one conventionally milled, one additively manufactured with a milled die land, and one additively manufactured with a polished die land. They found that while the AM profile die with a milled die land was suitable for extrudates with low requirements regarding the surface quality, the best results overall were obtained with the AM profile die with polished die land. This suggests that the surface quality of additively manufactured parts might be less prohibitive than initially thought, especially when appropriate post-processing methods are considered. A separate investigation showed that the increased surface roughness of additively manufactured extrusion dies in some scenarios improves purge times and might, therefore, be beneficial [Goe17, YHW+16]. *Yesildag et al.* demonstrated that AM enables topology optimisation of extrusion dies, in particular for incorporating temperature control channels and reducing the thermal mass of the die [YHW+16].

In conventional manufacturing, extrusion dies for complex extrudate geometries are often assembled using multiple die plates, resulting in sharp transition points. Using AM, flow channels in extrusion dies can be streamlined and manufactured in one continuous die, potentially improving the die's performance. However, a benefit in melt distribution is not guaranteed.

Lieber et al. investigated the potential of AM for a U-profile extrusion die, where they kept most of the existing die design and only fused the individual die plates to one continuous die and streamlined the melt flow channel. They concluded that under these limited DOF, there was no significant advantage of AM over conventional manufacturing [LVTT23].

Neubrech et al. investigated the potential of AM for the thermal and rheological optimisation of extrusion dies for blown film extrusion [Neu21, NWS+20]. Their goal was the optimisation of the primary distribution for a mandrel die. While the melt flow channel in conventionally manufactured 2^n pre-distributors often has a 90° bend before the split into the separate flow channels, *Neubrech* investigated the potential to exploit the design flexibility of AM by adjusting the path line and angles of the distributing flow channels. The resulting partially additively manufactured die improved the residence time and thermal regime within the die but could not provide a homogeneous mass flow distribution [US24]. The rheological and thermal performance of the melt flow channels were evaluated using Ansys Fluent and validated in lab trials. *Eber and Schiffers* performed a CFD-based optimisation of the geometry using a genetic algorithm in Ansys Fluent, building on the methodology developed by *te Heesen* [Hee15]. They considered multiple quality criteria for assessing each iteration of the geometry and identified candidate geometries for further investigations [ES23].

Biedermann et al. developed an automatic routing algorithm for multiple flow channels in co-extrusion dies using an iterative cable simulation in the CAD software Rhinoceros (Robert McNeel & Associates, Seattle, WA, USA) [BBM22, BM20]. In another algorithm, they were able to optimise the generated part for manufacturing under consideration of overhang constraint, where they performed an overhang analysis followed by line generation and automatic generation of both integrated (part of the final design) and sacrificial (meant to be removed after printing) support structures [BBM21]. The algorithm's efficacy was demonstrated when they used it to design the flow channels in a co-extrusion die with a complex profile, where the generated flow channel design was validated in simulations in Ansys CFX (Ansys Inc., Canonsburg, PA, USA) and in lab trials [BMWS20, BWSM21].

2.3.2 Additively Manufactured Mixing Elements

One of the advantages of AM is the newly introduced design flexibility in the design of polymer melt flow channels. For instance, it enables internal structures that cannot be milled or cast in conventional manufacturing as long as they fulfil the requirements for support-free manufacturing.

Groß demonstrated the suitability for the design of melt coolers in foam extrusion [Gro20a, Gro20b, Gro21]. *Hopmann and Hohlweck* as well as *Kietzmann and Astbury* investigated the potential of AM when designing cooling channels in injection moulding tools [HH21, KA24].

Hopmann et al. successfully demonstrated that this method could also be used to introduce temperature control channels into static mixers, thus combining mixing and cooling capabilities [HGL+18, HP19, HPSS19, HS18, HSB+20, HST+19, Sch16]. In a joint research project with the Institute for Laser Technology (ILT), they developed a design

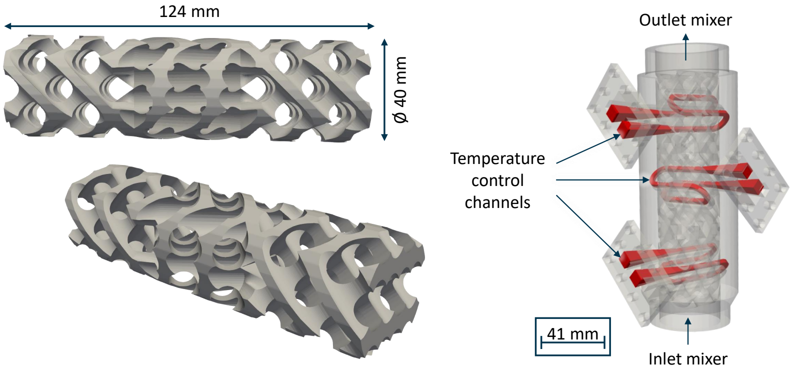


Figure 2.7: Left: base structure of additively manufacturable static mixer without temperature control channels, right: additively manufacturable static mixer with integrated temperature control channels (adapted from [HSB+20]).

framework for support-free additively manufactured static mixers (Figure 2.7 left) as well as the manufacturing constraints for integrated temperature control channels, resulting in a new static mixer design with integrated temperature control channels depicted in Figure 2.7 right [HGL+18, HPSS19, HST+19]. They investigated different cooling strategies and found different configurations that can benefit both cooling and mixing [HP19, HSB+20]. They also investigated the influence of the mixer's length on the mixing performance relative to the pressure drop [HP19, HSB+20]. While their results showed great potential, the manual design and optimisation of these static mixers with integrated temperature control channels were very costly, raising the question of whether automatic optimisation methods could help accelerate the design process.

In recent years, multiple research groups have embraced AM in developing optimised static mixers. *Kwon et al.* demonstrated the suitability of AM for improving heat transfer and material transport within a static mixer using both simulations and lab trials. In addition, they found that AM was beneficial regarding a reduction in mass and volume and simplified handling of the device [KLJK19]. *Hildner et al.* improved the pressure drop in an impeller spiral static mixer by modifying the helix elements exploiting the design flexibility of AM [HLZS23]. *Pereira et al.* used a generative design approach to create entirely novel geometries for mixing elements. Using an evolutionary algorithm, new candidates for mixing element geometries were identified and evaluated using CFD simulations, although no check for manufacturing constraints was performed during optimisation [PHL+24].

Although all of these case studies improved the mixing performance of their respective geometries and often demonstrate the improvement in simulations and lab trials, the manufacturing constraints of AM were manually enforced or assumed as given. However, a genuinely automatic optimisation algorithm requires an enforcement of the manufacturing constraints during runtime.

2.4 Research Questions and Hypotheses

Automatic optimisation has become more relevant for the design of extrusion dies. However, most optimisation algorithms in previous studies operated on fixed parameterisations of the flow channel geometry and had limited DOF. Additively manufactured extrusion dies do not require these limited DOF; thus, this application should be more suitable for optimisation methods that can exploit the whole design space provided by AM. However, these DOF can only be utilised if the manufacturing constraints from AM are respected during optimisation. Therefore, the first over-arching research hypothesis is:

The design flexibility of additive manufacturing enables novel structures for polymer melt flow channels. At the same time, additional constraints in the adjoint topology optimisation algorithm ensure the manufacturability of the optimised geometries.

This thesis investigates two different use cases: the optimisation of static mixers and the optimisation of profile extrusion dies. While the first research hypothesis applies to both use cases, the following research hypotheses are specific to either static mixers or profile extrusion dies.

The first, more narrow use case concerns the adjoint topology optimisation of a static mixer, where the optimisation objectives are limited but also specialised regarding the mixer's performance. The resulting first research hypothesis specific to the optimisation of additively manufacturable static mixers is:

Adjoint topology optimisation algorithms successfully optimise the flow channel geometry of static mixers for different optimisation objectives, where the different optimisation objectives are either to
(1) minimise pressure drop,
(2) maximise thermal mixing in addition to pressure drop reduction or
(3) maximise material mixing in addition to pressure drop reduction.

Adjoint topology optimisation algorithms optimise the geometry for a specific operating point and objective function. This can be beneficial in some cases, e.g., when optimising for a specific inhomogeneity pattern. However, an economical static mixer or extrusion die needs to display a certain degree of operating point independence, where the operating point is characterised by the throughput, the extruded material, and the thermal and material inlet inhomogeneity pattern. Due to the laminar flow regime, the throughput and extruded material do not significantly change the flow pattern. However, materials with a low power law index might show a different sensitivity to changes in the static mixer geometry [LFC96, LHW06, ZSMJ02]. Therefore, the second research hypothesis specific to the optimisation of additively manufacturable static mixers is:

The adjoint topology optimisation algorithm is only operating point dependent for operating points with different thermal or material inhomogeneities and optimisation objectives regarding the thermal or material mixing.

Profile extrusion dies are a complex example of polymer melt flow channels. Therefore, the second use case focuses on a low-complexity L-profile extrusion die, which still has many DOF that can be utilised to optimise the flow channel for different optimisation objectives. The research hypothesis specific to the optimisation of additively manufacturable profile extrusion dies is:

For certain scenarios, adjoint topology optimisation algorithms are a method with low computational complexity to optimise a profile extrusion die's flow channel geometry for specific optimisation objectives, such as minimal pressure drop or maximal flow balance.

3 Numerical Model of Polymer Melt Flows

This chapter discusses the numerical models used to describe polymer melt flows.

Dimensionless numbers can facilitate the analysis of transport phenomena by characterising the relative influence of multiple, sometimes conflicting, physical effects. Table 3.1 lists the most relevant characteristic numbers for the analysis of polymer melt flows within this thesis.

Name	**Definition**	**Interpretation**
Deborah	$\mathrm{De} = \frac{\mathrm{t_r}}{\mathrm{t_p}}$	$\frac{\text{relaxation time}}{\text{process time}}$
Graetz	$\mathrm{Gz} = \frac{\rho \mathrm{c_p} \mathbf{u} \mathrm{H}^2}{\kappa \mathrm{L}}$	$\frac{\text{lengthwise convection}}{\text{transverse conduction}}$
Reynolds	$\mathrm{Re} = \frac{\rho \mathbf{u} \mathrm{L}}{\eta}$	$\frac{\text{inertia forces}}{\text{viscous forces}}$

Table 3.1: Definition of dimensionless numbers with relaxation time t_r, process time t_p, density ρ, specific heat capacity c_p, velocity **u**, characteristic height H, thermal conductivity κ, characteristic length L, and dynamic viscosity η [OR14].

The Deborah number De characterises the degree of viscoelasticity of a fluid [OR14]. It is defined as the ratio between a characteristic relaxation time and an observed process time, e.g., a residence time. A low Deborah number (De = 0) indicates pure viscous flow, while higher Deborah numbers (De > 1) indicate a high degree of viscoelasticity [MHMS11].

The Graetz number Gz indicates the thermal regime in a flow channel [Köp14]. A small Graetz number (Gz < 1) results from dominant transverse conduction, establishing a radial temperature profile. Meanwhile, a large Graetz number (Gz > 100) implies dominant lengthwise convection, where the thermal regime is largely independent of the wall temperature. Instead, the melt temperature increases linearly with the pressure drop [HM16]. 1 < Gz < 100 is called the transition regime. In this thesis, the Graetz number is most relevant for characterising the thermal regime in static mixers.

The Reynolds number Re indicates whether a flow is primarily laminar or turbulent, where turbulence occurs for Re > 4000 [CS20]. In polymer melt flow channels, the Reynolds number is typically much smaller (Re << 1) [Hub23]. As a result, the flows in this thesis can be modelled as strictly laminar.

A complete description of the thermal and rheological state in polymer melt flow channels requires spatially and temporally resolved information regarding the pressure, velocity and temperature across the entire domain. When modelling polymer melt flows in extrusion dies and mixing elements, the following general assumptions can be made:

- The flow is laminar and steady [HM16].
- Polymer melt is an incompressible fluid with constant density, thermal diffusivity and specific heat capacity independent from pressure and temperature [HM16]. The rheological models used are discussed in Section 3.3.
- Gravitational forces can be neglected [HM16].
- The computational domain follows the rules of *ducted flows*, i.e., the computational domain has one inlet, one outlet and one wall with a specific set of boundary conditions, which will be discussed in more detail in Section 4.2.1.

3.1 Computational Fluid Dynamics Using the Immersed Boundary Method

Polymer melt flows can be described by the laws of conservation of mass (Equation 3.1) and momentum (Equation 3.2), forming the incompressible Navier-Stokes equations (NSE) which have been normalised by the constant density ρ:

$$0 = \nabla \cdot \mathbf{u} \tag{3.1}$$

$$0 = \nabla \cdot \mathbf{u}\mathbf{u} + \nabla \mathrm{p} - \nabla \cdot \left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)\right) \tag{3.2}$$

Here, $\mathbf{u}$ and p denote the velocity and pressure, while $\nu = \eta/\rho$ is the kinematic viscosity, calculated from the dynamic viscosity η and the density ρ. The rate of strain tensor is defined as $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}}\right)$.

For the design of flow channels, it is necessary to model both the flow channel as well as the geometry surrounding the flow channel. Generally, this can be approached in two ways:

1. Generation of two separate *boundary conforming* meshes for the fluid and solid domains (Figure 3.1 left).
2. Generation of one *boundary non-conforming* mesh for the whole domain (Figure 3.1 right).

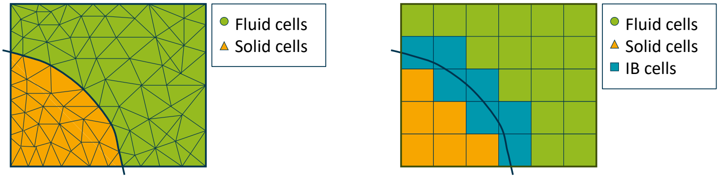


Figure 3.1: Left: boundary conforming mesh, right: boundary non-conforming mesh with an immersed boundary approach.

In boundary conforming methods, such as the Arbitrary Eulerian-Lagrangian (ALE) approach, the computational mesh is fitted to the fluid-solid interface and moves with the motion of the interface. While this ensures a better grid resolution at the interface, it is also computationally expensive since the interface motion has to be computed in a separate mesh motion solver, sometimes requiring re-meshing. Boundary non-conforming methods, such as Immersed Boundary (IB) methods, on the other hand, have the advantage that they simplify the task of grid generation as they do not require a well-resolved fluid-solid interface [GP20, MI05]. Furthermore, boundary non-conforming methods enable a computationally cheap method for modifying the fluid-solid interface, as no re-meshing is required during runtime.

The IB method was first introduced by *Peskin*, who used it to model flow patterns around heart valves [Pes72]. Today, the term IB is used to describe different types of methods. While they use different approaches, they all impose the boundary conditions at the interface by adding a source term (or forcing function) to the governing equations, mimicking the physical boundary [MI05]. Continuous forcing methods operate directly on the continuous model equations and can be used for both elastic and rigid boundaries, where different formulations have been proposed [MI05]. For rigid boundaries, such as the interface between steel die and polymer melt, the domain can be modelled as a porous medium in a so-called fictitious domain approach, where the whole domain can be modelled using the Navier-Stokes equations and an additional forcing term using the local permeability. Conversely, discrete forcing methods impose the boundary conditions directly or indirectly on the discrete model equations. In these cases, the immersed cells are virtually cut, thereby creating a virtual quasi-boundary-conforming mesh, where the laws of conservation of mass and momentum are ensured locally. While it can lead to improved outcomes, this method is much more computationally expensive.

In plastics processing, IB methods are used for multiple reasons. *Hopmann and Schön* as well as *Negrini* have shown its utility for modelling moving geometries in the extrusion process, such as screws and dynamic mixing elements [HS20, Neg23]. *Hopmann et al.* also demonstrate that IB methods can help reduce the computational complexity for the thermal design of extrusion dies [HLSW22]. *Kettemann et al.* demonstrate the utility of a cut-cell immersed boundary approach for static and moving geometries in plastics processing [KB20, KGB21], but this method can also introduce new numerical instabilities [SSH24].

For this thesis, the fictitious domain approach, as presented by *Khadra et al.*, is used, where the domain is modelled as a porous medium [KAPC00]. For the implementation, the momentum equation (Equation 3.2) is modified by an additional force term that models the permeability of porous medium and locally suppresses flow using the inverse Darcy coefficient α (Equation 3.3). This method is straightforward to implement and enables a computationally cheap way of modifying the fluid-structure interface during optimisation.

$$0 = \nabla \cdot \mathbf{uu} + \nabla \mathrm{p} - \nabla \cdot (2\nu \boldsymbol{\epsilon}(\mathbf{u})) + \alpha \mathbf{u} \tag{3.3}$$

For $\alpha_{\text{fluid}} = \alpha_0 = 0$, the flow is not suppressed, and the cell behaves like a fluid, while for $\alpha_{\text{solid}} > 0$, the flow is suppressed, and the cell behaves like a solid. Polymer melt flows exhibit relatively high inertial and viscous forces, and therefore α_{solid} needs to be set to a high value to truly suppress the flow. In this thesis, that value was set to $\alpha_{\text{solid}} = \alpha_{\text{max}} = 2 \cdot 10^9$.

3.2 Thermal and Material Transport Models

The incompressible and steady transport theorem of a scalar quantity ϕ (normalised by the constant density ρ) can be stated as

$$0 = \nabla \cdot (\mathbf{u}\phi) - \nabla \cdot \left(\mathrm{D}_\phi \nabla \phi\right) - \mathrm{Q}^\phi, \tag{3.4}$$

with a diffusivity constant D_ϕ and source term Q^Φ [MMD16].

The thermal regime (ϕ in Equation 3.4 is replaced by the temperature T) is modelled as in Equation 3.5, including an additional source term modelling the viscous dissipation of the polymer melt [Cat12, VRHB99, YCM+22].

$$0 = \nabla \cdot (\mathbf{u}\mathrm{T}) - \nabla \cdot (\mathrm{D_T} \nabla \mathrm{T}) - \frac{1}{\mathrm{c_p}} (2\nu \boldsymbol{\epsilon}(\mathbf{u}) : \nabla \mathbf{u}) \tag{3.5}$$

The thermal diffusivity D_T and specific heat capacity c_p depend on the IB parameter α and set for each cell with the values corresponding to the current material in that cell (Equation 3.6). The thermal diffusivity D_T can be calculated using the thermal conductivity κ, density ρ and specific heat capacity c_p as $D_T = \kappa/(\rho \cdot c_p)$.

$$\left(\mathrm{D_T}, \mathrm{c_p}\right) = \begin{cases} \left(\mathrm{D_{T,solid}}, \mathrm{c_{p,solid}}\right) & \text{, for } \alpha \geq 0.5 \cdot \alpha_{\mathrm{max}} \\ \left(\mathrm{D_{T,fluid}}, \mathrm{c_{p,fluid}}\right) & \text{, for } \alpha < 0.5 \cdot \alpha_{\mathrm{max}} \end{cases} \tag{3.6}$$

The material transport model (ϕ in Equation 3.4 is replaced by the concentration C) is needed to quantify distributive mixing. To avoid computationally expensive particle tracking methods, the material transport is modelled as in Equation 3.7, which interprets the concentration C as a passive scalar, such as coloured master batch [HK14, HSRF20], neglecting additional source terms. The diffusion coefficient D_C is set to a very low value, so the transport of the passive scalar is purely convective.

$$0 = \nabla \cdot (\mathbf{u}\mathrm{C}) - \nabla \cdot (\mathrm{D_C} \nabla \mathrm{C}) \tag{3.7}$$

3.3 Material Model

Polymer melt exhibits several non-Newtonian properties, the most prominent being shear-thinning and viscoelastic behaviour.

Viscoelastic fluids have both a viscous and an elastic force component during deformation. Polymer melt elasticity leads to shear-induced normal stresses, which results in swelling effects at the die exit. This so-called *die swell* is characterised by the Deborah number through the ratio of the melt's characteristic relaxation time and the melt's residence time in the die. If this ratio is too high, the melt 'remembers' past deformations and normal stresses, resulting in a higher degree of die swell. Viscoelastic models can predict this time-dependent and extensional rheological behaviour of polymer melt [Aal22, OR14]. However, they also increase the computational complexity of the material model and can decrease the numerical stability of the overall simulation [ACN22b, Hab14]. As a result, viscoelasticity is often neglected whenever the research focus does not strictly require it, which is also the case in this thesis. Shear-thinning fluids have a high viscosity at low shear rates, while higher shear rates decrease the overall viscosity. In the case of polymer melts, this can result in a viscosity difference spanning multiple orders of magnitude during processing. In addition, the polymer melt's viscosity decreases with increasing temperature.

Thus, the polymer melt is considered a shear-thinning fluid, where the viscosity η is modelled as a function of the shear rate $\dot{\gamma}$ and temperature T. This is achieved by using a Carreau model (Equation 3.8) including a shift factor for temperature from the Williams-Landel-Ferry (WLF) equation (Equation 3.9) [Car68, WLF55]:

$$\eta(\dot{\gamma}, \mathrm{T}) = \frac{\alpha_{\mathrm{T}}(\mathrm{T}) \cdot \mathrm{A}_{\mathrm{Carreau}}}{\left(1 + \alpha_{\mathrm{T}}(\mathrm{T}) \cdot \mathrm{B}_{\mathrm{Carreau}} \cdot \dot{\gamma}\right)^{\mathrm{C}_{\mathrm{Carreau}}}}. \tag{3.8}$$

In the Carreau model, $\mathrm{A}_{\mathrm{Carreau}}$ denotes the zero-shear viscosity, while $\mathrm{B}_{\mathrm{Carreau}}$ is the reciprocal transition rate and $\mathrm{C}_{\mathrm{Carreau}}$ is the slope of the viscosity curve at high shear rates. The WLF shift factor α_{T} can be calculated from Equation 3.9 using the standard temperature T_s, which is around 50 K higher than the glass transition temperature [OR14]. α_{T} shifts the viscosity curve from a reference measure temperature T_m to the desired temperature T.

$$\log\left(\alpha_{\mathrm{T}}(\mathrm{T})\right) = \frac{-8.86 \cdot (\mathrm{T} - \mathrm{T}_s)}{101.6\mathrm{K} + (\mathrm{T} - \mathrm{T}_s)} - \frac{-8.86 \cdot (\mathrm{T}_m - \mathrm{T}_s)}{101.6\mathrm{K} + (\mathrm{T}_m - \mathrm{T}_s)} \tag{3.9}$$

3.4 Implementation

All models are implemented and solved in OpenFOAM v6 (OpenFOAM Foundation Ltd., London, UK), an open source software originally based on C++. OpenFOAM is capable of solving various types of CFD problems as well as problems from structural mechanics, combustion, heat transfer, direct numerical simulation, particle-tracking flows, discrete methods, electromagnetics, and finance [Gre18].

Figure 3.2 depicts the workflow for obtaining a numerical solution for the governing equations presented. The process contains three main components, which are the *domain discretisation*, the *equation discretisation* and the *solution method*.

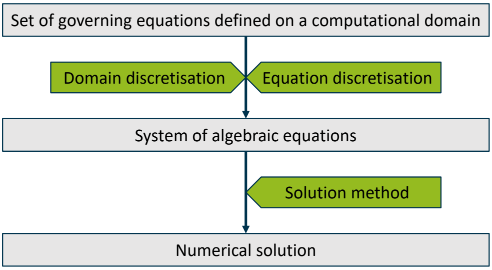

Set of governing equations defined on a computational domain
Domain discretisation
Equation discretisation
System of algebraic equations
Solution method
Numerical solution

Figure 3.2: Discretisation process for obtaining a numerical solution of the governing equations (adapted from [MMD16]).

The *domain discretisation* entails discretisation both in space and time. In OpenFOAM, the spatial discretisation is performed using an unstructured grid consisting of hexahedral cells [Gre18]. Steady flows require no temporal discretisation. Instead, the time steps displayed during simulation correspond to iterations towards a converged steady solution.

The *equation discretisation* of the partial differential equations for CFD, thermal and material transport is performed using the Finite Volume Method (FVM). A short description of FVM is given in the following. For a detailed description, the reader is referred to [MMD16]. The finite volume approach operates on a computational domain that is discretised into finite volume cells. On each of the so-called control volumes Ω_C, the local balance of the conservation equations can be obtained by integration. For example, the scalar transport model in Equation 3.4 can be integrated to

$$0 = \int_{\Omega_C} \nabla \cdot (\mathbf{u}\phi)\, \mathrm{d}\Omega - \int_{\Omega_C} \nabla \cdot \left(\mathrm{D}_\phi \nabla \phi\right) \mathrm{d}\Omega - \int_{\Omega_C} \mathrm{Q}^\phi \mathrm{d}\Omega. \tag{3.10}$$

This local balance can be turned into a local balance of fluxes over the control volume boundaries Γ_C through the application of the divergence theorem:

$$0 = \int_{\Gamma_C} (\mathbf{u}\phi) \cdot \mathbf{n} \mathrm{d}\Gamma - \int_{\Gamma_C} \left(\mathrm{D}_\phi \nabla \phi\right) \cdot \mathbf{n} \mathrm{d}\Gamma - \int_{\Omega_C} \mathrm{Q}^\phi \mathrm{d}\Omega. \tag{3.11}$$

Further discretisation of the flux boundary integrals in Equation 3.11 into sums is performed through numerical discretisation schemes like Gaussian quadrature.

The result is the linear system of equations in Equation 3.12 for a coefficient matrix D, the unknown vector **u** and a known right-hand side vector **f**.

$$\mathrm{D}\mathbf{u} = \mathbf{f} \tag{3.12}$$

The resulting set of discretised algebraic equations over discrete control volumes is solved using the Semi-Implicit Method for Pressure Linked Equations (SIMPLE). This is a *solution method* that was first developed by *Patankar and Spalding* for solving the incompressible NSE [PS72]. Its approach is based on the central idea of decoupling the velocity and pressure fields and solving them separately in an iterative process. Each iteration consists of three steps:

1. Momentum predictor: Solve the momentum equation using the pressure field from the last step.
2. Pressure corrector: Update the pressure field using the predicted velocity field to ensure continuity.
3. Momentum corrector: Adjust the velocity field by applying a pressure-dependent correction.

The thermal and material transport models are solved at the end of each iteration after the velocity and pressure fields are updated.

4 Adjoint Topology Optimisation Algorithm

This chapter discusses the adjoint topology optimisation algorithm and how it is implemented in OpenFOAM. First, a general introduction to the method is given, followed by a literature review that shows the current state of the art. Finally, the adjoint optimisation algorithm and its cost functions for the optimisation objectives in this thesis are presented. Parts of the algorithm have already been demonstrated in the following publications: [HS21, SH24a, SH24b, SSH22].

Flow channel geometries can be optimised by two methods: Topology Optimisation (TO) and Shape Optimisation (SO). In topology optimisation, the geometry's general shape and size are unknown. Shape optimisation, in contrast, operates on a fixed general topology without, e.g., splitting bodies or creating new holes. While SO is suitable for a wide range of problems, its variation of the geometry is limited through the parameterisation of the geometry, thus being prone to result in local minima [ADJ21, OT10]. Consequently, shape optimisation can be viewed as a restricted form of topology optimisation. Given the limitations imposed on shape optimisation through the generally complex shape parameterisation, utilising topology optimisation can be beneficial even for cases where the topology is either expected or forced to remain identical to the initial guess. For this reason, topology optimisation is also applied in this thesis and introduced in detail in the following.

The general optimisation problem for a cost function J is stated in Equation 4.1 [SLY04].

$$\begin{aligned} \text{min} \quad & \mathrm{J}(\mathbf{b}, \boldsymbol{\chi}(\mathbf{b})) \\ \text{such that} \quad & \mathfrak{R}(\boldsymbol{\chi}) = 0 \\ & \mathrm{g}(\mathbf{b}) \geq 0 \\ & \mathrm{h}(\boldsymbol{\chi}) \geq 0 \end{aligned} \tag{4.1}$$

This cost function can take many forms but usually aims to optimise a physical quantity, such as the minimisation of energy dissipation or drag, or a specified distribution of a quantity at a defined location. In this general optimisation problem, the cost function J depends on the design variable(s) **b** as well as the state variables $\boldsymbol{\chi}(\mathbf{b}) = (\mathbf{u}, p, T, C)$. The constraints for this optimisation problem consist of Partial Differential Equation (PDE) constraints $\mathfrak{R}$, e.g., from the NSE as well as geometric constraints g and physical constraints h.

The adjoint PDE-constrained optimisation algorithm was pioneered by *Pironneau*, who used it to optimise drag problems in fluid mechanics [Pir74]. Two different methods can be used to apply adjoints: While the discrete adjoint method works with the algebraic equations that come from the discretisation of the PDE, the continuous adjoint method is first applied directly to the PDE and then discretised.

Central to the adjoint optimisation method is the concept of duality [GP00]. In the following, this concept is demonstrated on a discrete adjoint example using the linear system of algebraic equations obtained in Equation 3.12 in Section 3.4. The goal is to evaluate $\mathbf{f}_a^T \chi^*$, where $\mathbf{f}_a^T$ represents the sensitivity of an objective function with respect to the discretised state variables and χ^* satisfies the linear system of equations coming from the linearised state equations. Equation 3.12 is called the *primal formulation* of the problem and, consequently, χ^* is called the primal solution of the problem. The *dual (adjoint) formulation* of this problem is to evaluate $\boldsymbol{\lambda}^{*T} \mathbf{f}$, where the adjoint solution $\boldsymbol{\lambda}^*$ satisfies the linear system of equations $D^T\boldsymbol{\lambda}^* = \mathbf{f}_a$ using the transposed matrix D^T. The two formulations are equivalent since

$$\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f} = \boldsymbol{\lambda}^{\mathrm{T}}\mathrm{D}\boldsymbol{\chi} = \left(\mathrm{D}^{\mathrm{T}}\boldsymbol{\lambda}\right)^{\mathrm{T}}\boldsymbol{\chi} = \mathbf{f}_{\mathrm{a}}^{\mathrm{T}}\boldsymbol{\chi}. \tag{4.2}$$

While the primal variables χ of the problem hold the physical significance typically assigned to them, the physical interpretation of adjoint variables is less straightforward. One interpretation is that they signal the influence of perturbation from an arbitrary source on the primal variable [GP00]. Alternatively, the adjoint variables can be interpreted as Lagrange multipliers for constrained optimisation problems. This thesis will exploit the latter interpretation, using the adjoint variables to obtain a gradient-based optimisation algorithm.

Using adjoint topology optimisation in this application has several advantages: Due to the design variables coming from the IB field, the optimisation problem inherently has many DOF. At the same time, it has a relatively low computational complexity since D is already assembled, and only two solver calls are needed to obtain the primal and adjoint solution and perform the gradient descent algorithm with respect to the design variables. For high-dimensional systems, the cost of the vector dot products is negligible compared to solving the linear systems of equations [GP00]. This makes the computational cost almost independent from the number of design variables and the adjoint approach much cheaper compared to methods that rely on the evaluation of multiple design points [GP97].

The adjoint approach also has some limitations. As stated in Equation 4.1, many engineering design applications require a set of geometric constraints. While these can be incorporated into the algorithm, the geometric constraints can pose a problem if imposed too strictly. As a result, some of the constraints have to be applied using softer penalty terms [GP00]. The gradient-based optimisation approach can only be applied if the design variables are continuous. These issues must be considered in the presented algorithm and are discussed in Section 5.3.5. Furthermore, the algorithm tends to converge to the nearest local minimum at the risk of missing global optima that are further removed from the initial configuration. However, in applications where the initial configuration is already a fairly good solution, such as optimising existing and manually optimised flow channels for polymer melt flows, a significant restriction from this limitation is not expected.

Adjoint topology optimisation algorithms can operate using two different kinds of sensitivities. With *volume sensitivities*, the changes proposed by the algorithm are directly applied to the volume cells of the computational mesh. In contrast, *surface sensitivities* are computed at the fluid-solid interface and indicate the proposed direction of the mesh motion at the interface. The former is a more direct approach and is utilised in this thesis, while the latter has advantages in applications with restricted DOF.

4.1 Literature Review of Adjoint Optimisation in OpenFOAM

The first continuous adjoint shape optimisation solver in OpenFOAM using an immersed boundary method for the design variable was published by *Othmer et al.* [Oth08, OVW07]. Starting in Version 2.0.0, the solver with the optimisation objective for minimal pressure drop has been included in the official OpenFOAM repositories under the name *adjointShapeOptimizationFoam*, with the modifications necessary for optimisation for homogeneous velocity distribution at the outlet being published in [Oth08]. The equations in *adjointShapeOptimizationFoam* are explained in more detail in Section 4.2. The solver was developed for both volume and surface sensitivities and has since been further developed for the application of aerodynamics in car development [Oth14a, Oth14b, Sih14]. In the case of optimisation with surface sensitivities, the mesh morphing is performed by an external mesh motion tool like ANSA (BETA CAE Systems International AG, Root, CH) [Oth14b]. *Robinson et al.* built on this work by applying the solver to optimise parameterised CAD geometries of an air duct [RAC+12].

Verboom modified the *adjointShapeOptimizationFoam* solver developed by *Othmer et al.* by extending it with a routine checking for manufacturing restrictions for an aerospace engineering application [Ver17].

Mosca modified the *adjointShapeOptimizationFoam* solver developed by *Othmer et al.* by extending it with a solver for a passive scalar to optimise a micro T-mixer using the surface sensitivities given by the adjoint optimisation algorith [Mos17]. He also demonstrated how to modify the solver for external flow shape optimisation problems [Mos20].

Ruberto modified the *adjointShapeOptimizationFoam* solver developed by *Othmer et al.* by extending it with a heat transfer problem and including new objective functions for a target temperature distribution at the outlet [Rub17]. *Faruoli et al.* used a similar approach and demonstrated the algorithm's effectiveness on both two-dimensional and three-dimensional geometries [FAKB17].

Subramaniam et al. modified the *adjointShapeOptimizationFoam* solver developed by *Othmer et al.* by extending it with a heat transfer system and applying it for a multi-objective optimisation, where the target was to balance heat transfer and pressure drop [SDH19].

Towara et al. improved the performance of *adjointShapeOptimizationFoam* by applying algorithmic differentiation on equations to obtain adjoints in a new discrete adjoint solver, including parallelisation in Message Passing Interface (MPI) [Tow18, TSN15]. *He et al.* also developed a discrete adjoint solver based on *adjointShapeOptimizationFoam* and demonstrated its applicability to the aerodynamic optimisation of a car [HMMM18].

The continuous adjoint optimisation algorithm can also be coupled with other types of topology or shape optimisation methods.

Lincke and Rung used a similar approach to the surface sensitivity optimisation in [Oth08], using an adjoint solver in OpenFOAM in combination with mesh morphing in ANSA, to solve heat transfer problems [LR12].

Helgason and Krajnovic extended the adjoint implementation by *Othmer et al.* with a scalar transport equation to optimise for a species concentration at the outlet or a different surface selection. For the interface motion, they used an ALE approach, and the final algorithm was demonstrated for weakly compressible, ducted flows in vehicles [Hel15, HK14].

Gallorini et al. included a heat transfer solver into their implementation based on *adjointShapeOptimizationFoam* and extended it also to include a routine for adaptive mesh refinement. Their adaptive mesh refinement routine was based on a hierarchical non-conforming h-refinement (refinement of cell size in computational mesh) strategy. It was combined with a flux correction to enforce conservation even with topology changes [GHP23a, GHP23b].

In recent years, the continuous adjoint formulation has also been coupled to the mesh motion solvers present in OpenFOAM® (OpenCFD Ltd., Paris, FR) [PG16], resulting in the solver *adjointOptimisationFoam*, which was first published in OpenFOAM-v1906.

Alexias and de Villiers investigated different implicit gradient smoothing algorithms along with a surface mesh regularisation algorithm to improve on the continuous adjoint formulation by *Papoutsis-Kiachagias and Giannakoglou* in OpenFOAM [AV19]. They demonstrated the effectiveness of their algorithm on different applications, power drop minimisation of an air S-bend duct and drag force minimisation of the DrivAer car model.

Alessi et al. used the continuous adjoint formulation by *Papoutsis-Kiachagias and Giannakoglou* to perform a node-based adjoint surface optimisation of a three-dimensional geometry under geometric constraints [AKVB19].

Alexias and Giannakoglou have demonstrated the effectiveness of *adjointOptimisationFoam* to optimise the shape of a two-phase static mixer [AG20a, AG20b]. The flow was modelled as a laminar flow of two miscible fluids, and the mixture density and viscosity were computed using a linear combination of the individual fluids' properties. For the SO, both a node-based parameterisation and a positional angle parameterisation of the geometry were used to obtain an optimised shape and positioning of the baffles within the mixing device.

4.2 Optimisation Problem

This thesis' optimisation algorithm is based on the implementation of *adjointShapeOptimizationFoam* in OpenFOAM [OVW07]. This continuous adjoint optimisation algorithm uses a so-called Lagrange function for a gradient descent algorithm to optimise the IB representation of the geometry as the design variable. While the algorithm's name suggests a SO method, it is a TO algorithm at its core.

Based on the equations necessary to describe polymer melt flows (Section 3), the optimisation problem can be formulated as

$$\min \quad \mathrm{L} = \mathrm{L}(\mathbf{b}, \mathbf{u}, \mathrm{p}, \mathrm{T}, \mathrm{C}) \qquad \text{such that} \quad \mathfrak{R}(\mathbf{b}, \mathbf{u}, \mathrm{p}, \mathrm{T}, \mathrm{C}) = 0, \quad \mathrm{g}(\mathbf{b}) \leq 0, \tag{4.3}$$

where L is the Lagrange function and $\mathfrak{R} = (\mathrm{R}^{\mathrm{p}}, \mathrm{R}^{\mathbf{u}}, \mathrm{R}^{\mathrm{T}}, \mathrm{R}^{\mathrm{C}})^{\mathrm{T}}$ are the constraints of the optimisation problem resulting from the state equations presented in Section 3. The geometric constraints g(**b**) are discussed in more detail in Section 5.

To obtain the Lagrange function or augmented cost function L, the product of the constraints $\mathfrak{R}$ with the vector of Lagrange multipliers $\boldsymbol{\lambda} = (\mathbf{u}_a, \mathrm{p}_a, \mathrm{T}_a, \mathrm{C}_a)$ is integrated over the flow domain Ω and added to the cost function J from the optimisation objective.

$$\mathrm{L} = \mathrm{J} + \int_{\Omega} \boldsymbol{\lambda} \cdot \mathfrak{R} \mathrm{d}\Omega \tag{4.4}$$

For the optimisation problem, the sensitivity of the augmented cost function L with respect to the design variables **b** and the state variables $\chi = (\mathbf{u}, \mathrm{p}, \mathrm{T}, \mathrm{C})$ is calculated:

$$\delta \mathrm{L} = \frac{\delta \mathrm{L}}{\delta \mathbf{b}} + \frac{\delta \mathrm{L}}{\delta \mathbf{u}} + \frac{\delta \mathrm{L}}{\delta \mathrm{p}} + \frac{\delta \mathrm{L}}{\delta \mathrm{T}} + \frac{\delta \mathrm{L}}{\delta \mathrm{C}} \tag{4.5}$$

The Lagrange multipliers are defined such that the variation of L with respect to the state variables vanishes:

$$\frac{\delta \mathrm{L}}{\delta \mathbf{u}} + \frac{\delta \mathrm{L}}{\delta \mathrm{p}} + \frac{\delta \mathrm{L}}{\delta \mathrm{T}} + \frac{\delta \mathrm{L}}{\delta \mathrm{C}} = 0 \tag{4.6}$$

Plugging the condition in Equation 4.6 into the variation of the Lagrange function (Equation 4.5) results in

$$\begin{aligned} &\frac{\delta \mathrm{J}}{\delta \mathbf{u}} + \frac{\delta \mathrm{J}}{\delta \mathrm{p}} + \frac{\delta \mathrm{J}}{\delta \mathrm{T}} + \frac{\delta \mathrm{J}}{\delta \mathrm{C}} \\ &+ \int_{\Omega} \boldsymbol{\lambda} \cdot \frac{\delta}{\delta \mathrm{p}} \mathfrak{R} \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\lambda} \cdot \frac{\delta}{\delta \mathbf{u}} \mathfrak{R} \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\lambda} \cdot \frac{\delta}{\delta \mathrm{T}} \mathfrak{R} \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\lambda} \cdot \frac{\delta}{\delta \mathrm{C}} \mathfrak{R} \mathrm{d}\Omega = 0 \end{aligned} \tag{4.7}$$

The sensitivity of the augmented cost function can, therefore, be computed from

$$\delta \mathrm{L} = \frac{\delta \mathrm{L}}{\delta \mathbf{b}} = \frac{\delta \mathrm{J}}{\delta \mathbf{b}} + \int_\Omega \boldsymbol{\lambda} \cdot \frac{\delta}{\delta \mathbf{b}} \mathfrak{R} \mathrm{d}\Omega \tag{4.8}$$

Given the assumption of fixed boundaries, i.e., the outer boundaries of the domain cannot move during optimisation, this can be simplified using the partial derivative $\partial/\partial\mathbf{b}$ [GHP23b]:

$$\begin{aligned}
\frac{\delta \mathrm{L}}{\delta \mathbf{b}} &= \frac{\partial \mathrm{J}}{\partial \mathbf{b}} + \int_\Omega \boldsymbol{\lambda} \cdot \frac{\partial \mathfrak{R}}{\partial \mathbf{b}} \mathrm{d}\Omega \\
&= \frac{\partial \mathrm{J}}{\partial \mathbf{b}} + \int_\Omega \mathrm{p_a} \frac{\partial \mathrm{R^p}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathbf{u}_\mathrm{a} \frac{\partial \mathrm{R}^\mathbf{u}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathrm{T_a} \frac{\partial \mathrm{R^T}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathrm{C_a} \frac{\partial \mathrm{R^C}}{\partial \mathbf{b}} \mathrm{d}\Omega \\
&= \int_\Omega \mathrm{R}^{\mathrm{p_a}} \frac{\partial \mathrm{p}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathrm{R}^{\mathbf{u}_\mathrm{a}} \frac{\partial \mathbf{u}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathrm{R}^{\mathrm{T_a}} \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \mathrm{d}\Omega + \int_\Omega \mathrm{R}^{\mathrm{C_a}} \frac{\partial \mathrm{C}}{\partial \mathbf{b}} \mathrm{d}\Omega \\
&\quad + \mathrm{BC}^{\mathrm{p_a}} + \mathrm{BC}^{\mathbf{u}_\mathrm{a}} + \mathrm{BC}^{\mathrm{T_a}} + \mathrm{BC}^{\mathrm{C_a}} + \int_\Omega \mathbf{u} \cdot \mathbf{u}_\mathrm{a} \frac{\partial \alpha}{\partial \mathbf{b}} \mathrm{d}\Omega
\end{aligned} \tag{4.9}$$

The constraints $\mathfrak{R}_\mathrm{a} = (\mathrm{R}^{\mathrm{p_a}}, \mathrm{R}^{\mathbf{u}_\mathrm{a}}, \mathrm{R}^{\mathrm{T_a}}, \mathrm{R}^{\mathrm{C_a}})^\mathrm{T}$, from now on called adjoint equations, as well as the adjoint boundary conditions $\mathrm{BC_a} = (\mathrm{BC}^{\mathrm{p_a}}, \mathrm{BC}^{\mathbf{u}_\mathrm{a}}, \mathrm{BC}^{\mathrm{T_a}}, \mathrm{BC}^{\mathrm{C_a}})^\mathrm{T}$ can be computed from the state equations. The full derivation of the adjoint equations and their boundary conditions is detailed in the works of *Gallorini et al.* and *Helgason* [GHP23b, Hel15] and are omitted here, but can be found in Appendix A.1. Their further treatment is discussed in Section 4.2.1 and Section 4.2.2.

In this algorithm, the design variable **b** corresponds to the IB representation of the geometry through the porosity α, which can be optimised using a steepest descent algorithm (Equation 4.10). The index n refers to the iteration step, i.e., the ”Time” in OpenFOAM (Section 3.4).

$$\alpha_{\mathrm{n}+1} = \alpha_\mathrm{n} - \Delta \frac{\partial \mathrm{L}(\alpha_\mathrm{n})}{\partial \alpha_\mathrm{n}} \tag{4.10}$$

During shape optimisation, the porosity α serves as an auxiliary variable to describe the continuous transition from fluid to solid with no explicit dependence of the cost function on the porosity, therefore $\partial \mathrm{J}/\partial \alpha_\mathrm{i} = 0$ [Oth08]. In the primal state equations, the porosity α is only found in the Darcy term $\alpha\mathbf{u}$, which results in the sensitivity for each cell with respect to the flow variable $\sigma_{\mathrm{u,i}} = \mathbf{u}_\mathrm{i} \cdot \mathbf{u}_{\mathrm{a,i}}$ since

$$\frac{\partial \mathfrak{R}}{\partial \alpha_\mathrm{i}} = (\mathbf{u} \quad 0 \quad 0 \quad 0)^\mathrm{T} \boldsymbol{\lambda}_\mathrm{i}. \tag{4.11}$$

When optimising for thermal or material homogeneity, two terms are added for the sensitivities from the temperature and concentration, $\sigma_{\mathrm{T,i}}$ and $\sigma_{\mathrm{C,i}}$, which are determined empirically. Each of the local sensitivities is multiplied by the cell volume. The sensitivity components regarding the temperature and concentration should only be used if their optimisation objectives are active. Therefore, weight variables w_T and w_C are included, which activate/deactivate these terms.

$$\frac{\delta L}{\delta \alpha_i} = \sigma_{u,i} V_i + w_T \sigma_{T,i} V_i + w_C \sigma_{C,i} V_i \tag{4.12}$$

The resulting equation for the steepest descent algorithm update of the IB field, modified from Equation 4.10, is therefore given by

$$\begin{aligned} \alpha_{n+1} = {} & \alpha_n + \\ & \gamma_r \cdot \left(\min\left(\max\left(\alpha_n + \lambda_u \sigma_u + \lambda_T w_T \sigma_T + \lambda_C w_C \sigma_C, \alpha_0\right), \alpha_{max}\right) - \alpha_n\right), \end{aligned} \tag{4.13}$$

where γ_r is a field relaxation factor. λ_u, λ_T and λ_C are the product of the cell volume V_i and the sensitivity response step size, respectively, and therefore control the speed/aggressiveness of the steepest descent algorithm, while α_0 and α_{max} represent the minimum and maximum IB values for the fluid and solid domain, respectively.

4.2.1 Stabilised Adjoint Equations for Ducted Flows

The complete system of adjoint equations is

$$R^{p_a} = \nabla \cdot \mathbf{u}_a + \frac{\partial J_\Omega}{\partial p} = 0 \tag{4.14}$$

$$\begin{aligned} R^{\mathbf{u}_a} = {} & -\nabla \cdot (\mathbf{u}\mathbf{u}_a) - (\nabla \mathbf{u}_a)\,\mathbf{u} + \nabla p_a - \nabla \cdot (2\nu \boldsymbol{\epsilon}(\mathbf{u}_a)) + \alpha(\mathbf{b})\,\mathbf{u}_a - T\nabla T_a \\ & + \frac{1}{c_p} \nabla T_a 2\nu \boldsymbol{\epsilon}(\mathbf{u}) - C\nabla C_a + \frac{\partial J_\Omega}{\partial \mathbf{u}} = 0 \end{aligned} \tag{4.15}$$

$$R^{T_a} = -\nabla \cdot (\mathbf{u} T_a) - \nabla (D_T \nabla T_a) + \frac{\partial J_\Omega}{\partial T} = 0 \tag{4.16}$$

$$R^{C_a} = -\nabla \cdot (\mathbf{u} C_a) - \nabla (D_C \nabla C_a) + \frac{\partial J_\Omega}{\partial C} = 0 \tag{4.17}$$

together with the adjoint boundary conditions

$$\int_\Gamma \left(\mathbf{n} \cdot \mathbf{u}_a + \frac{\partial J_\Gamma}{\partial p}\right) \frac{\partial p}{\partial \mathbf{b}} = 0 \tag{4.18}$$

$$\begin{aligned} \int_\Gamma \Bigg(& \mathbf{n} \cdot (\mathbf{u} \cdot \mathbf{u}_a) + \mathbf{u}_a (\mathbf{u} \cdot \mathbf{n}) - p_a \mathbf{n} - \mathbf{n} \cdot (2\nu \boldsymbol{\epsilon}(\mathbf{u}_a)) + T_a T \mathbf{n} + C_a C \mathbf{n} \\ & - \frac{1}{c_p} 2 T_a \mathbf{n} \cdot (2\nu \boldsymbol{\epsilon}(\mathbf{u})) + \frac{\partial J_\Gamma}{\partial \mathbf{u}} \Bigg) \frac{\partial \mathbf{u}}{\partial \mathbf{b}} + \int_\Gamma \mathbf{u}_a \cdot \left(2\nu \boldsymbol{\epsilon}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right) \cdot \mathbf{n}\right) = 0 \end{aligned} \tag{4.19}$$

$$\int_\Gamma \left(\mathbf{n} \cdot \mathbf{u} T_a + D_T \mathbf{n} \cdot \nabla T_a + \frac{\partial J_\Gamma}{\partial T}\right) \frac{\partial T}{\partial \mathbf{b}} - \int_\Gamma \mathbf{n} \cdot \left(T_a D_T \nabla \frac{\partial T}{\partial \mathbf{b}}\right) = 0 \tag{4.20}$$

$$\int_\Gamma \left(\mathbf{n} \cdot \mathbf{u} C_a + D_C \mathbf{n} \cdot \nabla C_a + \frac{\partial J_\Gamma}{\partial C}\right) \frac{\partial C}{\partial \mathbf{b}} - \int_\Gamma \mathbf{n} \cdot \left(C_a D_C \nabla \frac{\partial C}{\partial \mathbf{b}}\right) = 0 \tag{4.21}$$

This full set of adjoint equations (Equations 4.14 to 4.17) can be simplified by specialising it to ducted flows, where the computational domain has the following characteristics [Oth08]:

- one inlet with prescribed velocity and zero pressure gradient
- one outlet with zero velocity gradient and zero pressure
- wall with no-slip condition and zero pressure gradient

The cost functions for ducted flows used in this thesis are all surface integrals over the domain boundary, meaning they do not contain a contribution from the domain interior ($J_\Omega = 0$). Therefore, the adjoint equations do not depend on the cost function anymore. This means that when switching between different cost functions, the solver itself remains the same (Equations 4.22 to 4.25), and only the boundary conditions need to be adjusted.

$$\begin{aligned}
\mathrm{R}^{\mathrm{p_a}} &= \nabla \cdot \mathbf{u}_\mathrm{a} = 0 && (4.22)\\
\mathrm{R}^{\mathbf{u}_\mathrm{a}} &= -\nabla \cdot (\mathbf{u}\mathbf{u}_\mathrm{a}) - (\nabla \mathbf{u}_\mathrm{a})\,\mathbf{u} + \nabla \mathrm{p_a} - \nabla \cdot (2\nu\boldsymbol{\epsilon}(\mathbf{u}_\mathrm{a})) + \alpha(\mathbf{b})\,\mathbf{u}_\mathrm{a} && (4.23)\\
&\quad - \mathrm{T}\nabla\mathrm{T_a} + \frac{1}{\mathrm{c_p}}\nabla\mathrm{T_a} 2\nu\boldsymbol{\epsilon}(\mathbf{u}) - \mathrm{C}\nabla\mathrm{C_a} = 0 &&\\
\mathrm{R}^{\mathrm{T_a}} &= -\nabla \cdot (\mathbf{u}\mathrm{T_a}) - \nabla(\mathrm{D_T}\nabla\mathrm{T_a}) = 0 && (4.24)\\
\mathrm{R}^{\mathrm{C_a}} &= -\nabla \cdot (\mathbf{u}\mathrm{C_a}) - \nabla(\mathrm{D_C}\nabla\mathrm{C_a}) = 0 && (4.25)
\end{aligned}$$

In Equation 4.19, there are multiple terms involving the rate of strain tensor $\boldsymbol{\epsilon}(\mathbf{u})$, which can be simplified for divergenceless fields (Equation 3.1), where $\mathbf{u}_\mathrm{a}$ and $\partial\mathbf{u}/\partial\mathbf{b}$ vanish on $\partial\Gamma$.

$$\begin{aligned}
&\int_\Gamma \left(-\mathbf{n} \cdot (2\nu\boldsymbol{\epsilon}(\mathbf{u}_\mathrm{a})) - \frac{1}{\mathrm{c_p}} 2\mathrm{T_a}\mathbf{n} \cdot (2\nu\boldsymbol{\epsilon}(\mathbf{u})) \right) \frac{\partial \mathbf{u}}{\partial \mathbf{b}} + \int_\Gamma \mathbf{u}_\mathrm{a} \cdot \left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right) \cdot \mathbf{n} \right) \qquad (4.26)\\
&= \int_\Gamma \nu \left(-(\mathbf{n}\cdot\nabla)\,\mathbf{u}_\mathrm{a} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}} - \frac{1}{\mathrm{c_p}} 2\mathrm{T_a}(\mathbf{n}\cdot\nabla)\,\mathbf{u}\frac{\partial \mathbf{u}}{\partial \mathbf{b}} + (\mathbf{n}\cdot\nabla)\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \mathbf{u}_\mathrm{a} \right)\\
&\quad + \int_\Gamma \nabla\nu \cdot \left((\mathbf{u}_\mathrm{a}\cdot\mathbf{n}) \frac{\partial \mathbf{u}}{\partial \mathbf{b}} - \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \mathbf{n} \right) \mathbf{u}_\mathrm{a} \right)
\end{aligned}$$

For laminar Newtonian flows, the viscosity term ν only contains the molecular viscosity. As a result, the gradient of the viscosity $\nabla\nu$ is zero, which means that the second term on the right hand side vanishes. For the shear-thinning flows present in this thesis, $\nabla\nu$ is assumed to be small enough that this simplification still holds.

$$\int_\Gamma \left(\mathbf{n}\cdot\mathbf{u}_\mathrm{a} + \frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{p}}\right)\frac{\partial \mathrm{p}}{\partial \mathbf{b}} = 0 \tag{4.27}$$

$$\int_\Gamma \Bigg(\mathbf{n}\cdot(\mathbf{u}\cdot\mathbf{u}_\mathrm{a}) + \mathbf{u}_\mathrm{a}(\mathbf{u}\cdot\mathbf{n}) - \mathrm{p}_\mathrm{a}\mathbf{n} - (\mathbf{n}\cdot\nabla)\mathbf{u}_\mathrm{a} + \mathrm{T}_\mathrm{a}\mathrm{T}\mathbf{n} + \mathrm{C}_\mathrm{a}\mathrm{C}\mathbf{n} \\ - \frac{1}{\mathrm{c_p}}2\mathrm{T}_\mathrm{a}(\mathbf{n}\cdot\nabla)\mathbf{u} + \frac{\partial \mathrm{J}_\Gamma}{\partial \mathbf{u}}\Bigg)\frac{\partial \mathbf{u}}{\partial \mathbf{b}} + \int_\Gamma (\mathbf{n}\cdot\nabla)\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\cdot\mathbf{u}_\mathrm{a} = 0 \tag{4.28}$$

$$\int_\Gamma \left(\mathbf{n}\cdot\mathbf{u}\mathrm{T}_\mathrm{a} + \mathrm{D_T}\mathbf{n}\cdot\nabla\mathrm{T}_\mathrm{a} + \frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{T}}\right)\frac{\partial \mathrm{T}}{\partial \mathbf{b}} - \int_\Gamma \mathbf{n}\cdot\left(\mathrm{T}_\mathrm{a}\mathrm{D_T}\nabla\frac{\partial \mathrm{T}}{\partial \mathbf{b}}\right) = 0 \tag{4.29}$$

$$\int_\Gamma \left(\mathbf{n}\cdot\mathbf{u}\mathrm{C}_\mathrm{a} + \mathrm{D_C}\mathbf{n}\cdot\nabla\mathrm{C}_\mathrm{a} + \frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{C}}\right)\frac{\partial \mathrm{C}}{\partial \mathbf{b}} - \int_\Gamma \mathbf{n}\cdot\left(\mathrm{C}_\mathrm{a}\mathrm{D_C}\nabla\frac{\partial \mathrm{C}}{\partial \mathbf{b}}\right) = 0 \tag{4.30}$$

The adjoint boundary conditions in Equations 4.27 to 4.30 are evaluated for the individual domain boundaries. In many of the following evaluations, the velocity is split into its normal and tangential components. For clarity, these are referred to as u_n = $\mathbf{u}\cdot\mathbf{n}$ and $\mathbf{u}_t$ = $\mathbf{u}$ - $\mathbf{n}$ u_n, respectively.

The velocity is at a fixed value at the **inlet and wall**, i.e., it is constant in time. As a result, $\partial\mathbf{u}/\partial\mathbf{b}$ is zero, and the first integral of Equation 4.28 vanishes. The same effect can be observed for temperature and concentration, where the values at the inlet and wall can be non-uniform but are still constant in time and independent of the design variables. In this case, the adjoint boundary conditions are independent of p_a. Since p_a has a similar role in the adjoint NSE to that of p in the primal NSE, the same zero-gradient boundary condition is set for p_a at the inlet.

$$(\mathbf{n}\cdot\nabla)\mathrm{p}_\mathrm{a} = 0 \tag{4.31}$$

$$\mathbf{u}_\mathrm{a,t} = 0, \quad \mathrm{u}_\mathrm{a,n} = -\frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{p}} \tag{4.32}$$

$$\mathrm{T}_\mathrm{a} = 0 \tag{4.33}$$

$$\mathrm{C}_\mathrm{a} = 0 \tag{4.34}$$

At the **outlet**, the value for the pressure is fixed at zero and therefore $\partial p/\partial\mathbf{b}$ = 0, while zero Neumann conditions are imposed for velocity, temperature and concentration.

$$\mathrm{p}_\mathrm{a} = \mathbf{u}\cdot\mathbf{u}_\mathrm{a} + \mathrm{u}_\mathrm{n}\mathrm{u}_\mathrm{a,n} - \nu(\mathbf{n}\cdot\nabla)\mathrm{u}_\mathrm{a,n} + \frac{\partial \mathrm{J}}{\partial \mathrm{u}_\mathrm{n}} + \mathrm{T}_\mathrm{a}\mathrm{T} + \mathrm{C}_\mathrm{a}\mathrm{C} - \frac{2}{\mathrm{c_p}}\mathrm{T}_\mathrm{a}(\mathbf{n}\cdot\nabla)\mathrm{u}_\mathrm{n} \tag{4.35}$$

$$\mathrm{u}_\mathrm{n}\mathbf{u}_\mathrm{a,t} + \nu(\mathbf{n}\cdot\nabla)\mathbf{u}_\mathrm{a,t} = -\frac{\partial \mathrm{J}}{\partial \mathbf{u}_\mathrm{t}} \tag{4.36}$$

$$\mathrm{u}_\mathrm{n}\mathrm{T}_\mathrm{a} + \mathrm{D_T}\mathbf{n}\cdot\nabla\mathrm{T}_\mathrm{a} = -\frac{\partial \mathrm{J}}{\partial \mathrm{T}} \tag{4.37}$$

$$\mathrm{u}_\mathrm{n}\mathrm{C}_\mathrm{a} + \mathrm{D_C}\mathbf{n}\cdot\nabla\mathrm{C}_\mathrm{a} = -\frac{\partial \mathrm{J}}{\partial \mathrm{C}} \tag{4.38}$$

In practice, the algorithm's stability is improved by neglecting all terms containing derivatives of first order or higher, therefore, this approach was chosen in this thesis.

4.2.2 Optimisation Objectives

The cost function formulations for the minimisation of pressure drop or power dissipation as well as the optimisation for flow uniformity at the outlet have been taken from *Othmer* [Oth08], while the cost functions for the optimisation regarding a target temperature or concentration distribution at the outlet have been inspired by the works of *Helgason and Krajnovic* and *Ruberto* [Hel15, HK14, Rub17].

Minimal pressure drop

The cost function used to minimise the pressure drop (Equation 4.39) computes the net inward flux of energy through the domain boundaries [Oth08].

$$\mathrm{J} = \int_{\Gamma} -\left(\mathrm{p} + \frac{1}{2}\mathbf{u}^2\right)\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}\Gamma \tag{4.39}$$

Its derivatives for the boundary conditions (Equation 4.40) depend on the domain boundary type as it is the only cost function depending on the pressure.

$$\frac{\partial \mathrm{J}}{\partial \mathrm{p}} = -\mathrm{u_n} = \begin{cases} 0 & \text{at wall} \\ -\mathrm{u_n} & \text{at inlet} \end{cases} \tag{4.40}$$
$$\frac{\partial \mathrm{J}}{\partial \mathrm{u_n}} = -\frac{1}{2}|\mathrm{u}|^2 - \mathrm{u_n^2}, \quad \frac{\partial \mathrm{J}}{\partial \mathbf{u}_\mathrm{t}} = -\mathrm{u_n}\mathbf{u}_\mathrm{t}, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{T}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{C}} = 0$$

Maximal flow balance

The cost function for a uniform velocity distribution at the outlet is given in Equation 4.41, where $\tilde{c} = 1\ \mathrm{m^{-2}s^{-1}}$ is a constant for unit consistency. The target velocity $\mathbf{u}_{\mathrm{target}}$ can be set to any desired value. For flow balance, $\mathbf{u}_{\mathrm{target}}$ should be set to the average velocity at the flow channel outlet before optimisation. The resulting derivatives for the boundary conditions are listed in Equation 4.42.

$$\mathrm{J} = \int_{\mathrm{outlet}} \frac{\tilde{\mathrm{c}}}{2}\left(\mathbf{u} - \mathbf{u}_{\mathrm{target}}\right)^2 \mathrm{d}\Gamma \tag{4.41}$$

$$\frac{\partial \mathrm{J}}{\partial \mathrm{p}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{u_n}} = \tilde{\mathrm{c}}\left(\mathrm{u_n} - \mathrm{u_{n,target}}\right), \quad \frac{\partial \mathrm{J}}{\partial \mathbf{u}_\mathrm{t}} = \tilde{\mathrm{c}}\left(\mathbf{u}_\mathrm{t} - \mathbf{u}_{\mathrm{t,target}}\right) \tag{4.42}$$
$$\frac{\partial \mathrm{J}}{\partial \mathrm{T}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{C}} = 0$$

Maximal thermal homogeneity

The cost function to optimise for a target temperature at the outlet is given in Equation 4.43, with its derivatives for the boundary conditions stated in Equation 4.44. For static mixers, the goal is not only to homogenise the melt's temperature distribution but also to minimise the additional shear heating and cool the melt. As a result, it is most beneficial to set the target temperature $\mathrm{T}_{\mathrm{target}}$ to the lowest observed temperature at the inlet.

$$\mathrm{J} = \int_{\mathrm{outlet}} \frac{1}{2} \left(\mathrm{T} - \mathrm{T}_{\mathrm{target}}\right)^2 \mathrm{d}\Gamma \tag{4.43}$$

$$\frac{\partial \mathrm{J}}{\partial \mathrm{p}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{u}_{\mathrm{n}}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathbf{u}_{\mathrm{t}}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{T}} = \left(\mathrm{T} - \mathrm{T}_{\mathrm{target}}\right), \quad \frac{\partial \mathrm{J}}{\partial \mathrm{C}} = 0 \tag{4.44}$$

Maximal material homogeneity

The cost function to optimise for a target concentration at the outlet is given in Equation 4.45, with its derivatives for the boundary conditions stated in Equation 4.46. To homogenise the concentration distribution, the target concentration $\mathrm{C}_{\mathrm{target}}$ should be set to the expected value for perfect mixing, i.e., the average concentration at the inlet.

$$\mathrm{J} = \int_{\mathrm{outlet}} \frac{1}{2} \left(\mathrm{C} - \mathrm{C}_{\mathrm{target}}\right)^2 \mathrm{d}\Gamma \tag{4.45}$$

$$\frac{\partial \mathrm{J}}{\partial \mathrm{p}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{u}_{\mathrm{n}}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathbf{u}_{\mathrm{t}}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{T}} = 0, \quad \frac{\partial \mathrm{J}}{\partial \mathrm{C}} = \left(\mathrm{C} - \mathrm{C}_{\mathrm{target}}\right) \tag{4.46}$$

Based on the shape of the cost functions Equation 4.43 and Equation 4.45 and the sensitivity proposed by *Ruberto* [Rub17], the sensitivities for the steepest descent algorithm update of the IB field in Equation 4.13 are

$$\sigma_{\mathrm{u}} = \mathbf{u} \cdot \mathbf{u}_{\mathrm{a}}, \quad \sigma_{\mathrm{T}} = \left(\mathrm{T} - \mathrm{T}_{\mathrm{target}}\right) \cdot \mathrm{T}_{\mathrm{a}}, \quad \sigma_{\mathrm{C}} = \left(\mathrm{C} - \mathrm{C}_{\mathrm{target}}\right) \cdot \mathrm{C}_{\mathrm{a}} \tag{4.47}$$

5 Additive Manufacturing Constraints Algorithm

In this chapter, the design restrictions of AM are discussed, and an algorithm incorporating these design flexibilities and design restrictions as geometric constraints into the optimisation algorithm in Section 4 is presented. Parts of this algorithm were already published in [SH24a, SH24b].

5.1 Manufacturing Restrictions for Self-Supporting Structures

While AM enables new types of geometries, it also has manufacturing constraints, which need to be addressed in the optimisation algorithm.

The most commonly cited manufacturing constraint for LPBF is the overhang angle constraint [HGL+18, Lip18, Neu21, ZC20]. During the manufacturing process, the upper powder bed layer is melted and fused locally. The melted layer requires support so it can solidify in place. This support has two roles: The layer requires mechanical support, i.e., the powder bed itself is not strong enough to hold the melted layer in place. Instead, most of the melted layer should sit on top of an already solidified layer of the part. This is also critical due to the second requirement of thermal support, i.e., the heat transferred into the layer must be transferred out again. The angle between the structure within a layer and the structure in the layer below it is called the *overhang angle* β (Figure 5.1). Depending on the specific manufacturing method and geometric scenario, the exact critical overhang angle $\bar{\beta}$ can vary but is usually around $\bar{\beta} = 45°$ [HGL+18, Lip18, Neu21, ZC20], which is the value used in this thesis. The implementation of the overhang angle constraint for both cartesian and non-cartesian meshes is discussed in Section 5.3.1.

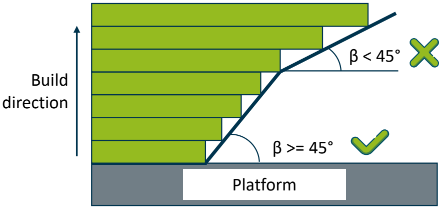


Figure 5.1: Definition of the overhang angle β with critical overhang angle $\bar{\beta} = 45°$.

While the overhang angle constraint already covers a lot of the structures occurring during optimisation, it lacks robustness concerning hanging features (Figure 5.2), where the overhang angle constraint is satisfied, but there is no material existing underneath the structural part [ZC20]. The horizontal minimum length constraint prevents hanging features by ensuring that each solid cell i that is part of the structure has a certain number of solid cells surrounding it within the same layer. The algorithm computing the horizontal minimum length constraint for cartesian and non-cartesian meshes is discussed in Section 5.3.2.

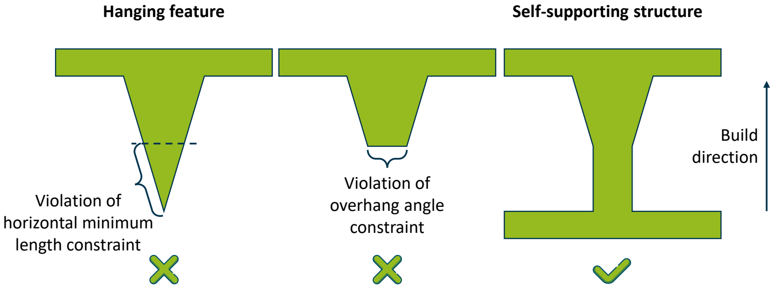

Hanging feature
Self-supporting structure
Violation of horizontal minimum length constraint
Violation of overhang angle constraint
Build direction

Figure 5.2: Definition of a hanging feature. The structure on the left fulfils the overhang angle constraint but violates the horizontal minimum length constraint. The structure in the middle fulfils the horizontal minimum length constraint but violates the overhang angle constraint. The structure on the right is self-supporting and fulfils both manufacturing constraints.

During the additive manufacturing process, two types of domains or structures occur: The part that is manufactured and the remaining domain that is filled with unmelted metal powder. The first is the solid structure or solid phase, while the latter is the void phase. After manufacturing, the unmelted powder within the void phase must be removable. This is only possible if every part of the void phase has a connection to the outside, i.e., connectivity is ensured (Figure 5.3). In addition, the void phase should not have dead angles that might hinder the powder flow. The implementation of the connectivity constraint is discussed in Section 5.3.3.

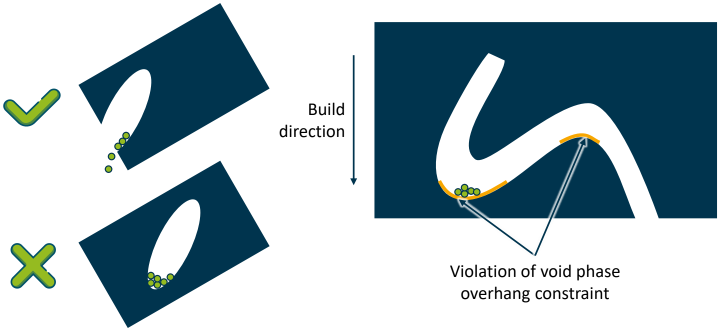


Figure 5.3: Definition of the connectivity constraint of the void phase (adapted from [ZC22]).

While the degrees of freedom and the design flexibility provided by the optimisation algorithm and the additive manufacturing method are extensive, they can also be counter-productive in some cases. For instance, during an optimisation for minimal pressure drop, it should be expected that the configuration with minimal pressure drop will be the one where any structure adding to the overall flow resistance is absent. In addition, not every geometry manufacturable with AM can withstand the forces within an extrusion line [HP19]. A minimum wall thickness independent of the constraints of the manufacturing process has to be enforced. The resulting changes to the algorithm are discussed in Section 5.3.4.

5.2 Literature Review of Algorithms for Additive Manufacturing Restrictions

Most publications in literature utilise the PDEs governing the structural mechanics of the part for the optimisation. This is the main difference between the algorithms in literature and the approach taken in this thesis since the PDE constraints for the topology optimisation problem in this thesis come from the CFD simulation of the domain *surrounding* the optimised part.

Langelaar developed an algorithm which optimises part topology and the support structure layout and build direction [Lan17, Lan18]. In his master thesis, *Verboom* integrated an algorithm checking for manufacturing constraints into OpenFOAM [Ver17].

Yamada et al. developed an algorithm optimising the part geometry in various ways. They considered assembly as a manufacturing criterion, and during the optimisation, they looked at geometric constraints, optimisation of build orientation, and multi-material topology optimisation [FY24, TY23].

Nardoni et al. developed a topology optimisation algorithm that can design appropriate support structures for LPBF using a level-set method [NMD+24].

Garcke et al. developed a phase field structural topology optimisation algorithm, where the critical overhangs in the structure are penalised using anisotropic energy functionals, resulting in a support-free structure [GLNS23].

To maximise the potential of AM for the design of extrusion dies, it is beneficial to build an algorithm that optimises the topology using a support-free approach, where the part does not require external supports meant to be removed after printing. The algorithm should also be compatible with the IB representation of the die geometry.

Many topology optimisation algorithms use a density-based approach to model the phase transition between the structure and its surroundings. One common approach for structural topology optimisation is the Solid Isotropic Material with Penalty (SIMP) method. This interpretation is compatible with an IB approach, where the domain is modelled as a porous medium, making algorithms based on this method well-suited for the presented problem.

Behrou et al. apply a SIMP method, where they used an adjoint-based approach to obtain the sensitivities for the optimisation under projection-based overhang constraints [BGG18].

Haveroth et al. developed a SIMP-based topology optimisation algorithm and investigated both a self-weight-based formulation and a thermal conductivity-based formulation for the manufacturing restrictions [HTC+22]. They found that both formulations improved the manufacturability of the structure.

Zhang et al. developed an optimisation algorithm for the design of self-supporting printable structures using a SIMP approach under consideration of multiple design restrictions. Alongside the overhang angle constraint, they also included a horizontal minimum length constraint and a connectivity constraint for both the part and the surrounding structure [ZCX19, ZC20, ZC22].

5.3 Geometric Constraints Algorithm

The manufacturing constraints formulation developed by *Zhang and Cheng* was used as the basis for the algorithm presented in this thesis. It was modified to adapt it to OpenFOAM's data structures and as well as non-cartesian meshes. In addition, a new method was used to incorporate the manufacturing constraints into the previously presented optimisation algorithm.

Whereas the traditional SIMP algorithm forming the basis of the algorithm by *Zhang and Cheng* operates on an artificial material density function as the design variable, the algorithm developed for this thesis adapts the formulas to the IB representation by using the porosity α_m of cell m normalised by α_{max} as the density ρ of cell m.

In the first step, a linear density filter (Equation 5.1) is applied to the material density, which avoids checkerboard distributions and improves mesh independence, relaxing the density field [ZCX19]. $\rho_{im,i}$ is the intermediate density of the i^{th} cell within the sphere of influence N_i. $\mathbf{x}_m$ is the coordinate of the centroid of the m^{th} cell within N_i and w denotes the weight for the cell.

$$\rho_{\mathrm{im,i}} = \frac{\sum_{\mathrm{m}\in \mathrm{N_i}} \mathrm{w}(\mathbf{x}_\mathrm{m}) \mathrm{V}_\mathrm{m} \frac{\alpha_\mathrm{m}}{\alpha_\mathrm{max}}}{\sum_{\mathrm{m}\in \mathrm{N_i}} \mathrm{w}(\mathbf{x}_\mathrm{m}) \mathrm{V}_\mathrm{m}}, \qquad (5.1)$$

$$\text{with} \quad \mathrm{w}(\mathbf{x}_\mathrm{m}) = \mathrm{R_i} - \|\mathbf{x}_\mathrm{m} - \mathbf{x}_\mathrm{i}\|$$

In the original paper by *Zhang and Cheng*, the neighbourhood N_i is defined as all cells within a certain radius R of cell i [ZCX19, ZC20]. While this is the most robust implementation, using the same implementation for N_i in OpenFOAM would lead to a prohibitively high runtime. In OpenFOAM, all information regarding the computational mesh is stored in a *polyMesh*, which consists of lists of points, faces and cells. For each cell i, the cells with a common face with cell i can be directly accessed via the function *cellCells*. Cells that border cell i by at least one common point can also be determined with relatively low computational complexity. However, with non-cartesian meshes in particular, assembling a data structure for N_i that spans more than two cells in each direction exceeds the computational complexity of performing the rest of the optimisation problem once the computational mesh exceeds roughly one million cells. Instead, N_i is determined to be the set of all cells that neighbour cell i by at least one common point, as well as their neighbours (i.e., cell i's second-degree neighbours). This decreases the sphere of influence but was found to have little impact on the algorithm's performance. For the computation of $w(\mathbf{x}_m)$, the radius R is calculated for each cell i to account for the varying cell sizes within the computational mesh and was manually fitted to $R_i = 0.2 \cdot \sqrt[4]{V_i}$, which gave the most robust results across the non-uniform cell sizes present in the mesh.

Next, a volume-preserving nonlinear density filter is applied to obtain a discrete structure with a sharp fluid-solid interface [ZCX19]. The physical density $\boldsymbol{\rho}_{\text{physical}}$ is calculated using the intermediate density obtained in the first step. The constant β_f indicates how aggressive the filter is, while η_f is the threshold parameter [Bou01, XCC10].

$$\rho_{\mathrm{physical,i}} = \begin{cases} \eta_f \left[e^{-\beta_f (1-\rho_{\mathrm{im,i}}/\eta_f)} - \left(1 - \rho_{\mathrm{im,i}}/\eta_f\right) e^{-\beta_f} \right], & 0 \le \rho_{\mathrm{im,i}} \le \eta_f \\ (1-\eta_f) \left[1 - e^{-\beta_f (\rho_{\mathrm{im,i}} - \eta_f)/(1-\eta_f)} + \right. & \\ \left. +(\rho_{\mathrm{im,i}} - \eta_f) e^{-\beta_f} / (1-\eta_f)\right] + \eta_f, & \eta_f \le \rho_{\mathrm{im,i}} \le 1 \end{cases} \tag{5.2}$$

For the algorithm presented, both β_f and η_f have been manually calibrated to 0.2.

5.3.1 Overhang Angle Constraint

The following algorithm enforcing the overhang angle constraint is based on the 18 element scheme (Figure 5.4). It was developed for cartesian meshes and presented for 2D in [ZCX19] and for 3D in [ZC20]. In the 18 element scheme, each cell i in the computational mesh is surrounded by eight neighbours (a, ⋯, h) within the same layer and nine neighbouring cells (j, ⋯, r) in the layer below, which is the layer where its structural support comes from. The 18 cell IDs form the vector $S_i = (\text{ i, a, } \cdots \text{, r })^T$.

Generally, the overhang angle β is defined as

$$\cos \beta = \frac{1}{\|\nabla \boldsymbol{\rho}_{\mathrm{physical}}\|} \nabla \boldsymbol{\rho}_{\mathrm{physical}}^{\mathrm{T}} \mathbf{n}, \tag{5.3}$$

where $\mathbf{n}$ is the unit vector of the build direction. In the original publications by *Zhang and Cheng*, the build direction is defined as $\mathbf{n} = (0, 0, 1)^T$ (build in z-direction) [ZCX19, ZC20]. Since all geometries considered in this thesis are based on a cylindrical shape, the build direction coincides with the direction of extrusion, which is along the x-axis, i.e., $\mathbf{n} = (1, 0, 0)^T$. In the following, both versions will be presented if not stated otherwise.

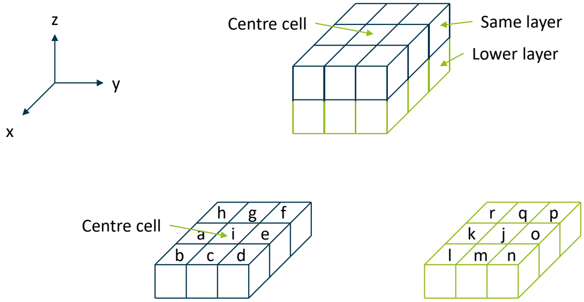


Figure 5.4: 18 element scheme for a cartesian mesh with z as build direction (adapted from [ZC20]).

For the computation of the local overhang angle for cell i, a local density distribution function $\hat{\rho}$ is introduced which replaces $\boldsymbol{\rho}_{\text{physical}}$ in Equation 5.3:

$$\hat{\rho}_i (x, y, z) \approx a_i x + b_i y + c_i z + d_i \tag{5.4}$$

This linear density function spans a hyperplane in $\mathbb{R}^4$, and its gradient, which corresponds to the structural boundary surface inward normal, can be computed as

$$\nabla \hat{\rho}_i = (a_i, b_i, c_i)^T . \tag{5.5}$$

To fit the densities of the 18 cells with the hyperplane, the densities and coordinates are substituted into Equation 5.4. The result is a set of linear algebraic equations, which can be represented in matrix form as

$$\mathbf{A}_i \boldsymbol{\xi}_i = \boldsymbol{\rho}_{\text{physical},i}, \tag{5.6}$$

$$\text{where} \quad \mathbf{A}_i = \begin{pmatrix} x_i & y_i & z_i & 1 \\ x_a & y_a & z_a & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_r & y_r & z_r & 1 \end{pmatrix}, \boldsymbol{\xi}_i = \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix}, \boldsymbol{\rho}_{\text{physical},i} = \begin{pmatrix} \rho_{\text{physical},i} \\ \rho_{\text{physical},a} \\ \vdots \\ \rho_{\text{physical},r} \end{pmatrix}.$$

(x_k, y_k, z_k) is the coordinate of the kth cell, k $\in S_i$. To obtain ξ_i, the least squares method is applied to get

$$\boldsymbol{\xi}_i = \mathbf{B}_i \boldsymbol{\rho}_{\text{physical},i}, \quad \text{where } \mathbf{B}_i = \left(\mathbf{A_i}^T \mathbf{A_i} \right)^{-1} \mathbf{A_i}^T. \tag{5.7}$$

For cartesian meshes with build in x-direction, the matrix $\mathbf{B}_i$ reads

$$\mathbf{B}_{\mathrm{i,cartesian}} = \tag{5.8}$$
$$\frac{1}{36}\begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 & -4 \\ 0 & -3 & -3 & 0 & 3 & 3 & 3 & 0 & -3 & 0 & -3 & -3 & 0 & 3 & 3 & 3 & 0 & -3 \\ 0 & 0 & 3 & 3 & 3 & 0 & -3 & -3 & -3 & 0 & 0 & 3 & 3 & 3 & 0 & -3 & -3 & -3 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \end{pmatrix}.$$

The last row of $\mathbf{B}_{\text{i, cartesian}}$ in Equation 5.8 was omitted here as it is unnecessary to determine the density gradient. For a build in z-direction, the first and third row of $\mathbf{B}_{\text{i, cartesian}}$ must be switched.

All optimisations in this thesis are performed on cylindrical geometries. The cells are layered in the build direction, i.e., each layer can be assumed to have the same number of cells, and these cells are stacked on top of each other. However, the cells within a single layer are not arranged in a purely cartesian pattern. $\mathbf{B}_{\text{i, cartesian}}$ needs to be deconstructed for generalisation to non-cartesian meshes.

For the cartesian mesh, each cell i has eight neighbours in the same layer ($n_1 = 8 + 1 = 9$ including cell i) and $n_2 = 9$ neighbours in the layer below. In addition, there are $n_3 = 6$ neighbours with $y_{\text{neigh}} > y_i$ and $n_4 = 6$ neighbours with $y_{\text{neigh}} < y_i$, while there are $n_5 = 6$ neighbours with $z_{\text{neigh}} > z_i$ and $n_6 = 6$ neighbours with $z_{\text{neigh}} < z_i$. Because all neighbours with $y_{\text{neigh}} = y_i$ and $z_{\text{neigh}} = z_i$ are assigned the value 0 in the rows that depend on this, their set size is not explicitly calculated.

The calculation for the individual coefficients of $\mathbf{B}_{\text{i, non-cartesian}}$ is listed in Table 5.1. Because the number of cell neighbours can differ depending on the direction, the rows of $\mathbf{B}_{\text{i, non-cartesian}}$ are assembled in individual data structures.

		Cartesian		Non-Cartesian	
		no. neighs	$\mathbf{B}_i$[row,neigh]	no. neighs	$\mathbf{B}_i$[row,neigh]
Row 1	$x_{\text{neigh}} = x_i$	8+1=9	+1/9	n_1	$+1/n_1$
	$x_{\text{neigh}} < x_i$	9	-1/9	n_2	$-1/n_2$
Row 2	$y_{\text{neigh}} > y_i$	6	+1/3	n_3	$+0.5/n_3$
	$y_{\text{neigh}} = y_i$	6	0	-	0
	$y_{\text{neigh}} < y_i$	6	-1/3	n_4	$-0.5/n_4$
Row 3	$z_{\text{neigh}} > z_i$	6	+1/3	n_5	$+0.5/n_5$
	$z_{\text{neigh}} = z_i$	6	0	-	0
	$z_{\text{neigh}} < z_i$	6	-1/3	n_6	$-0.5/n_6$

Table 5.1: Construction of $\mathbf{B}_i$.

In the row of $\mathbf{B}_{\text{i, non-cartesian}}$ corresponding to the build direction, cell i and all neighbours in the same layer as cell i are assigned $+1/n_1$, and all neighbours in the layer below cell i are assigned $-1/n_2$, provided such a layer exists.

The row corresponding to the y-coordinate of $\mathbf{B}_{\text{i, non-cartesian}}$ is constructed by assigning the value $+0.5/n_3$ to all neighbours with $y_{\text{neigh}} > y_i$ and $-0.5/n_4$ to all neighbours with $y_{\text{neigh}} < y_i$, while all cells with $y_{\text{neigh}} = y_i$ including cell i are assigned 0.

For the z-coordinate, the assembly corresponds to that of row 3 in $\mathbf{B}_{\text{i, non-cartesian}}$. All neighbours with $z_{\text{neigh}} > z_i$ get assigned with $+0.5/n_5$ and all neighbours with $z_{\text{neigh}} < z_i$ get assigned with $-0.5/n_6$, while all cells with $z_{\text{neigh}} = z_i$ including cell i get assigned 0.

Using this assembly, Equation 5.7 can be evaluated for each cell i. Thus, the overhang angle can be computed as

$$\cos\beta_i = \frac{1}{\|\nabla\hat{\rho}_i\|}\nabla\hat{\rho}_i^{T}\mathbf{n} = \begin{cases} \dfrac{a_i}{\sqrt{a_i^2 + b_i^2 + c_i^2}}, & \text{for } \mathbf{n} = (1,0,0)^{T}. \\ \dfrac{c_i}{\sqrt{a_i^2 + b_i^2 + c_i^2}}, & \text{for } \mathbf{n} = (0,0,1)^{T}. \end{cases} \tag{5.9}$$

Thus, the first geometric constraint $g_1(i)$ for our optimisation problem Equation 4.3 is given by the overhang angle constraint, where $\bar{\beta}$ is the critical overhang angle.

$$g_1(i) = \left(\cos\beta_i - \cos\bar{\beta}\right)\|\nabla\hat{\rho}_i\| \leq 0 \quad (i = 1, \dots, n_{\text{cells}}) \tag{5.10}$$

5.3.2 Horizontal Minimum Length Constraint

The original algorithm enforcing the horizontal minimum length constraint was developed for cartesian meshes and can be found in [ZC20]. Around each cell i, four support sets $M_{i,1}$, …, $M_{i,4}$ are defined (Figure 5.5). Each support set spans a quadrant that reaches up to m cells from cell i. Assuming a build in x-direction, $M_{i,1}$, …, $M_{i,4}$ can be calculated using the cell size Δx as follows:

$$M_{i,1} = \{k \mid x_k = x_i, \quad 0 < y_k - y_i \leq m\Delta x, \quad 0 < z_k - z_i \leq m\Delta x\} \tag{5.11}$$

$$M_{i,2} = \{k \mid x_k = x_i, -m\Delta x \leq y_k - y_i < 0, \quad 0 < z_k - z_i \leq m\Delta x\} \tag{5.12}$$

$$M_{i,3} = \{k \mid x_k = x_i, -m\Delta x \leq y_k - y_i < 0, \quad -m\Delta x \leq z_k - z_i < 0\} \tag{5.13}$$

$$M_{i,4} = \{k \mid x_k = x_i, \quad 0 < y_k - y_i \leq m\Delta x, -m\Delta x \leq z_k - z_i < 0\}, \tag{5.14}$$

where (x_k, y_k, z_k) is the coordinate of the k^th cell. The union of $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, $M_{i,4}$ is a horizontal square area of (2m + 1) × (2m + 1).

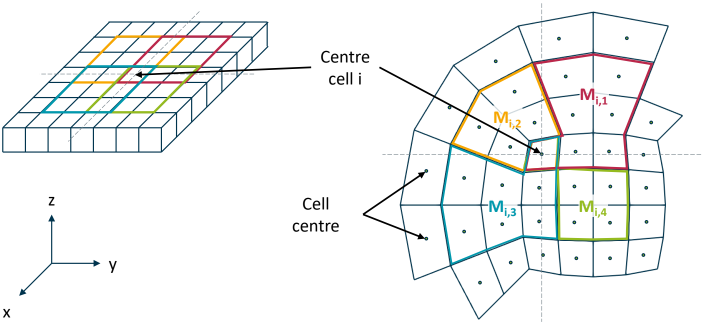


Figure 5.5: Left: horizontal square scheme for a cartesian mesh (adapted from [ZC20]), right: horizontal square scheme for a non-cartesian mesh. The four coloured areas denote the support sets $M_{i,1}$, ..., $M_{i,4}$. The grey dashed lines depict the y- and z-coordinate extension through the central cell, respectively.

The horizontal minimum length constraint is fulfilled if, for each cell in the structure, at least one of the average cell densities in its four support sets is not smaller than the density of the cell itself:

$$\begin{gathered}\tau_{i,1}\left(\boldsymbol{\rho}_{\text{physical}}\right) \le 0 \quad \text{or} \quad \tau_{i,2}\left(\boldsymbol{\rho}_{\text{physical}}\right) \le 0 \quad \text{or} \\ \tau_{i,3}\left(\boldsymbol{\rho}_{\text{physical}}\right) \le 0 \quad \text{or} \quad \tau_{i,4}\left(\boldsymbol{\rho}_{\text{physical}}\right) \le 0 \\ i = 1, \dots, n_{\text{cells}}\end{gathered} \tag{5.15}$$

For cartesian meshes, this translates to

$$\tau_{i,k,\text{cartesian}} = \rho_{\text{physical},i} - \frac{\sum_{j \in M_{i,k}} \rho_{\text{physical},j}}{(m+1)^2 - 1}, \qquad k = 1, 2, 3, 4. \tag{5.16}$$

For non-cartesian meshes, the support sets have different sizes depending on their location in the mesh. Therefore, the support constraint has to be generalised to

$$\tau_{i,k,\text{non-cartesian}} = \rho_{\text{physical},i} - \frac{\sum_{j \in M_{i,k}} \rho_{\text{physical},j}}{\max\left(1, \text{size}\left(M_{i,k}\right)\right)}, \qquad k = 1, 2, 3, 4. \tag{5.17}$$

As a result, the second geometric constraint for our optimisation problem $g_2(i)$ for cell i reads

$$g_2(i) = \min_k \tau_{i,k}\left(\rho_{\text{physical},i}\right) \le 0. \tag{5.18}$$

5.3.3 Connectivity Constraint

The connectivity constraint ensures that the void phase is connected to the outside of the domain and that dead angles that might prevent the removal of residue powder are absent. Therefore, for the first time, the connectivity constraint introduces a manufacturing constraint that operates on both the structure and the domain's void phase.

It can be demonstrated that the connectivity constraint is fulfilled if the geometric constraints g_1 and g_2 are fulfilled for the solid phase in build direction **and** the void phase against build direction [ZC22].

As a result, the horizontal minimum length constraint needs to be modified to accommodate both the solid phase and the void phase. For this purpose, the concept of a phase density $\rho_{\text{phase,i}}$ for cell i is introduced, where

$$\rho_{\text{phase,i}} = \begin{cases} 1 - \rho_{\text{physical,i}}, & \text{for } \rho_{\text{physical,i}} \leq 0.5. \\ \rho_{\text{physical,i}}, & \text{for } \rho_{\text{physical,i}} > 0.5. \end{cases} \tag{5.19}$$

This enables the calculation of the horizontal minimum length constraint for both phases in one step. Thus, the final geometric constraint $g_3(i)$ for cell i replacing $g_2(i)$ reads

$$g_3(i) = \min_k \tau_{i,k}\left(\rho_{\text{phase,i}}\right) \leq 0. \tag{5.20}$$

5.3.4 Application-Specific Geometric Constraint

In the algorithm presented, the additional constraints regarding the design restrictions from the application are met by embedding a scaffold structure α_{fixed} in the initial geometry. $\alpha_{\text{fixed, i}}$ is assigned the value 1 for each cell i that is part of the scaffold structure and 0 everywhere else. This scaffold structure fulfils all constraints regarding manufacturing and operation and will remain untouched during optimisation. A drawback of this approach is that it limits the DOF of the topology optimisation algorithm presented in Section 4, effectively turning it into a shape optimisation algorithm for some choices of α_{fixed}. The exact nature and purpose of the scaffold structure depends on the specific use case and will be explained in Section 6 and Section 7, respectively.

5.3.5 Incorporation of Geometric Constraints into Optimisation Algorithm

The optimisation algorithm presented in Section 4 proposes an updated IB representation of the geometry (Equation 4.13). The manufacturing constraints g_1 and g_3 cannot be directly incorporated into the algorithm in the same way the PDE constraints are embedded via the sensitivity of the cost functions. Instead, the algorithm splits the proposed update of the design variables and the actual update of the design variables into two steps. It then checks the proposed update's compliance with the geometric constraints for each cell individually. The proposed change in iteration n of the design variables $\alpha_{\text{diff, n}}$ is calculated from

$$\alpha_{\mathrm{diff,n}} = \gamma_{\mathrm{r}} \cdot \left(\min\left(\max\left(\alpha_{\mathrm{n}} + \lambda_{\mathrm{u}}\sigma_{\mathrm{u}} + \lambda_{\mathrm{T}} w_{\mathrm{T}} \sigma_{\mathrm{T}} + \lambda_{\mathrm{C}} w_{\mathrm{C}} \sigma_{\mathrm{C}}, \alpha_0\right), \alpha_{\max}\right) - \alpha_{\mathrm{n}}\right) \quad (5.21)$$

Next, the algorithm visits each cell of the computational domain. If the cell is designated as part of the scaffold structure ($\alpha_{\mathrm{fixed}} = 1$), any proposed changes are automatically rejected. If no change is proposed for a cell ($\alpha_{\mathrm{diff,\,n}} = 0$), the cell is skipped. The remaining scenarios are listed in Table 5.2 and explained in detail in the following. For simplicity, every cell i is designated as currently fluid if $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$ and as currently solid if $\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$.

Proposed change of design variable $\alpha_{\mathrm{diff,\,n}} > 0$			
No.	Constraint g_1	Constraint g_3	Action for current assignment of α_n
1	< 0	≤ 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: accept $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: accept
2	< 0	> 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: reject $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: accept
3	≥ 0	≤ 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: reject $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: reject
4	≥ 0	> 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: reject $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: reject
Proposed change of design variable $\alpha_{\mathrm{diff,\,n}} < 0$			
No.	Constraint g_1	Constraint g_3	Action for current assignment of α_n
5	< 0	≤ 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: accept $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: accept
6	< 0	> 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: accept $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: reject
7	≥ 0	≤ 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: accept $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: accept
8	≥ 0	> 0	$\alpha_{n,i} \geq 0.5 \cdot \alpha_{max}$: accept $\alpha_{n,i} < 0.5 \cdot \alpha_{max}$: reject

Table 5.2: Criteria used to accept or reject proposed changes by optimisation algorithm to ensure additive manufacturability.

According to Table 5.2, different scenarios have to be distinguished:

1: The proposed change of the design variables implies that the flow in this cell should be blocked. If the cell is already designated as solid, this change to increase α even further should be accepted. If the cell was previously designated as fluid, the combination of constraint functions does not explicitly oppose this change, and therefore, it should be confirmed for now. If this change turns out to have been wrong, the algorithm can correct it in the next iteration.

2: In this scenario, the algorithm proposes to increase α, but the horizontal minimum length constraint is violated. If the cell is currently assigned solid, this change is rejected. However, for a void phase cell, this combination of constraint functions implies that the algorithm wants to change the cell's (wrong) assignment. Therefore, this change should be confirmed.

3+4: In scenarios 3 and 4, the overhang angle constraint is violated. As a result, the proposed change to increase α is automatically rejected regardless of the cell's current assignment.

5: Here, the proposed change of the design variables implies that polymer melt flow should be allowed in this cell. At the same time, no manufacturing constraints are violated. As a result, a cell that is currently assigned as solid phase could stay solid but will be changed to void phase for now. If it turns out that the manufacturing constraints are then violated, the algorithm can correct it in the next iteration. If the cell is already assigned as void phase, this change is trivial and can be confirmed.

6: In scenario 6, the algorithm proposes a decrease of α towards an assignment as a void phase cell. While the horizontal minimum length constraint is violated, the overhang constraint is fulfilled. As a result, the change should be confirmed for a cell currently assigned as solid but rejected for a cell currently assigned as part of the void phase.

7: In this scenario, the overhang angle constraint is violated in the current configuration, and the algorithm proposes to change the design variable in this cell towards a void phase assignment. If that cell is currently part of the solid phase, this change should be confirmed, and the algorithm will decide in the next iteration whether this change has improved the manufacturing constraints in this cell. This change can be confirmed if the cell is already assigned as a void phase cell.

8: In the final scenario, all manufacturing constraints are currently violated, and the algorithm proposes changing this cell's design variable towards a void phase assignment. This change should be rejected if the cell is already assigned as void phase. However, the change should be confirmed if the cell is currently designated as a solid phase cell.

The pseudocode of the resulting adjoint topology optimisation algorithm under consideration of the design constraints from AM is depicted in Figure 5.6.

```
1   Compute converged solution of state equations for initial geometry
    (without geometry update)

2   for i = 1 to $n_{Cells}$ do
        Determine neighbourhood $N_i$ and weights $w_i$
        Compute $\mathbf{B}_i$, $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, $M_{i,4}$

3   for n = 1 to $n_{designIterationSteps}$ do
        Compute intermediate and physical density $\boldsymbol{\rho}_{im}$ and $\boldsymbol{\rho}_{physical}$
        Compute overhang angle constraint $g_1$
        Compute horizontal minimum length constraint $g_3$
        Compute proposed change of the design variables $\alpha_{diff,n}$
        for i = 1 to $n_{Cells}$ do
            if $\alpha_{fixed,i} = 1$ or $\alpha_{diff,n,i} = 0$ then
                Continue
            Check geometric constraints for cell i
            Confirm/reject proposed change of design variable for cell i
        Update solution of primal and adjoint equations

4   Compute converged solution of state equations for optimised geometry
    (without geometry update)
```

Figure 5.6: Pseudocode of the optimisation algorithm under consideration of design constraints from AM.

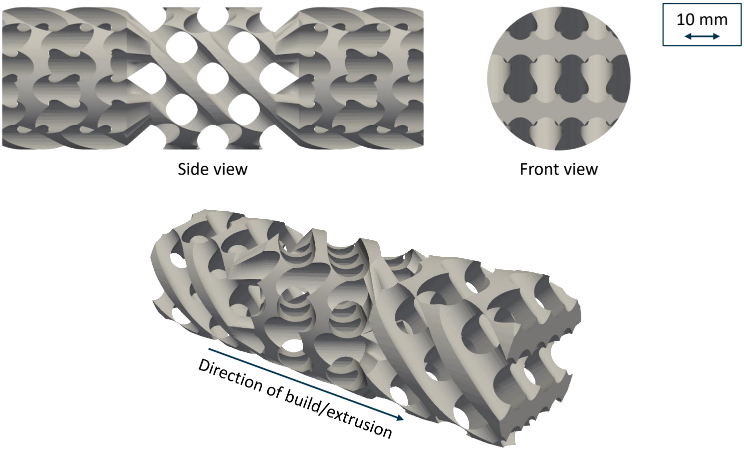
Note that steps 2 and 3 cannot be run in parallel using MPI at this point as the connectivity data structures for the geometric constraints do not work across partition boundaries. The files regarding the AM-compatibility algorithm are listed in Appendix A.2, the full source code is published on GitHub and can be accessed at [URL24].

6 Application to Static Mixers

This chapter presents and discusses the application of the optimisation algorithm to a static mixer. First, the static mixer before optimisation is described, and the optimisation objectives, along with the setup of the simulation, are discussed before the optimisation results are presented. Furthermore, the operating point dependency is investigated, and the validation of the optimisation results in lab trials is presented. The chapter closes with a discussion of the findings.

The X-mixer that formed the basis of this investigation was originally based on an SMB plus 40/3 (Promix Solutions GmbH, Wetzlar, DE), a commercially available static mixer manufactured by casting. The construction of an additively manufacturable variation of this mixer has been detailed in [HGL+18, HST+19]. This already additively manufacturable geometry consists of basic repetition units that have been optimised to support themselves. Between the individual blocks, which have a length of 1 D, a transition element helps smooth the overall geometry.

In the simulations, the static mixer's geometry was virtually imposed on the computational mesh using the immersed boundary method. The static mixer's parameterisation was extended by 1 mm in the radial direction to avoid unstable cell-cutting operations at the interface of the static mixer with the outer cylinder wall. The resulting static mixer is depicted in Figure 6.1 and has a length of 124 mm and a target diameter of 40 mm, which is extended to 42 mm by the overhangs.



Side view

Front view

Figure 6.1: Geometry of the static mixer before optimisation including overhangs.

6.1 Optimisation of an Additively Manufacturable Static Mixer

Three optimisation objectives were investigated for the optimisation of the static mixer. The first was minimising the pressure drop in the mixer, while the second and third were maximising either thermal or material mixing.

6.1.1 Simulation Setup

The simulations were performed on a cylinder with a diameter of 41 mm and a length of 140 mm. The computational mesh was created with *blockMesh* and consisted of 3.26 million hexahedral cells. The initial volume of the static mixer before optimisation was 50,103 mm^3. The scaffold structure (Figure 6.2 top left) followed the same build pattern but had a mixing bar thickness of 1 mm. Besides ensuring manufacturability, the scaffold structure also has a practical function. During optimisation for minimal pressure drop, the algorithm is expected to minimise the resistance in the flow channel, including the inherent flow resistance introduced by a mixing element, disregarding the purpose of the mixer geometry as a necessary component [SSH22]. In addition, the static mixer was encased by a fixed cylinder that ensures support for the static mixer at the domain's boundaries. The cylinder's wall

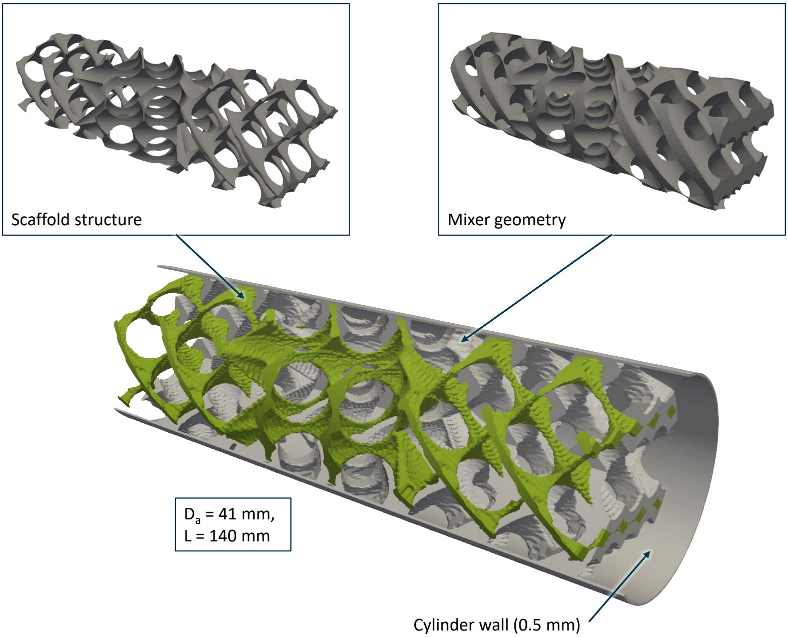


Figure 6.2: Shape and position of static mixer (depicted in grey) and scaffold structure (depicted in green) within the cylinder before optimisation.

thickness was 0.5 mm, corresponding to the computational mesh's eight outermost cells. During initialisation, the geometry of the static mixer, the scaffold and the outside cylinder were imposed on the mesh using *setFields*. The resulting structure within the computational domain is depicted in Figure 6.2.

The operating point for the optimisation was characterised by the throughput, extruded material, inlet temperature profile and inlet concentration profile.

The throughput during optimisation was set to 20 kg/h based on previous investigations [SSH22].

The plastic melt in this setup was Hostalen GD 9550 F (LyondellBasell GmbH, Wesseling, DE), which is a blown film type Polyethylene of High Density (PE-HD) with a Melt Flow Rate (MFR) of 3.1 g/10 min (190 °C/5.0 kg) [NN18]. Its material data can be found in Table 6.1. In addition, the thermal material data for the static mixer were set to $D_{T,steel} = 1 \cdot 10^{-5}$ m^2/s, $c_{p,steel} = 466$ J/(kg·K) and $D_{C,steel} = 0$ m^2/s.

Parameter	**Unit**	**Value**
Carreau parameter $A_{Carreau}$	m^2/s	12.87
Carreau parameter $B_{Carreau}$	s	0.1871
Carreau parameter $C_{Carreau}$	-	0.655
Standard temperature T_s	K	237
Reference temperature T_m	K	473
Specific heat capacity c_p	J/(kg · K)	2900
Thermal diffusivity D_T	m^2/s	$1.1997 \cdot 10^{-7}$
Melt density ρ	kg/m^3	736
Diffusion coefficient D_C	m^2/s	$1 \cdot 10^{-12}$

Table 6.1: Material model parameters for PE-HD [Yes17].

The definition of geometry, throughput, and extruded material allows for the calculation of the Graetz number for this operating point. The static mixer with a diameter of 40 mm had an inlet area of $1.2566 \cdot 10^{-3}$ m^2. With a melt density of 736 kg/m^3 and a throughput of 20 kg/h, this results in an average inlet velocity of $\mathbf{u} = 6.0068 \cdot 10^{-3}$ m/s. The length of the mixer was L = 124 mm, and the distance between the mixing bars was assumed to be H = 10 mm. The thermal diffusivity of PE-HD was $1.1997 \cdot 10^{-7}$ m^2/s. The resulting Graetz number was

$$\mathrm{Gz} = \frac{\mathbf{u} \cdot \mathrm{H}^2}{\mathrm{D_T} \cdot \mathrm{L}} = \frac{6.0068 \cdot 10^{-3}\,\mathrm{m/s} \cdot (10\,\mathrm{mm})^2}{1.1997 \cdot 10^{-7}\,\mathrm{m^2/s} \cdot 124\,\mathrm{mm}} = 40.37. \tag{6.1}$$

Therefore, this operating point falls under the so-called transitional regime, implying that both lengthwise convection and transverse conduction influence the thermal regime within the static mixer.

The inlet temperature profile (Equation 6.2 and Figure 6.3 left) corresponded to a hot core inhomogeneity. Its parameters have been chosen to correspond to the inhomogeneities typically observed at the extrusion line used in the validation lab trials.

$$\mathrm{T(r)} = \left(-1 \cdot 10^8 \cdot \mathrm{r}^4 - 44430 \cdot \mathrm{r}^2 + 31.615\right) \cdot 1.5 + 473.15 \tag{6.2}$$

The variable r in Equation 6.2 corresponds to the radius within the flow channel, which can be calculated through the Pythagorean theorem.

$$\mathrm{r} = \sqrt{\mathrm{y}^2 + \mathrm{z}^2} \tag{6.3}$$

The inlet concentration profile (Equation 6.4 and Figure 6.3 right) was a half profile similar to the works by *Singh et al.* [MSA12, Sin08]. The profile was parameterised using a logistic function to achieve a smoother transition between the two phases.

$$\mathrm{C(z)} = \frac{1}{1 + \exp\left(-2000 \cdot \mathrm{z}\right)} \tag{6.4}$$

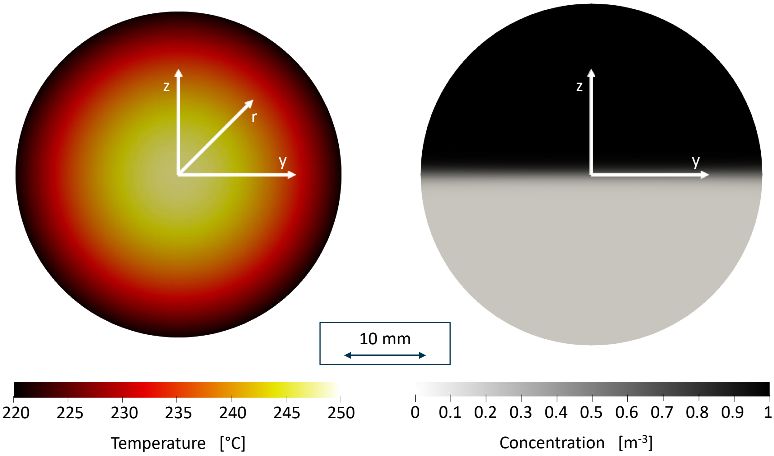


Figure 6.3: Thermal and material inlet inhomogeneity pattern during optimisation.

The complete set of initial and boundary conditions for the state variables is listed in Table 6.2.

State variable	**Inlet**	**Wall**	**Outlet**	**Initial field**
u	flowRateInletVelocity $7.5483 \cdot 10^{-6}\ \mathrm{m^3/s}$	noSlip	zeroGradient	$(0\ 0\ 0)\ \frac{\mathrm{m}}{\mathrm{s}}$
p	zeroGradient	zeroGradient	fixedValue $10^{-5}\ \mathrm{m^2/s^2}$	$10^5\ \frac{\mathrm{m^2}}{\mathrm{s^2}}$
T	codedFixedValue see Equation 6.2	fixedValue 493.15 K	zeroGradient	493.15 K
C	codedFixedValue see Equation 6.4	zeroGradient	zeroGradient	$0\ \frac{1}{\mathrm{m^3}}$
$\mathbf{u}_a$	fixedValue (-1 0 0) m/s	noSlip	adjointOutletVelocity (0 0 0) m/s	$(0\ 0\ 0)\ \frac{\mathrm{m}}{\mathrm{s}}$
p_a	zeroGradient	zeroGradient	adjointOutletPressure $0\ \mathrm{m^2/s^2}$	$10^5\ \frac{\mathrm{m^2}}{\mathrm{s^2}}$
T_a	fixedValue $-1\ \mathrm{m^2/(s^2K)}$	fixedValue $0\ \mathrm{m^2/(s^2K)}$	adjointOutletTemperature $0\ \mathrm{m^2/(s^2K)}$	$0\ \frac{\mathrm{m^2}}{\mathrm{s^2K}}$
C_a	fixedValue $-1\ \mathrm{m^5/s^2}$	fixedValue $0\ \mathrm{m^5/s^2}$	adjointOutletConcentration $0\ \mathrm{m^5/s^2}$	$0\ \frac{\mathrm{m^5}}{\mathrm{s^2}}$

Table 6.2: Initial and boundary conditions.

For the optimisation for thermal and material mixing, T_{target} and C_{target} were set to 493.15 K and 0.5 $1/m^3$, respectively. The number of optimisation steps as well as the sensitivity response step sizes were determined manually, ensuring that no more than 20% of the initial geometry was removed.

For the evaluation of the static mixers' performance characteristics after optimisation, all state variable fields were recalculated using the original initial and boundary conditions and the new optimised static mixer IB fields.

6.1.2 Optimisation Results

This section opens with a presentation of the new geometries and the changes resulting from the optimisation algorithm, followed by a discussion of the effects on the performance characteristics of the static mixers. An in-depth interpretation and discussion of the observed results will follow in Section 6.4.

The optimisation for minimal pressure drop reduced the steel volume of the static mixer by 15% to 42,672 mm^3. Figure 6.4 top depicts the changes to the static mixer made by the algorithm, where the orange regions represent areas where material has been removed, leaving only the reference geometry visible, while the blue regions represent where material has been added, while the geometry remains unchanged in the grey area. A different way to look at the performance of the algorithm is depicted in Figure 6.4 at the bottom, where all blue surfaces show candidate areas for the removal of material, while red surfaces show candidate areas for the addition of material, not considering manufacturability. A view from a different perspective with a focus on the inlet of the static mixer is available in Figure A.1 in Appendix A.3. Both representations show that the optimisation for minimal pressure drop resulted mainly in material removal except at the outside wall, where the manufacturing constraints algorithm prevented many changes. The most material removal occured near the centreline, where the highest flow velocities and, thus, the highest impact on the overall pressure drop were observed. The only areas where material could be added were found behind the final cross bars. These areas are typical spots for dead zones, where recirculation vortexes increase the energy dissipation within the flow channel.

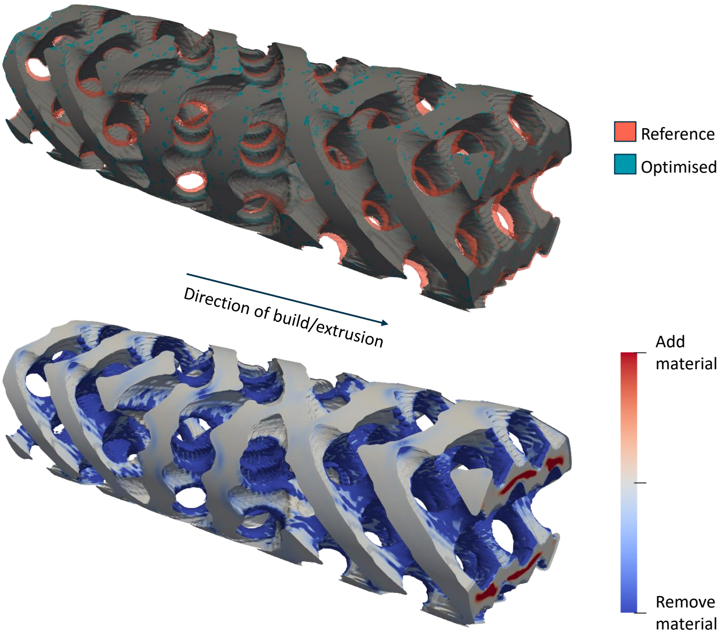


Figure 6.4: Changes to the static mixer geometry after optimisation for minimal pressure drop.

Optimisation to maximise thermal mixing reduced the overall steel volume of the static mixer by 7% to 46,811 mm^3. The changes to the geometry of the static mixer are depicted in Figure 6.5. A view from a different perspective with a focus on the inlet of the static mixer is available in Figure A.2 in Appendix A.3. The changes were less uniform compared to the optimisation for minimal pressure drop. Near the inlet, more material was removed in the outer parts closer to the cylinder wall and the up- and downstream-facing flanks of the mixing bars, while the out- and inward-facing flanks mainly remained untouched. In addition, material was added at some of the downstream-facing flanks of the mixing bars.

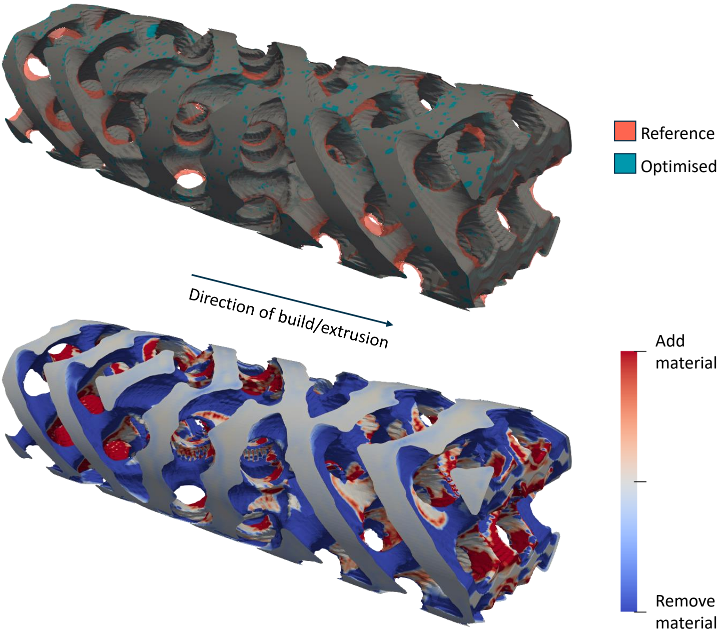


Figure 6.5: Changes to the static mixer geometry after optimisation for maximal thermal mixing.

Optimisation to maximise material mixing reduced the overall steel volume of the static mixer by 19% to 40,688 mm^3. The observed changes to the geometry of the static mixer (Figure 6.6) were the most asymmetric out of the three optimisation objectives. A view from a different perspective with a focus on the inlet of the static mixer is available in Figure A.3 in Appendix A.3. The algorithm removed material from the mixing bars near the cylinder wall as well as near the centreline. However, at the inlet of the mixer, material was removed on one side and added on the other, corresponding to the material inhomogeneity at the inlet of the domain. On the side where the initial concentration was high, material was added, and the static mixer was also extended near the cylinder wall. In contrast, on the side where the initial concentration was low, material was removed to the extent that the upstream-facing flank of the first mixing bar was almost completely removed, leaving only the scaffold structure. This asymmetry in the changes to the geometry reduced towards the back of the static mixer, where the concentration difference was less severe.

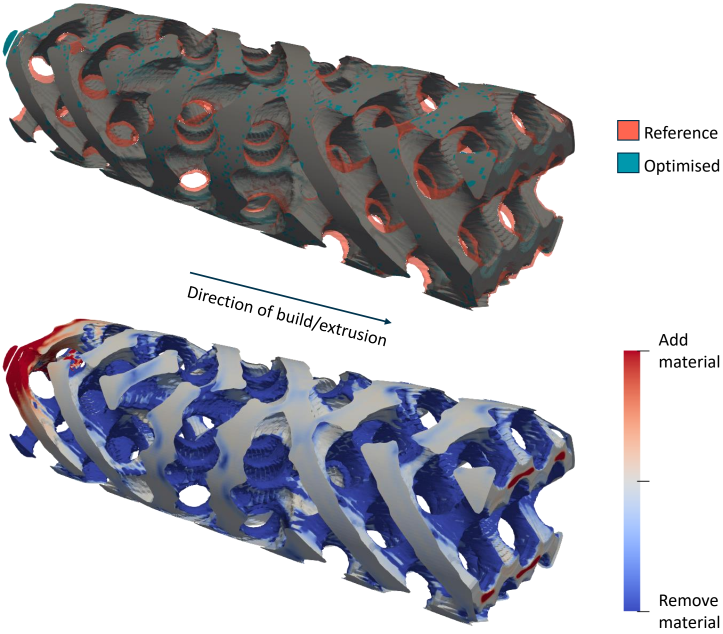


Figure 6.6: Changes to the static mixer geometry after optimisation for maximal material mixing.

In the following, the static mixer before optimisation will be referred to as *Geometry "M"*, while the static mixers optimised for minimal pressure drop, maximal thermal mixing and maximal material mixing will be referred to as *Geometry "P"*, *Geometry "T"* and *Geometry "C"*, respectively.

The performance of each of the geometries was quantified by evaluating the pressure drop as well as the local thermal and material inhomogeneity at cut planes positioned 5 mm before and after the mixer.

Table 6.3 lists the steel volume of each static mixer, the pressure drop and the thermal and material mixing performance for all four geometries.

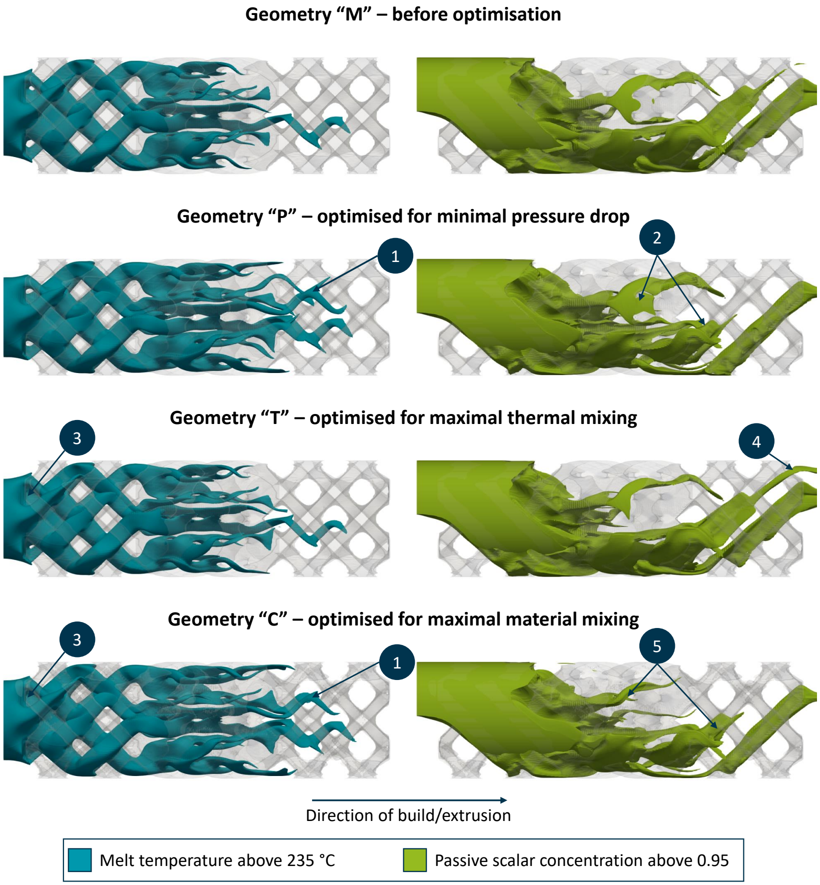
	V [mm^3]	**Δp [bar]**	**$e_{thermal}$ [-]**	**$e_{material}$ [-]**
Geometry ”M”	50,103	33.58	0.78	0.67
Geometry ”P”	42,672 (-15%)	27.45 (-18%)	0.79 (+1%)	0.70 (+5%)
Geometry ”T”	46,811 (-7%)	31.56 (-6%)	0.81 (+3%)	0.69 (+3%)
Geometry ”C”	40,688 (-19%)	28.95 (-14%)	0.81 (+3%)	0.72 (+8%)

Table 6.3: Steel volume, pressure drop, thermal and material mixing performance for the static mixers before and after optimisation, where the best performance is highlighted in green.

Overall, material removal dominated for all optimisation objectives, resulting in a reduced steel volume for all optimised static mixers. During optimisation for thermal mixing, the ratio of material addition and material removal was more balanced, resulting in a smaller change in the overall steel volume for geometry ”T” compared to geometries ”P” and ”C”.

The initial geometry resulted in a pressure drop of $\Delta p = 33.58$ bar. Optimisation for minimal pressure drop reduced the flow resistance of the static mixer, reducing the pressure drop by 18% to 27.45 bar. A similar effect was observed with the other two optimisation objectives: Optimisation to maximise thermal mixing reduced the pressure drop by 6% to 31.56 bar, while the optimisation to maximise material mixing reduced the pressure drop by 14% to 28.95 bar.

Figure 6.7 depicts the flow patterns in the static mixer geometries in two different ways: on the left side, the areas which have a melt temperature above 235 °C are shown in blue, while the right depicts areas with a passive scalar concentration above 0.95 in green. In both cases, the goal is to minimise the amount of plastic melt with these properties reaching the outlet. The most prominent changes are highlighted in the figure using the labels (1) - (5). The mixing bars in geometry ”P” are much thinner than in geometry ”M”, especially near the centreline of the flow channel. The thinner mixing bars reduce the cooling capacity, and melt pools with high temperatures can pass through the mixer undisrupted (1). However, thinner mixing bars towards the centreline of the flow channel promote transverse flows near the centreline, where the highest concentration gradient is found. As a result, geometry ”P” spreads melt pools with high concentration early (2), resulting in a 5% increase in material mixing. The changes in geometry ”T” increase early transverse flows, which help to distribute the hot core (3), promoting increased heat exchange in the melt. As a result, the hot melt



Geometry “M” – before optimisation

Geometry “P” – optimised for minimal pressure drop

Geometry “T” – optimised for maximal thermal mixing

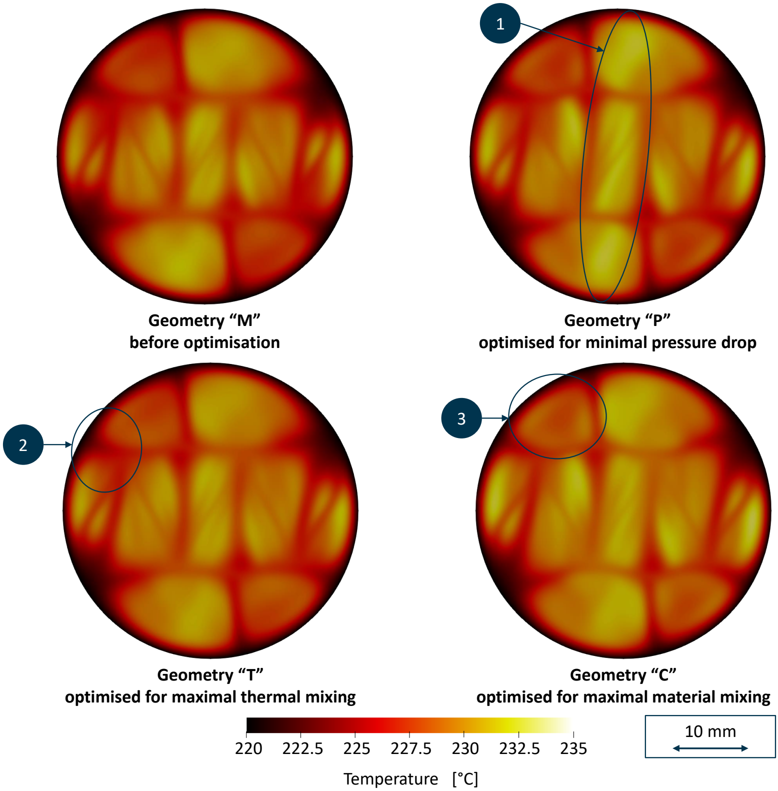
Geometry “C” – optimised for maximal material mixing

1
2
3
4
3
1
5
Direction of build/extrusion
🟦 Melt temperature above 235 °C
🟩 Passive scalar concentration above 0.95

Figure 6.7: Flow patterns through the static mixer geometries before and after optimisation for plastic melt with high temperature (left) and high passive scalar concentration (right).

pools cool faster. While this configuration benefits purely radial distributive mixing, it is not as ideal for the half profile material inhomogeneity pattern. As a result, multiple melt pools with high concentration survive until the outlet (4) compared to the single melt pools in geometry ”M”, which also benefits material mixing overall. The changes in geometry ”C” increase early transverse flows to maximise material exchange between the depicted upper and lower half of the flow channel while also widening the flow channels between the mixing bars overall. As a result, geometry ”C” combines the early transverse flows (3) of geometry ”T” with the reduced melt cooling of geometry ”P” (1), leading to overall improved thermal mixing. The full potential of these changes is apparent when observing melt pools with high concentrations. These are broken up early (5), improving material mixing by 8%.

The resulting temperature distributions at the outlet for all four mixer geometries are depicted in Figure 6.8, with the most prominent changes being highlighted with the labels (1) - (3). Since the thermal mixing number $e_{thermal}$ also takes the local flow velocity into account, the velocity distributions are visualised in Figure A.4 in Appendix A.3.



Geometry “M”
before optimisation

Geometry “P”
optimised for minimal pressure drop

Geometry “T”
optimised for maximal thermal mixing

Geometry “C”
optimised for maximal material mixing

Figure 6.8: Comparison of temperature distribution at the outlet of the mixer for the different objective functions.

Geometry ”M” (Figure 6.8 top left) achieved a thermal homogenisation of $e_{thermal} = 0.78$. While the optimisation for minimal pressure drop improved thermal mixing by 1% to 0.79, the melt exited the mixer at a higher temperature (1) (Figure 6.8 top right). Optimisation for thermal mixing improved the thermal homogenisation by 3% to 0.81, avoiding the hot spots (1) observed with the optimisation for minimal pressure drop while reducing the cold spots observed with the initial geometry (2) (Figure 6.8 bottom left). Optimisation for material mixing also improved the thermal homogenisation by 3% to 0.81, reducing the size of local cold spots (3) (Figure 6.8 bottom right).

The concentration distributions achieved by the four mixer geometries are depicted in Figure 6.9, with the most prominent changes being highlighted with the labels (1) - (3). Since the material mixing number e_{material} also takes the local flow velocity into account, the velocity distributions are visualised in Figure A.4 in Appendix A.3.

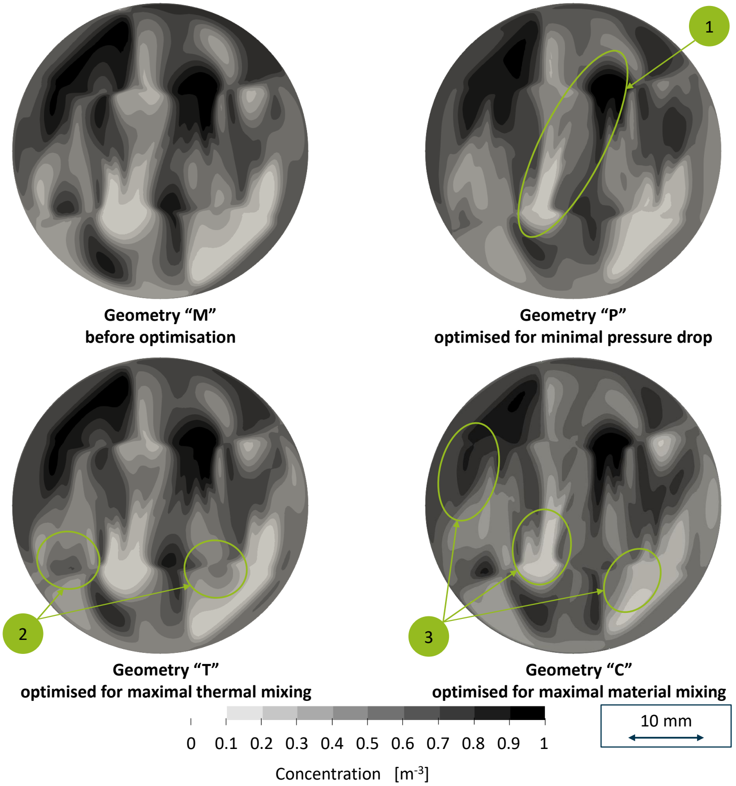


Figure 6.9: Comparison of material distribution at the outlet of the mixer for the different objective functions.

Geometry "M" (concentration profile in Figure 6.9 top left) achieved a material homogenisation of $e_{\text{material}} = 0.67$. Optimisation for minimal pressure drop reduced the size of the areas (1) where the concentration deviated from the average value (Figure 6.9 top right), improving material mixing by 5% to 0.70. A different pattern emerged during optimisation for thermal mixing, where the size of these inhomogeneous areas stayed the same, but their deviation from the average concentration reduced (2) (Figure 6.9 bottom left), resulting in an improvement of material mixing by 3% to 0.69. Optimisation for material mixing improved material mixing by 8% to 0.72 by combining both approaches (3) (Figure 6.9 bottom right).

6.2 Simulative Investigation of the Operating Point Dependency of the Algorithm

The investigation of the Operating Point (OP) dependency was conducted by varying the throughput, extruded material, thermal and material inlet inhomogeneity and evaluating the performance of the static mixer geometries ”M”, ”P”, ”T” and ”C” at these new OPs. Similar investigations with different OPs and an older version of the algorithm were already published in [SH24b], where additional investigations into static mixers that were optimised for different OPs can be found.

6.2.1 Operating Points

Table 6.4 depicts the investigated operating points.

	Polymer material	Throughput	Temperature profile	Concentration profile
OP in optimisation	PE-HD	20 kg/h	Core	Half
Variation of throughput	PE-HD	40 kg/h	Core	Half
Variation of material	PS	20 kg/h	Core	Half
Variation of thermal inhomogeneity	PE-HD	20 kg/h	Ring	Half
Variation of material inhomogeneity	PE-HD	20 kg/h	Core	Ring

Table 6.4: Trial plan for the investigation of the operating point dependency.

The influence of the throughput was investigated by evaluating the static mixers’ performance at a throughput of 40 kg/h, which is double the amount during optimisation, thus doubling the Graetz number.

The plastic melt was replaced with a Polystyrene (PS) (PS158K, Ineos Styrolution GmbH, Frankfurt/Main, DE) for the investigation of the material dependence. Its material data are listed in Table 6.5. PS has a lower zero-viscosity A_{Carreau} and higher shear-thinning properties (higher C_{Carreau}) than PE-HD. These characteristics are associated with a lower pressure drop and improved mixing quality [LHW06]. In addition, it has a lower thermal diffusivity D_T.

The influence of the thermal inhomogeneity on the algorithm’s performance was investigated by evaluating the static mixers’ performance for a ring type inhomogeneity (Equation 6.5 and Figure 6.10 left), which is representative of excessive viscous dissipation within long melt pipes [Cat13].

$$\mathrm{T(r)} = \left(-9 \cdot 10^{7} \cdot \mathrm{r}^{4} + 10116 \cdot \mathrm{r}^{3} + 37414 \cdot \mathrm{r}^{2} - 42.33 \cdot \mathrm{r}\right) \cdot 8 + 493.15 \qquad (6.5)$$

The change in the inlet thermal inhomogeneity has two consequences. On the one hand,

Parameter	**Unit**	**Value**
Carreau parameter $A_{Carreau}$	m^2/s	4.58
Carreau parameter $B_{Carreau}$	s	0.1812
Carreau parameter $C_{Carreau}$	-	0.7146
Standard temperature T_s	K	395.63
Reference temperature T_m	K	503.15
Specific heat capacity c_p	J/(kg · K)	2300
Thermal diffusivity D_T	m^2/s	$7.1999 \cdot 10^{-8}$
Melt density ρ	kg/m^3	936
Diffusion coefficient D_C	m^2/s	$1 \cdot 10^{-12}$

Table 6.5: Material model parameters for PS.

the different temperature profile changes the performance of the cost function for thermal mixing. On the other hand, a ring temperature profile also changes the viscosity distribution in the melt flow, which affects the velocity distribution in the flow channel and, therefore, has the potential to affect the pressure drop and material mixing.

Finally, the influence of the inlet material inhomogeneity was investigated using a ring profile inhomogeneity for the inlet concentration (Equation 6.6 and Figure 6.10 right), which is a common material inhomogeneity resulting from a lower concentration of additives in the centre of the melt pool which forms during single-screw plastication [SH03].

$$C(r) = (9 \cdot 10^7 \cdot r^4 - 10116 \cdot r^3 - 37414 \cdot r^2 + 42.33 \cdot r) \cdot 0.2 + 0.95 \qquad (6.6)$$

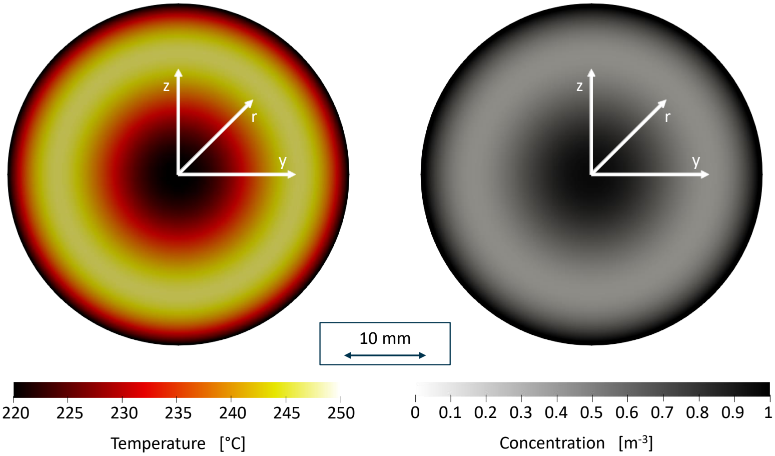


Figure 6.10: Thermal and material inlet inhomogeneity pattern for OP dependency investigation.

6.2.2 Results

Table 6.6 depicts all results of the OP dependency investigation; the first block shows the performance characteristics at the original operating point for reference.

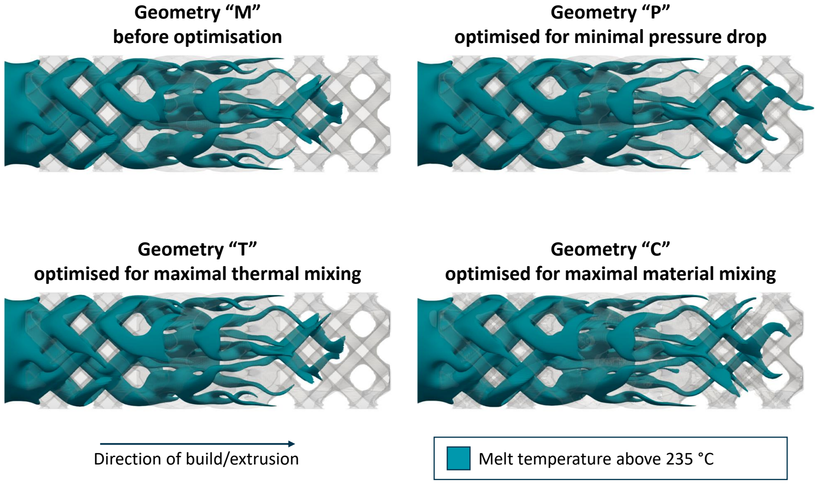
The second block in Table 6.6 depicts the performance characteristics for higher throughput than the mixers were subject to during optimisation. All objective functions still led to a similar, if slightly lower, reduction in pressure drop. Thermal mixing e_{thermal} now increased by 4% with geometry ”T”, which was optimised for thermal mixing, but still by 3% with geometry ”C”, which was optimised for material mixing, and by 1% with geometry ”P”, which was optimised for its pressure drop. This change might imply that geometry ”T” is even more beneficial regarding thermal mixing at higher throughputs, where more shear heating is present, outweighing the reduced residence time within the static mixer. Material mixing e_{material} was unaffected by changes in throughput, independent of the optimisation objective.

The third block in Table 6.6 depicts the performance characteristics when the flow through the static mixers was simulated for a different material than initially optimised for, in this case, a low-viscosity PS with more shear-thinning properties. While the absolute values for the pressure drops associated with each of the geometries were a lot lower with PS than with PE-HD, the reduction in pressure drop caused by the optimisation performed with PE-HD was in the same order of magnitude for all optimisation objectives. Geometry ”P” had the same thermal mixing performance as geometry ”M” before optimisation, while geometries ”T” and ”C” still improved thermal mixing by 3%. Both observations were consistent with the general effects of more shear-thinning polymers described by *Liu et al.* in [LHW06]. Material mixing was unaffected by changes in throughput and independent of the optimisation objective.

Performance at original operating point				
	V [mm^3]	**Δp [bar]**	**e$_{\text{thermal}}$ [-]**	**e$_{\text{material}}$ [-]**
Geometry ”M”	50,103	33.58	0.78	0.67
Geometry ”P”	42,672 (-15%)	27.45 (-18%)	0.79 (+1%)	0.70 (+5%)
Geometry ”T”	46,811 (-7%)	31.56 (-6%)	0.81 (+3%)	0.69 (+3%)
Geometry ”C”	40,688 (-19%)	28.95 (-14%)	0.81 (+3%)	0.72 (+8%)
Performance at higher throughput (40 kg/h) than mixers were optimised for				
	V [mm^3]	**Δp [bar]**	**e$_{\text{thermal}}$ [-]**	**e$_{\text{material}}$ [-]**
Geometry ”M”	50,103	51.87	0.78	0.67
Geometry ”P”	42,672 (-15%)	43.25 (-17%)	0.79 (+1%)	0.70 (+4%)
Geometry ”T”	46,811 (-7%)	49.06 (-5%)	0.81 (+4%)	0.69 (+3%)
Geometry ”C”	40,688 (-19%)	45.50 (-12%)	0.80 (+3%)	0.72 (+7%)
Performance for different material (PS) than mixers were optimised for				
	V [mm^3]	**Δp [bar]**	**e$_{\text{thermal}}$ [-]**	**e$_{\text{material}}$ [-]**
Geometry ”M”	50,103	18.28	0.77	0.67
Geometry ”P”	42,672 (-15%)	15.08 (-17%)	0.77 (±0%)	0.70 (+5%)
Geometry ”T”	46,811 (-7%)	17.24 (-6%)	0.79 (+3%)	0.69 (+3%)
Geometry ”C”	40,688 (-19%)	15.84 (-13%)	0.79 (+3%)	0.72 (+9%)
Performance for other thermal inhomogeneity (ring) than mixers were optimised for				
	V [mm^3]	**Δp [bar]**	**e$_{\text{thermal}}$ [-]**	**e$_{\text{material}}$ [-]**
Geometry ”M”	50,103	33.00	0.76	0.68
Geometry ”P”	42,672 (-15%)	26.94 (-18%)	0.75 (-1%)	0.71 (+4%)
Geometry ”T”	46,811 (-7%)	30.99 (-6%)	0.77 (+1%)	0.70 (+3%)
Geometry ”C”	40,688 (-19%)	28.41 (-14%)	0.76 (-1%)	0.73 (+8%)
Performance for other material inhomogeneity (ring) than mixers were optimised for				
	V [mm^3]	**Δp [bar]**	**e$_{\text{thermal}}$ [-]**	**e$_{\text{material}}$ [-]**
Geometry ”M”	50,103	33.54	0.78	0.76
Geometry ”P”	42,672 (-15%)	27.40 (-18%)	0.79 (+1%)	0.77 (+2%)
Geometry ”T”	46,811 (-7%)	31.51 (-6%)	0.81 (+3%)	0.73 (-3%)
Geometry ”C”	40,688 (-19%)	28.91 (-14%)	0.81 (+3%)	0.77 (+1%)

Table 6.6: Overview over off-design performances of reference and optimised static mixers, where the best performance is highlighted in green, and all declines are highlighted in orange.

The fourth block in Table 6.6 depicts the performance characteristics for a different thermal inhomogeneity than the mixers were subject to during optimisation. The pressure drop results were similar to those for the original OP in all geometries independent of the objective function. Geometry ”M”’s thermal mixing capability was slightly lower for the ring profile inhomogeneity than for the core profile inhomogeneity (0.76 vs. 0.78). Thermal mixing was reduced by 1% in geometries ”P” and ”C”, which were optimised for minimal pressure drop and maximal material mixing, respectively. This is the first instance where an optimised geometry causes a decline in the performance characteristics compared to the static mixer before optimisation. The flow patterns depicted in Figure 6.11 suggest that this is partly due to the reduced cooling in these static mixers. While geometry ”T”, which was optimised for a core profile thermal inhomogeneity, still improved thermal mixing by 1% to 0.77, this improvement is lower than at the original OP. For the ring profile, the flow patterns for geometry ”M” and geometry ”T” (Figure 6.11) are very similar. This suggests that the thermal inlet inhomogeneity must be considered if the goal is to optimise a static mixer for a specific machine setup. The material mixing performances of the different geometries were very similar to the results from the initial OP. While the absolute values of e_{material} were slightly higher, the differences between the geometries were not affected by the different thermal inhomogeneity pattern.


Geometry “M”
before optimisation

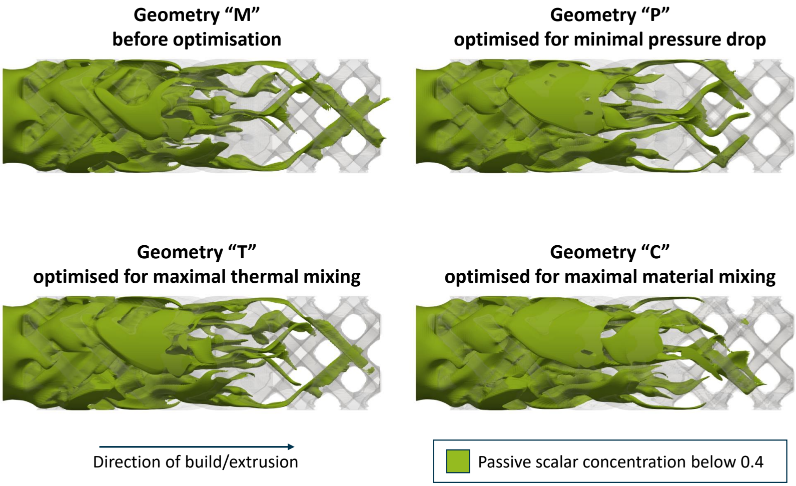
Geometry “P”
optimised for minimal pressure drop

Geometry “T”
optimised for maximal thermal mixing

Geometry “C”
optimised for maximal material mixing

Figure 6.11: Flow patterns in static mixers for a ring thermal inhomogeneity.

The last block in Table 6.6 depicts the performance characteristics for a different material inhomogeneity than the mixers were subject to during optimisation. Pressure drop improvement was largely unaffected by the material inhomogeneity, independent of the objective function. Geometry ”P” still improved thermal mixing by 1%, while geometries ”T” and ”C” still improved the thermal homogenisation by 3%. This confirms the assumption that the material inhomogeneity pattern acts as a passive scalar without affecting the flow or thermal regime. Geometry ”T”, which was optimised for a core profile thermal inhomogeneity, reduced material mixing for a ring profile material inhomogeneity by 3%. The best material mixing was observed with geometry ”P”, which was optimised for minimal pressure drop. However, at 2%, this improvement was considerably worse than for the initial OP. Geometry ”C”, which was optimised for the half profile material inhomogeneity, improved material mixing by 1%. The flow patterns in Figure 6.12, which show areas with a passive scalar concentration below 0.4, support these quantitative observations. While geometries ”P” and ”C” reduce melt pools with low concentration within the length of the static mixer, these inhomogeneous melt pools persist in the static mixer geometries ”M” and ”T”. In addition, geometry ”T” concentrates melt pools with low concentration near the centreline of the flow channel, thus reducing the overall material mixing performance.


Geometry “M”
before optimisation

Geometry “P”
optimised for minimal pressure drop

Geometry “T”
optimised for maximal thermal mixing

Geometry “C”
optimised for maximal material mixing

Direction of build/extrusion

🟩 Passive scalar concentration below 0.4

Figure 6.12: Flow patterns in static mixers for a ring material inhomogeneity.

6.3 Manufacturing and Practical Evaluation

The static mixer geometries before and after optimisation were manufactured to validate the simulation results. All geometries fulfilled the criteria for additive manufacturability. However, static mixer geometry ”C” was optimised for an inlet material inhomogeneity that could not be replicated in lab trials; therefore, it was omitted from manufacturing and practical evaluation.

The remaining geometries ”M”, ”P”, and ”T” were extracted from their IB representation in Paraview (Kitware, Inc., Clifton Park, NY, USA) and exported as STL files. Subsequently, they were post-processed in Meshmixer (Autodesk Inc., San Rafael, CA, USA), where the STL files were checked for errors and standardised in their resolution. Subsequently, the static mixer geometries were merged with an outer cylinder with a wall thickness of 1 mm, which was designed with notches to position the static mixer within the hollow cylinder that connects to the extruder (Figure 6.13). As a result, the static mixer could be reproducibly positioned in the centre of the melt flow channel.

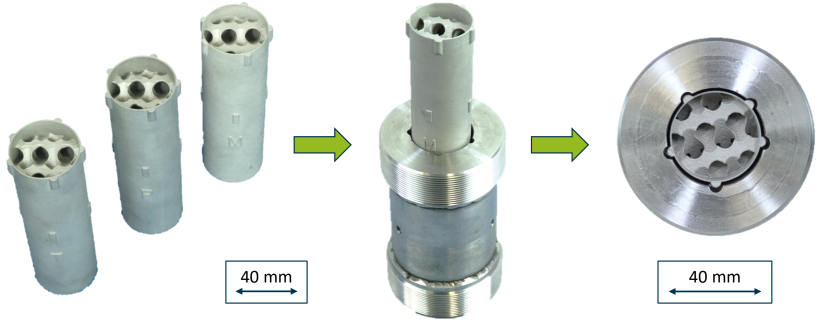


Figure 6.13: Additively manufactured static mixers (left) for validation trials, during which they were mounted in a hollow cylinder (right).

The static mixer geometries were additively manufactured through LPBF on an EOS M 290 machine (EOS GmbH, Krailling/Munich, DE) from tooling steel 1.2709 [NN20] with a layer thickness of 50 µm. After printing, the surfaces were bead-blasted during post-processing.

6.3.1 Lab Trial Setup

The setup for the validation trials was already published in [HSRF20, SSH22], the description here will focus on the most relevant aspects. Validation trials were performed using a 60 mm single-screw extruder (Oerlikon Barmag GmbH & Co. KG, Remscheid, DE). The extruded material was PE-HD type Hostalen GD 9550 F, the same material used during optimisation. Two different throughputs were investigated. A throughput of 20 kg/h was achieved by setting the extruder speed to 30 rpm, while the rotational speed of the extruder was set to 61 rpm for the high throughput of 40 kg/h. The material was processed at a nominal melt temperature of 220 °C, and the temperatures in the barrel zones were set up as a rising temperature profile (180 °C, 200 °C, 220 °C). The setup's other melt conveying parts had a nominal temperature of 220 °C controlled by heating bands. For the high throughput, this results in the inlet temperature profile used during optimisation. Both upstream and downstream of the static mixer, a pressure sensor and an immersed temperature sensor were positioned to measure both the static mixer's pressure drop and the radial temperature profiles before and after the mixer (Figure 6.14). The immersed temperature sensor measured the radial melt temperature at r = 0 mm, 7 mm and 14 mm, with r = 20 mm being the temperature at the barrel wall. Ten measurements were taken at each operating point, with a 5-minute difference between them. This minimises the influence of transient effects often associated with the extrusion process, which typically occur with either a higher or a lower frequency than five minutes [Sch23b]. For evaluation, the average value and the standard deviation of the ten measurements were taken.

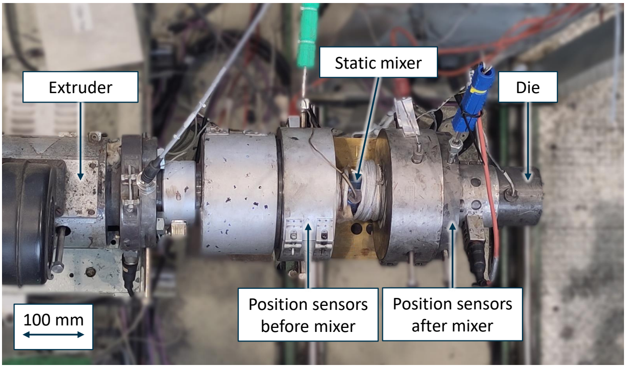


Figure 6.14: Extrusion line setup for practical evaluation.

6.3.2 Lab Trial Results

The complete data set from the validation trials is listed in Appendix A.4.

Figure 6.15 depicts the pressure drop for each geometry, both in the simulations (left) and in the lab trials (right). The difference between the simulated and measured pressure drop was around 29%, 38% and 32% for the geometry before optimisation ”M”, optimised for minimal pressure drop ”P”, and optimised for thermal mixing ”T”, respectively, independent of throughput. However, these results appear to contain a systematic error, where the absolute pressure drop values differ, while the differences between the geometries are comparable. In all cases, the improvement in pressure drop predicted by the simulation was either matched or underestimated. In the simulations, the optimisation for minimal pressure drop improved the pressure drop by 17-18%, depending on the throughput. In contrast, the pressure drop improvement in the lab trials was measured at 29% for the low throughput and 22% for the high throughput. A similar pattern emerged for the geometry ”T” that was optimised for thermal mixing, where the predicted pressure drop improvement was 5-6%, while the measured pressure drop improvement was 9-11%.

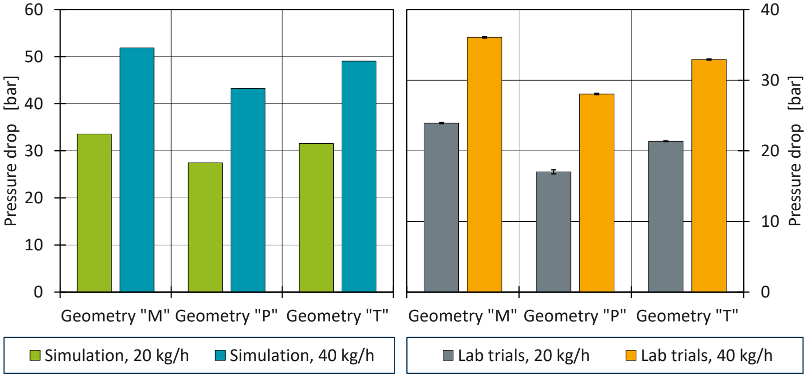


Figure 6.15: Comparison of pressure drop results obtained from simulations (left) and lab trials (right) for all investigated operating points.

Figure 6.16 depicts the temperature profiles from the lab trials and the corresponding temperature probes taken at the inlet and outlet in the simulations. At the high throughput, the static mixers resulted in cooling of the melt by 2-3 °C near the centreline (r = 0 mm) and by 6-7 °C at r = 14 mm. However, there was no significant difference between geometries ”M”, ”P” and ”T”. At the low throughput, no significant differences were found in the lab trial data between the three static mixer geometries. The simulations systematically overestimated the cooling of the melt within the mixer. The observed differences will be discussed in more detail in Section 6.4.

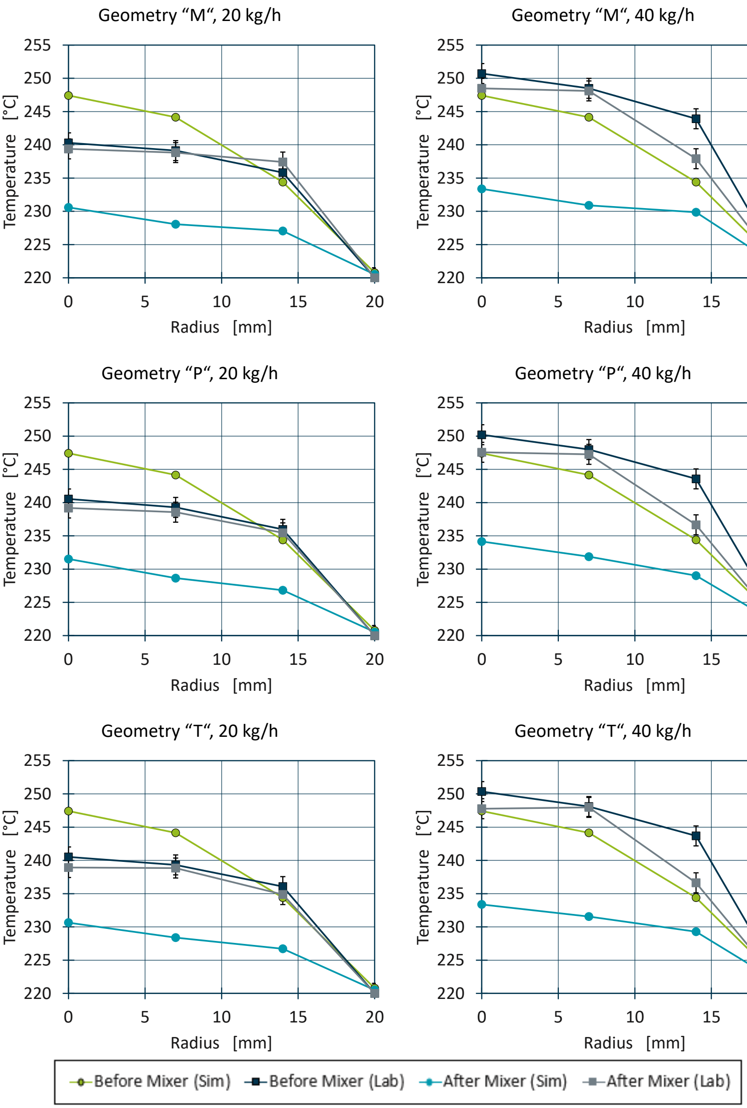


Figure 6.16: Temperature profiles from simulations and lab trials for all investigated operating points, where a radius of 0 mm refers to the centreline of the flow channel. Note that the error bars on the lab trial data refer to the uncertainties in the temperature sensors since they outweigh the standard deviations in the recorded data.

6.4 Discussion

This section first discusses the performance of the new optimisation algorithm, followed by a discussion of the results concerning the first three research hypotheses stated in Section 2.4.

The algorithm's run-time depends, of course, on the utilised hardware. As a result, a comparison to genetic algorithms, the most common alternative optimisation algorithm, is only possible in abstract terms. Each solver call/design point evaluation is in the order of magnitude of a few hours, depending on the solver and hardware. Genetic algorithms evaluate hundreds of design points in a process that is only partially parallelisable, thus resulting in hundreds of solver calls [Hee15, JSVS22]. In contrast, the presented adjoint optimisation algorithm requires at most the equivalent of five solver calls, where two account for the computation of the initial primal and adjoint CFD solution, two account for the solution of the primal and adjoint fields during optimisation and the remaining solver call accounts for the computation of a ”clean” CFD solution for the optimised geometry. In some cases, depending on the geometry, operating point and optimisation objective, the first two solver calls might not be required, reducing the number of solver calls to three.

Most genetic algorithms benefit from a reduced run-time through parallelisation, e.g., with MPI. The algorithm checking the AM constraints during optimisation requires detailed information regarding the mesh connectivity and current IB assignment of the surrounding cells. However, this information is lost during parallelisation with MPI. As a result, during optimisation, which accounts for two of those five solver calls, the algorithm cannot be parallelised at this point, increasing the overall run-time. For successful parallelisation, the implementation would require preserving information regarding the location and current IB assignment of all second-degree neighbour cells across processor boundaries, which is possible to implement but out of scope for this thesis. The solver calls that are only used for the computation of the initial primal and adjoint CFD solutions before or after optimisation are not affected by this and can be parallelised with MPI. The resulting speed-up factor varies based on the number of processes. However, even without parallelisation during optimisation, the adjoint optimisation algorithm would beat a genetic algorithm that benefits from parallelisation since parallelised genetic algorithms would need to cut down the computing time of hundreds of solver calls below the computing time of these two serial solver calls.

The optimisation algorithm does not require user interaction during run-time. However, the correct settings of the case and optimisation algorithm require some experience to dial in efficiently. The sensitivity response step sizes λ_u, λ_T and λ_C control the aggressiveness, i.e., the rate of change per iteration step in the optimisation and, therefore, need to be chosen carefully. While they need to fit their respective sensitivities, they must also balance the individual state variable sensitivities since their ratio can influence the overall optimisation result. The algorithm is highly sensitive to the primal and adjoint state variable fields' boundary conditions. In addition, the target values for the optimisation of thermal or material mixing must be chosen carefully. In most cases, the average field value at the outlet is a good starting point. However, the target temperature for thermal mixing can also be set to the minimum temperature at the outlet to encourage cooling alongside mixing, although it should not be set lower than that. The final challenge in handling the optimisation algorithm concerns the termination criterion. Since the optimisation algorithm does not converge in the conventional sense, the endpoint must be determined manually. In practice, this end can be identified through additional constraints that are not embedded in the algorithm, such as the mechanical stability of the optimised structure.

The first over-arching research hypothesis was

The design flexibility of additive manufacturing enables novel structures for polymer melt flow channels. At the same time, additional constraints in the adjoint topology optimisation algorithm ensure the manufacturability of the optimised geometries.

During optimisation, the algorithm created new structures with varying degrees of novelty. While the static mixer optimised for minimal pressure drop displays design features that could have been realised manually in a CAD software, the static mixers that were optimised for mixing capability have design features that are harder to represent in a CAD software and are unlikely to have been the result of manual optimisation. However, while these structures exploit the design flexibility of AM, they are probably also manufacturable by casting. The full extent of this design flexibility has been demonstrated (using an earlier version of the algorithm) in a previous publication in [SSH22], where the static mixers featured integrated oil channels. The geometric constraints included in the optimisation algorithm were not explicitly enforced as optimisation criteria but implicitly enforced by checking them whenever a cell's assignment was changed, potentially risking decisions by the algorithm that were only sensible on a local level. Yet, the algorithm still created geometries which required minimal manual post-processing before manufacturing.

The limitations of the DOF, mainly from the scaffold structure, also result in some weaknesses of this implementation for the geometric constraints. Previous investigations have shown that it can be beneficial to vary the overall length of the static mixer [HSB+20, Sin08]. However, the scaffold structure prohibits the variation of the overall mixer length in the current model. In addition, the scaffold structure limits the DOF in general since it enforces a certain base symmetry, number and location of mixing bars, preventing completely novel structures [PHL+24]. Despite these weaknesses, the algorithm could still optimise the geometry for its objective. Therefore, these current weaknesses only highlight the potential for further improvements in the algorithm. Consequently, the investigations regarding the additive manufacturability of the geometries obtained from the optimisation algorithm confirm the first research hypothesis.

The current implementation of the manufacturing constraints within the optimisation algorithm limits its versatility. However, implementations of geometric constraints for other manufacturing methods, such as milling, turning, or casting, are available in literature [VLL+16, WHXR23]. Due to the absence of a structural mechanics solver for the static mixer itself, the current implementation relies on the scaffold structure, user experience, and manual calculation after optimisation to ensure the mechanical stability of the optimised geometries.

The first research hypothesis specific to the optimisation of additively manufacturable static mixers was

Adjoint topology optimisation algorithms successfully optimise the flow channel geometry of static mixers for different optimisation objectives, where the different optimisation objectives are either to
(1) minimise pressure drop,
(2) maximise thermal mixing in addition to pressure drop reduction or
(3) maximise material mixing in addition to pressure drop reduction.

Each objective function led to different changes in the geometry of the static mixer, resulting in different effects on the pressure drop as well as the thermal and material distributive mixing. In general, a relationship exists between flow resistance and pressure drop in a pipe established by the Hagen-Poisseuille law of hydrodynamics [HM16]:

$$\Delta \mathrm{p} = \frac{8 \cdot \mathrm{L} \cdot \eta\left(\dot{\gamma}\right)}{\pi \cdot \mathrm{R}^4} \cdot \dot{\mathrm{V}}, \qquad (6.7)$$

where L and R denote the flow channel's length and the radius, respectively. Removing material from the static mixer geometry reduces the shear rate $\dot{\gamma}$, which increases the viscosity η. However, it also increases the radius R of the flow channel, which decreases the pressure drop by the fourth power. As a result, removing material from the static mixer geometry is generally expected to reduce the overall pressure drop. While that was observed for the most part, there was one notable exception: While the optimisation for material mixing removed the most material from the mixer, it did so in places with relatively little influence on the overall flow resistance and, thus, on the overall pressure drop. This resulted in the lowest pressure drop being observed with the optimisation for minimal pressure drop, while the static mixer optimised for material mixing had a reduced, but not minimal, pressure drop.

During optimisation for minimal pressure drop, most changes to the geometry occurred closer to the centreline of the flow channel, where the velocities and, therefore, contributors to the flow resistance are highest. In contrast, the changes during optimisation for thermal mixing were primarily in areas with cold spots near the cylinder wall, maximising convective transport there. Similarly, the changes that resulted from optimisation for material mixing maximised transverse flows in areas with high concentration gradients. Ultimately, all objective functions reduced the thickness of the mixing bars. This seems counter-intuitive since material mixing is traditionally associated with increasing the shear forces on the melt. However, the results in this thesis have shown that there are different ways to combat material inhomogeneities, provided the static mixer is optimised for the specific material inhomogeneity in question.

However, reducing the volume of the static mixer by up to 19% bears the risk of inhibiting the mechanical stability of the optimised static mixers. Conventionally, all melt conveying parts of extrusion lines, including mixers, are over-designed, incorporating high safety factors to account for pressure spikes during startup or sudden obstruction of the flow channel [HM16]. A simple structural analysis was performed using an approach similar to *Hopmann et al.* in [HGL+18]. Neglecting gravitational forces, the pressure forces F_p (in both radial and axial direction) and viscous forces F_Z were calculated on a single mixing bar using the pressures and wall shear stresses obtained with the higher throughput of 40 kg/h in the simulations. Since the mixing bars also support each other in a real scenario and the pressures were overestimated by roughly 30% in the simulations, this results in a reasonably conservative estimate. Before optimisation, the estimated maximal projection area for a 45° angle attack of a mixing bar with a square cross section, a thickness of 5 mm and a mixing bar length of 40 mm can be calculated as 282.8 mm^2 [HGL+18]. With a calculated pressure drop of 52 bar and a maximum wall shear stress of 0.196 MPa, the total force F_{total} on a single mixing bar is around 118 N. Assuming a central point of attack (Figure 6.17 left), the von Mises stress in the connection of the mixing bar and barrel is around 133 MPa. In the simulations, the static mixer that was optimised for minimal pressure drop had a

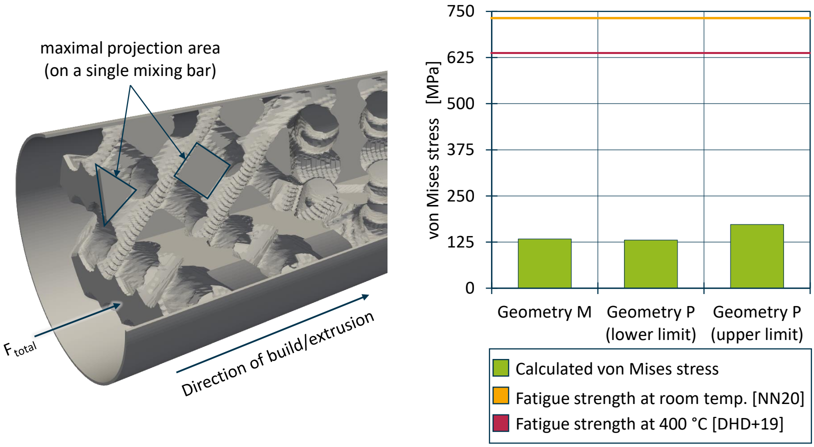


Figure 6.17: Left: Calculation of the maximal von Mises stress on a single mixing bar, right: structural analysis of the static mixer geometries regarding their fatigue strength.

pressure drop of 45 bar and a maximum wall shear stress of 0.125 MPa. The mixing bar thickness ranged from roughly 3 mm to 4 mm for this optimised geometry. Assuming a 4 mm mixing bar thickness, this results in a total force of 80 N and a von Mises stress of 130 MPa, while a 3 mm mixing bar thickness leads to a total force of 60 N and a von Mises stress of 173 MPa. The static mixer optimised for thermal mixing lies somewhere between those two geometries and pressure drops and was, therefore, not calculated explicitly. The fatigue strength of additively manufactured and heat-treated tooling steel 1.2709 is 732 MPa [NN20]. This value has been reported to decrease by 13% at 400 °C [DHD+19], giving an estimate for the lower limit (Figure 6.17 right). Therefore, the optimised static mixer geometry can be assumed to hold up during the expected operational loads. However, a more detailed analysis should be performed for operating points outside the investigated scope.

A reduction in flow resistance tends to result in reduced shear heating. Since all optimisation objectives reduced the flow resistance in some capacity, this can explain why all optimisation objectives improved thermal homogenisation to some degree. However, the thinner mixing bars of geometry ”P” reduced the heat exchange between the colder cylinder wall of the mixer and the melt, resulting in overall higher melt temperatures. The improvement in thermal mixing capability was very low at 3-4%, depending on the operating point. However, the initial mixing performance before optimisation was already quite good. This might imply that the initial geometry has already maximised its mixing capability for this type of thermal inlet inhomogeneity and that further improvement of the geometry is only possible for more severe or different types of inlet inhomogeneities [Imh04, Sch23a]. In contrast, the improvement in material homogenisation was much higher at 8%, although this was also the performance indicator where the highest operating point dependence was observed.

During the practical evaluation in lab trials, the measured pressure drop was around 29% to 38% lower than in the simulations. However, while the absolute pressure drop data did not match, the (relative) changes between the geometries confirmed the results obtained in the simulations. The differences in the data obtained from the immersed temperature sensors for the different static mixer geometries were within the sensor accuracy range and, thus, not significant. However, this is consistent with the findings in the simulations, where the temperature differences were also minimal. In previous investigations, the optimised static mixers had a more considerable impact on the thermal regime of the melt [SSH22]. This was not replicated in the lab trials detailed in Section 6.3. However, the investigations in [SSH22] were performed at a higher throughput of 80 kg/h, where the Graetz number implies dominant lengthwise convection and the melt temperature only depends on shear heating through pressure drop. Therefore, the differing impact on the thermal regime within the static mixers in [SSH22] does not contradict the findings in this thesis. Furthermore, it cannot be ruled out at this point that the temperature data were influenced by the melt film between the static mixer and hollow cylinder in the lab trial setup. This might also explain the differences in the temperature curves between the simulation and lab trials since the simulations assumed ideal heat transfer between the cylinder wall and the plastic melt. This would require further experimental investigations, e.g., through static mixers that were printed in one piece for direct installation at the extrusion line, similar to the setup used in [SSH22].

The optimised static mixer geometries obtained by the adjoint topology optimisation algorithm fulfil their respective optimisation objectives to at least some degree, confirming the first use case specific research hypothesis.

The second research hypothesis specific to the optimisation of additively manufacturable static mixers was

The adjoint topology optimisation algorithm is only operating point dependent for operating points with different thermal or material inhomogeneities and optimisation objectives regarding the thermal or material mixing.

All optimised geometries performed similarly concerning the pressure drop, thermal and material mixing independent of the throughput, confirming the research hypothesis and previous findings by *Zalc et al.* and *Li et al.* [LFC96, ZSMJ02]. Similarly, the optimised geometries still improved pressure drop and thermal and material mixing, no matter if the polymer material used during optimisation or different, confirming previous findings by *Liu et al.* [LHW06]. In contrast, the thermal or material inhomogeneity at the inlet significantly impacted the optimisation algorithm’s performance when optimising for thermal or material mixing. In some cases, a wrong choice for an inlet inhomogeneity worsened the algorithm’s performance. This suggests that the inlet inhomogeneity pattern is the only aspect of the operating point that should be chosen carefully for optimisation. However, the overall lowest operating point dependency was observed with the optimisation for minimal pressure drop. Geometry ”P” consistently had the lowest pressure drop and improved material mixing with little impact on thermal mixing compared to geometry ”M” before optimisation. The underlying X-mixer geometry that formed the basis for geometry ”M” is mainly known for its good mixing performance, with its biggest drawback being the high pressure drop. Therefore, the optimisation objective to minimise pressure drop is the safest and recommended option if the inlet inhomogeneity is unknown since it improves the mixer’s drawbacks without compromising its strengths. However, that does not mean that this is the only viable option for OP specific static mixer optimisation. Consequently, the investigations regarding the operating point dependency of the optimisation algorithm confirm the second use case specific research hypothesis.

In conclusion, the newly developed adjoint TO algorithm for additively manufacturable flow channels worked as intended for the chosen narrow use case. Based on these results, the next section will demonstrate it on a second use case that is less defined in its DOF.

7 Application to Profile Extrusion Dies

This chapter presents and discusses the application of the developed adjoint topology optimisation algorithm to a simple L-profile extrusion die. First, the extrusion die before optimisation is described, and the optimisation objectives, along with the setup of the simulation, are discussed before the optimisation results are presented. The chapter closes with a discussion of the findings.

The investigations were conducted on a flow channel of an extrusion die for an L-shaped profile (Figure 7.1) known to be manufacturable using AM [HWK+14, Win15].

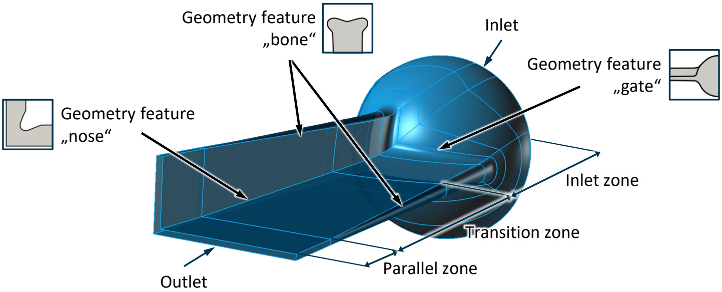


Figure 7.1: Flow channel of the L-profile die before optimisation (adapted from [HWK+14, YHW+16]).

The flow channel is divided into three zones. The inlet zone predistributes the cylindrical melt stream from the extruder, while the transition zone forms the profile. At the end of the transition zone, the final extrudate profile geometry is reached, and the constant parallel zone reduces die swell from stored elongational energy. This particular flow channel design exhibits several design features, which have been described in detail in [Win15]. At the end of the inlet zone, the *gate* defines an inlet angle for each leg of the geometry, optimising the melt predistribution. The end pieces of each leg are enlarged to increase the influx in those regions prone to under-feeding, resulting in the design feature *bone*. Finally, the design feature *nose* denotes a constriction at the interface between the two legs, increasing the resistance to transverse flows and decreasing the flow velocity.

Figure 7.2 depicts the individual dimensions of the flow channel. The build direction for AM was defined against the direction of extrusion since this maximises the layer support during printing for dies with convergent flow channels.

Two optimisation objectives were investigated for the optimisation of the L-profile extrusion die: The first was minimising the pressure drop in the flow channel, while the second was maximising the flow balance, i.e., homogeneous velocity distribution at the die outlet.

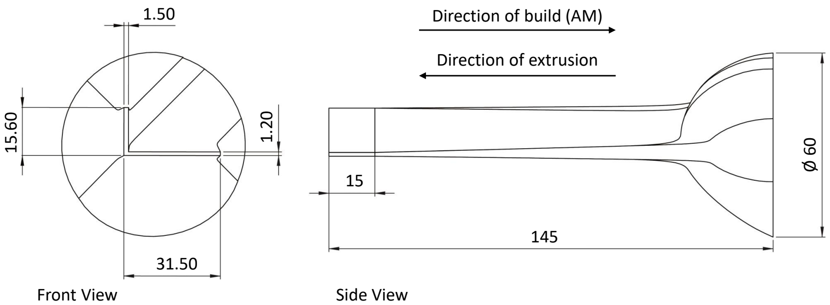


Figure 7.2: Dimensions of the die for a L-shaped profile (adapted from [Win15]).

7.1 Case Setup and Optimisation Objectives

The simulations were performed on a cylinder with a diameter of 60 mm and a length of 144 mm, again using 1 mm of the overall length as an overhang for *setFields*. The computational mesh was created with *blockMesh* and consisted of 6.765 million hexahedral cells. In the simulations, the flow channel geometry was virtually imposed on the computational mesh using the immersed boundary method (Figure 7.3).

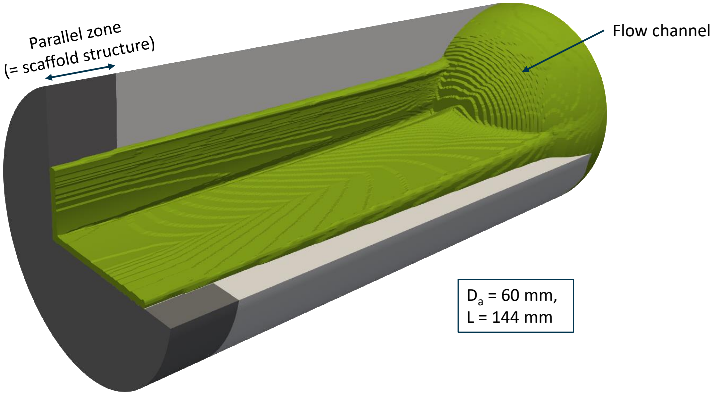


Figure 7.3: Position of the initial flow channel (depicted in green) within the cylinder before optimisation, the fixed parallel zone is depicted in grey.

Since the die's parallel zone serves the relaxation of the melt, it should not be modified and was, therefore, fixed using the scaffold structure implementation. The optimisation objective should be applied only to the outlet of the flow channel, not to the outlet face of the extrusion die since all flows are prohibited there, and this might trigger unwanted changes upstream of the parallel zone. This was achieved by splitting the outlet face into two patches using *topoSet* and *createPatch*. The new patch at the outlet behaved like the outside wall and is listed as part of the wall patch when discussing the boundary conditions below. During initialisation, the geometry of the profile extrusion die and the scaffold structure were imposed on the mesh using *setFields*.

The operating point was chosen to reflect the works by *Windeck* [Win15], who used the same flow channel geometry. Therefore, the plastic melt was extruded with a throughput of 6 kg/h at a nominal processing temperature of 200 °C.

The plastic melt in this setup was Moplen HP400H (LyondellBasell GmbH, Wesseling, DE), which is a Polypropylene (PP). Its material data are listed in Table 7.1.

Parameter	**Unit**	**Value**
Carreau parameter $A_{Carreau}$	m^2/s	14.752
Carreau parameter $B_{Carreau}$	s	0.8798
Carreau parameter $C_{Carreau}$	-	0.6238
Standard temperature T_s	K	263
Reference temperature T_m	K	473
Melt density ρ	kg/m^3	750

Table 7.1: Material model parameters for PP [Win15].

The complete set of initial and boundary conditions for the state variables is listed in Table 7.2.

State variable	**Inlet**	**Wall**	**Outlet**	**Initial field**
u	flowRateInletVelocity $2.22 \cdot 10^{-6}\ m^3/s$	noSlip	zeroGradient	$(0\ 0\ 0)\ \frac{m}{s}$
p	zeroGradient	zeroGradient	fixedValue $0\ m^2/s^2$	$0\ \frac{m^2}{s^2}$
T	fixedValue 473.15 K	zeroGradient	zeroGradient	473.15 K
$\mathbf{u}_a$	fixedValue (-1 0 0) m/s	noSlip	adjointOutletVelocity (0 0 0) m/s	$(0\ 0\ 0)\ \frac{m}{s}$
p_a	zeroGradient	zeroGradient	adjointOutletPressure $0\ m^2/s^2$	$0\ \frac{m^2}{s^2}$

Table 7.2: Initial and boundary conditions for optimisation of profile extrusion die.

At the specified operating point, the initial flow channel had an average velocity of 25.98 mm/s, with velocities ranging from 0.99 mms/s to 55.66 mm/s. The initial flow distribution (Figure 7.4) favoured the short leg, while the outer part of the long leg was under-fed. The area with the second-highest flow velocities was at the intersection of both legs.

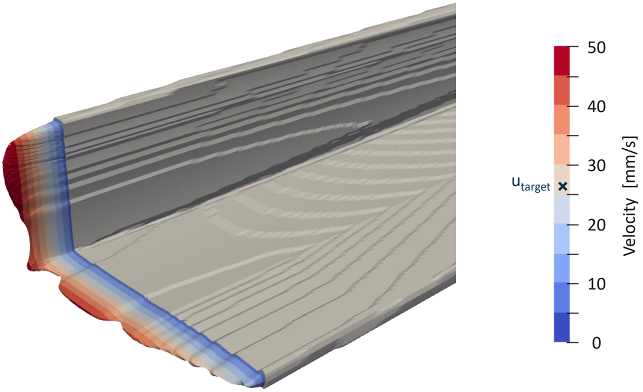


Figure 7.4: Flow balance of the original flow channel before optimisation.

To quantify the flow balance, the flow balance ratio FB (Equation 7.1) is defined as the ratio of the average velocities in the long leg and short leg of the profile. FB = 1 indicates a perfect flow balance, while FB < 1 means the short leg is over-fed, and FB > 1 means the long leg is over-fed.

$$\mathrm{FB} = \frac{\mathbf{u}_{\text{ave,long leg}}}{\mathbf{u}_{\text{ave,short leg}}} \tag{7.1}$$

With an average velocity in the short leg of 29.53 mm/s and an average velocity in the long leg of 23.93 mm/s, the initial flow balance FB is 0.81.

During optimisation for flow balance, the target velocity $\mathbf{u}_{\text{target}}$ was set to (26 0 0) mm/s, corresponding to the average velocity at the outlet, with the goal of reducing the overall variance in the velocity distribution.

7.2 Optimisation Results

This section presents the results by visualising the changes made to the flow channel geometry. For manufacturing, these changes are inverted and applied to the corresponding extrusion die. The visualisation of the flow channels before and after optimisation is followed by a listing and description of the flow channels' performance characteristics. An in-depth interpretation and discussion of the observed results will follow in Section 7.3.

Figure 7.5 top depicts the changes to the flow channel made by the algorithm during optimisation for minimal pressure drop, where the orange areas (if present) represent areas where the flow channel has been narrowed, and the blue areas represent where the flow channel has been widened, while the geometry remains unchanged in the grey area. A different way to look at the performance of the algorithm is depicted in Figure 7.5 at the bottom, where all blue surfaces show candidate areas for narrowing the flow channel, while red surfaces show candidate areas for widening the flow channel, not considering manufacturability. Both representations show that the optimisation for minimal pressure drop mainly resulted in a widening of the melt flow channel in both legs across the transition zone. Moreover, the flow channel was widened at the intersection of both legs, removing the design feature *nose*. However, the inlet zone and early part of the transition zone, which are governed by the design feature *gate*, were mostly unchanged. Another way of phrasing this observation is that the algorithm primarily affected areas with high shear rates/ shear stresses, which are down-stream closer to the die exit.

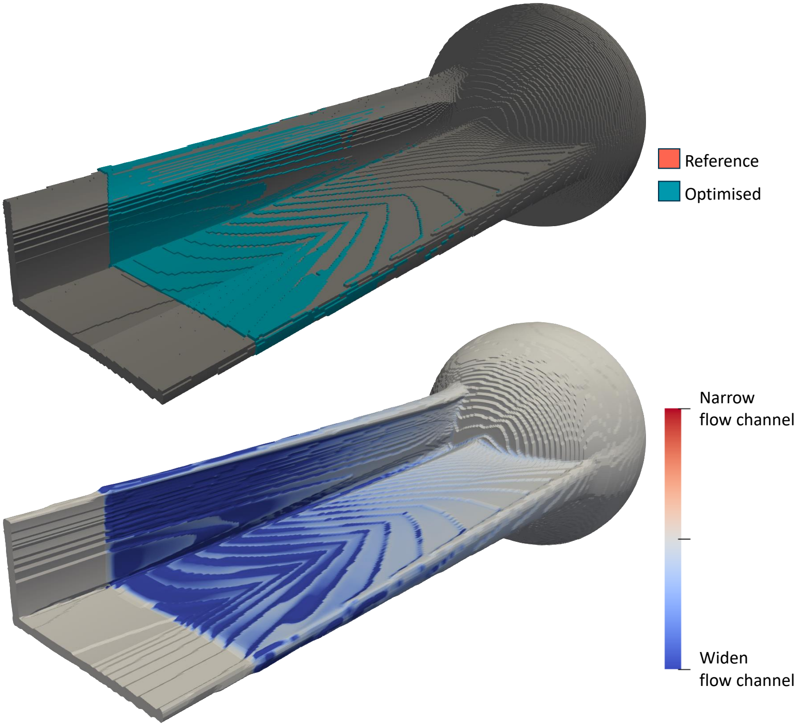


Figure 7.5: Changes to the flow channel geometry after optimisation for minimal pressure drop.

The changes to the geometry of the flow channel during optimisation for flow balance are depicted in Figure 7.6. Again, the design feature *gate* at the early transition zone was not adjusted. However, the optimisation objective for flow balance affected both the design features *nose* and *bone*. After optimisation, the *nose* was more pronounced, partly reaching into the short leg (red spot in Figure 7.6 bottom). In addition, the *bone* at the short leg was extended near the outlet. In contrast, the whole flow channel as well as the *bone* were widened at the long leg, corresponding to the areas with low flow velocities before optimisation.

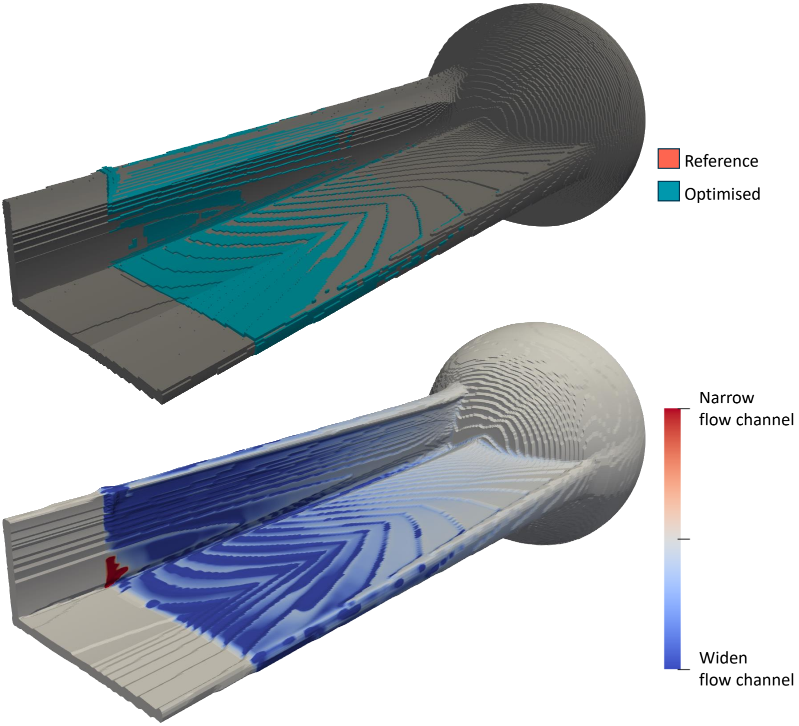


Figure 7.6: Changes to the flow channel geometry after optimisation for flow balance.

In the following, the L-Profile flow channel before optimisation will be referred to as *Geometry ”L”*, while the flow channels optimised for minimal pressure drop and maximal flow balance will be referred to as *Geometry ”P”* and *Geometry ”F”*, respectively.

The pressure drop and the flow balance were evaluated at cut planes positioned 1 mm behind the inlet and 1 mm before the outlet of the computational domain.

Table 7.3 lists the die’s steel volume, the pressure drop, the average velocity at the outlet and in each leg, as well as the flow balance ratio FB for all three flow channel geometries.

	V **[mm^3]**	**Δp** **[bar]**	**u_{ave}** **[mm/s]**	**$u_{ave, short leg}$** **[mm/s]**	**$u_{ave, long leg}$** **[mm/s]**	**FB** **[-]**
Geometry ”L”	334,870	64.99	25.98	29.53	23.93	0.81
Geometry ”P”	333,714	58.74 (-10%)	25.98	31.08	22.99	0.74 (-9%)
Geometry ”F”	333,809	60.97 (-6%)	25.86	27.79	24.76	0.89 (+10%)

Table 7.3: Performance characteristics for the flow channels before and after optimisation.

The steel volume of the die did not change significantly during optimisation. The initial flow channel geometry ”L” resulted in a pressure drop of 64.99 bar and an average velocity of 25.98 mm/s with a flow balance ratio of 0.81. Optimisation for minimal pressure drop (geometry ”P”) reduced the pressure drop by 10% to 58.74 bar but decreased the flow balance ratio by 9%. In contrast, optimisation for flow balance with $\mathbf{u}_{target} = \mathbf{u}_{ave}$ (geometry ”F”) reduced the pressure drop by 6% to 60.97 bar and the average velocity at the outlet to 25.86 mm/s, while improving the flow balance ratio by 10%. This pressure drop reduction can be explained since widening the flow channel near the design feature *bone* outweighs the constriction near the design feature *nose*.

The changes in the flow balance are also visualised in Figure 7.7. The original flow channel design (Figure 7.7 left) caused an imbalance between the legs of the profile by over-supplying the short leg and under-supplying the long leg, especially near the outside end of the legs. Optimisation for minimal pressure drop worsened this imbalance since the short leg is also wider and, thus, has a lower flow resistance (Figure 7.7 middle). In contrast, optimisation for flow balance reduced the maximum flow velocity in the short leg and increased melt supply in the outer end of the long leg (Figure 7.7 right).

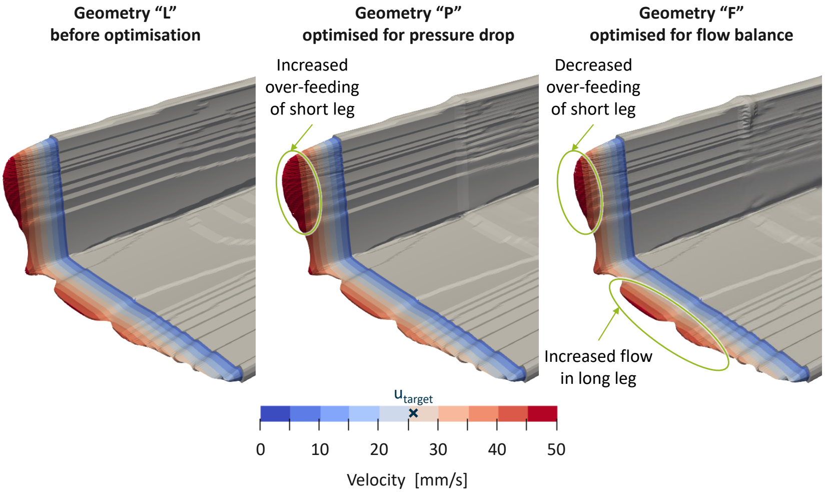

Geometry "L"
before optimisation
Geometry "P"
optimised for pressure drop
Geometry "F"
optimised for flow balance
Increased over-feeding of short leg
Decreased over-feeding of short leg
Increased flow in long leg
u_{target}
0
10
20
30
40
50
Velocity [mm/s]

Figure 7.7: Velocity distributions obtained with the flow channels before and after optimisation.

7.3 Discussion

This section first discusses the differences in the algorithm's performance between this use case and the static mixer use case before addressing the research hypotheses posed at the beginning of the thesis.

Initially, optimisations were performed for the cost functions as described in the previous chapter. However, the resulting geometries were identical for all optimisation objectives, although the values of their cost functions differed considerably. As a solution, the cost functions were artificially enhanced before their addition to the adjoint boundary conditions at the outlet. This artificial Cost Function Inflation (CFI) factor had to be manually calibrated. An excessively low CFI led to identical results independent of the cost function, while a too high CFI broke the optimisation algorithm. This is illustrated in Figure 7.8, where the results of an optimisation run for flow balance with a CFI of $5 \cdot 10^7$ are visualised. Here, the algorithm was too aggressive and blocked the flow completely in areas with $\mathbf{u} > \mathbf{u}_{ave}$. Eventually, the cost function inflation was calibrated to $2 \cdot 10^7$, and all results presented in Section 7.2 were obtained with this setting.

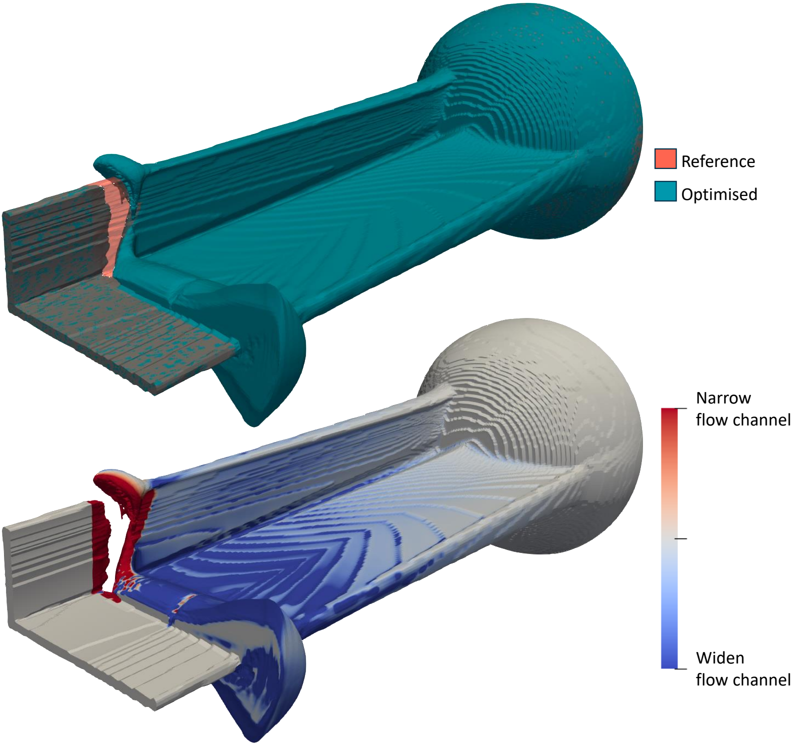


Figure 7.8: Optimisation result obtained with an overestimated cost function inflation.

One explanation for this behaviour may be found in the nature of polymer melt flows: The implementation of the immersed boundary method already required an artificial inflation of $\alpha_{\text{solid}} = \alpha_{\text{max}} = 2 \cdot 10^9$ to block the polymer melt flow (Section 3.1). A similar effect might be happening here, where the other terms of the adjoint outlet boundary condition dominate, effectively suppressing the cost function's influence on the sensitivity. This was not observed during the optimisation of the static mixer in Section 6 since the optimisation objectives influenced different state variable sensitivities, i.e., optimisation for minimal pressure drop affected the velocity sensitivity, while the optimisations for thermal and material mixing affected the temperature and concentration sensitivity, respectively. In contrast, the optimisations in this chapter both affect the velocity sensitivity $\sigma_{\mathbf{u}}$.

Future investigations should focus on this phenomenon and determine whether the cause for the inflation of α_{solid} in the immersed boundary representation is the same cause for the CFI. Ideally, one can derive a fixed inflation ratio for both phenomena.

The research hypothesis specific to the optimisation of additively manufacturable profile extrusion dies was

For certain scenarios, adjoint topology optimisation algorithms are a method with low computational complexity to optimise a profile extrusion die's flow channel geometry for specific optimisation objectives, such as minimal pressure drop or maximal flow balance.

For the reasons already stated in Section 6.4, the algorithm performed the optimisation at a relatively low computational cost.

Previous optimisation algorithms were based on CFD simulations that suggested a higher flow imbalance even with the same geometry and operating point [HWK+14, Win15]. Therefore, a quantitative comparison of the results obtained by *Windeck* with the results obtained in this thesis is not feasible.

When *Windeck* investigated the influence of the design features in this L-profile geometry, he found the following trends to achieve flow balance and minimal pressure drop [Win15]: The design feature *gate* was most relevant for the flow resistance, i.e., the pressure drop. He found that less aggressive convergence of the flow channel was the best strategy to reduce flow resistance. *Windeck* found that the design feature *bone* was only relevant for the final distribution of the melt and that widening it near the outlet can be beneficial for the melt distribution. The design feature *nose* was primarily relevant to reducing transverse flows, thus being most influential regarding the flow balance of the geometry.

During optimisation, the optimisation objectives for minimal pressure drop and maximal flow balance were investigated in isolation. After calibration of the CFI factor, both objective functions lead to the desired changes in the geometry. However, optimising for minimal pressure drop came at the expense of reducing the outlet's flow balance. This can be explained by the fact that many design features are introduced to improve the flow balance but happen to increase the flow resistance as a side effect. In contrast, the optimisation for maximal flow balance improved the overall flow balance at the outlet while reducing the pressure drop as well. In general, two approaches can be used to improve the flow balance:

1. Increase the flow resistance (and thus pressure) locally in extrusion direction to force redirection of flow in transverse direction.
2. Reduce the flow resistance in transverse direction while keeping the flow resistance in extrusion direction constant.

The optimisation results produced by this algorithm favour option 2, resulting in reduced shear rates within the transverse flow and even reducing the pressure drop as a side effect.

These observed effects of the cost functions on the optimisation objectives can be expressed in terms of the design feature changes made by the optimisation algorithm. The differences between the changes made by the two objective functions are highlighted in Figure 7.9, where the changes made during the optimisation for flow balance are highlighted in blue, the changes made during the optimisation for minimal pressure drop are highlighted in orange and the coinciding areas remain grey.

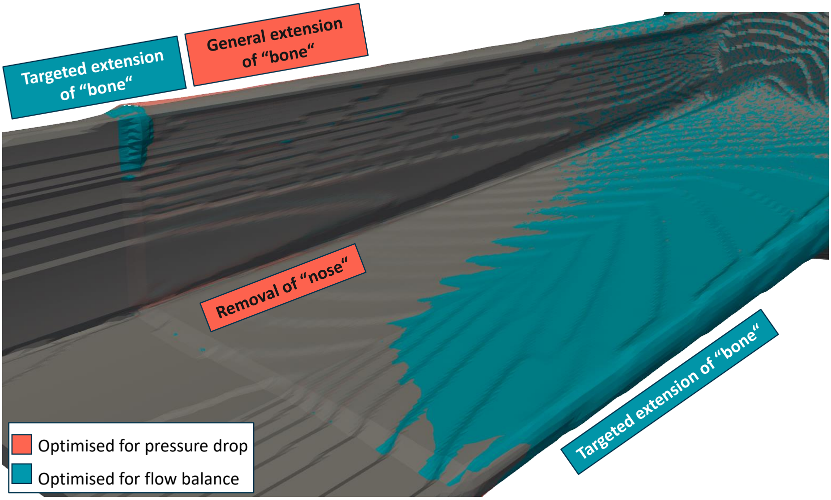

Targeted extension of "bone"
General extension of "bone"
Removal of "nose"
Targeted extension of "bone"
🟥 Optimised for pressure drop
🟦 Optimised for flow balance

Figure 7.9: Difference in changes caused by the optimisation objectives.

This algorithm confirmed the proposed change for the design feature *gate* to delay the flow channel's convergence, independent of the objective function. The relevance of the design feature *bone* for the final distribution of the melt was also observed in this study during the optimisation for flow balance. In contrast, the design feature *bone* was also widened to a lesser extent during optimisation for minimal pressure drop. Still, in this case, the changes occurred across the whole transition zone to reduce the overall flow resistance. Since the algorithm did not consider transverse flows during optimisation for minimal pressure drop, the design feature *nose* was only preserved during optimisation for flow balance.

In general, *Windeck* observed that deformations of the flow channel, which were far from the outlet, only had a small influence on the minimisation of the cost function. The same effect was observed in this algorithm, where most changes occurred in the parts near the outlet which were not fixed by the scaffold structure. Consequently, the results obtained in this thesis confirm the use case specific research hypothesis.

The first over-arching research hypothesis for both use cases was

The design flexibility of additive manufacturing enables novel structures for polymer melt flow channels. At the same time, additional constraints in the adjoint topology optimisation algorithm ensure the manufacturability of the optimised geometries.

Although both objective functions were able to improve the flow channel for their respective purpose, the overall improvement was limited. One reason for this might lie in the DOF given to the optimisation algorithm. While the parallel zone near the die outlet was fixed with a scaffold structure to control for die swell effects, this also limited the DOF for pressure drop and flow balance optimisation potential. This ties in with the findings by *Lieber et al.*, who investigated the potential of AM for a U-profile extrusion die [LVTT23]. They limited their DOF to streamlining the flow channel in the transition zone and adding fillets to the design feature *nose* in the corners of the profile. These changes improved the thermal homogeneity of the extrudate at the die outlet by reducing shear heating. However, the streamlined transition zone reduced the cross section of the flow channel early, increasing pressure drop and reducing the overall flow balance. They concluded that while AM enables manufacturing of a streamlined die out of one piece, the benefits were outweighed by the drawbacks of AM. Another reason for the limited improvement might lie in the use of a well-designed initial flow channel geometry. The case studies by *Lieber et al.* as well as in this thesis were based on flow channel geometries that had already been manually optimised, leaving relatively few viable DOF.

The results in this thesis demonstrated an improvement in the pressure drop and flow balance of the optimised flow channel geometries, and the resulting flow channel geometries were additively manufacturable, thus confirming the research question addressed in both use cases. Still, the findings of both profile extrusion die case studies imply that simple geometries that were already designed for conventional manufacturing with few remaining DOF, like the L-profile, may not be the ideal use case to investigate the full potential of the optimisation algorithm.

8 Conclusion and Outlook

The goal of this thesis was to investigate the potential of the adjoint topology optimisation method to optimise the flow channel geometry in extrusion dies and static mixing elements, exploiting the low computational complexity and high degrees of freedom enabled by the optimisation algorithm and design flexibility of additive manufacturing.

An algorithm was developed to perform adjoint topology optimisation in the open source software OpenFOAM. The algorithm was extended by a routine that checked additional constraints ensuring additive manufacturability on non-cartesian meshes. It used an immersed boundary representation of the structure within the flow channel to enable quick adaptation of the geometry. Different, in part newly included, optimisation objectives facilitated the targeted optimisation for either minimal pressure drop or maximal flow balance, maximal thermal mixing, or material mixing. The algorithm was demonstrated in two use cases. In the first use case, a static mixer was optimised for minimal pressure drop, maximal thermal mixing, and material mixing. The operating point dependency of the optimisation algorithm was investigated by evaluating the performance of the optimised static mixers at new off-design operating points. The best candidate static mixers were manufactured and evaluated in lab trials. In the second use case, a simple L-profile extrusion die flow channel was optimised for either minimal pressure drop or maximal flow balance.

In each case, the algorithm improved the flow channels' functionality for their optimisation objective. All optimised geometries fulfilled the manufacturing constraints for AM and required minimal post-processing for manufacturing. Comparing the results from the optimisation runs for minimal pressure drop and maximal flow balance of the L-profile, it was observed that cost functions applied to the boundary conditions of the same state variable need to be artificially inflated to impact the sensitivity for the design update significantly.

In the static mixer use case, individualised static mixers were obtained for the specific operating point, inlet inhomogeneity and optimisation objective. All optimisation objectives achieved a pressure drop reduction of up to 20%, which was qualitatively confirmed and even exceeded in lab trials. The slight improvement in thermal homogeneity in the simulations was not significant in lab trials.

During the investigation into the operating point dependency, the algorithm was found to be relatively insensitive to throughput and polymer material. This is desirable since these parameters of the OP are changed more frequently in an existing extrusion line than the thermal or material inhomogeneity pattern, which is usually fundamentally linked to the machine configuration and the extruded material. Optimisation for minimal pressure drop displayed the lowest operating point dependency. Furthermore, in most cases where the inlet inhomogeneity differed from the original operating point, the static mixer that was optimised for minimal pressure drop prevailed as the optimal static mixer. Consequently, the research hypotheses regarding the suitability of an adjoint topology optimisation algorithm for additively manufacturable static mixers as well as the algorithm's operating point dependency were confirmed. Based on the results, it is generally recommended to focus on the optimisation objective for minimal pressure drop, especially if the inlet inhomogeneity is unknown.

In the extrusion die use case, both optimisation objectives reduced the pressure drop by up to 10%, but only the optimisation for flow balance improved the flow balance, resulting in a pressure drop improvement by 6% and a flow balance improvement by 10%. The results confirm the research hypothesis that the newly developed algorithm can also be applied to profile extrusion dies. However, these results were obtained using a pre-designed initial flow channel and are still in early stages.

Recommendations for future work can be divided into two aspects. The first concerns future research questions regarding the nature of adjoint topology optimisation algorithms for polymer melt flow channel design applications.

Further research into adjoint topology optimisation for plastics processing applications should include a closer investigation of the cost function inflation phenomenon. The results in this thesis suggest a connection between the cost function inflation and the already established inflation of the immersed boundary representation for polymer melt flows as a viable research hypothesis.

Both use cases in this thesis perform optimisation of a geometry that has already been designed and optimised using other methods. Future research should consider the minimum initial geometry required for a successful optimisation. This also includes a formal investigation into whether the initial geometry has to already meet the criteria regarding manufacturability or whether the algorithm is also suitable for making a non-conforming geometry manufacturable.

The second aspect of future work based on this thesis concerns the recommended next features required to make this algorithm commercially viable.

For economic considerations, the geometric constraints algorithm must be modified such that it can be run in parallel using the OpenFOAM-native MPI.

Coupling the algorithm with a structural mechanics solver could ensure mechanical stability at expected operating points during run-time. This would also reduce the need for the scaffold structure, which was included to ensure a minimum wall thickness. The scaffold structure's purpose would be reduced to ensuring that the optimisation algorithm does not remove the entire static mixer to minimise the flow resistance at all cost. This could be circumvented using multiple approaches. One solution might be to move the entire scaffold structure to enforce minimum wall thickness. The other solution involves avoiding a unidirectional optimisation for minimal pressure drop and running the optimisation algorithm using a combination of optimisation objectives. This would result in the mixing capability being automatically considered alongside the pressure drop, although this increases the algorithm's operating point dependency. Extending the optimisation algorithm with geometric constraints of other manufacturing methods, such as milling, turning, or casting, could improve its versatility.

The optimisation objectives in this thesis were limited to well-documented objective functions. However, the complex profile extrusion die design requires the conformation of many more criteria. Optimisation objectives for the mechanical design could be embedded in a structural mechanics solver. Similarly, this thesis does not fully exploit the possible optimisation objectives for extrusion dies' rheological and thermal design. This includes, but is not limited to, optimising for a short residence time and narrow residence time spectrum, specified shear rates and shear stresses for the polymer melt, controlled die swell, and reducing the thermal mass of the extrusion die.

The optimisation algorithm in this thesis utilised volume sensitivities to directly affect the flow channel geometry. However, an alternative implementation using surface sensitivities which indicate the preferred direction of change at the structure's surface may open the algorithm's applicability to new use cases that cannot exploit the full DOF. For instance, the algorithm could be used to design the metering zones of screws in single-screw extruders. An alternative application could be in optimising flat film or spiral mandrel extrusion dies where the die has already been manufactured but still has DOF through the configuration of an adjustable die gap.

9 Zusammenfassung and Summary

9.1 Zusammenfassung

Die Auslegung von Fließkanälen für Polymerschmelzen in Extrusionsanlagen stellt weiterhin eine Herausforderung dar, die zu einem verstärkten Einsatz von Optimierungsalgorithmen führt. Die additive Fertigung erhöht die Freiheitsgrade bei der Gestaltung von Fließkanälen, doch müssen die damit verbundenen Fertigungsrestriktionen berücksichtigt werden. Ziel dieser Arbeit ist die Untersuchung des Potenzials der adjungierten Topologieoptimierung für die Gestaltung von Fließkanälen für Kunststoffschmelzen im Hinblick auf spezifische Optimierungsziele bei gleichzeitiger Gewährleistung der additiven Fertigbarkeit. Dazu gehört auch eine Untersuchung der Betriebspunktabhängigkeit des Optimierungsalgorithmus.

Es wird ein Algorithmus entwickelt, der eine adjungierte Topologieoptimierung in OpenFOAM (OpenFOAM Foundation Ltd., London, UK) durchführt. Er nutzt eine Immersed Boundary Methode, um eine schnelle Anpassung der Geometrie zu ermöglichen. In den Algorithmus werden zusätzliche Randbedingungen, die die additive Fertigbarkeit auf nicht-kartesischen Rechengittern gewährleisten, implementiert. Verschiedene Optimierungsziele ermöglichen die gezielte Optimierung bezüglich Druckverlust, rheologischer Balanciertheit, thermischer oder stofflicher Mischwirkung. Der Algorithmus wird in zwei Anwendungsfällen demonstriert. Im ersten Anwendungsfall wird ein statischer Mischer für minimalen Druckverlust, maximale thermische sowie stoffliche Mischwirkung optimiert. Die Betriebspunktabhängigkeit des Optimierungsalgorithmus wird untersucht, indem die Leistung der optimierten statischen Mischer für Betriebspunkte außerhalb der Auslegung bewertet wird. Die vielversprechendsten statischen Mischer werden gefertigt und praktisch erprobt. Im zweiten Anwendungsfall wird der Fließkanal eines L-Profil-Werkzeugs auf minimalen Druckverlust sowie auf maximale rheologische Balanciertheit optimiert.

Der Algorithmus verbessert die Funktionalität der Fließkanäle in Bezug auf ihr Optimierungsziel und stellt gleichzeitig die additive Fertigbarkeit sicher. Im Anwendungsfall des statischen Mischers wird bei allen Optimierungszielen eine Reduktion des Druckverlusts erreicht, was in Laborversuchen qualitativ bestätigt wird. Die Optimierungsergebnisse sind hinsichtlich des Durchsatzes und Kunststoffs betriebspunktunabhängig, jedoch stark abhängig von der ursprünglichen thermischen und stofflichen Inhomogenität. Die Optimierung für minimalen Druckverlust zeigt die geringste Betriebspunktabhängigkeit. Im Anwendungsfall des L-Profil-Werkzeugs verringern beide Optimierungsziele den Druckverlust, aber nur die Optimierung für homogene Geschwindigkeitsverteilung verbessert die rheologische Balanciertheit.

Diese Forschungsarbeit zeigt das Optimierungspotenzial für individualisierte Mischer auf und demonstriert das Potenzial der adjungierten Topologieoptimierung als Methode für die rheologische Auslegung komplexer Profilextrusionswerkzeuge.

9.2 Summary

The design of polymer melt flow channels in extrusion lines remains challenging, leading to an increased use of optimisation algorithms. Additive manufacturing increases the degrees of freedom in flow channel design, but its manufacturing constraints must be considered. This thesis aims to investigate the potential of adjoint topology optimisation for designing polymer melt flow channels for specific optimisation objectives while ensuring additive manufacturability. This includes an investigation of the operating point dependency of the optimisation algorithm.

An algorithm is developed to perform adjoint topology optimisation in OpenFOAM (OpenFOAM Foundation Ltd., London, UK). It uses an immersed boundary representation to enable quick adaptation of the geometry. The algorithm is extended with additional constraints ensuring additive manufacturability on non-cartesian meshes. Different optimisation objectives facilitate the targeted optimisation for minimal pressure drop or maximal flow balance, thermal mixing, or material mixing. The algorithm is demonstrated in two use cases. In the first use case, a static mixer is optimised for minimal pressure drop, maximal thermal mixing, and material mixing. The operating point dependency of the optimisation algorithm is investigated by evaluating the performance of the static mixers before and after optimisation for off-design operating points. The best candidate static mixers are manufactured and evaluated in lab trials. In the second use case, a simple L-profile extrusion die flow channel is optimised for either minimal pressure drop or maximal flow balance.

The algorithm improves the flow channels' functionality with respect to their optimisation objective while ensuring suitability for additive manufacturing. In the static mixer use case, all optimisation objectives achieve a pressure drop reduction, which is confirmed qualitatively in lab trials. The optimisation results are independent of the operating point regarding the throughput and polymer material but highly dependent on the inlet inhomogeneity. Optimisation for minimal pressure drop is the most reliable independent of the operating point. In the extrusion die use case, both optimisation objectives reduce the pressure drop, but only the optimisation for flow balance improves flow balance.

This research highlights the optimisation potential for individualised mixing elements and demonstrates the potential of adjoint topology optimisation as a method for designing the flow channels of complex profile extrusion dies.

Abbreviations

Abbreviation	**Meaning**
ALE	Arbitrary Lagrangian-Eulerian
AM	Additive Manufacturing
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFI	Cost Function Inflation
DOF	Degrees of Freedom
EMUM	Elastic Mesh Update Method
FVM	Finite Volume Method
IB	Immersed Boundary
LLPD	Low Low Pressure Drop
LPBF	Laser Powder Bed Fusion
LPD	Low Pressure Drop
MFR	Melt Flow Rate
MPI	Message Passing Interface
NSE	Navier-Stokes Equations
NURBS	Non-Uniform Rational B-Splines
OP	Operating Point
PDE	Partial Differential Equation
PE-HD	Polyethylene of High Density
PP	Polypropylene
PS	Polystyrene
SIMP	Solid Isotropic Material with Penalty
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
SLM	Selective Laser Melting
SMX	Static Mixers using cross bars X
SO	Shape Optimisation
TO	Topology Optimisation

Abbreviation	**Meaning**
WLF	Williams-Landel-Ferry

Symbols

Symbol	**Unit**	**Meaning**
a	$-$	Parameter of local density function
$\mathbf{A}$	$-$	Linear algebraic equation matrix for overhang angle constraint
$\mathrm{A}_{\mathrm{Carreau}}$	$\mathrm{Pa \cdot s \mid m^2/s}$	Carreau parameter representing zero-shear viscosity
b	$-$	Parameter of local density function
$\mathbf{B}$	$-$	Linear algebraic equation matrix for overhang angle constraint
$\mathrm{B}_{\mathrm{Carreau}}$	s	Carreau parameter representing reciprocal transition rate
$\mathbf{b}$	$-$	Design variable
c	$-$	Parameter of local density function
C	$\mathrm{1/m^3}$	(Primal) Concentration of a passive scalar
$\mathrm{C_a}$	$\mathrm{m^5/s^2}$	Adjoint concentration of a passive scalar
$\mathrm{C}_{\mathrm{Carreau}}$	$-$	Carreau parameter representing slope of the viscosity curve
$\mathrm{c_p}$	$\mathrm{J/(kg \cdot K)}$	Specific heat capacity
$\mathrm{C}_{\mathrm{target}}$	$\mathrm{1/m^3}$	Target concentration for optimisation
$\tilde{\mathrm{c}}$	$\mathrm{1/(m^2 \cdot s)}$	Constant for unit consistency
d	$-$	Parameter of local density function
D	$-$	Coefficient maxtrix (system of algebraic equations)
$\mathrm{D_C}$	$\mathrm{m^2/s}$	Diffusion coefficient
D_{ϕ}	$-$	Diffusivity for a scalar quantity ϕ
$\mathrm{D_T}$	$\mathrm{m^2/s}$	Thermal diffusivity
e	$-$	Mixing number
$\mathrm{e}_{\mathrm{material}}$	$-$	Material mixing number

Symbol	**Unit**	**Meaning**
e_{thermal}	–	Thermal mixing number
$\mathbf{f}$	–	Primal right hand side vector (system of algebraic equations)
$\mathbf{f}_{\text{a}}$	–	Dual/adjoint right hand side vector (system of algebraic equations)
FB	–	Flow balance ratio
F_p	N	Pressure force on a single mixing bar
F_{total}	N	Total force on a single mixing bar
F_z	N	Viscous force on a single mixing bar
g	–	Geometric constraints of optimisation problem
g_1	–	Overhang angle constraint
g_2	–	Minimum horizontal length constraint
g_3	–	Minimum horizontal length constraint under consideration of connectivity
h	–	Physical constraints of optimisation problem
H	m	Characteristic height
J	–	Cost function
L	m	Characteristic length
m	–	Reach of horizontal square area
M	–	Support set of horizontal square scheme
$\mathbf{n}$	–	Normal
N	–	Sphere of influence/"Neighbourhood"
n_{cells}	–	Number of cells
p	$\text{bar} \mid \text{Pa} \mid \frac{\text{m}^2}{\text{s}^2}$	(Primal) Pressure
p_a	$\text{bar} \mid \text{Pa} \mid \frac{\text{m}^2}{\text{s}^2}$	Adjoint pressure
Q^{ϕ}	–	Source term for a scalar quantity ϕ
R	m	Radius
R^C	–	Constraint from advection-diffusion equation
R^{C_a}	–	Adjoint advection-diffusion equation
R^p	–	Constraint from continuity equation

Symbol	**Unit**	**Meaning**
$\mathrm{R}^{\mathrm{p_a}}$	$-$	Adjoint continuity equation
R^{T}	$-$	Constraint from energy equation
$\mathrm{R}^{\mathrm{T_a}}$	$-$	Adjoint energy equation
$\mathrm{R}^{\mathbf{u}}$	$-$	Constraint from momentum equation
$\mathrm{R}^{\mathbf{u}_\mathrm{a}}$	$-$	Adjoint momentum equation
S	$-$	Set of cell neighbours in same and lower layer
T	K	(Primal) Temperature
$\mathrm{T_a}$	$\mathrm{m^2/(s^2 \cdot K)}$	Adjoint temperature
$\mathrm{T_m}$	K	Reference measure temperature from WLF equation
$\mathrm{t_p}$	s	Process time
$\mathrm{t_r}$	s	Relaxation time
$\mathrm{T_s}$	K	Standard temperature from WLF equation
$\mathrm{T_{target}}$	K	Target temperature for optimisation
$\mathbf{u}$	$\mathrm{m/s}$	(Primal) Velocity
$\mathbf{u}_\mathrm{a}$	$\mathrm{m/s}$	Adjoint velocity
$\mathrm{u_{a,n}}$	$\mathrm{m/s}$	Normal component of adjoint velocity
$\mathbf{u}_\mathrm{a,t}$	$\mathrm{m/s}$	Tangential component of adjoint velocity
$\mathrm{u_n}$	$\mathrm{m/s}$	Normal component of (primal) velocity
$\mathrm{u_{n,target}}$	$\mathrm{m/s}$	Normal component of target velocity
$\mathbf{u}_\mathrm{t}$	$\mathrm{m/s}$	Tangential component of (primal) velocity
$\mathbf{u}_\mathrm{target}$	$\mathrm{m/s}$	Target velocity for optimisation
$\mathbf{u}_\mathrm{t,target}$	$\mathrm{m/s}$	Tangential component of target velocity
V	$\mathrm{m^3}$	Volume
$\dot{\mathrm{V}}$	$\mathrm{m^3/s}$	Volumetric flow rate
w	$-$	Weight for intermediate density calculation
$\mathrm{w_c}$	$-$	Weight for optimisation objective regarding concentration
$\mathrm{w_T}$	$-$	Weight for optimisation objective regarding temperature
x	m	x-coordinate of cell

Symbol	**Unit**	**Meaning**
$\Delta\mathrm{x}$	–	Cell size for computation of horizontal square area
$\mathbf{x}$	–	Coordinate of cell centroid
y	m	y-coordinate of cell
z	m	z-coordinate of cell
α	$1/\mathrm{s}$	Inverse Darcy coefficient
α_{diff}	$1/\mathrm{s}$	Proposed change of inverse Darcy coefficient value during optimisation
α_{fixed}	–	Binary field denoting scaffold structure
α_{fluid}	$1/\mathrm{s}$	Inverse Darcy coefficient value for fluid cell
α_{max}	$1/\mathrm{s}$	Inverse Darcy coefficient value for solid cell
α_{solid}	$1/\mathrm{s}$	Inverse Darcy coefficient value for solid cell
α_{T}	–	Shift factor for temperature from WLF equation
α_0	$1/\mathrm{s}$	Inverse Darcy coefficient value for fluid cell
β	$^\circ$	Overhang angle
$\bar{\beta}$	$^\circ$	Critical overhang angle
β_{f}	–	Nonlinear density filter parameter
Γ	m^2	Surface
$\dot{\gamma}$	$1/\mathrm{s}$	Shear rate
γ_{r}	–	Field relaxation factor
δ	–	Local inhomogeneity
$\boldsymbol{\epsilon}$	$1/\mathrm{s}$	Rate of strain tensor
η	$\mathrm{Pa}\cdot\mathrm{s}$	Dynamic viscosity
η_{f}	–	Nonlinear density filter parameter
κ	$\mathrm{W}/(\mathrm{m}\cdot\mathrm{K})$	Thermal conductivity
$\boldsymbol{\lambda}$	–	Adjoint variables (vector of Lagrange multipliers)
λ_{C}	s/m^2	Sensitivity response step size for concentration
$\boldsymbol{\lambda}^*$	–	Adjoint solution (vector of Lagrange multipliers)
λ_{T}	s/m^2	Sensitivity response step size for temperature
λ_{u}	s/m^2	Sensitivity response step size for velocity

Symbol	Unit	Meaning
ν	$\mathrm{m^2/s}$	Kinematic viscosity
ξ	—	Set of parameters of local density function
ρ	$\mathrm{kg/m^3}$	Density
$\hat{\rho}$	—	Local density function
ρ_{im}	—	Intermediate density field
ρ_{phase}	—	Phase density field
ρ_{physical}	—	Physical density field
σ_{c}	$\mathrm{m^2/s^2}$	Local sensitivity with respect to concentration
σ_{T}	$\mathrm{m^2/s^2}$	Local sensitivity with respect to temperature
σ_{u}	$\mathrm{m^2/s^2}$	Local sensitivity with respect to velocity
τ	—	Horizontal minimum length constraint for a single support set
ϕ	—	Scalar quantity
χ	—	Primal flow variables
χ^*	—	Primal solution vector
Ω	$\mathrm{m^3}$	Volume
δ	—	Total derivative
∂	—	Partial derivative
∇	—	Differential operator
$\mathfrak{R}$	—	Constraints of optimisation problem

Indices

Index	Meaning
after	Location after mixer
ave	Average value
before	Location before mixer
C	Control volume

Index	Meaning
fluid	Cell is part of polymer melt domain
i	Cell index
j	Cell index
k	Cell index
n	Time step index
neigh	Neighbour cell index
r	Cell index
solid	Cell is part of steel domain
target	Target value for optimisation

References

[Aal22] Aali, M.: *Viscoelastic Fluids in Profile Extrusion: Relevance and Characterization*. University of Minho, PhD Thesis, 2022

[ACN22a] Aali, M.; Carneiro, O. S.; Nóbrega, J. M.: The Effect of Fluid Elasticity on the Pressure Drop and Flow Distribution in Profile Extrusion Dies. *17th OpenFOAM Workshop*. Cambridge, UK, 2022

[ACN22b] Aali, M.; Carneiro, O. S.; Nóbrega, J. M.: Profile extrusion die design: A comparative study between elastic and inelastic fluids. *Polymer Engineering & Science* 62 (2022) 2, p. 497–509

[ADJ21] Allaire, G.; Dapogny, C.; Jouve, F.: Shape and topology optimization. In: Bonito, A.; Nochetto, R. H. (Editor): *Handbook of Numerical Analysis: Geometric Partial Differential Equations - Part II*. Amsterdam: Elsevier, 2021

[AG20a] Alexias, P.; Giannakoglou, K. C.: Optimization of a static mixing device using the continuous adjoint to a two-phase mixing model. *Optimization and Engineering* 21 (2020) 2, p. 631–650

[AG20b] Alexias, P.; Giannakoglou, K. C.: Shape Optimization of a Two-Fluid Mixing Device Using Continuous Adjoint. *Fluids* 5 (2020) 1, p. 11

[AKVB19] Alessi, G.; Koloszar, L.; Verstraete, T.; van Beeck, J.: Node-Based Adjoint Surface Optimization of U-Bend Duct for Pressure Loss Reduction. In: Andrés-Pérez, E.; González, L. M.; Periaux, J.; Gauger, N.; Quagliarella, D.; Giannakoglou, K. C. (Editor): *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*. Cham: Springer, 2019

[AM20] Alekseev, K. A.; Mukhametzyanova, A. G.: Classification, Function, and Construction of Modern Static Mixers. *Chemical and Petroleum Engineering* 55 (2020) 11-12, p. 934–942

[AMK04] Alemaskin, K.; Manas-Zloczower, I.; Kaufman, M.: Index for Simultaneous Dispersive and Distributive Mixing Characterization in Processing Equipment. *International Polymer Processing* 19 (2004) 4, p. 327–334

[AMK05] Alemaskin, K.; Manas-Zloczower, I.; Kaufman, M.: Color mixing in the metering zone of a single screw extruder: Numerical simulations and experimental validation. *Polymer Engineering and Science* 45 (2005) 7, p. 1011–1020

[AV19] Alexias, P.; de Villiers, E.: Gradient Projection, Constraints and Surface Regularization Methods in Adjoint Shape Optimization. In: Andrés-Pérez, E.; González, L. M.; Periaux, J.; Gauger, N.; Quagliarella, D.; Giannakoglou, K. C. (Editor): *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*. Cham: Springer, 2019

[BBM21] Biedermann, M.; Beutler, P.; Meboldt, M.: Automated design of additive manufactured flow components with consideration of overhang constraint. *Additive Manufacturing* 46 (2021), p. 102119

[BBM22] Biedermann, M.; Beutler, P.; Meboldt, M.: Routing multiple flow channels for additive manufactured parts using iterative cable simulation. *Additive Manufacturing* 56 (2022), p. 102891

[BGG18] Behrou, R.; Guest, J. K.; Gaynor, A. T.: Projection-Based Overhang Constraints: Implementing an Efficient Adjoint Formulation for Sensitivity Analysis. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Quebec City, Canada, 2018

[BHHB12] Brandl, E.; Heckenberger, U.; Holzinger, V.; Buchbinder, D.: Additive manufactured AlSi10Mg samples using Selective Laser Melting (SLM): Microstructure, high cycle fatigue, and fracture behavior. *Materials & Design* 34 (2012), p. 159–169

[BHK+20] Bajaj, P.; Hariharan, A.; Kini, A.; Kürnsteiner, P.; Raabe, D.; Jägle, E. A.: Steels in additive manufacturing: A review of their microstructure and properties. *Materials Science and Engineering A* 772 (2020), p. 138633

[BKC04] Brown, E. C.; Kelly, A. L.; Coates, P. D.: Melt temperature field measurement in single screw extrusion using thermocouple meshes. *Review of Scientific Instruments* 75 (2004) 11, p. 4742–4748

[BM20] Biedermann, M.; Meboldt, M.: Computational design synthesis of additive manufactured multi-flow nozzles. *Additive Manufacturing* 35 (2020), p. 101231

[BMWS20] Biedermann, M.; Meboldt, M.; Walker, S.; Schwendemann, D.: Innovation in der Profil Co-Extrusion. *KunststoffXtra* 10 (2020) 11, p. 18–20

[Bou01] Bourdin, B.: Filters in topology optimization. *International Journal For Numerical Methods in Engineering* 50 (2001) 9, p. 2143–2158

[Bre22] Breuer, R.: *Additive Fertigung von Extrusionswerkzeugen - eine Analyse aus technischer und wirtschaftlicher Sicht*. RWTH Aachen University, unpublished PhD presentation, 2022

[BS99] Byrde, O.; Sawley, M.: Optimization of a Kenics static mixer for non-creeping flow conditions. *Chemical Engineering Journal* 72 (1999) 2, p. 163–169

[BWSM21] Biedermann, M.; Walker, S.; Schwendemann, D.; Meboldt, M.: Rethinking Co-Extrusion: Additive Manufacturing and Simulation-Based Design as a Duo for Extrusion Dies. *Kunststoffe International* 15 (2021) 1, p. 26–29

[Car68] Carreau, P. J.: *Rheological equations from molecular network theories*. University of Wisconsin, PhD thesis, 1968

[Cat12] Catherine, O.: Effect Of Typical Melt Temperature Non-Uniformity On Flow Distribution in Flat Dies. *Proceedings of the 70th Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Orlando, Florida, United States of America, 2012

[Cat13] Catherine, O.: Evaluation Of The Flow Performance Of A Static Mixer For Non-Uniform Incoming Melt Temperature With Computational Fluid Dynamics (CFD). *Proceedings of SPE EUROTEC*. Lyon, France, 2013

[Chu19] Chung, C. I.: *Extrusion of polymers: Theory and practice*. Munich and Cincinnati: Hanser Publishers and Hanser Publications, 3rd edition, 2019

[CN12] Carneiro, O. S.; Nóbrega, J. M.: *Design of Extrusion Forming Tools*. London: Smithers Rapra Technology, 2012

[CS20] Campbell, G. A.; Spalding, M. A.: *Analyzing and Troubleshooting Single-Screw Extruders*. Munich, Vienna: Hanser, 2020

[Dan52] Danckwerts, P. V.: The definition and measurement of some characteristics of mixtures. *Applied Scientific Research* 3 (1952) 4, p. 279–296

[DHD+19] Damon, J.; Hanemann, T.; Dietrich, S.; Graf, G.; Lang, K.-H.; Schulze, V.: Orientation dependent fatigue performance and mechanisms of selective laser melted maraging steel X3NiCoMoTi18-9-5. *International Journal of Fatigue* 127 (2019), p. 395–402

[Dom11] Domingues, N.: *Mixing in Single Screw Extrusion: Modelling and Optimization*. University of Minho, PhD thesis, 2011

[EB13] Erb, T.; Bonten, C.: A Novel Method For The Evaluation Of Particle Tracking Simulations Of Mixing Processes. *Proceedings of the 71st Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Cincinnati, Ohio, United States Of America, 2013

[Elg11] Elgeti, S.: *Free-surface flows in shape optimization of extrusion dies*. RWTH Aachen University, PhD thesis, 2011

[EPW+12] Elgeti, S.; Probst, M.; Windeck, C.; Behr, M.; Michaeli, W.; Hopmann, C.: Numerical shape optimization as an approach to extrusion die design. *Finite Elements in Analysis and Design* 61 (2012), p. 35–43

[Erb18] Erb, T.: *Simulationsgestützte Optimierung dynamischer Mischer für Hochleistungsextruder*. University of Stuttgart, PhD thesis, 2018

[ES23] Eber, C.; Schiffers, R.: CFD-based automatic optimization of an extrusion die's primary distribution. *Proceedings of the 37th International Conference of the Polymer Processing Society (PPS-37)*. Fukuoka City, Japan, 2023

[FAKB17] Faruoli, M.; Alessi, G.; Koloszar, L.; van Beeck, J.: Adjoint optimization with thermal constraints. *Student Submission for the 5th OpenFOAM User Conference*. Wiesbaden, Germany, 2017

[FM06] Fischer, P.; Michels, R.: Moderne Schlauchköpfe mit Wendelverteilern. *Fachtagung "Blasformtechnik 2006"*. Würzburg, Germany, 2006

[Fra24] Frank, M.: *Simulationsgestützte Bestimmung und Optimierung der Mischgüte in der Einschneckenextrusion*. Universität Paderborn, PhD thesis, 2024

[FSR+18] Fayazfar, H.; Salarian, M.; Rogalsky, A.; Sarker, D.; Russo, P.; Paserin, V.; Toyserkani, E.: A critical review of powder-based additive manufacturing of ferrous alloys: Process parameters, microstructure and mechanical properties. *Materials & Design* 144 (2018), p. 98–128

[FY24] Feng, Y.; Yamada, T.: Multi-material topology optimization for additive manufacturing considering maximum build volume and assembly process. *Engineering Analysis with Boundary Elements* 163 (2024), p. 616–640

[Gal09] Gale, M.: *Mixing in Single Screw Extrusion*. London: Smithers Rapra Technology, 2009

[GCS22] Gaspar-Cunha, A.; Covas, J. A.; Sikora, J.: Optimization of Polymer Processing: A Review (Part I—Extrusion). *Materials* 15 (2022) 1, p. 384

[GGC+24] Ghnatios, C.; Gravot, E.; Champaney, V.; Verdon, N.; Hascoët, N.; Chinesta, F.: Polymer extrusion die design using a data-driven autoencoders technique. *International Journal of Material Forming* 17 (2024) 4

[GHP23a] Gallorini, E.; Hèlie, J.; Piscaglia, F.: A multi region adjoint-based solver for topology optimization in conjugate heat transfer problems. *Computers & Fluids* 266 (2023), p. 106042

[GHP23b] Gallorini, E.; Hèlie, J.; Piscaglia, F.: An adjoint-based solver with adaptive mesh refinement for efficient design of coupled thermal-fluid systems. *International Journal for Numerical Methods in Fluids* (2023), p. 1–27

[GHZ14] Gebhardt, A.; Hötter, J.-S.; Ziebura, D.: Impact of SLM build parameters on the surface quality. *RTejournal - Forum für Rapid Technologie* 2014 (2014)

[GLNS23] Garcke, H.; Lam, K.; Nürnberg, R.; Signori, A.: Overhang Penalization in Additive Manufacturing via Phase Field Structural Topology Optimization with Anisotropic Energies. *Applied Mathematics & Optimization* 87 (2023) 3, p. 44

[GMS11] Geiger, K.; Martin, G.; Sobotta, A.: Relaxationszonen in Werkzeugen. *Kunststoffe* 101 (2011) 6, p. 44–49

[Goe17] Goeres, J.: *Spülverhalten verschiedener Polymere in additiv gefertigten Extrusionswerkzeugen*. Institute for Plastics Processing, RWTH Aachen University, unpublished student project, 2017 - Supervisor: Malte Schön

[GP97] Giles, M. B.; Pierce, N. A.: Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA Paper* 97-1850 (1997)

[GP00] Giles, M. B.; Pierce, N. A.: An Introduction to the Adjoint Approach to Design. *Flow, Turbulence and Combustion* 65 (2000) 3/4, p. 393–415

[GP20] Griffith, B. E.; Patankar, N. A.: Immersed Methods for Fluid-Structure Interaction. *Annual Review of Fluid Mechanics* 52 (2020), p. 421–448

[Gre18] Greenshields, C. J.: *OpenFOAM v6 User Guide*. London, UK: The OpenFOAM Foundation, 2018

[Gro20a] Gross, H.: Additiv hergestellte Schmelzekühler für die Schaumextrusion. *Blasformen & Extrusionswerkzeuge* 16 (2020) 5, p. 7–12

[Gro20b] Gross, H.: Maßgeschneidert kühlen und mischen: Konstruktionskonzept für die additive Fertigung von Schmelzekühlern. *Kunststoffe* 74 (2020) 2, p. 72–73

[Gro21] Gross, H.: Erweiterte verfahrenstechnische Möglichkeiten durch generativ gefertigte Mischschlauchköpfe (Teil 1). *Blasformen & Extrusionswerkzeuge* 17 (2021) 3, p. 7–9

[Hab14] Habla, F.: *Modeling and CFD simulation of viscoelastic single and multiphase flow*. Technische Universität München, PhD thesis, 2014

[HBE+22] Hube, S.; Behr, M.; Elgeti, S.; Schön, M.; Sasse, J.; Hopmann, C.: Numerical design of distributive mixing elements. *Finite Elements in Analysis and Design* 204 (2022), p. 103733

[Hee15] te Heesen, O.: *Ganzheitlicher Ansatz zur Dimensionierung und Optimierung von Extrusionswerkzeugen am Beispiel von Wendelverteilern*. Universität Duisburg-Essen, PhD thesis, 2015

[Hel15] Helgason, E.: *Development of adjoint-based optimization methods for ducted flows in vehicles*. Chalmers University Of Technology, PhD thesis, 2015

[Heu09] Heusser, R.: Perfect mixing with minimal pressure drop - The new static mixer. *Sulzer Technical Review* 91 (2009) 2+3, p. 23–25

[Heu18] Heusser, R.: Cool gemischt: Intensives statisches Mischen mit effektivem Wärmeaustausch verbinden. *Kunststoffe* 108 (2018) 1, p. 65–67

[HGL+18] Hopmann, C.; Gärtner, R.; Lohn, J.; Onken, J.; Schön, M.; Kreimeier, S.; Dahlmann, R.: Konstruktionsmethodik für die additive Fertigung. *Umdruck zum 29. Internationalen Kolloquium Kunststofftechnik*. Aachen, Germany, 2018

[HH21] Hopmann, C.; Hohlweck, T.: Path-planning Algorithm for the automatic generation of a cooling channel layout in injection moulds. *Proceedings of the 2021 SPE Annual Technical Conference*. online, 2021

[HK14] Helgason, E.; Krajnovic, S.: Implementation of an Adjoint-Based Optimization With Scalar Transport. *Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition*. Montreal, Quebec, Canada, 2014

[HKMS09] Hirschberg, S.; Koubek, R.; Moser, F.; Schöck, J.: An improvement of the Sulzer SMX™ static mixer significantly reducing the pressure drop. *Chemical Engineering Research and Design* 87 (2009) 4, p. 524–532

[HLSW22] Hopmann, C.; Leuchtenberger, L.; Schön, M.; Wallhorn, L.: Effect of Local Heat Pipe Cooling on Throughput Distribution and Thermal Homogeneity in a Binary Melt Pre-Distributor for Polyolefin Extrusion. *Polymers* 14 (2022) 11, p. 2271

[HLZS23] Hildner, M.; Lorenz, J.; Zhu, B.; Shih, A.: Pressure drop reduction of the impeller spiral static mixer design enabled by additive manufacturing. *Chemical Engineering and Processing - Process Intensification* 191 (2023), p. 109486

[HM16] Hopmann, C.; Michaeli, W.: *Extrusion dies for plastics and rubber: Design and engineering computations*. Munich and Vienna: Hanser, 4th edition, 2016

[HMMM18] He, P.; Mader, C. A.; Martins, J. R.; Maki, K. J.: An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. *Computers and Fluids* 168 (2018), p. 285–303

[HP19] Hopmann, C.; Poprawe, R.: *Simulative Optimierung und generative Fertigung von statischen Mischern für die Extrusion von Kunststoffen*. Institute for Plastics Processing, RWTH Aachen University, Abschlussbericht zum IGF-Vorhaben Nr. 18712 N, 2019

[HPSS19] Hopmann, C.; Poprawe, R.; Schön, M.; Schmidhüsen, T.: Bessere statische Mischer dank Simulation und additiver Fertigung. *Blasformen & Extrusionswerkzeuge* 15 (2019) 05, p. 5–9

[HS18] Hopmann, C.; Schön, M.: Thermische Homogenisierung von Schmelze durch simulativ optimierte statische Mischer. *Extrusion* 24 (2018) 5, p. 36–38

[HS20] Hopmann, C.; Schön, M.: Evaluation Of Mesh Interface And Immersed Boundary Models For The Optimisation Of Mixing Elements. *Proceedings of the 78th Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Digital Edition, 2020

[HS21] Hopmann, C.; Sasse, J.: Automatische Optimierung von additiv gefertigten statischen Mischern. *Blasformen & Extrusionswerkzeuge* 17 (2021) 4, p. 3–5

[HSB+20] Hopmann, C.; Schön, M.; Backmann, M.; Petzinka, F.; Leuchtenberger, L.: Simulative optimisation of mixing and die technology in extrusion. In: *Proceedings of 30th International Colloquium Plastics Technology 2020, pp. 93-128*. Aachen: Shaker, 2020

[HSRB20] Han, S.; Salvatore, F.; Rech, J.; Bajolet, J.: Abrasive flow machining (AFM) finishing of conformal cooling channels created by selective laser melting (SLM). *Precision Engineering* 64 (2020), p. 20–33

[HSRF20] Hopmann, C.; Schön, M.; Reul, M. M.; Facklam, M.: A Method for the Validation of Simulated Mixing Characteristics of Two Dynamic Mixers in Single-Screw Extrusion. *Polymers* 12 (2020) 10, p. 2234

[HST+19] Hopmann, C.; Schön, M.; Theunissen, M.; Meiners, W.; Shen, L.: Simulative evaluation of the optimization potential of additively manufactured static mixing elements for extrusion. *Proceedings of PPS-33: The 33rd International Conference of the Polymer Processing Society*. Cancun, Mexico, 2019

[HTC+22] Haveroth, G. A.; Thore, C.-J.; Correa, M. R.; Ausas, R. F.; Jakobsson, S.; Cuminato, J. A.; Klarbring, A.: Topology optimization including a model of the layer-by-layer additive manufacturing process. *Comput. Methods Appl. Mech. Engrg.* 398 (2022), p. 115203

[Hua98] Huang, C.-C.: A systematic approach for the design of a spiral mandrel die. *Polymer Engineering & Science* 38 (1998) 4, p. 573–582

[Hub23] Hube, S.: *Numerical Shape Optimization for Dynamic Mixing Elements in Single-Screw Extruders*. RWTH Aachen University, PhD thesis, 2023

[HWK+14] Hopmann, C.; Windeck, C.; Kurth, K.; Behr, M.; Siegbert, R.; Elgeti, S.: Improving the automated optimization of profile extrusion dies by applying appropriate optimization areas and strategies. *Proceedings of PPS-29: The 29th International Conference of the Polymer Processing Society - Conference Papers*. Nuremberg, Germany, 2014

[HWM01] van der Hoeven, J. C.; Wimberger-Friedl, R.; Meijer, H. E. H.: Homogeneity of multilayers produced with a static mixer. *Polymer Engineering & Science* 41 (2001) 1, p. 32–42

[HY15] Hopmann, C.; Yesildag, N.: Numerical investigation of the temperature influence on the melt predistribution in a spiral mandrel die with different polyolefins. *Journal of Polymer Engineering* 36 (2015) 7, p. 751–760

[HY17] Hopmann, C.; Yesildag, N.: Simulative evaluation of the temperature influence on different types of pre-distributors in spiral mandrel dies. *Proceedings of the 75th Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Anaheim, California, USA, 2017

[HYB+15] Hopmann, C.; Yesildag, N.; Bremen, S.; Wissenbach, K.; Merkt, S.: Surface quality of profile extrusion dies manufactured by Selective Laser Melting. *RTeJournal - Forum für Rapid Technologie* 2015 (2015)

[HYS17] Hopmann, C.; Yesildag, N.; Schön, M.: Thermal influences in the star-pre-distributor of a spiral mandrel die. *Journal of Polymer Engineering* 37 (2017) 8, p. 837–844

[HZD23] Hao, X.; Zhang, G.; Deng, T.: Improved Optimization of a Coextrusion Die with a Complex Geometry Using the Coupling Inverse Design Method. *Polymers* 15 (2023) 15, p. 3310

[Imh04] Imhoff, A.: *Dreidimensionale Beschreibung der Vorgänge in einem Einschneckenextruder*. RWTH Aachen University, PhD thesis, 2004 - ISBN: 3-86130-499-6

[JSVS22] Janssen, M.; Schulz, L.; Vorjohann, F.; Schiffers, R.: Automatisierte Optimierung von dynamischen Mischelementen. *Zeitschrift Kunststofftechnik/Journal Of Plastics Technology* 18 (2022) 1, p. 24–61

[KA24] Kietzmann, C.; Astbury, D.: Cooling channel layout optimization for injection molding. *Proceedings of the 2024 SPE Annual Technical Conference*. St. Louis, USA, 2024

[KAPC00] Khadra, K.; Angot, P.; Parneix, S.; Caltagirone, J.-P.: Fictitious domain approach for numerical modelling of Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* 34 (2000) 8, p. 651–684

[Kau04] Kaul, S.: *Rechnergestützte Optimierungsstrategien für Profil-Extrusionswerkzeuge*. RWTH Aachen University, PhD thesis, 2004 - ISBN: 3-86130-492-9

[KB20] Kettemann, J.; Bonten, C.: Simulation Of A Saxton-Mixer In High-Performance Extruders Using The Immersed Boundary Method. *Proceedings of the 78th Annual Technical Conference of the Society of Plastic Engineers (ANTEC) - Digital Edition*. digital, 2020

[KEB19] Kast, O.; Epple, S.; Bonten, C.: Unbestechlich mischen: Die Mischgüte von Extrudaten mittels digitaler Bildauswertung bestimmen. *Kunststoffe* 109 (2019) 4, p. 60–63

[KGB21] Kettemann, J.; Gatin, I.; Bonten, C.: Verification and validation of a finite volume immersed boundary method for the simulation of static and moving geometries. *Journal of Non-Newtonian Fluid Mechanics* 290 (2021), p. 104510

[KGPM01] Kruijt, P. G. M.; Galaktionov, O. S.; Peters, G. W. M.; Meijer, H. E. H.: The Mapping Method for Mixing Optimization Part I: The Multiflux Static Mixer. *International Polymer Processing* 16 (2001) 2, p. 151–160

[Kie10] Kienzl, W.: *Mischteilberechnungen mittels FEM*. Montanuniversität Leoben, Master thesis, 2010

[KLJK19] Kwon, B.; Liebenberg, L.; Jacobi, A. M.; King, W. P.: Heat transfer enhancement of internal laminar flows using additively manufactured static mixers. *International Journal of Heat and Mass Transfer* 137 (2019), p. 292–300

[Klo15] Klocke, F.: *Fertigungsverfahren 5 - Gießen, Pulvermetallurgie, Additive Manufacturing*. Berlin Heidelberg: Springer Verlag, 4th edition, 2015

[KMBK24] Kornely, M.; Matzerath, T.; Bonten, C.; Kreutzbruck, M.: Analysis of the Mixing Quality in Polymer Blends by Computed Tomography. *e-Journal of Nondestructive Testing* 29 (2024) 3

[Köp14] Köpplmayr, T.: *Computational and Experimental Modelling of Microlayer Forming Processes*. Johannes Kepler Universität Linz, PhD thesis, 2014

[Lan17] Langelaar, M.: An additive manufacturing filter for topology optimization of print-ready designs. *Structural and Multidisciplinary Optimization* 55 (2017), p. 871 – 883

[Lan18] Langelaar, M.: Combined optimization of part topology, support structure layout and build orientation for additive manufacturing. *Structural and Multidisciplinary Optimization* 57 (2018), p. 1985–2004

[LFC96] Li, H. Z.; Fasol, C.; Choplin, L.: Hydrodynamics and heat transfer of rheologically complex fluids in a Sulzer SMX static mixer. *Chemical Engineering Science* 51 (1996) 10, p. 1947–1955

[LHE24] Lee, J.; Hube, S.; Elgeti, S.: Neural networks vs. splines: advances in numerical extruder design. *Engineering with Computers* 40 (2024) 2, p. 989–1004

[LHW06] Liu, S.; Hrymak, A. N.; Wood, P. E.: Laminar mixing of shear thinning fluids in a SMX static mixer. *Chemical Engineering Science* 61 (2006) 6, p. 1753–1759

[Lip18] Lippert, R. B.: *Restriktionsgerechtes Gestalten gewichtsoptimierter Strukturbauteile für das Selektive Laserstrahlschmelzen*. Gottfried Wilhelm Leibniz Universität Hannover, PhD thesis, 2018

[LL17] Lachmayer, R.; Lippert, R.: *Additive Manufacturing Quantifiziert: Visionäre Anwendungen und Stand der Technik*. Hannover: Springer, 2017

[LP14] Lee, S.-T.; Park, C. B.: *Polymeric foams series: Foam extrusion: Principles and practice*. Boca Raton, FL, USA: CRC Press, 2nd edition, 2014

[LR12] Lincke, A.; Rung, T.: Adjoint-Based Sensitivity Analysis for Buoyancy-Driven Incompressible Navier-Stokes Equations with Heat Transfer. *Proceedings of the Eighth International Conference on Engineering Computational Technology*. Stirlingshire, Scotland, 2012

[LVTT23] Lieber, S. C.; Varghese, A. P.; Tarantino, R.; Tafuni, A.: Additive manufacturing for plastic extrusion die tooling: A numerical investigation. *CIRP Journal of Manufacturing Science and Technology* 41 (2023), p. 401–412

[MBKO24] Morales Ubal, C.; Beishuizen, N.; Kusch, L.; van Oijen, J.: Adjoint-based design optimization of a Kenics static mixer. *Results in Engineering* 21 (2024), p. 101856

[MBS07] Michaeli, W.; Blömer, P.; Scharf, M.: Aufbiegung von Radialwendelverteilern. *Zeitschrift Kunststofftechnik/Journal Of Plastics Technology* 3 (2007) 1, p. 1–19

[Mei99] Meiners, W.: *Direktes Selektives Laser Sintern einkomponentiger metallischer Werkstoffe*. RWTH Aachen University, PhD thesis, 1999

[MHMS11] Menges, G.; Haberstroh, E.; Michaeli, W.; Schmachtenberg, E.: *Menges Werkstoffkunde Kunststoffe*. München and Wien: Hanser, 6th edition, 2011

[MI05] Mittal, R.; Iaccarino, G.: Immersed Boundary Methods. *Annual Review of Fluid Mechanics* 37 (2005) 1, p. 239–261

[MKW01] Michaeli, W.; Kaul, S.; Wolff, T.: Computer-aided optimization of extrusion dies. *Journal of Polymer Engineering* 21 (2001) 2-3, p. 225–237

[MLM+15] Mihailova, O.; Lim, V.; McCarthy, M.; McCarthy, K.; Bakalis, S.: Laminar mixing in a SMX static mixer evaluated by positron emission particle tracking (PEPT) and magnetic resonance imaging (MRI). *Chemical Engineering Science* 137 (2015), p. 1014–1023

[MMD16] Moukalled, F.; Mangani, L.; Darwish, M.: *The Finite Volume Method in Computational Fluid Dynamics*. Cham: Springer International Publishing, 2016

[Mos17] Mosca, R.: *Adjoint-Based Passive Optimization of a Micro T-mixer*. Politecnico di Milano, Master thesis, 2017

[Mos20] Mosca, R.: *Implementation of new boundary conditions for external flow adjoint-based shape optimization*. Chalmers University Of Technology, Student Work, 2020

[MS16] Maier, R. D.; Schiller, M.: *Handbuch Kunststoff-Additive*. München: Hanser, 4th, completely revised edition, 2016

[MSA12] Meijer, H. E. H.; Singh, M. K.; Anderson, P. D.: On the performance of static mixers: A quantitative comparison. *Progress in Polymer Science* 37 (2012) 10, p. 1333–1349

[Neg23] Negrini, G.: *Non-conforming methods for the simulation of industrial polymer mixing processes*. Politecnico di Milano, PhD thesis, 2023

[Neu21] Neubrech, B.: *Thermo-rheologische Optimierung additiv gefertigter Extrusionswerkzeuge*. Universität Duisburg-Essen, PhD thesis, 2021

[NMD+24] Nardoni, C.; Mang, C.; Danan, D.; Bordeu Weldt, F. E.; Betbeder-Lauque, X.; Cortial, J.; Ploscaru, C.: Topology optimization of supports for metal powder bed additive manufacturing. *9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*. Lisbon, Portugal, 2024

[NN01] N.N.: *US6215093B1: Selective laser sintering at melting temperature*. Patent, US Patent and Trademark Office, 10.04.2001

[NN14] N.N.: *EP2851118A1: Device for mixing and heat exchange and method for its production*. Patent, European Patent Office, 29.08.2014

[NN18] N.N.: *Hostalen GD 9550 F Technical Data Sheet*. Rotterdam: LyondellBasell Industries Holdings, B.V., 2018

[NN20] N.N.: *EOS ToolSteel 1.2709 Material Data Sheet*. Krailling/Munich: EOS GmbH, 2020

[NN21] N.N.: *DE 10 2021 112 007.1: Mischteil einer Schneckenmaschine mit verbesserter radialer Homogenisierungswirkung*. Patent, Deutsches Patent- und Markenamt, 07.05.2021

[NN23a] N.N.: *Plastics - the fast Facts 2023*. Frankfurt: Plastics Europe e.V., 2023

[NN23b] N.N.: *Plastics Transitions Roadmap - Executive Summary*. Brussels: Plastics Europe e.V., 2023

[NN24] N.N.: *AMPOWER Report 2024 - Management Summary*. Hamburg: AMPOWER GmbH & Co. KG, 2024

[NWS+20] Neubrech, B.; Wortberg, J.; Schiffers, R.; Kleszczynski, S.; Witt, G.: Neuartige Werkzeuggeometrie für die Blasfolienextrusion unter Verwendung additiver Fertigungstechnik. *Zeitschrift Kunststofftechnik* 16 (2020) 03, p. 82–106

[OR14] Osswald, T. A.; Rudolph, N.: *Polymer rheology: Fundamentals and applications*. Munich and Cincinnati: Hanser, 2014

[OT10] Olason, A.; Tidman, D.: *Methodology for Topology and Shape Optimization in the Design Process*. Chalmers University of Technology, Master thesis, 2010

[Oth08] Othmer, C.: A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids* 58 (2008) 8, p. 861–877

[Oth14a] Othmer, C.: Adjoint Based Topology and Shape Optimization for Car Development. *AMS Seminar Series*. NASA Ames Research Center, 2014

[Oth14b] Othmer, C.: Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry* 4 (2014) 1, p. 6

[OVW07] Othmer, C.; de Villiers, E.; Weller, H. G.: Implementation of a Continuous Adjoint for Topology Optimization of Ducted Flows. *18th AIAA Computational Fluid Dynamics Conference*. Reston, Virginia, 2007

[PBE13] Pauli, L.; Behr, M.; Elgeti, S.: Towards shape optimization of profile extrusion dies with respect to homogeneous die swell. *Journal of Non-Newtonian Fluid Mechanics* 200 (2013), p. 79–87

[Pes72] Peskin, C. S.: Flow patterns around heart valves: A numerical method. *Journal of Computational Physics* 10 (1972) 2, p. 252–271

[PG16] Papoutsis-Kiachagias, E. M.; Giannakoglou, K. C.: Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications. *Archives of Computational Methods in Engineering* 23 (2016) 2, p. 255–299

[PHL+24] Pereira, G. G.; Howard, D.; Lahur, P.; Breedon, M.; Kilby, P.; Hornung, C. H.: Freeform generative design of complex functional structures. *Scientific Reports* 14 (2024) 1, p. 11918

[Pir74] Pironneau, O.: On optimum design in fluid mechanics. *Journal of Fluid Mechanics* 64 (1974), p. 97–110

[Pit11] Pittman, J. F.: Computer-aided design and optimization of profile extrusion dies for thermoplastics and rubber: A review. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 225 (2011), p. 280–321

[PM79] Pahl, M. H.; Muschelknautz, E.: Einsatz und Auslegung statischer Mischer. *Chemie Ingenieur Technik* 51 (1979) 5, p. 347–364

[PR19] del Pilar Noriega Escobar, M.; Rauwendaal, C.: *Troubleshooting the extrusion process: A systematic approach to solving plastic extrusion problems*. Munich: Hanser Publishers, 3rd edition, 2019

[PS72] Patankar, S.; Spalding, D.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer* 15 (1972) 10, p. 1787–1806

[RAC+12] Robinson, T.; Armstrong, C.; Chua, H.; Othmer, C.; Grahs, T.: Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Computer-Aided Design and Applications* 9 (2012) 3, p. 253–268

[Rat15] Rathod, M. L.: *Evaluation Of Dispersive Mixing, Extension Rate And Bubble Size Distribution Using Numerical Simulation Of A Non-Newtonian Fluid In A Twin- Screw Mixer*. Rutgers University, PhD thesis, 2015

[Rau19] Rauwendaal, C.: *Understanding extrusion*. Munich, Germany: Hanser Publishers, 3rd edition, 2019

[RBBS10] Rezaei Shahreza, A.; Behravesh, A. H.; Bakhshi Jooybari, M.; Soury, E.: Design, optimization, and manufacturing of a multiple–thickness profile extrusion die with a cross flow. *Polymer Engineering & Science* 50 (2010) 12, p. 2417–2424

[RBBT00] Rauline, D.; Le Blévec, J.-M.; Bousquet, J.; Tanguy, P. A.: A Comparative Assessment of the Performance of the Kenics and SMX Static Mixers. *Chemical Engineering Research and Design* 78 (2000) 3, p. 389–396

[RFF+17] Rajkumar, A.; Ferrás, L. L.; Fernandes, C.; Carneiro, O. S.; Becker, M.; Nóbrega, J. M.: Design Guidelines to Balance the Flow Distribution in Complex Profile Extrusion Dies. *International Polymer Processing* 32 (2017) 1, p. 58–71

[RFF+18] Rajkumar, A.; Ferrás, L. L.; Fernandes, C.; Carneiro, O. S.; Nóbrega, J. M.: Guidelines for balancing the flow in extrusion dies: the influence of the material rheology. *Journal of Polymer Engineering* 38 (2018) 2, p. 197–211

[RG15] Ryckebosh, K.; Gupta, M.: Optimization of a profile coextrusion die using a three-dimensional flow simulation software. *Proceedings of the 73rd Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Orlando, Florida, United States of America, 2015

[RTBB98] Rauline, D.; Tanguy, P. A.; Le Blévec, J.-M.; Bousquet, J.: Numerical investigation of the performance of several static mixers. *The Canadian Journal of Chemical Engineering* 76 (1998) 3, p. 527–535

[Rub17] Ruberto, E.: *An adjoint based topology optimization for flows including heat transfer*. Politecnico di Milano, Master Thesis, 2017

[SAAJ21] Shahbazi, A.; Ashtiani, H. A. D.; Afshar, H.; Jafarkazemi, F.: Optimization of the SMX static mixer types thermal and hydraulic performance by coupling CFD-Genetic Algorithm. *International Communications in Heat and Mass Transfer* 126 (2021), p. 105388

[SAM09] Singh, M. K.; Anderson, P. D.; Meijer, H. E. H.: Understanding and Optimizing the SMX Static Mixer. *Macromolecular Rapid Communications* 30 (2009) 4-5, p. 362–736

[Sau11] Saul, K.: *Automatisierte Auslegung von Extrusionswerkzeugen*. Universität Duisburg-Essen, PhD thesis, 2011

[SBE16] Siegbert, R.; Behr, M.; Elgeti, S.: Die swell as an objective in the design of polymer extrusion dies. *ESAFORM 2016: Proceedings of the 19th International ESAFORM Conference on Material Forming*. Nantes, France, 2016

[Sch16] Schön, M.: Statikmischer optimieren durch Simulation und additive Fertigung. *Plastverarbeiter* 12 (2016), p. 57

[Sch23a] Schön, M.: *Eine simulationsgestützte Methodik zur Dimensionierung von statischen und dynamischen Mischteilen für die Extrusion unter Berücksichtigung der Eingangsinhomogenität*. RWTH Aachen University, PhD thesis, 2023

[Sch23b] Schön, M.: *Transiente Effekte in stationär betriebenen Schneckenmaschinen*. RWTH Aachen University, unpublished PhD presentation, 2023

[SDH19] Subramaniam, V.; Dbouk, T.; Harion, J.-L.: Topology optimization of conjugate heat transfer systems: A competition between heat transfer enhancement and pressure drop reduction. *International Journal of Heat and Fluid Flow* 75 (2019), p. 165–184

[SEB+13] Siegbert, R.; Elgeti, S.; Behr, M.; Kurth, K.; Windeck, C.; Hopmann, C.: Design Criteria in Numerical Design of Profile Extrusion Dies. *Key Engineering Materials* 554-557 (2013), p. 794–800

[Seh10] Sehrt, J.: *Möglichkeiten und Grenzen bei der generativen Herstellung metallischer Bauteile durch das Strahlschmelzverfahren*. Universität Duisburg-Essen, PhD thesis, 2010

[SH03] Spalding, M. A.; Hyun, K. S.: Troubleshooting mixing problems in single-screw extruders. *Proceedings of the 61st Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Nashville, Tennessee, United States of America, 2003

[SH24a] Sasse, J.; Hopmann, C.: Automatic optimisation of additively manufactured static mixers under consideration of design restrictions. *Umdruck zum 32. Internationalen Kolloquium Kunststofftechnik*. Aachen, 2024

[SH24b] Sasse, J.; Hopmann, C.: Static mixers producible by additive manufacturing: Operating point specific geometries through automatic optimisation. *Proceedings of the 38th International Conference of the Polymer Processing Society (PPS-38)*. St. Gallen, CH, 2024

[SHA20] Spanjaards, M.; Hulsen, M. A.; Anderson, P. D.: Computational analysis of the extrudate shape of three-dimensional viscoelastic, non-isothermal extrusion flows. *Journal of Non-Newtonian Fluid Mechanics* 282 (2020), p. 104310

[SHA21] Spanjaards, M.; Hulsen, M. A.; Anderson, P. D.: Die shape optimization for extrudate swell using feedback control. *Journal of Non-Newtonian Fluid Mechanics* 293 (2021), p. 104552

[SHL11] Spierings, A. B.; Herres, N.; Levy, G.: Influence of the particle size distribution on surface quality and mechanical properties in AM steel parts. *Rapid Prototyping Journal* 17 (2011) 3, p. 195–202

[Sie18] Siegbert, R.: *Quantification of viscoelastic die swell as a design objective in the numerical design of profile extrusion dies*. RWTH Aachen University, PhD thesis, 2018

[Sih14] Sihvo, C.: *Numerical Method for Shape Optimization from Wind Noise Perspective*. Luleå University of Technology, Master thesis, 2014

[Sin08] Singh, M. K.: *Design, analysis, and optimization of distributive mixing: with applications to micro and industrial flow devices*. Technische Universiteit Eindhoven, PhD thesis, 2008

[SKA+09] Singh, M. K.; Kang, T. G.; Anderson, P. D.; Meijer, H. E. H.; Hrymak, A. N.: Analysis and optimization of low-pressure drop static mixers. *AIChE Journal* 55 (2009) 9, p. 2208–2216

[SKL+04] Spalding, M. A.; Kuhman, J. A.; Larson, D.; Kuhman, J.; Prettyman, H. L.: Performance of a distributive melt-mixing screw with an advanced mixing tip. *Proceedings of the 62nd Annual Technical Conference of the Society of Plastic Engineers (ANTEC)*. Chicago, Illinois, United States of America, 2004

[SLY04] Soto, O.; Löhner, R.; Yang, C.: An adjoint–based design methodology for CFD problems. *International Journal of Numerical Methods for Heat & Fluid Flow* 14 (2004) 6, p. 734–759

[Som16] Soman, S.: *Study of Effects of Design Modification in Static Mixer Geometry and its Applications*. University of Waterloo, Master thesis, 2016

[Spa22] Spanjaards, M.: *Modeling and optimization of polymer extrusion*. Technische Universiteit Eindhoven, PhD thesis, 2022

[SSG+20] Sharma, S.; Sarkar, K.; Goswami, M.; Deb, A.; Dcunha, S.; Chattopadhyay, S.: An approach to design extrusion dies for complex shaped rubber profiles using finite element analysis. *Journal of Manufacturing Processes* 57 (2020), p. 700–711

[SSH22] Sasse, J.; Schön, M.; Hopmann, C.: Static Mixers Producible by Additive Manufacturing: Novel Rapid Automatic Optimisation and Practical Evaluation. *Polymers* 14 (2022) 21, p. 4646

[SSH24] Sasse, J.; Schön, M.; Hopmann, C.: Fluid-Structure Interaction Framework for the Design of Extrusion Dies using the Immersed Boundary Surface Method in foam-extend. *9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*. Lisbon, Portugal, 2024

[SSPH00] Szarvasy, I.; Sienz, J.; Pittman, J. F.; Hinton, E.: Computer Aided Optimisation of Profile Extrusion Dies: Definition and Assessment of the Objective Function. *International Polymer Processing* 15 (2000) 1, p. 28–39

[SVJS23] Schulz, L.; Vorjohann, F.; Janssen, M.; Schiffers, R.: Evaluating distributive mixing potential by mean distance of streamlines and a passive scalar. *Proceedings of the 37th International Conference of the Polymer Processing Society (PPS-37)*. Fukuoka City, Japan, 2023

[SW51] Spencer, R.; Wiley, R.: The mixing of very viscous liquids. *Journal of Colloid Science* 6 (1951) 2, p. 133–145

[SYF+15] Siegbert, R.; Yesildag, N.; Frings, M.; Schmidt, F.; Elgeti, S.; Sauerland, H.; Behr, M.; Windeck, C.; Hopmann, C.; Queudeville, Y.; Vroomen, U.; Bührig-Polaczek, A.: Individualized production in die-based manufacturing processes using numerical optimization. *The International Journal of Advanced Manufacturing Technology* 80 (2015) 5-8, p. 851–858

[TMU+22] Tariq, S.; Mushtaq, A.; Ullah, A.; Qamar, R.; Ali, Z.; Hassan, M.; Ahmed, U.; Alam, S.; Sadiq, M.: Numerical Simulation of Low-Pressure Drop Static Mixers for Mixing Enhancement. *Iranian Journal of Chemistry and Chemical Engineering* 41 (2022) 9, p. 3141–3167

[Töb05] Többen, W.: *Rechnergestützte Prozessbeschreibung von Scher- und Mischelementen*. Universität Paderborn, PhD thesis, 2005

[Tow18] Towara, M.: *Discrete Adjoint Optimization with OpenFOAM*. RWTH Aachen University, PhD thesis, 2018

[TSN15] Towara, M.; Schanen, M.; Naumann, U.: MPI-Parallel Discrete Adjoint OpenFOAM. *Procedia Computer Science* 51 (2015), p. 19–28

[TY23] Tajima, M.; Yamada, T.: Topology optimization with geometric constraints for additive manufacturing based on coupled fictitious physical model. *Computer Methods in Applied Mechanics and Engineering* 417 (2023), p. 116415

[URL17] N.N.: *Variantenreiche Fertigung mit 3D-Druck-Verfahren (22.11.2017)*. URL: https://www.additive-fertigung.com/bericht/lpbf—laser-powder-bed-fusion_3239/varianten-reiche_fertigung_mit_3d_druck_verfahren-2017-11-22, 12.09.2024

[URL24] N.N.: *GitHub repository: source code PhD thesis*. URL: https://github.com/janasasse/phd-thesis-sasse, 21.08.2024

[US24] Ulbrich, C.; Schiffers, R.: A retrofit solution for substituting a conventional primary distribution by using additive manufacturing in a spiral mandrel die assembly in blown film extrusion. *Proceedings of the 38th international conference of the Polymer Processing Society (PPS-38)*. St. Gallen, CH, 2024

[Ver17] Verboom, J.: *Design and Additive Manufacturing of Manifolds for Navier-Stokes Flow: A Topology Optimisation Approach*. Delft University of Technology, Master Thesis, 2017

[VLL+16] Vatanabe, S. L.; Lippi, T. N.; de Lima, C. R.; Paulino, G. H.; Silva, E. C.: Topology optimization with manufacturing constraints: A unified projection-based approach. *Advances in Engineering Software* 100 (2016), p. 97–112

[VRHB99] Visser, J. E.; Rozendal, P. F.; Hoogstraten, H. W.; Beenackers, A. A.: Three-dimensional numerical simulation of flow and heat transfer in the Sulzer SMX static mixer. *Chemical Engineering Science* 54 (1999) 13-14, p. 2491–2500

[VSJS22] Vorjohann, F.; Schulz, L.; Janssen, M.; Schiffers, R.: Automated Optimization of a Block-Head-Mixer With an Innovative Algorithm. *Conference Proceedings of the 2022 SPE Annual Technical Conference (ANTEC)*. Charlotte, NC, United States of America, 2022

[Wen22] Wenz, K.: Kombination aus künstlicher Intelligenz und additiver Fertigung. *Kunststoffe* 112 (2022) 10, p. 190–192

[WHXR23] Wang, C.; Han, W.; Xu, B.; Rong, J.: Casting-oriented structural topology optimization with dimensional shrinkage. *Applied Mathematical Modelling* 117 (2023), p. 625–651

[Win15] Windeck, C.: *Simulative und experimentelle Bestimmung des viskoelastischen Aufschwellens und numerische Fließkanaloptimierung in der Extrusion*. RWTH Aachen University, PhD thesis, 2015 - ISBN: 978-3-95886-084-1

[WLF55] Williams, M. L.; Landel, R. F.; Ferry, J. D.: The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-Forming Liquids. *Journal of the American Chemical Society* 77 (1955) 7, p. 3701–3707

[WMK01] Wang, W.; Manas-Zloczower, I.; Kaufman, M.: Characterization of Distributive Mixing in Polymer Processing Equipment using Renyi Entropies. *International Polymer Processing* 16 (2001) 4, p. 315–322

[WMK03] Wang, W.; Manas-Zloczower, I.; Kaufman, M.: Entropic characterization of distributive mixing in polymer processing equipment. *AIChE Journal* 49 (2003) 7, p. 1637–1644

[Wol23] Wolff, D.: *Learning-based approaches for the analysis and optimization of profile extrusion dies and bioreactors*. RWTH Aachen University, PhD thesis, 2023

[Wol00] Wolff, T.: *Rechnerunterstützte Optimierung von Fließkanälen für Profilextrusionswerkzeuge*. RWTH Aachen University, PhD thesis, 2000 - ISBN: 3-89653-445-9

[XCC10] Xu, S.; Cai, Y.; Cheng, G.: Volume preserving nonlinear density filter based on heaviside functions. *Structural and Multidisciplinary Optimization* 41 (2010) 4, p. 495–505

[XYT18] Xu, B.; Yu, H.; Turng, L.-S.: Distributive mixing in a corotating twin screw channel using Lagrangian particle calculations. *Advances in Polymer Technology* 37 (2018) 6, p. 2215–2229

[YCM+22] Yu, Y.; Chen, Y.; Meng, H.; Yao, Y.; Liu, D.; Wu, J.: Numerical analysis of thermal dynamics and mixing performance in the blade-type static mixers. *Journal of Mechanical Science and Technology* 36 (2022) 7, p. 3701–3716

[Yes17] Yesildag, N.: *Simulative Auslegung von thermisch-rheologisch homogenen Vorverteilern in Wendelverteilerwerkzeugen*. RWTH Aachen University, PhD thesis, 2017 - ISBN: 978-3-95886-171-8

[YGK14] Yilmaz, O.; Gunes, H.; Kirkkopru, K.: Optimization of a Profile Extrusion Die for Flow Balance. *Fibers and Polymers* 15 (2014), p. 753–761

[YHW+16] Yesildag, N.; Hopmann, C.; Windeck, C.; Bremen, S.; Wissenbach, K.; Merkt, S.: Opportunities and challenges of profile extrusion dies produced by additive manufacturing processes. *Proceedings of PPS-32*. Lyon, France, 2016

[ZC20] Zhang, K.; Cheng, G.: Three-dimensional high resolution topology optimization considering additive manufacturing constraints. *Additive Manufacturing* 35 (2020), p. 101224

[ZC22] Zhang, K.; Cheng, G.: Structural topology optimization with four additive manufacturing constraints by two-phase self-supporting design. *Structural and Multidisciplinary Optimization* 65 (2022) 11, p. 339

[ZCX19] Zhang, K.; Cheng, G.; Xu, L.: Topology optimization considering overhang constraint in additive manufacturing. *Computers & Structures* 212 (2019), p. 86–100

[ZEE22] Zwar, J.; Elber, G.; Elgeti, S.: Shape optimization for temperature regulation in extrusion dies using microstructures. *Journal of Mechanical Design* (2022), p. 1–14

[ZHL+20] Zhang, G.; Huang, X.; Li, S.; Xia, C.; Deng, T.: Improved inverse design method for thin-wall hollow profiled polymer extrusion die based on FEM-CFD simulations. *The International Journal of Advanced Manufacturing Technology* 106 (2020) 7-8, p. 2909–2919

[ZHSD19] Zhang, G.; Huang, X.; Shuanggao, L.; Deng, T.: Optimized Design Method for Profile Extrusion Die Based on NURBS Modeling. *Fibers and Polymers* 20 (2019), p. 1733–1741

[ZSL+24] Zhou, Y.; Shen, S.; Liu, T.; Li, P.; Duan, F.: Effective heat conduction evaluation of lattice structures from selective laser melting printing. *International Journal of Heat and Mass Transfer* 218 (2024), p. 124790

[ZSMJ02] Zalc, J. M.; Szalai, E. S.; Muzzio, F. J.; Jaffer, S.: Characterization of flow and mixing in an SMX static mixer. *AIChE Journal* 48 (2002) 3, p. 427–436

During the preparation of this work the author used Grammarly (Grammarly Inc., San Francisco, USA) in order to check spelling and grammar and improve readability. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of this thesis.

A Appendix

A.1 Derivation of Adjoint Equations

Governing equations for primary problem:

$$\mathrm{R}^{\mathrm{p}} = \nabla \cdot \mathbf{u} \tag{A.1}$$

$$\mathrm{R}^{\mathbf{u}} = \nabla \cdot \mathbf{u}\mathbf{u} + \nabla \mathrm{p} - \nabla \cdot \left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)\right) + \alpha\mathbf{u} \tag{A.2}$$

$$\mathrm{R}^{\mathrm{T}} = \nabla \cdot \left(\mathbf{u}\mathrm{T}\right) - \nabla \cdot \left(\mathrm{D}_{\mathrm{T}}\nabla\mathrm{T}\right) - \frac{1}{\mathrm{c}_{\mathrm{p}}}\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right) : \nabla\mathbf{u}\right) \tag{A.3}$$

$$\mathrm{R}^{\mathrm{C}} = \nabla \cdot \left(\mathbf{u}\mathrm{C}\right) - \nabla \cdot \left(\mathrm{D}_{\mathrm{C}}\nabla\mathrm{C}\right) \tag{A.4}$$

Adjoint terms for mass conservation from Equation A.1

$$\begin{aligned}
\int_\Omega \mathrm{p}_\mathrm{a} \cdot \frac{\partial \mathrm{R}^{\mathrm{p}}}{\partial \mathbf{b}} &= -\int_\Omega \mathrm{p}_\mathrm{a} \frac{\partial}{\partial \mathbf{b}} \left(\nabla \cdot \mathbf{u}\right) = -\int_\Omega \mathrm{p}_\mathrm{a} \frac{\partial}{\partial \mathbf{b}} \left(\frac{\partial \mathrm{u}_\mathrm{i}}{\partial \mathrm{x}_\mathrm{i}}\right) \\
&= -\int_\Omega \frac{\partial}{\partial \mathrm{x}_\mathrm{i}} \left(\mathrm{p}_\mathrm{a} \frac{\partial \mathrm{u}_\mathrm{i}}{\partial \mathbf{b}}\right) + \int_\Omega \frac{\partial \mathrm{u}_\mathrm{i}}{\partial \mathbf{b}} \frac{\partial \mathrm{p}_\mathrm{a}}{\partial \mathrm{x}_\mathrm{i}} \\
&= -\int_\Gamma \mathrm{n}_\mathrm{i} \left(\mathrm{p}_\mathrm{a} \frac{\partial \mathrm{u}_\mathrm{i}}{\partial \mathbf{b}}\right) + \int_\Omega \frac{\partial \mathrm{u}_\mathrm{i}}{\partial \mathbf{b}} \frac{\partial \mathrm{p}_\mathrm{a}}{\partial \mathrm{x}_\mathrm{i}} \\
&= -\int_\Gamma \mathbf{n} \cdot \left(\mathrm{p}_\mathrm{a} \frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right) + \int_\Omega \frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \nabla \mathrm{p}_\mathrm{a}
\end{aligned} \tag{A.5}$$

Adjoint terms for momentum balance from Equation A.2 with $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}}\right)$

$$\begin{aligned}
\int_\Omega \mathbf{u}_\mathrm{a} \cdot \frac{\partial \mathrm{R}^{\mathrm{u}}}{\partial \mathbf{b}} = &\int_\Omega \mathbf{u}_\mathrm{a} \cdot \left(\nabla \cdot \left(\mathbf{u}\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right)\right) + \int_\Omega \mathbf{u}_\mathrm{a} \cdot \left(\nabla \cdot \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\mathbf{u}\right)\right) \\
&+ \int_\Omega \mathbf{u}_\mathrm{a} \cdot \left(\nabla \frac{\partial \mathrm{p}}{\partial \mathbf{b}}\right) - \int_\Omega \mathbf{u}_\mathrm{a} \cdot \left(\nabla \cdot \left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right)\right)\right) \\
&+ \int_\Omega \mathbf{u}_\mathrm{a} \cdot \mathbf{u}\frac{\partial \alpha}{\partial \mathbf{b}} + \int_\Omega \alpha\mathbf{u}_\mathrm{a} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}}
\end{aligned} \tag{A.6}$$

First convective term of Equation A.6:

$$\begin{aligned}
\int_\Omega \mathbf{u}_\mathrm{a} \cdot \left(\nabla \cdot \left(\mathbf{u}\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right)\right) &= \int_\Omega \mathrm{u}_{\mathrm{a,j}} \frac{\partial}{\partial \mathrm{x}_\mathrm{i}} \left(\frac{\partial \mathrm{u}_\mathrm{j}}{\partial \mathbf{b}} \mathrm{u}_\mathrm{i}\right) \\
&= \int_\Gamma \frac{\partial \mathrm{u}_\mathrm{j}}{\partial \mathbf{b}} \mathrm{u}_\mathrm{i} \mathrm{u}_{\mathrm{a,j}} \mathrm{n}_\mathrm{i} - \int_\Omega \frac{\partial \mathrm{u}_\mathrm{j}}{\partial \mathbf{b}} \mathrm{u}_\mathrm{i} \frac{\partial \mathrm{u}_{\mathrm{a,j}}}{\partial \mathrm{x}_\mathrm{i}} \\
&= \int_\Gamma \left(\mathbf{u} \cdot \mathbf{n}\right) \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \mathbf{u}_\mathrm{a}\right) - \int_\Omega \left(\left(\mathbf{u} \cdot \nabla\right) \mathbf{u}_\mathrm{a}\right) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}} \\
&\overset{\text{cont.}}{=} \int_\Gamma \left(\mathbf{u} \cdot \mathbf{n}\right) \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \mathbf{u}_\mathrm{a}\right) - \int_\Omega \left(\nabla \cdot \left(\mathbf{u}\mathbf{u}_\mathrm{a}\right)\right) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}}
\end{aligned} \tag{A.7}$$

Second convective term of Equation A.6:

$$\begin{aligned}\int_\Omega \mathbf{u}_\mathrm{a}\cdot\left(\nabla\cdot\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\mathbf{u}\right)\right) &= \int_\Omega \mathbf{u}_\mathrm{a}\cdot\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\cdot\nabla\mathbf{u}\right)\\ &= \int_\Gamma \mathrm{n}_\mathrm{j}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\mathrm{u}_\mathrm{a,i}\mathrm{u}_\mathrm{i} - \int_\Omega\left(\mathrm{u}_\mathrm{i}\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\\ &= \int_\Gamma\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\cdot\mathbf{n}\right)(\mathbf{u}\cdot\mathbf{u}_\mathrm{a}) - \int_\Omega(\nabla\mathbf{u}_\mathrm{a}\cdot\mathbf{u})\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\end{aligned}\tag{A.8}$$

Third (pressure) term of Equation A.6:

$$\int_\Omega \mathbf{u}_\mathrm{a}\cdot\left(\nabla\frac{\partial \mathrm{p}}{\partial\mathbf{b}}\right) = \int_\Gamma(\mathbf{n}\cdot\mathbf{u}_\mathrm{a})\frac{\partial \mathrm{p}}{\partial\mathbf{b}} - \int_\Omega\frac{\partial \mathrm{p}}{\partial\mathbf{b}}\nabla\cdot\mathbf{u}_\mathrm{a}\tag{A.9}$$

Fourth (diffusive) term of Equation A.6:

$$\begin{aligned}&-\int_\Omega \mathbf{u}_\mathrm{a}\cdot\left(\nabla\cdot\left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\right)\right)\right)\\ &= -\int_\Omega \nu\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\left(\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}} + \frac{\partial}{\partial \mathrm{x}_\mathrm{i}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\mathrm{u}_\mathrm{a,i}\\ &= \int_\Gamma \nu \mathrm{n}_\mathrm{j}\left(\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}} + \frac{\partial}{\partial \mathrm{x}_\mathrm{i}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\mathrm{u}_\mathrm{a,i} - \int_\Omega \nu\left(\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}} + \frac{\partial}{\partial \mathrm{x}_\mathrm{i}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\end{aligned}\tag{A.10}$$

Second term of Equation A.10:

$$\begin{aligned}&-\int_\Omega \nu\left(\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}} + \frac{\partial}{\partial \mathrm{x}_\mathrm{i}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\\ &= -\int_\Gamma \nu\left(\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\mathrm{n}_\mathrm{j} + \frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\mathrm{n}_\mathrm{i}\right) + \int_\Omega \nu\left(\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial^2 \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{j}} + \frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\frac{\partial^2 \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{i}\partial \mathrm{x}_\mathrm{j}}\right)\\ &= -\int_\Gamma \nu\left(\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}}\mathrm{n}_\mathrm{j} + \frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{i}}\mathrm{n}_\mathrm{j}\right) + \int_\Omega \nu\left(\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial^2 \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{j}} + \frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\frac{\partial^2 \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{i}}\right)\\ &= -\int_\Gamma \nu\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\left(\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}} + \frac{\partial \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{i}}\right)\mathrm{n}_\mathrm{j} + \int_\Omega \nu\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\left(\frac{\partial^2 \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{j}} + \frac{\partial^2 \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{i}}\right)\end{aligned}\tag{A.11}$$

Thus, the diffusive term can be written as

$$\begin{aligned}&-\int_\Omega \mathbf{u}_\mathrm{a}\cdot\left(\nabla\cdot\left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\right)\right)\right)\\ &= \int_\Gamma \nu \mathrm{n}_\mathrm{j}\left(\frac{\partial}{\partial \mathrm{x}_\mathrm{j}}\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}} + \frac{\partial}{\partial \mathrm{x}_\mathrm{i}}\frac{\partial \mathrm{u}_\mathrm{j}}{\partial\mathbf{b}}\right)\mathrm{u}_\mathrm{a,i} - \int_\Gamma \nu\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\left(\frac{\partial \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}} + \frac{\partial \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{i}}\right)\mathrm{n}_\mathrm{j}\\ &\quad + \int_\Omega \nu\frac{\partial \mathrm{u}_\mathrm{i}}{\partial\mathbf{b}}\left(\frac{\partial^2 \mathrm{u}_\mathrm{a,i}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{j}} + \frac{\partial^2 \mathrm{u}_\mathrm{a,j}}{\partial \mathrm{x}_\mathrm{j}\partial \mathrm{x}_\mathrm{i}}\right)\\ &= \int_\Gamma \mathbf{u}_\mathrm{a}\cdot\left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\right)\cdot\mathbf{n}\right) - \int_\Gamma \mathbf{n}\cdot\left(2\nu\boldsymbol{\epsilon}(\mathbf{u}_\mathrm{a})\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\right) + \int_\Omega \nabla\cdot(2\nu\boldsymbol{\epsilon}(\mathbf{u}_\mathrm{a}))\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{b}}\end{aligned}\tag{A.12}$$

Adjoint terms for temperature transport diffusion from Equation A.3

$$\int_{\Omega} \mathrm{T_a} \cdot \frac{\partial \mathrm{R^T}}{\partial \mathbf{b}} = \int_{\Omega} \mathrm{T_a} \frac{\partial}{\partial \mathbf{b}} \left(\nabla \cdot (\mathbf{u}\mathrm{T}) - \nabla \cdot (\mathrm{D_T} \nabla \mathrm{T}) - \frac{1}{\mathrm{c_p}} \left(2\nu \boldsymbol{\epsilon} (\mathbf{u}) : \nabla \mathbf{u} \right) \right) \tag{A.13}$$

First term of Equation A.13:

$$\int_{\Omega} \mathrm{T_a} \frac{\partial}{\partial \mathbf{b}} \left(\nabla \cdot (\mathbf{u}\mathrm{T}) \right) = \int_{\Omega} \mathrm{T_a} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \nabla \mathrm{T} \right) + \int_{\Omega} \mathrm{T_a} \left(\mathbf{u} \cdot \nabla \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \right) \tag{A.14}$$

First part of Equation A.14:

$$\begin{aligned} \int_{\Omega} \mathrm{T_a} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \nabla \mathrm{T} \right) &= \int_{\Omega} \mathrm{T_a} \frac{\partial \mathrm{u_i}}{\partial \mathbf{b}} \frac{\partial \mathrm{T}}{\partial \mathrm{x_i}} \\ &= \int_{\Omega} \frac{\partial}{\partial \mathrm{x_i}} \left(\mathrm{T_a} \frac{\partial \mathrm{u_i}}{\partial \mathbf{b}} \mathrm{T} \right) - \int_{\Omega} \mathrm{T} \frac{\partial}{\partial \mathrm{x_i}} \left(\mathrm{T_a} \frac{\partial \mathrm{u_i}}{\partial \mathbf{b}} \right) \\ &= \int_{\Gamma} \mathrm{n_i} \left(\mathrm{T_a} \frac{\partial \mathrm{u_i}}{\partial \mathbf{b}} \mathrm{T} \right) - \int_{\Omega} \mathrm{T} \frac{\partial}{\partial \mathrm{x_i}} \left(\mathrm{T_a} \frac{\partial \mathrm{u_i}}{\partial \mathbf{b}} \right) \\ &= \int_{\Gamma} (\mathrm{T_a} \mathrm{T} \mathbf{n}) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}} - \int_{\Omega} \mathrm{T} \nabla \mathrm{T_a} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}}, \end{aligned} \tag{A.15}$$

because $\partial\, \partial \mathrm{u_i} / \partial \mathbf{b}\, \partial \mathrm{x_i} = 0$ due to continuity equation.

Second part of Equation A.14:

$$\begin{aligned} \int_{\Omega} \mathrm{T_a} \left(\mathbf{u} \cdot \nabla \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \right) &= \int_{\Omega} \mathrm{T_a} \left(\mathrm{u_i} \frac{\partial}{\partial \mathrm{x_i}} \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \right) \\ &= \int_{\Gamma} (\mathrm{n_i} \mathrm{u_i} \mathrm{T_a}) \frac{\partial \mathrm{T}}{\partial \mathbf{b}} - \int_{\Omega} \frac{\partial (\mathrm{T_a} \mathrm{u_i})}{\partial \mathrm{x_i}} \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \\ &= \int_{\Gamma} (\mathbf{n} \cdot \mathbf{u} \mathrm{T_a}) \frac{\partial \mathrm{T}}{\partial \mathbf{b}} - \int_{\Omega} \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \nabla \cdot (\mathrm{T_a} \mathbf{u}) \end{aligned} \tag{A.16}$$

Second term in Equation A.13:

$$\begin{aligned} &- \int_{\Omega} \mathrm{T_a} \frac{\partial}{\partial \mathbf{b}} \left(\nabla \cdot (\mathrm{D_T} \nabla \mathrm{T}) \right) \\ &= \int_{\Omega} \mathrm{T_a} \nabla \cdot \left(\frac{\partial \mathrm{D_T}}{\partial \mathbf{b}} \nabla \mathrm{T} \right) - \int_{\Omega} \mathrm{T_a} \nabla \cdot \left(\mathrm{D_T} \nabla \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \right) \\ &= - \int_{\Gamma} (\mathrm{T_a} \mathbf{n} \cdot \nabla \mathrm{T}) \frac{\partial \mathrm{D_T}}{\partial \mathbf{b}} - \int_{\Omega} \nabla \mathrm{T} \cdot \nabla \mathrm{T_a} \frac{\partial \mathrm{D_T}}{\partial \mathbf{b}} \\ &\quad - \int_{\Gamma} \mathbf{n} \cdot \left(\mathrm{T_a} \mathrm{D_T} \nabla \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \right) + \int_{\Gamma} (\mathrm{D_T} \mathbf{n} \cdot \nabla \mathrm{T_a}) \frac{\partial \mathrm{T}}{\partial \mathbf{b}} - \int_{\Omega} \nabla (\mathrm{D_T} \nabla \mathrm{T_a}) \frac{\partial \mathrm{T}}{\partial \mathbf{b}} \end{aligned} \tag{A.17}$$

Third term in Equation A.13:

$$\begin{aligned}
&-\int_\Omega \mathrm{T_a}\frac{\partial}{\partial \mathbf{b}}\left(\frac{1}{\mathrm{c_p}}\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right):\nabla\mathbf{u}\right)\right) \\
&\overset{\mathrm{c_p=const.}}{=} -\frac{1}{\mathrm{c_p}}\int_\Omega \mathrm{T_a}\frac{\partial}{\partial \mathbf{b}}\left(\nu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}}\right):\nabla\mathbf{u}\right) \\
&= -\frac{1}{\mathrm{c_p}}\int_\Omega \mathrm{T_a}\frac{\partial}{\partial \mathbf{b}}\left(\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\right)\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\right) \\
&= -\frac{1}{\mathrm{c_p}}\int_\Omega \mathrm{T_a}\left(\frac{\partial \nu}{\partial \mathbf{b}}\left(\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\right)\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\nu\left(\frac{\partial}{\partial \mathrm{x_i}}\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}+\frac{\partial}{\partial \mathrm{x_j}}\frac{\partial \mathrm{u_i}}{\partial \mathbf{b}}\right)\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\right. \\
&\qquad\left.+\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\right)\frac{\partial\partial \mathrm{u_j}}{\partial \mathbf{b}\partial \mathrm{x_i}}\right)
\end{aligned} \tag{A.18}$$

First term of Equation A.18:

$$\int_\Omega \mathrm{T_a}\left(\frac{\partial \nu}{\partial \mathbf{b}}\left(\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\right)\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\right)=\int_\Omega \mathrm{T_a}\left(2\boldsymbol{\epsilon}\left(\mathbf{u}\right):\nabla\mathbf{u}\right)\frac{\partial \nu}{\partial \mathbf{b}} \tag{A.19}$$

Second term of Equation A.18:

$$\begin{aligned}
&\int_\Omega \mathrm{T_a}\left(\nu\left(\frac{\partial}{\partial \mathrm{x_i}}\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}+\frac{\partial}{\partial \mathrm{x_j}}\frac{\partial \mathrm{u_i}}{\partial \mathbf{b}}\right)\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\right) \\
&= \int_\Gamma \mathrm{T_a}\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\mathrm{n_i}+\frac{\partial \mathrm{u_i}}{\partial \mathbf{b}}\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\mathrm{n_j}\right)+\int_\Omega \mathrm{T_a}\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x_i}\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathbf{b}}\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x_j}\partial \mathrm{x_i}}\right) \\
&= \int_\Gamma \mathrm{T_a}\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}\mathrm{n_i}+\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\mathrm{n_i}\right)+\int_\Omega \mathrm{T_a}\nu\left(\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x_i}\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_i}\partial \mathrm{x_j}}\right) \\
&= \int_\Gamma \mathrm{T_a}\nu\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\left(\frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}}+\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}}\right)\mathrm{n_i}+\int_\Omega \mathrm{T_a}\nu\frac{\partial \mathrm{u_j}}{\partial \mathbf{b}}\left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x_i}\partial \mathrm{x_i}}+\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_i}\partial \mathrm{x_j}}\right) \\
&= \int_\Gamma \mathrm{T_a}\mathbf{n}\cdot\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)\cdot\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right)+\int_\Omega \mathrm{T_a}\nabla\cdot\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)\right)\cdot\frac{\partial \mathbf{u}}{\partial \mathbf{b}}
\end{aligned} \tag{A.20}$$

Third term of Equation A.18:

$$\begin{aligned}
&\int_\Omega T_a \left(\nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial \partial u_j}{\partial \mathbf{b} \partial x_i}\right) \\
&= \int_\Omega \frac{\partial}{\partial x_i} \left(T_a \nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}}\right) - \int_\Omega \frac{\partial T_a}{\partial x_i} \nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}} \\
&\quad - \int_\Omega T_a \nu \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}} \\
&= \int_\Gamma n_i T_a \nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}} - \int_\Omega \frac{\partial T_a}{\partial x_i} \nu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}} \\
&\quad - \int_\Omega T_a \nu \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_j}{\partial \mathbf{b}} \\
&= \int_\Gamma T_a \mathbf{n} \cdot \left(2\nu \boldsymbol{\epsilon}(\mathbf{u}) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right) - \int_\Omega \nabla T_a 2\nu \boldsymbol{\epsilon}(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial \mathbf{b}} - \int_\Omega T_a \nabla \cdot (2\nu \boldsymbol{\epsilon}(\mathbf{u})) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}}
\end{aligned} \tag{A.21}$$

Adjoint terms for advection-diffusion problem for a passive scalar from Equation A.4

$$\int_\Omega C_a \cdot \frac{\partial R^C}{\partial \mathbf{b}} = \int_\Omega C_a \frac{\partial}{\partial \mathbf{b}} \left(\nabla \cdot (\mathbf{u}C) - \nabla \cdot (D_C \nabla C)\right) \tag{A.22}$$

First term of Equation A.22:

$$\int_\Omega C_a \frac{\partial}{\partial \mathbf{b}} (\nabla \cdot (\mathbf{u}C)) = \int_\Omega C_a \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \nabla C\right) + \int_\Omega C_a \left(\mathbf{u} \cdot \nabla \frac{\partial C}{\partial \mathbf{b}}\right) \tag{A.23}$$

First part of Equation A.23:

$$\begin{aligned}
\int_\Omega C_a \left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \cdot \nabla C\right) &= \int_\Omega C_a \frac{\partial u_i}{\partial \mathbf{b}} \frac{\partial C}{\partial x_i} \\
&= \int_\Omega \frac{\partial}{\partial x_i} \left(C_a \frac{\partial u_i}{\partial \mathbf{b}} C\right) - \int_\Omega C \frac{\partial}{\partial x_i} \left(C_a \frac{\partial u_i}{\partial \mathbf{b}}\right) \\
&= \int_\Gamma n_i \left(C_a \frac{\partial u_i}{\partial \mathbf{b}} C\right) - \int_\Omega C \frac{\partial}{\partial x_i} \left(C_a \frac{\partial u_i}{\partial \mathbf{b}}\right) \\
&= \int_\Gamma (C_a C \mathbf{n}) \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}} - \int_\Omega C \nabla C_a \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{b}},
\end{aligned} \tag{A.24}$$

because $\partial\, \partial u_i / \partial \mathbf{b}\, \partial x_i = 0$ due to continuity equation.

Second part of Equation A.23:

$$\begin{aligned}
\int_\Omega C_a \left(\mathbf{u} \cdot \nabla \frac{\partial C}{\partial \mathbf{b}}\right) &= \int_\Omega C_a \left(u_i \frac{\partial}{\partial x_i} \frac{\partial C}{\partial \mathbf{b}}\right) \\
&= \int_\Gamma (n_i u_i C_a) \frac{\partial C}{\partial \mathbf{b}} - \int_\Omega \frac{\partial (C_a u_i)}{\partial x_i} \frac{\partial C}{\partial \mathbf{b}} \\
&= \int_\Gamma (\mathbf{n} \cdot \mathbf{u} C_a) \frac{\partial C}{\partial \mathbf{b}} - \int_\Omega \frac{\partial C}{\partial \mathbf{b}} \nabla \cdot (C_a \mathbf{u})
\end{aligned} \tag{A.25}$$

Second term in Equation A.22:

$$\begin{aligned}
&-\int_\Omega \mathrm{C_a}\frac{\partial}{\partial \mathbf{b}}\left(\nabla\cdot\left(\mathrm{D_C}\nabla\mathrm{C}\right)\right) \\
&=\int_\Omega \mathrm{C_a}\nabla\cdot\left(\frac{\partial \mathrm{D_C}}{\partial \mathbf{b}}\nabla\mathrm{C}\right)-\int_\Omega \mathrm{C_a}\nabla\cdot\left(\mathrm{D_C}\nabla\frac{\partial \mathrm{C}}{\partial \mathbf{b}}\right) \\
&=-\int_\Gamma\left(\mathrm{C_a}\mathbf{n}\cdot\nabla\mathrm{C}\right)\frac{\partial \mathrm{D_C}}{\partial \mathbf{b}}-\int_\Omega \nabla\mathrm{C}\cdot\nabla\mathrm{C_a}\frac{\partial \mathrm{D_C}}{\partial \mathbf{b}} \\
&\quad-\int_\Gamma \mathbf{n}\cdot\left(\mathrm{C_a}\mathrm{D_C}\nabla\frac{\partial \mathrm{C}}{\partial \mathbf{b}}\right)+\int_\Gamma\left(\mathrm{D_C}\mathbf{n}\cdot\nabla\mathrm{C_a}\right)\frac{\partial \mathrm{C}}{\partial \mathbf{b}}-\int_\Omega\nabla\left(\mathrm{D_C}\nabla\mathrm{C_a}\right)\frac{\partial \mathrm{C}}{\partial \mathbf{b}}
\end{aligned} \tag{A.26}$$

As a result, there are three types of terms: volume integrals containing a partial derivative of the primary variables χ = (p, **u**, T, C) with respect to the design variable **b**, which form the adjoint equations, surface integrals containing a partial derivative of χ with respect to the design variable **b**, which form the adjoint boundary conditions, and terms independent from a partial derivative of χ with respect to the design variable **b**, forming the augmented sensitivity.

Thus, the resulting system of adjoint equations is

$$\mathrm{R}^{\mathrm{p_a}}=\nabla\cdot\mathbf{u}_\mathrm{a}+\frac{\partial \mathrm{J}_\Omega}{\partial \mathrm{p}}=0 \tag{A.27}$$

$$\begin{aligned}
\mathrm{R}^{\mathbf{u}_\mathrm{a}}&=-\nabla\cdot\left(\mathbf{u}\mathbf{u}_\mathrm{a}\right)-\left(\nabla\mathbf{u}_\mathrm{a}\right)\mathbf{u}+\nabla\mathrm{p_a}-\nabla\cdot\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}_\mathrm{a}\right)\right)+\alpha\mathbf{u}_\mathrm{a}-\mathrm{T}\nabla\mathrm{T_a} \\
&\quad+\frac{1}{\mathrm{c_p}}\nabla\mathrm{T_a}2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)-\mathrm{C}\nabla\mathrm{C_a}+\frac{\partial \mathrm{J}_\Omega}{\partial \mathbf{u}}=0
\end{aligned} \tag{A.28}$$

$$\mathrm{R}^{\mathrm{T_a}}=-\nabla\cdot\left(\mathbf{u}\mathrm{T_a}\right)-\nabla\left(\mathrm{D_T}\nabla\mathrm{T_a}\right)+\frac{\partial \mathrm{J}_\Omega}{\partial \mathrm{T}}=0 \tag{A.29}$$

$$\mathrm{R}^{\mathrm{C_a}}=-\nabla\cdot\left(\mathbf{u}\mathrm{C_a}\right)-\nabla\left(\mathrm{D_C}\nabla\mathrm{C_a}\right)+\frac{\partial \mathrm{J}_\Omega}{\partial \mathrm{C}}=0 \tag{A.30}$$

together with the adjoint boundary conditions

$$\int_\Gamma\left(\mathbf{n}\cdot\mathbf{u}_\mathrm{a}+\frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{p}}\right)\frac{\partial \mathrm{p}}{\partial \mathbf{b}}=0 \tag{A.31}$$

$$\begin{aligned}
&\int_\Gamma\Bigg(\mathbf{n}\cdot\left(\mathbf{u}\cdot\mathbf{u}_\mathrm{a}\right)+\mathbf{u}_\mathrm{a}\left(\mathbf{u}\cdot\mathbf{n}\right)-\mathrm{p_a}\mathbf{n}-\mathbf{n}\cdot\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}_\mathrm{a}\right)\right)+\mathrm{T_a}\mathrm{T}\mathbf{n}+\mathrm{C_a}\mathrm{C}\mathbf{n} \\
&\quad-\frac{1}{\mathrm{c_p}}2\mathrm{T_a}\mathbf{n}\cdot\left(2\nu\boldsymbol{\epsilon}\left(\mathbf{u}\right)\right)+\frac{\partial \mathrm{J}_\Gamma}{\partial \mathbf{u}}\Bigg)\frac{\partial \mathbf{u}}{\partial \mathbf{b}}+\int_\Gamma\mathbf{u}_\mathrm{a}\cdot\left(2\nu\boldsymbol{\epsilon}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{b}}\right)\cdot\mathbf{n}\right)=0
\end{aligned} \tag{A.32}$$

$$\int_\Gamma\left(\mathbf{n}\cdot\mathbf{u}\mathrm{T_a}+\mathrm{D_T}\mathbf{n}\cdot\nabla\mathrm{T_a}+\frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{T}}\right)\frac{\partial \mathrm{T}}{\partial \mathbf{b}}-\int_\Gamma\mathbf{n}\cdot\left(\mathrm{T_a}\mathrm{D_T}\nabla\frac{\partial \mathrm{T}}{\partial \mathbf{b}}\right)=0 \tag{A.33}$$

$$\int_\Gamma\left(\mathbf{n}\cdot\mathbf{u}\mathrm{C_a}+\mathrm{D_C}\mathbf{n}\cdot\nabla\mathrm{C_a}+\frac{\partial \mathrm{J}_\Gamma}{\partial \mathrm{C}}\right)\frac{\partial \mathrm{C}}{\partial \mathbf{b}}-\int_\Gamma\mathbf{n}\cdot\left(\mathrm{C_a}\mathrm{D_C}\nabla\frac{\partial \mathrm{C}}{\partial \mathbf{b}}\right)=0 \tag{A.34}$$

A.2 Source Code Checking Geometric Constraints

A.2.1 initDataStructure.H

```
Info << "Precomputations for geometric constraints algorithm" << endl;

/* ------------------------------------------------------------------ */
scalar yNeigh, zNeigh, xCell, yCell, zCell;
scalar yNeighNeigh, zNeighNeigh, radius, distance, init;
label n1, n2, n3, n4, n5, n6;

// list for all neighbours of all cells
DynamicList<List<label>> neighbour(mesh.nCells());
DynamicList<List<label>> allNeighbours(mesh.nCells());
DynamicList<List<scalar>> B_xRow(mesh.nCells());
DynamicList<List<scalar>> B_yRow(mesh.nCells());
DynamicList<List<scalar>> B_zRow(mesh.nCells());
DynamicList<List<label>> M_1(mesh.nCells());
DynamicList<List<label>> M_2(mesh.nCells());
DynamicList<List<label>> M_3(mesh.nCells());
DynamicList<List<label>> M_4(mesh.nCells());
DynamicList<List<label>> sameLayer(mesh.nCells());

const scalarField&      Vcell  = mesh.V();
DynamicList<List<scalar>> weights(mesh.nCells());

forAll (alpha, celli)
{
    /* -------- determine neighbours and weights of celli -------- */
    DynamicList<label> N_i_sameLayer;
    DynamicList<label> N_i_lowerLayer;
    DynamicList<label> N_i_upperLayer;
    DynamicList<scalar> w_i_sameLayer;
    DynamicList<scalar> w_i_lowerLayer;
    DynamicList<scalar> w_i_upperLayer;
    xCell = mesh.C()[celli].x();
    yCell = mesh.C()[celli].y();
    zCell = mesh.C()[celli].z();

    // filter radius ”R”
    radius = 0.2 * Foam::sqrt(Foam::sqrt(Vcell[celli]));

    // find points corresponding to celli
    const labelList& cellPoints = mesh.cellPoints()[celli];
    forAll (cellPoints, pointi)
    {
        // identify all cells that use this point
        const labelList& pointCells = mesh.pointCells()[cellPoints[pointi]];
        forAll (pointCells, pointCelli)
        {
            // check for each cell using this point if ...
            // (a) ... it is not already in our list of neighbours of celli
            // (b) ... this cell is not celli itself
            // (c) ... cell is in same layer as celli
            if (
                (findIndex(N_i_sameLayer, pointCells[pointCelli]) == −1)
             && (pointCells[pointCelli] != celli)
             && (mesh.C()[pointCells[pointCelli]].x() > (xCell − SMALL))
             && (mesh.C()[pointCells[pointCelli]].x() < (xCell + SMALL))
        )
        {
            N_i_sameLayer.append(pointCells[pointCelli]);
            distance = Foam::sqrt
                (
                    sqr(mesh.C()[pointCells[pointCelli]].x()-xCell)
                + sqr(mesh.C()[pointCells[pointCelli]].y()-yCell)
                + sqr(mesh.C()[pointCells[pointCelli]].z()-zCell)
                );
            w_i_sameLayer.append(radius-distance);
        }

        // check for each cell using this point if ...
        // (a) ... it is not already in our list of neighbours of celli
        // (b) ... this cell is not celli itself
        // (c) ... cell is in the layer below celli
        if (
            (findIndex(N_i_lowerLayer, pointCells[pointCelli]) == -1)
         && (pointCells[pointCelli] != celli)
         && (mesh.C()[pointCells[pointCelli]].x() < (xCell - SMALL))
        )
        {
            N_i_lowerLayer.append(pointCells[pointCelli]);
            distance = Foam::sqrt
                (
                    sqr(mesh.C()[pointCells[pointCelli]].x()-xCell)
                + sqr(mesh.C()[pointCells[pointCelli]].y()-yCell)
                + sqr(mesh.C()[pointCells[pointCelli]].z()-zCell)
                );
            w_i_lowerLayer.append(radius-distance);
        }

        // check for each cell using this point if ...
        // (a) ... it is not already in our list of neighbours of celli
        // (b) ... this cell is not celli itself
        // (c) ... cell is in the layer above celli
        if (
            (findIndex(N_i_upperLayer, pointCells[pointCelli]) == -1)
         && (pointCells[pointCelli] != celli)
         && (mesh.C()[pointCells[pointCelli]].x() > (xCell + SMALL))
        )
        {
            N_i_upperLayer.append(pointCells[pointCelli]);
            distance = Foam::sqrt
                (
                    sqr(mesh.C()[pointCells[pointCelli]].x()-xCell)
                + sqr(mesh.C()[pointCells[pointCelli]].y()-yCell)
                + sqr(mesh.C()[pointCells[pointCelli]].z()-zCell)
                );
            w_i_upperLayer.append(radius-distance);
        }
    }

}

// NOTE: we now have to remain in the order defined by sameLayer and lowerLayer
// first cell in neighbour is i
neighbour[celli].append(celli);
neighbour[celli].append(N_i_sameLayer);
neighbour[celli].append(N_i_lowerLayer);
```

```
allNeighbours[celli].append(celli);
allNeighbours[celli].append(N_i_sameLayer);
allNeighbours[celli].append(N_i_lowerLayer);
allNeighbours[celli].append(N_i_upperLayer);
weights[celli].append(0.0);
weights[celli].append(w_i_sameLayer);
weights[celli].append(w_i_lowerLayer);
weights[celli].append(w_i_upperLayer);

// need this for Part II of minLengthConstraint precomputations
sameLayer[celli].append(N_i_sameLayer);

/* ---- Assembly of B_i for (in-plane) non-cartesian mesh ---- */

// assign B_xRow
// n1 = number of cells that get + Label (add 1 to include celli)
n1 = N_i_sameLayer.size() + 1;
init = 1.0/n1;
List<scalar> B_xRow_sameLayer(n1, init);
B_xRow[celli].append(B_xRow_sameLayer);
// n2 = number of cells that get - Label
n2 = N_i_lowerLayer.size();
if (n2 > 0)
{
    init = -1.0/n2;
    List<scalar> B_xRow_lowerLayer(n2, init);
    B_xRow[celli].append(B_xRow_lowerLayer);
}

// assign B_yRow, B_zRow
n3=0, n4=0, n5=0, n6=0;
init = 0.0;
List<scalar> B_yRow_sameLayer(n1-1, init);
List<scalar> B_yRow_lowerLayer(n2, init);
List<scalar> B_zRow_sameLayer(n1-1, init);
List<scalar> B_zRow_lowerLayer(n2, init);
forAll (N_i_sameLayer, neighi)
{
    yNeigh = mesh.C()[N_i_sameLayer[neighi]].y();
    zNeigh = mesh.C()[N_i_sameLayer[neighi]].z();

    if (yNeigh > (yCell + SMALL))
    {
        n3++;
        B_yRow_sameLayer[neighi] = 1.0;
    }
    if (yNeigh < (yCell - SMALL))
    {
        n4++;
        B_yRow_sameLayer[neighi] = -1.0;
    }
    if (zNeigh > (zCell + SMALL))
    {
        n5++;
        B_zRow_sameLayer[neighi] = 1.0;
    }
    if (zNeigh < (zCell - SMALL))
    {
        n6++;
        B_zRow_sameLayer[neighi] = -1.0;
```

```
}

//precomputations for minLengthConstraint PART I
const labelList& faceNeighs = mesh.cellCells()[celli];
const labelList& neighFaceNeighs
    = mesh.cellCells()[N_i_sameLayer[neighi]];
if ((yNeigh > (yCell - SMALL)) && (zNeigh < (zCell + SMALL)))
{
    if(findIndex(M_1[celli], N_i_sameLayer[neighi]) == -1)
    {
        M_1[celli].append(N_i_sameLayer[neighi]);
    }
    forAll(faceNeighs, faceNeighi)
    {
        // if the cell is in faceNeighs list of both celli and neighi
        // AND the cell has not been added to M yet
        if
        (
         (findIndex(neighFaceNeighs, faceNeighs[faceNeighi]) != -1)
         && (findIndex(M_1[celli], faceNeighs[faceNeighi]) == -1)
        )
        {
            M_1[celli].append(faceNeighs[faceNeighi]);
        }
    }
}
if ((yNeigh < (yCell + SMALL)) && (zNeigh < (zCell + SMALL)))
{
    if(findIndex(M_2[celli], N_i_sameLayer[neighi]) == -1)
    {
        M_2[celli].append(N_i_sameLayer[neighi]);
    }
    forAll(faceNeighs, faceNeighi)
    {
        // if the cell is in faceNeighs list of both celli and neighi
        // AND the cell has not been added to M yet
        if
        (
         (findIndex(neighFaceNeighs, faceNeighs[faceNeighi]) != -1)
         && (findIndex(M_2[celli], faceNeighs[faceNeighi]) == -1)
        )
        {
            M_2[celli].append(faceNeighs[faceNeighi]);
        }
    }
}
if ((yNeigh < (yCell + SMALL)) && (zNeigh > (zCell - SMALL)))
{
    if(findIndex(M_3[celli], N_i_sameLayer[neighi]) == -1)
    {
        M_3[celli].append(N_i_sameLayer[neighi]);
    }
    forAll(faceNeighs, faceNeighi)
    {
        // if the cell is in faceNeighs list of both celli and neighi
        // AND the cell has not been added to M yet
        if
        (
         (findIndex(neighFaceNeighs, faceNeighs[faceNeighi]) != -1)
         && (findIndex(M_3[celli], faceNeighs[faceNeighi]) == -1)
```

```
            )
            {
                M_3[celli].append(faceNeighs[faceNeighi]);
            }
        }
    }
    if ((yNeigh > (yCell - SMALL)) && (zNeigh > (zCell - SMALL)))
    {
        if(findIndex(M_4[celli], N_i_sameLayer[neighi]) == -1)
        {
            M_4[celli].append(N_i_sameLayer[neighi]);
        }
        forAll(faceNeighs, faceNeighi)
        {
            // if cell is in faceNeighs list of both celli and neighi
            // AND the cell has not been added to M yet
            if
            (
             (findIndex(neighFaceNeighs, faceNeighs[faceNeighi]) != -1)
             && (findIndex(M_4[celli], faceNeighs[faceNeighi]) == -1)
            )
            {
                M_4[celli].append(faceNeighs[faceNeighi]);
            }
        }
    }
}

forAll (N_i_lowerLayer, neighi)
{
    yNeigh = mesh.C()[N_i_lowerLayer[neighi]].y();
    zNeigh = mesh.C()[N_i_lowerLayer[neighi]].z();

    if (yNeigh > (yCell + SMALL))
    {
        n3++;
        B_yRow_lowerLayer[neighi] = 1.0;
    }
    if (yNeigh < (yCell - SMALL))
    {
        n4++;
        B_yRow_lowerLayer[neighi] = -1.0;
    }
    if (zNeigh > (zCell + SMALL))
    {
        n5++;
        B_zRow_lowerLayer[neighi] = 1.0;
    }
    if (zNeigh < (zCell - SMALL))
    {
        n6++;
        B_zRow_lowerLayer[neighi] = -1.0;
    }
}

// every cell found for n3 gets assigned +0.5/n3 for B_yRow,
// and every cell found for n4 gets assigned -0.5/n4 for B_yRow
// every cell found for n5 gets assigned +0.5/n5 for B_zRow,
// and every cell found for n6 gets assigned -0.5/n6 for B_zRow
B_yRow[celli].append(B_yRow_sameLayer);
```

```
    B_yRow[celli].append(B_yRow_lowerLayer);
    forAll (B_yRow[celli], neighi)
    {
        if (B_yRow[celli][neighi] > 0)
        {
            B_yRow[celli][neighi] *= 0.5/n3;
        }
        if (B_yRow[celli][neighi] < 0)
        {
            B_yRow[celli][neighi] *= 0.5/n4;
        }
    }
    B_zRow[celli].append(B_zRow_sameLayer);
    B_zRow[celli].append(B_zRow_lowerLayer);
    forAll (B_zRow[celli], neighi)
    {
        if (B_zRow[celli][neighi] > 0)
        {
            B_zRow[celli][neighi] *= 0.5/n5;
        }
        if (B_zRow[celli][neighi] < 0)
        {
            B_zRow[celli][neighi] *= 0.5/n6;
        }
    }

}

// precomputations for minLengthConstraint PART II;
// already found first-level neighs in loop above,
// now we need to append with second-level neighbours here
// (can't do it earlier because this requires all neighbours)
label currentNeigh, currentNeighNeigh;
forAll (alpha, celli)
{

    yCell = mesh.C()[celli].y();
    zCell = mesh.C()[celli].z();
    DynamicList<label> tmp1, tmp2, tmp3, tmp4;
    forAll (M_1[celli], neighi)
    {
        currentNeigh = M_1[celli][neighi];
        yNeigh = mesh.C()[currentNeigh].y();
        zNeigh = mesh.C()[currentNeigh].z();
        // these are first degree neighbours, now find their neighbours
        // and put them in some tmp-structure to append to M_1 later
        forAll (sameLayer[currentNeigh], neighNeighi)
        {
            currentNeighNeigh = sameLayer[currentNeigh][neighNeighi];
            yNeighNeigh = mesh.C()[currentNeighNeigh].y();
            zNeighNeigh = mesh.C()[currentNeighNeigh].z();
            // criteria 2, 3 and 4 remove celli and any duplicates
            if (((yNeighNeigh > (yNeigh-SMALL)) && (zNeighNeigh < (zNeigh+SMALL)))
                && (currentNeighNeigh != celli)
                && (findIndex(tmp1, currentNeighNeigh) == -1)
                && (findIndex(M_1[celli], currentNeighNeigh) == -1))
            {
                tmp1.append(currentNeighNeigh);
            }
        }
```

```
}
M_1[celli].append(tmp1);

forAll (M_2[celli], neighi)
{
    currentNeigh = M_2[celli][neighi];
    yNeigh = mesh.C()[currentNeigh].y();
    zNeigh = mesh.C()[currentNeigh].z();
    // these are first degree neighbours, now find their neighbours
    // and put them in some tmp-structure to append to M_2 later
    forAll (sameLayer[currentNeigh], neighNeighi)
    {
        currentNeighNeigh = sameLayer[currentNeigh][neighNeighi];
        yNeighNeigh = mesh.C()[currentNeighNeigh].y();
        zNeighNeigh = mesh.C()[currentNeighNeigh].z();
        // criteria 2, 3 and 4 remove celli and any duplicates
        if (((yNeighNeigh < (yNeigh+SMALL)) && (zNeighNeigh < (zNeigh+SMALL)))
            && (currentNeighNeigh != celli)
            && (findIndex(tmp2, currentNeighNeigh) == -1)
            && (findIndex(M_2[celli], currentNeighNeigh) == -1))
        {
            tmp2.append(currentNeighNeigh);
        }
    }
}
M_2[celli].append(tmp2);

forAll (M_3[celli], neighi)
{
    currentNeigh = M_3[celli][neighi];
    yNeigh = mesh.C()[currentNeigh].y();
    zNeigh = mesh.C()[currentNeigh].z();
    // these are first degree neighbours, now find their neighbours
    // and put them in some tmp-structure to append to M_3 later
    forAll (sameLayer[currentNeigh], neighNeighi)
    {
        currentNeighNeigh = sameLayer[currentNeigh][neighNeighi];
        yNeighNeigh = mesh.C()[currentNeighNeigh].y();
        zNeighNeigh = mesh.C()[currentNeighNeigh].z();
        // criteria 2, 3 and 4 remove celli and any duplicates
        if (((yNeighNeigh < (yNeigh+SMALL)) && (zNeighNeigh > (zNeigh-SMALL)))
            && (currentNeighNeigh != celli)
            && (findIndex(tmp3, currentNeighNeigh) == -1)
            && (findIndex(M_3[celli], currentNeighNeigh) == -1))
        {
            tmp3.append(currentNeighNeigh);
        }
    }
}
M_3[celli].append(tmp3);

forAll (M_4[celli], neighi)
{
    currentNeigh = M_4[celli][neighi];
    yNeigh = mesh.C()[currentNeigh].y();
    zNeigh = mesh.C()[currentNeigh].z();
    // these are first degree neighbours, now find their neighbours
    // and put them in some tmp-structure to append to M_4 later
    forAll (sameLayer[currentNeigh], neighNeighi)
    {
            currentNeighNeigh = sameLayer[currentNeigh][neighNeighi];
            yNeighNeigh = mesh.C()[currentNeighNeigh].y();
            zNeighNeigh = mesh.C()[currentNeighNeigh].z();
            // criteria 2, 3 and 4 remove celli and any duplicates
            if (((yNeighNeigh > (yNeigh-SMALL)) && (zNeighNeigh > (zNeigh-SMALL)))
                && (currentNeighNeigh != celli)
                && (findIndex(tmp4, currentNeighNeigh) == -1)
                && (findIndex(M_4[celli], currentNeighNeigh) == -1))
            {
                tmp4.append(currentNeighNeigh);
            }
        }
    }
    M_4[celli].append(tmp4);

}
```

A.2.2 optimiseGeometry.H

```
Info << "Running geometric constraints algorithm" << endl;

/* ------------------------------------------------------------------ */
/* Use the linear density filter and design variables to get the
 intermediate densities (Equation 5.1) */

scalar sum_numerator = 0.0, sum_denominator = 0.0, rho_m;
label cellm;

forAll (rho_im, celli)
{
    forAll (allNeighbours[celli], m)
    {
        cellm = allNeighbours[celli][m];
        // alpha is scaled to [0, 1]
        rho_m = alpha[cellm]/alphaMax.value();
        sum_numerator += weights[celli][m] * Vcell[cellm] * rho_m;
        sum_denominator += weights[celli][m] * Vcell[cellm];
    }

    rho_im[celli] = max(min(sum_numerator / sum_denominator, 1.0), 0.0);
    sum_numerator = 0.0, sum_denominator = 0.0;
}


/* ------------------------------------------------------------------ */
/* Use the nonlinear density filter and intermediate densities to get
 the physical densities (Equation 5.2) */

// eta: threshold parameter
scalar eta = 0.2;
// beta indicates how aggressive the filter is
scalar beta = 0.2;

forAll (rho_physical, celli)
{
    // NOTE [ZC20] instructs to include 0 and 1 as boundaries for rho_im,
    // but I found this leads to local errors, this one is more robust
    if (rho_im[celli] <= eta)
    {
        rho_physical[celli] =
        (
            eta *
            (
                Foam::exp(-beta * (1 - (rho_im[celli]/eta)))
              - (1 - (rho_im[celli]/eta)) * Foam::exp(-beta)
            )
        );
    }
    else
    {
        rho_physical[celli] =
        (
            (1-eta) *
            (
                1 - Foam::exp(-beta * (rho_im[celli]-eta) / (1-eta))
              + ((rho_im[celli]-eta) * Foam::exp(-beta) / (1-eta))
            ) + eta
        );
    }
}


/* ------------------------------------------------------------------- */
/* Calculate geometric constraints using rho_physical */

// Overhang angle constraint (Equation 5.10)

scalarField xi_a(mesh.nCells(), 0.0);
scalarField xi_b(mesh.nCells(), 0.0);
scalarField xi_c(mesh.nCells(), 0.0);
scalarField cosAngle(mesh.nCells());
scalar cos_betaCrit = Foam::cos(0.78); // 45 degrees
label neigh;

forAll (alpha, celli)
{
    xCell = mesh.C()[celli].x();
    yCell = mesh.C()[celli].y();
    zCell = mesh.C()[celli].z();
    forAll (neighbour[celli], neighi)
    {
        neigh = neighbour[celli][neighi];
        xi_a[celli] += B_xRow[celli][neighi] * rho_physical[neigh];
        xi_b[celli] += B_yRow[celli][neighi] * rho_physical[neigh];
        xi_c[celli] += B_zRow[celli][neighi] * rho_physical[neigh];
    }
    // compute cosAngle (alpha in literature) = c/sqrt(a²+b²+c²)
    cosAngle[celli] =
    (
        xi_a[celli] // we use xi_a here because we extrude in x-direction
      / (Foam::sqrt(sqr(xi_a[celli])+sqr(xi_b[celli])+sqr(xi_c[celli]))+SMALL)
    );

    // overhangCriterion < 0: these are the cells that are okay to be added
    // (based on the current structure)
    overhangCriterion[celli] =
    (
        (cosAngle[celli]-cos_betaCrit)
      * Foam::sqrt(sqr(xi_a[celli]) + sqr(xi_b[celli]) + sqr(xi_c[celli]))
    );
}


// Horizontal minimum length and connectivity constraint (Equation 5.20)

scalarField rho_phase = rho_physical;
List<scalar> horLength(4, 0.0);
scalar horLength_1, horLength_2, horLength_3, horLength_4;

forAll (alpha, celli)
{
    horLength_1 = 0, horLength_2 = 0, horLength_3 = 0, horLength_4 = 0;

    // (Equation 5.19)
    if (rho_physical[celli] <= 0.5)
    {
        rho_phase[celli] = 1 - rho_physical[celli];
    }

    forAll (M_1[celli], i)
    {
        horLength_1 += (1 - rho_phase[M_1[celli][i]]);
    }
    // M_1[celli].size() is NOT always >= 1! (boundary region)
    horLength[0] = rho_phase[celli] - (horLength_1/max(1,M_1[celli].size()));

    forAll (M_2[celli], i)
    {
        horLength_2 += (1 - rho_phase[M_2[celli][i]]);
    }
    // M_2[celli].size() is NOT always >= 1! (boundary region)
    horLength[1] = rho_phase[celli] - (horLength_2/max(1,M_2[celli].size()));

    forAll (M_3[celli], i)
    {
        horLength_3 += (1 - rho_phase[M_3[celli][i]]);
    }
    // M_3[celli].size() is NOT always >= 1! (boundary region)
    horLength[2] = rho_phase[celli] - (horLength_3/max(1,M_3[celli].size()));

    forAll (M_4[celli], i)
    {
        horLength_4 += (1 - rho_phase[M_4[celli][i]]);
    }
    // M_4[celli].size() is NOT always >= 1! (boundary region)
    horLength[3] = rho_phase[celli] - (horLength_4/max(1,M_4[celli].size()));

    // minLengthCriterion should be smaller than 0 for at least one k!
    minLengthCriterion[celli] = min(horLength);
}

/* ------------------------------------------------------------------- */
/* Update design variables */

optimizationProperties.lookup("lambdaU") >> lambdaU;
optimizationProperties.lookup("lambdaT") >> lambdaT;
optimizationProperties.lookup("lambdaC") >> lambdaC;
// compute individual sensitivities
volScalarField sensU = Ua & U;
volScalarField sensT = Ta * (T - average(T));
volScalarField sensC = Conca * (Conc - average(Conc));

// proposed design variable update
alphaDiff =
(
    mesh.fieldRelaxationFactor("alpha")
  * (
        min
        (
            max
            (
                alpha + lambdaU*sensU + lambdaT*wT*sensT + lambdaC*wC*sensC
              , zeroAlpha
            )
          , alphaMax
        )
      - alpha
    )
);

// automatically reject all changes for scaffold structure
forAll(alpha, i)
{
    if (alphaFixed[i] == 1)
    {
        alpha[i] = alphaMax.value();
        alphaDiff[i] = 0;
    }
}

label proposedAddedCells = 0;
label proposedRemovedCells = 0;
label confirmedAddedCells = 0;
label confirmedRemovedCells = 0;
label sizeSolid = 0;

forAll(alpha, i)
{
    if (alphaDiff[i] > 0)
    {
        proposedAddedCells++;
    }
    if (alphaDiff[i] < 0)
    {
        proposedRemovedCells++;
    }
    if (alpha[i] > (0.5*alphaMax.value()))
    {
        sizeSolid++;
    }
}
Info<< "Number of solid cells before update: " << sizeSolid << endl;

zeroCells(alpha, inletCells);
//zeroCells(alpha, outletCells);

// Use geometric constraints to confirm or deny the proposed changes
forAll(alpha, i)
{
    if (alphaFixed[i] == 1)
    {
        alpha[i] = alphaMax.value();
        alphaDiff[i] = 0;
        continue;
    }

    // solid cell wants to stay solid, it is okay to build there
    // and it is happy there: confirm
    // fluid cell wants to become solid, it is okay to build there,
    // but it was also happy as fluid: confirm for now
    if
    (
        (alphaDiff[i] > 0)
     && (overhangCriterion[i] < 0) && (minLengthCriterion[i] <= 0)
    )
    {
        alpha[i] = min(alpha[i] + 5 * alphaDiff[i], alphaMax.value());
        confirmedAddedCells++;
        continue;
    }
    // fluid cell wants to become solid, it is okay to build there
    // and it was not happy as a fluid anyway: confirm
    if
    (
        (alphaDiff[i] > 0) && (alpha[i] < 0.5*alphaMax.value())
     && (overhangCriterion[i] < 0) && (minLengthCriterion[i] > 0)
    )
    {
        alpha[i] = min(alpha[i] + 5 * alphaDiff[i], alphaMax.value());
        confirmedAddedCells++;
        continue;
    }
    // solid cell wants to become fluid, although AM restrictions
    // would be fine with it staying solid: confirm for now
    // fluid cell wants to stay fluid and AM restrictions don't care: confirm
    if
    (
        (alphaDiff[i] < 0)
     && (overhangCriterion[i] < 0) && (minLengthCriterion[i] <= 0)
    )
    {
        alpha[i] = max(alpha[i] + 5 * alphaDiff[i], zeroAlpha.value());
        confirmedRemovedCells++;
        continue;
    }
    // solid cell wants to become fluid, overhang was fine there
    // but it was not happy in hor. length: confirm
    if
    (
        (alphaDiff[i] < 0) && (alpha[i] > 0.5*alphaMax.value())
     && (overhangCriterion[i] < 0) && (minLengthCriterion[i] > 0)
    )
    {
        alpha[i] = max(alpha[i] + 5 * alphaDiff[i], zeroAlpha.value());
        confirmedRemovedCells++;
        continue;
    }
    // solid cell wants to become fluid, overhang was not fine anyway:
    // confirm for now
    // fluid cell wants to stay fluid and AM restrictions are happy
    // this way: confirm
    if
    (
        (alphaDiff[i] < 0)
     && (overhangCriterion[i] >= 0) && (minLengthCriterion[i] <= 0)
    )
    {
        alpha[i] = max(alpha[i] + 5 * alphaDiff[i], zeroAlpha.value());
        confirmedRemovedCells++;
        continue;
    }
    // solid cell wants to become fluid, overhang was not fine there
    // and it was not happy in hor. length: confirm
    if
    (
        (alphaDiff[i] < 0) && (alpha[i] > 0.5*alphaMax.value())
     && (overhangCriterion[i] >= 0) && (minLengthCriterion[i] > 0)
    )
    {
        alpha[i] = max(alpha[i] + 5 * alphaDiff[i], zeroAlpha.value());
        confirmedRemovedCells++;
        continue;
    }
    // reject all other changes automatically and continue
}

// print final number of solid cells after application of constraints
sizeSolid = 0;
forAll(alpha, i)
{
    if (alpha[i] > (0.5*alphaMax.value()))
    {
        sizeSolid++;
    }
}
label rejectedAddedCells = proposedAddedCells - confirmedAddedCells;
label rejectedRemovedCells = proposedRemovedCells - confirmedRemovedCells;

Info << "    "
    << proposedAddedCells << " cells want to become solid. It was confirmed for "
    << confirmedAddedCells << " cells and rejected for "
    << rejectedAddedCells << " cells." << endl;
Info << "    "
    << proposedRemovedCells << " cells want to become fluid. It was confirmed for "
    << confirmedRemovedCells << " cells and rejected for "
    << rejectedRemovedCells << " cells." << endl;
Info << "Number of solid cells after update: " << sizeSolid << "\n" << endl;
```

A.3 Additional Evaluations from Use Case "Static Mixer"

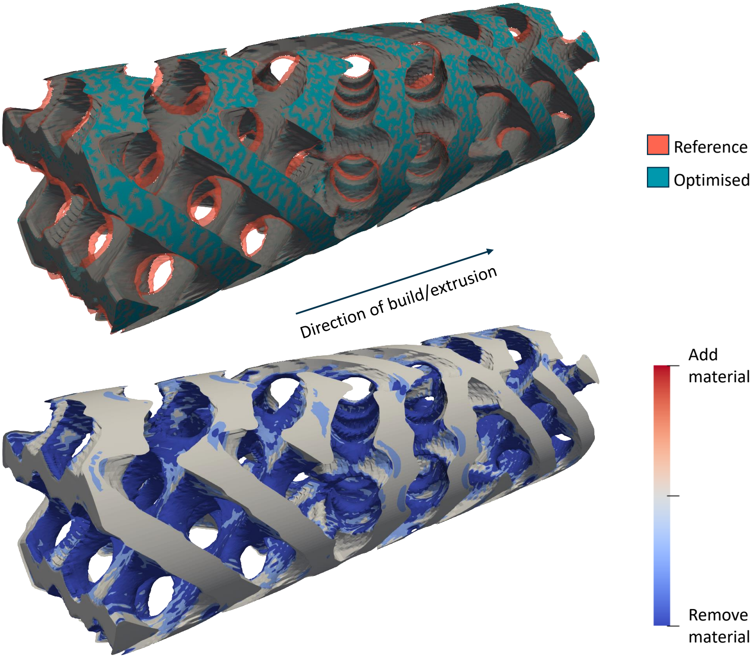


Figure A.1: Changes to the static mixer geometry after optimisation for minimal pressure drop.

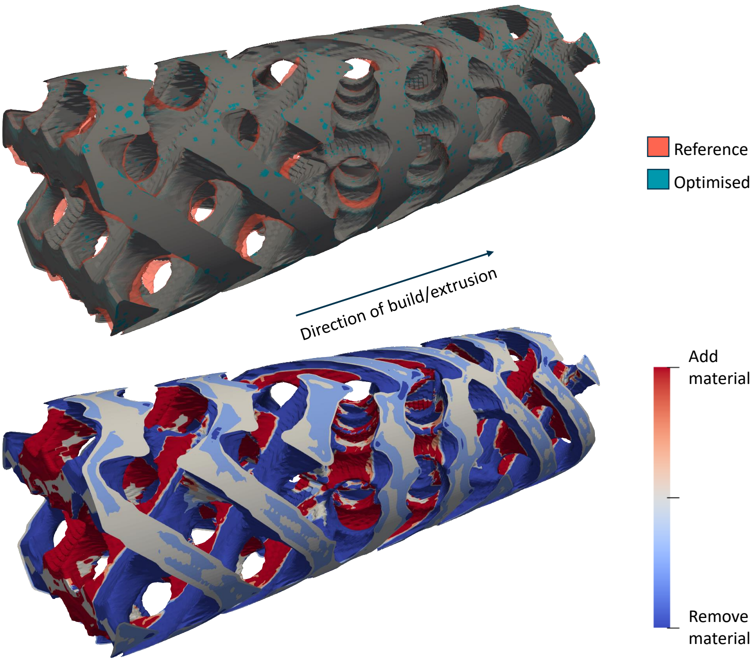


Figure A.2: Changes to the static mixer geometry after optimisation for maximal thermal mixing.

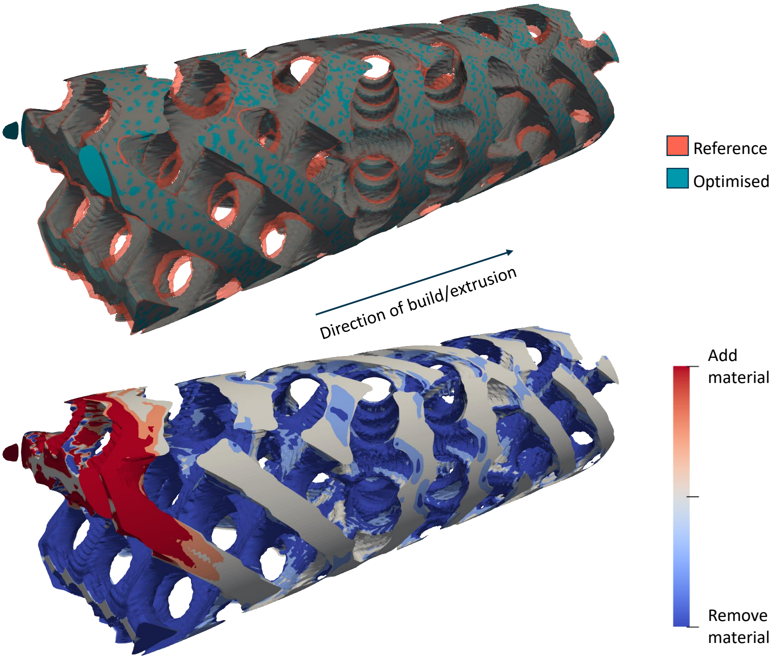
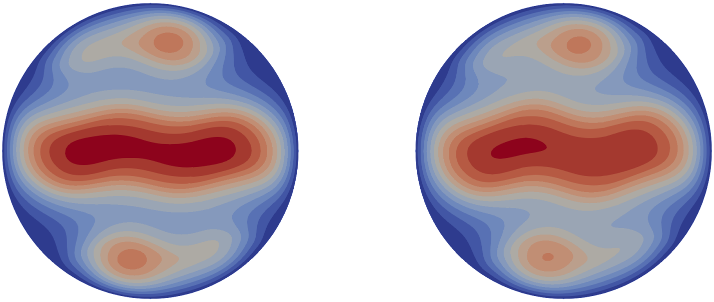
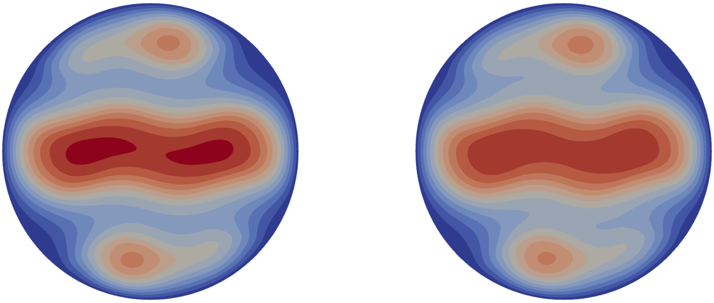


Figure A.3: Changes to the static mixer geometry after optimisation for maximal material mixing.



Geometry “M”
before optimisation

Geometry “P”
optimised for minimal pressure drop



Geometry “T”
optimised for maximal thermal mixing

Geometry “C”
optimised for maximal material mixing

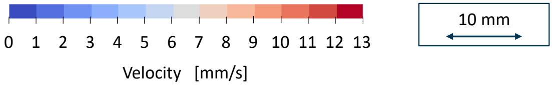


Figure A.4: Comparison of velocity distribution at the outlet of the mixer for the different objective functions.

A.4 Lab Trial Data from Use Case ”Static Mixer”

Time	Before mixer				After mixer				Throughput	Pressure drop
[hh:mm]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	[g/min]	[bar]
17:30	235.4	240.3	239.0	30.1	238.6	239.7	239.1	6.08	333.0	24.02
17:35	235.6	240.1	239.0	30.0	239.2	239.5	238.9	6.06		23.94
17:40	236.1	240.2	239.1	30.2	233.8	238.8	238.4	6.13	332.0	24.07
17:45	235.9	240.7	239.4	29.9	236.5	239.5	239.0	6.10	330.5	23.80
17:50	235.5	240.0	238.9	30.0	239.3	239.5	238.8	6.08	334.0	23.92
17:55	236.6	240.7	239.6	29.8	237.1	239.7	239.2	6.04		23.76
18:00	235.3	240.0	238.7	30.0	236.8	239.0	238.4	6.06	329.0	23.94
18:05	235.8	240.2	239.1	30.0	239.2	239.5	238.8	6.04	331.3	23.96
18:10	236.6	240.7	239.6	29.9	238.8	239.8	239.2	6.01	331.7	23.89
18:15	235.4	240.1	238.9	30.0	234.8	238.9	238.5	6.06		23.94

Table A.1: Lab trial data for static mixer geometry ”M” at 20 kg/h.

Time	Before mixer				After mixer				Throughput	Pressure drop
[hh:mm]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	[g/min]	[bar]
18:50	243.6	250.6	248.3	42.1	239.7	248.5	248.1	5.99	684.0	36.11
18:55	244.4	250.9	248.7	42.1	235.5	248.3	248.0	6.06	676.0	36.04
19:00	244.0	251.0	248.7	42.1	239.0	248.8	248.4	6.04	660.1	36.06
19:05	243.8	250.8	248.5	42.2	237.1	248.6	248.2	6.11	669.3	36.09
19:10	244.6	251.0	248.8	42.1	238.9	248.8	248.4	6.11	667.5	35.99
19:15	243.7	250.5	248.3	42.2	236.2	248.2	247.8	6.15	668.7	36.05
19:20	243.2	250.4	248.1	42.3	239.7	248.3	248.0	6.13	671.0	36.17
19:25	244.7	250.9	248.8	42.2	237.7	248.7	248.3	6.11	678.5	36.09
19:30	243.8	250.6	248.4	42.0	239.0	248.5	248.0	6.15	665.6	35.85
19:35	243.4	250.5	248.3	42.4	236.3	248.2	247.9	6.15	664.0	36.25

Table A.2: Lab trial data for static mixer geometry ”M” at 40 kg/h.

Time	Before mixer				After mixer				Throughput	Pressure drop
[hh:mm]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	[g/min]	[bar]
10:30	236.6	240.9	239.7	22.7	233.6	239.3	238.8	5.95	327.5	16.75
10:35	235.7	240.4	239.0	22.9	235.0	239.0	238.4	5.99	325.6	16.91
10:40	235.6	240.0	238.8	22.9	238.4	239.2	238.4	5.94	327.0	16.96
10:45	237.1	241.8	240.5	22.7	234.4	239.7	239.2	5.94	325.0	16.76
10:50	235.8	240.9	239.4	22.8	236.6	239.6	238.9	5.90	322.4	16.90
10:55	236.3	240.4	239.3	22.9	235.0	239.0	238.4	5.94	324.1	16.96
11:00	235.0	239.5	238.3	23.0	236.3	238.6	237.9	5.94	324.6	17.06
11:05	236.8	241.5	240.2	22.7	234.2	239.7	239.1	5.92	323.6	16.78
11:10	236.0	240.3	239.1	22.7	237.1	239.4	238.7	5.27	322.0	17.43
11:15	234.9	239.8	238.5	23.0	233.9	238.4	237.8	5.33	322.0	17.67

Table A.3: Lab trial data for static mixer geometry ”P” at 20 kg/h.

Time	Before mixer				After mixer				Throughput	Pressure drop
[hh:mm]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	[g/min]	[bar]
11:45	244.3	250.6	248.4	33.8	235.6	247.7	247.4	5.92	674.0	27.88
11:50	243.2	250.0	247.7	34.0	235.8	247.4	247.1	5.94	673.1	28.06
11:55	244.0	250.4	248.3	33.9	238.4	247.9	247.4	5.95	677.2	27.95
12:00	243.7	250.3	248.1	34.0	235.7	247.5	247.3	5.97	674.5	28.03
12:05	243.1	250.1	247.8	33.9	237.2	247.6	247.3	5.92	675.6	27.98
12:10	243.8	250.2	248.0	34.2	237.8	247.6	247.3	5.92	675.5	28.28
12:15	243.2	250.3	248.0	34.0	236.2	247.6	247.3	5.95	674.0	28.05
12:20	243.8	250.1	248.0	34.0	235.9	247.4	247.1	5.94	672.5	28.06
12:25	243.5	250.1	247.8	34.0	238.1	247.6	247.3	5.97	672.0	28.03
12:30	243.1	250.0	247.7	34.2	235.9	247.4	247.1	6.01	673.5	28.19

Table A.4: Lab trial data for static mixer geometry ”P” at 40 kg/h.

Time [hh:mm]	Before mixer				After mixer				Throughput [g/min]	Pressure drop [bar]
	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]		
14:20	236.2	240.6	239.4	27.4	231.4	238.8	238.8	6.17	333.3	21.23
14:25	235.1	239.9	238.7	27.6	234.7	238.4	238.3	6.20	327.3	21.40
14:30	236.1	239.9	238.9	27.5	236.8	238.8	238.6	6.19	331.5	21.31
14:35	235.6	240.3	239.0	27.5	235.9	238.9	238.8	6.17	330.5	21.33
14:40	236.1	240.2	239.2	27.6	234.6	238.5	238.4	6.20	329.0	21.40
14:45	235.9	240.4	239.1	27.6	235.9	238.9	238.8	6.17	328.5	21.43
14:50	235.6	240.4	239.1	27.6	234.4	238.8	238.7	6.19	333.3	21.41
14:55	237.2	241.7	240.5	27.5	235.7	239.7	239.7	6.17	331.8	21.33
15:00	236.2	241.1	239.7	27.4	235.3	239.6	239.5	6.13	331.3	21.27
15:05	236.6	240.7	239.6	27.5	233.9	238.9	238.9	6.13	330.0	21.37

Table A.5: Lab trial data for static mixer geometry ”T” at 20 kg/h.

Time	Before mixer				After mixer				Throughput	Pressure drop
[hh:mm]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	T_1 [°C]	T_2 [°C]	T_3 [°C]	p [bar]	[g/min]	[bar]
15:35	243.7	250.4	248.2	39.0	235.7	247.8	248.0	6.06	670.6	32.94
15:40	243.6	250.4	248.2	38.9	236.1	247.9	248.0	6.06	668.4	32.84
15:45	244.2	250.7	248.4	38.9	236.4	248.0	248.3	6.04	669.9	32.86
15:50	243.1	250.2	247.8	39.0	236.4	247.6	247.8	6.08	673.2	32.92
15:55	244.2	250.5	248.4	38.9	237.9	248.0	248.2	6.04	668.6	32.86
16:00	243.6	250.4	248.0	39.0	236.8	247.8	248.0	6.03	670.0	32.97
16:05	243.3	250.2	247.9	39.1	235.6	247.4	247.7	6.04	671.0	33.06
16:10	244.0	250.3	248.1	38.9	237.6	247.9	248.0	6.03	670.0	32.87
16:15	243.0	250.0	247.7	39.1	235.9	247.4	247.7	5.99	668.5	33.11
16:20	244.1	250.4	248.3	38.8	237.9	247.9	248.0	5.95	669.0	32.85

Table A.6: Lab trial data for static mixer geometry ”T” at 40 kg/h.

Lebenslauf

Persönliche Angaben

Name	Jana Sasse
Geburtsdatum	19. September 1995
Geburtsort	München

Bildungsweg

2020–2025	**Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen** Promotionsstudium
2018–2020	**Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen** Masterstudium (M.Sc.) Computational Engineering Science, Vertiefungsrichtung Ingenieurswissenschaften
2013–2018	**Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen** Bachelorstudium (B.Sc.) Computational Engineering Science, Vertiefungsrichtung Strömung und Technische Verbrennung
2005–2013	**Bischöfliches Gymnasium St. Ursula, Geilenkirchen** Allgemeine Hochschulreife

Beruflicher Werdegang

2020–2025	**Institut für Kunststoffverarbeitung (IKV) an der RWTH Aachen** Wissenschaftlicher Mitarbeiter und Leitung der Arbeitsgruppe ”Extrusionswerkzeuge | CAE”
2018–2020	**Institut für Angewandte Medizintechnik, Lehr- und Forschungsgebiet Kardiovaskuläre Technik (RWTH Aachen)** Wissenschaftliche Hilfskraft
2017	**enmodes GmbH** Praktikum
2016-2017	**Institut für Geometrie und Praktische Mathematik (RWTH Aachen)** Studentische Hilfskraft

Aachen, im Februar 2025