Travail de fin d'études et stage[BR]- Travail de fin d'études : Towards the Integration of Casting Simulations to a Structural Topology Optimization Framework[BR]- Stage d'insertion professionnelle

Auteur : Yi Mon Theint,
Promoteur(s) : Tromme, Emmanuel
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil mécanicien, à finalité spécialisée en technologies durables en automobile
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/19569

Towards the Integration of Casting Simulations to a Structural Topology Optimization Framework

Author: Yi Mon THEINT (S211680)

Academic supervisor: Emmanuel TROMME

Master thesis for obtaining the degree of Master of Science in "Mechanical Engineering"

University of Liège
Faculty of Applied Sciences
Academic year 2023-2024

Acknowledgments

This thesis work was carried out over a period of four months. I am very grateful for having this opportunity to work on an intriguing topic within the field of topology optimization. I would like to express my deepest appreciation to all those who provided me the possibility to fulfill this project.

I would first like to thank my thesis supervisor, Prof. Emmanuel Tromme. His knowledge, expertise and guidance throughout the project have been invaluable and has helped me to keep the project on the right track. I am particularly grateful for Prof. Tromme's patience and encouragement throughout this journey.

I would also like to thank Prof. Pierre Duysinx for providing me with all the necessary facilities and tools to carry out the work within the Faculty's research unit. Moreover, I am deeply appreciative of Prof. Duysinx's invaluable advice and support throughout my academic journey. His guidance has been transformative, enabling me to adapt effectively to the new academic environment.

I extend my sincere appreciation to Dr. Ir. Eduardo Fernandez for his invaluable contributions, his availability in moments of need, and his exceptional kindness and empathy. I am sincerely thankful for sharing his insights and valuable advice on my research endeavors.

I would like to extend my heartfelt gratitude to the other professors and dedicated teaching assistants I had the privilege of meeting and learning from during my Master's studies at the university. The guidance, knowledge, and expertise received from them have been invaluable in shaping my academic journey.

Lastly, I wish to convey my deepest gratitude to my parents, whose unwavering support and encouragement have been a constant source of strength throughout my years of study. Their boundless love, guidance, and sacrifices have been the cornerstone of my academic journey, and I am profoundly grateful for their belief in me.

Additionally, I extend my heartfelt appreciation to my siblings and friends for their enduring support and understanding. Their collective support, whether in moments of celebration or when faced with academic hurdles, has been a treasured gift.

Abstract

Cost-effective lightweight design emerges as a pivotal focus for the automotive industry's future. Global competitiveness, stringent regulatory standards, and the integration of weight-intensive electronic elements in modern propulsion systems require the development of lighter, more efficient chassis components.

With this perspective, topology optimization is extensively applied for the design of lightweight components. The casting process stands as a time and cost-efficient method for automotive mass production, widely adapted within the industry. Typically, weight optimization process does not consider castability, leading to later-stage modifications . These modifications incur additional time spent for manufacturability and often result in a heavier design than the initially optimized one. This thesis introduces an optimization process that optimizes weight and castability concurrently during the early design phase, offering a solution to this challenge.

The study focuses on incorporating casting simulations into previously developed topology optimization framework, which involves accommodating geometric casting constraints, including directional molding, split-drawing, minimum member size, and draft angle considerations. A previously established Python code, designed for topology optimization incorporating casting constraints, offers flexibility and scalability. This code utilizes the open-source FEniCS Project as its finite element software, enabling the utilization of PETSc as a backend for linear algebra operations to enhance efficiency. A casting simulation is performed using OpenFOAM, focusing on flows involving heat transfer. A dedicated solver, employing the continuous adjoint approach, is implemented within OpenFOAM to calculate sensitivities. These outcomes are then merged with the topology optimization optimizer in FEniCS, leading to the establishment of an integrated optimization approach.

The established solver undergoes validation by comparing the sensitivities computed with the finite difference method. Subsequently, the integrated approach's validation is carried out through a 2-dimensional cantilever beam problem.

Keywords : topology optimization, manufacturing, casting, constraints, flows with heat transfer, OpenFOAM

Contents

List of Figures

List of Tables

1 Introduction

Today's automotive industry is continuously evolving with a focus on enhancing efficiency, performance, and sustainability. One of the key frontiers driving innovation in this sector is the pursuit of lightweight design and product development. The need for lightweight solutions in automotive design is propelled by various factors from stringent regulatory requirements aimed at reducing emissions to the growing demand for energy-efficient vehicles. Lighter vehicles not only improve fuel efficiency but also contribute to enhanced flexibility, better handling, and overall enhanced performance. Over the years, there have been relentless innovation and transformative strategies driving the evolution of vehicles towards a lighter, more efficient future.

1.1 Background

Casting is a widely used manufacturing process in automotive industries to produce various parts and components which involves pouring molten material, usually metals or alloys like aluminum, iron, steel, or magnesium, into a mold cavity and allowing it to solidify into the desired shape. In order to maximize the advantages of topology optimization, it is crucial to comply the manufacturing requirements.

Figure 1.1 illustrates the current design process for cast components. Currently, optimization focuses separately on weight and the casting process without considering castability in the early design stages. This approach often leads to additional mass being added to the component, primarily benefiting the casting process but resulting in a heavier design overall.

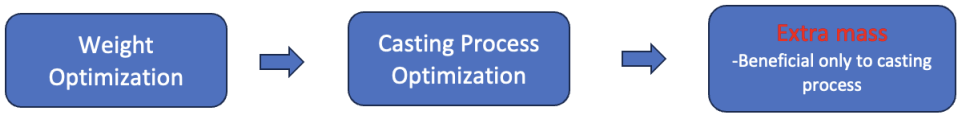


Figure 1.1: Current Design Process of Cast Components

In order to develop sustainable transport solutions, there is a need to enhance vehicle efficiency through weight reduction. However, there are numerous heavy casted components in vehicles, offering substantial opportunities for weight reduction through advanced computational-based design optimization. Nevertheless, optimizing design aspects separately for weight and the casting process will not sufficiently meet the weight reduction objectives. There is a necessity to simultaneously optimize the design for both weight and the casting process to effectively achieve the desired weight reduction targets.

Topology optimization is one of the key approaches for lightweight product development within the automotive industry. It plays a pivotal role in redefining how engineers approach the design process to achieve optimal weight reduction without compromising structural integrity or performance. Topology optimization utilizes advanced algorithms and computational tools to iteratively analyze and redistribute material within a given design space by systematically removing unnecessary material or strategically placing it where it is most needed. Although it achieves an optimal result with fulfilled constraints, it often results complex shapes that can be challenging to manufacture with traditional methods. Therefore, it is crucial to consider manufacturing constraints and manufacturability in the design process.

When focusing on cast parts, the initial constraints derived from optimization outcomes typically revolve around adhering to geometric guidelines. These guidelines involve preventing undercuts and hollow structures (18),(19), performing connectivity checks, ensuring minimal member sizes (20),(21), and controlling hole sizes (22). Nowadays, several of these limitations are integrated into topology optimization software, allowing for a fundamental usage of the optimization outcomes. However, a significant level of abstraction exists, demanding manual adjustments to the geometry to achieve a design suitable for casting. To fully leverage the potential of topology optimization and to capitalize on the expanded capabilities of simulation tools, emphasis is placed on employing casting simulations to optimize part designs. Initial approaches primarily focus on individual aspects, such as the filling or solidification behavior, without taking into account the mechanical properties simultaneously (23),(24). Subsequent methodologies involve multidisciplinary optimizations, integrating mechanical Finite Element Method (FEM) calculations with casting simulations through various interconnected methods. Compared to the filling behavior, there has been more approaches focusing on the solidification of the part. For instance, weighting factors are incorporated into the SIMP (Solid Isotropic Material with Penalization) method to optimize the cooling process and mitigate internal stresses in forming tools (27). Shape optimizations are executed to enhance both mechanical properties and solidification, utilizing thermal sensitivities (28). Additionally, the Level Set Method is applied to optimize cooling behavior (25),(26).

1.2 Objective

The objective of this thesis is to integrate casting simulations into the process of topology optimization while considering manufacturing constraints for the design of reliable cast components. By merging these methodologies, the aim is to apply their collective advantages. This integration intends to optimize part designs not only for mechanical performance but also for manufacturability, specifically focusing on cast parts within the automotive industry. The goal is to leverage advanced computational tools and algorithms to achieve optimal weight reduction and structural integrity while ensuring that the resultant designs are feasible for casting processes. Through this integration, the thesis aims to explore the potential of topology optimization enhanced by casting simulations to produce components that meet both performance criteria and manufacturing requirements, thereby advancing lightweight product development in automotive engineering.

The optimization process flow chart is shown in Figure 1.2. The topology optimization is integrated with geometric casting constraints and casting process simulation. The aim of an integrated optimization approach is to produce a conceptual design for a lightweight castable component. By effectively implementing this optimization-driven design process, there is a potential reduction in the time required for designing lightweight cast components. Simultaneously, it ensures adherence to strict structural standards and high-quality casting process requirements.

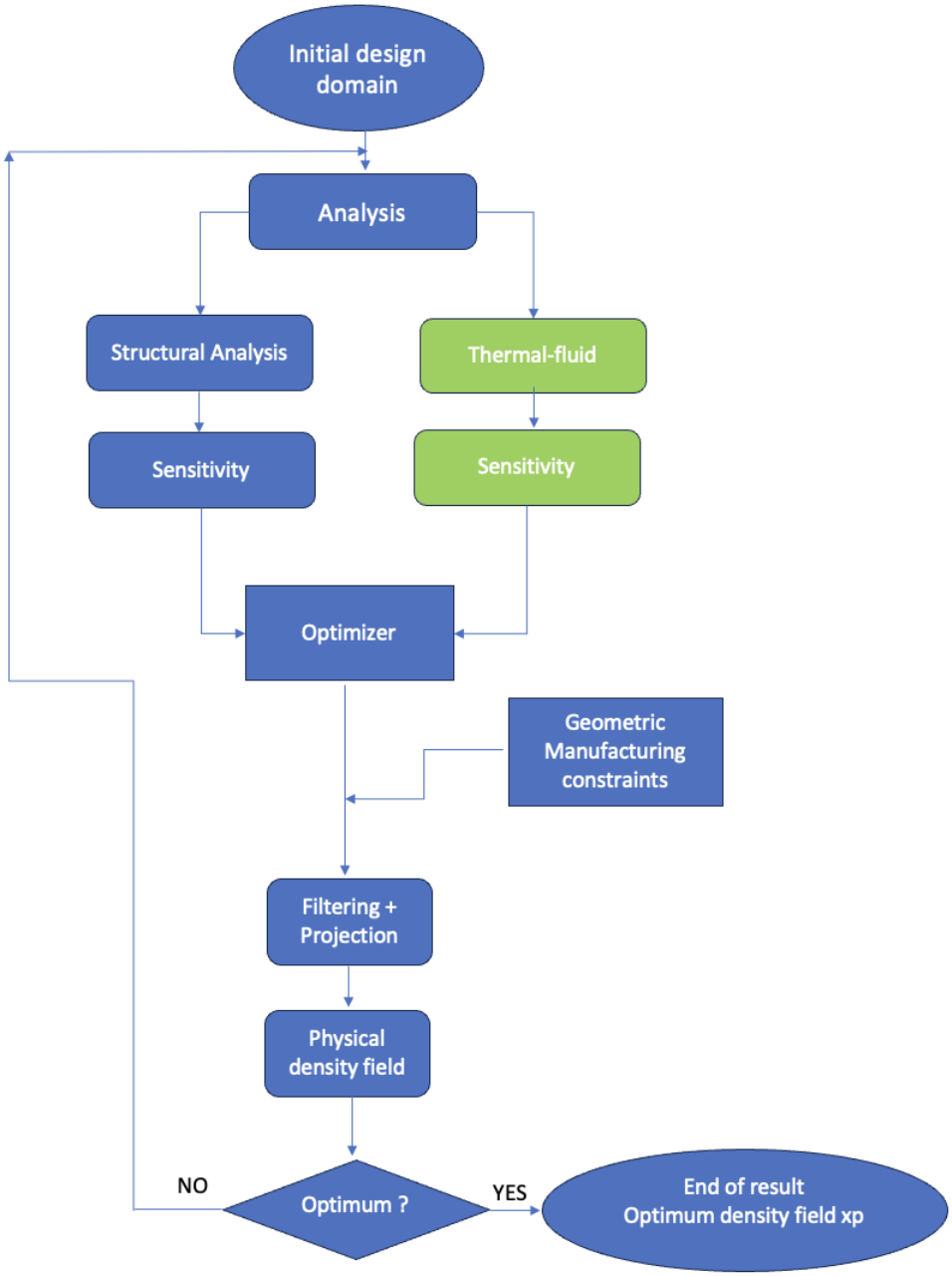


Figure 1.2: Overview of Optimization Driven Design Process

1.3 Methodologies

In the pursuit of ready-to-manufacture, lightweight components for mass production, a flexible and large-scale Python code had been developed for topology optimization with integrated casting constraints, the blue boxes in the flow chart, as in Figure 1.2. The method uses the open-source FEniCS Project as finite element software, allowing the usage of PETSc as linear algebra back-end for better efficiency and the geometric constraints for casting had been implemented in Python using the filter-based method proposed by M. Langelaar (35).

In order to perform casting related simulations with the purpose of enhancing manufacturability and design efficiency, the work is done in an open-source software package, OpenFOAM which allows for efficiently handling large-scale simulations and parallel processing capabilities. The computed sensitivities from OpenFOAM simulations will be integrated to Python code in order to further performing optimization with implemented geometric constraints.

1.4 Outline

The thesis report is organized into seven chapters. The first chapter introduces the thesis work, providing a comprehensive background to the research. The second chapter presents the theoretical aspects of topology optimization framework. In third chapter, it presents the casting process, its geometric constraints, and its integration with topology optimization methodologies, including casting simulation approaches. In fourth chapter, the mathematical formulation of thermal-fluid systems like casting is presented, detailing the equations and principles. Moving forward, the implementation of solvers within the OpenFOAM simulation software is discussed, demonstrating practical applications in fifth chapter. Subsequently, the sixth chapter focuses on validating the methods employed, outlining validation procedures and results analyses. Finally, the thesis is concluded by drawing conclusions from the findings and offering perspectives for future research work.

1.5 Limitations

Due to the complexed nature of the entire casting process, from filling to solidification, the simulation covers only a partial system, specifically, the focus is solely on heat transfer within the mold, restricted to a single material. Other components such as chillers, feeders, in-gates, and various mold elements are not incorporated due to project time constraints. Additionally, the method's validation is tested on a simple 2-dimensional cantilever beam problem.

2 Topology Optimization Framework

2.1 Topology optimization

Topology optimization is a structural optimization technique which is widely used in industry to design light weight components. A structural optimization problem is composed of an objective function (f), design variables, state variables and the optimization algorithm maximizes or minimizes the objective function to achieve an optimal solution. Some of the algorithms commonly applied include Sequential Linear or Quadratic Programming (2), Optimality Criteria (1) and Method of Moving Asymptotes (33). The topology optimization method employs some basis concepts such as a fixed design domain and relaxation of the optimization problem (1). In the literature, various computational models have been proposed (1), however, the so-called SIMP model (Solid Isotropic Material with Penalization) (4) is applied in the current scope.

2.1.1 Problem formulation and FE discretization

The purpose of topology optimization is to find the optimal design of a structure within a specified domain with the known quantities in the problem such as, the applied loads, the support conditions, the structure's volume and some additional design restrictions. Typically, the design space is divided into discrete sections using the finite element method, creating a mesh composed of finite elements.

In the context of this study, the density-based topology optimization relies on the finite element method as a fundamental tool, requiring a suitable mesh. To facilitate the iterative design modifications aligned with the objective, the finite element mesh typically remains unchanged, enabling faster optimization by eliminating the need for repeated remeshing and re-parameterization of the evolving geometry.

The problem formulation is constructed from the density approach of topology optimization, distributing material in every element of a regular finite element mesh through the use of design variables x representing relative densities. The continuous variation of the design variables $x_e \in [0, 1]$ within the discrete design domain is coupled with SIMP interpolation scheme in order to effectively penalize intermediate densities, allowing the optimizer greater flexibility to enact gradual adjustments to design variables. This approach enhances convergence properties and aligns well with gradient-based optimization methods.

In this work, the topology optimization algorithm aims to minimize structural compliance while adhering to a constraint on the maximum material usage. In mathematical terms, the problem formulation is as:

$$\begin{aligned} \underset{x}{min} \quad & c(x) \\ s.t.: \quad & v^T x \leq V^* \\ & 0 \leq x_e \leq 1 \end{aligned} \tag{2.1}$$

where the function $c(x)$ represents the compliance calculated in each iteration of the numerical topology optimization problem, derived from the density field of the previous iteration. V^* denotes the maximum allowable material volume, while v represents

the array of element volumes, determined by their respective density values.

In mathematical terms, the optimal subset of material distribution within the design domain ($\Omega_{optimal} \subset \Omega$), where Ω is the design domain. The optimization design variable, the density vector x contains densities for all elements x_e. The localized elemental stiffness matrix E is a function of density vector x as per Equation 2.1, where x_e is 0 or 1 depending on the optimal material distribution of the element;

$$E(x) = xE^0 \tag{2.2}$$

$$x_e = \begin{cases} 1 & if \ \ e \in \Omega_{mat} \\ 0 & if \ \ e \in \Omega \setminus \Omega_{mat} \end{cases} \tag{2.3}$$

with a volume constraint

$$\int_\Omega x d\Omega = Vol(\Omega_{mat}) \leq V \tag{2.4}$$

where V is the design domain volume. The elemental density values can either be 0 or 1 where 1 signifies the solid element and 0 is void. With values 0 or 1, the elemental density value is discrete function which makes the optimization problem unsolvable with gradient based optimization algorithm.

Although it should ideally create binary results with either x_e=0 or x_e=1, instabilities may be induced in the optimization problem due to large design jumps. Therefore, the binary problem mentioned above is converted into continuous problem by applying power law and the discrete density variable is converted into continuous variable, which is known as SIMP (Solid Isotropic Material with Penalization) method and elemental material property function is written as:

$$E = x^p E^0 \tag{2.5}$$

where p is penalizing factor which penalizes the elements with intermediate densities to convert them into 0 or 1, x_{min} is the lower limit of density value in order to avoid singularities. The method was developed by Bendsϕe in 1989 (4) and it has been accepted as a topology optimization technique of many advantages. As shown in Figure 2.1, the higher the exponent p, the lower the mechanical properties of intermediate densities, and thus the optimization algorithm has an incentive to use more and more full void or fully solid elements, leading to sharper results. With $p = 1$, there is no penalization and the element stiffness is proportional to the relative density. It shall be noted that when using high penalization factor (generally $p > 5$), the optimization problem becomes difficult to solve due to the sharp variation of material properties close to x_e=1.

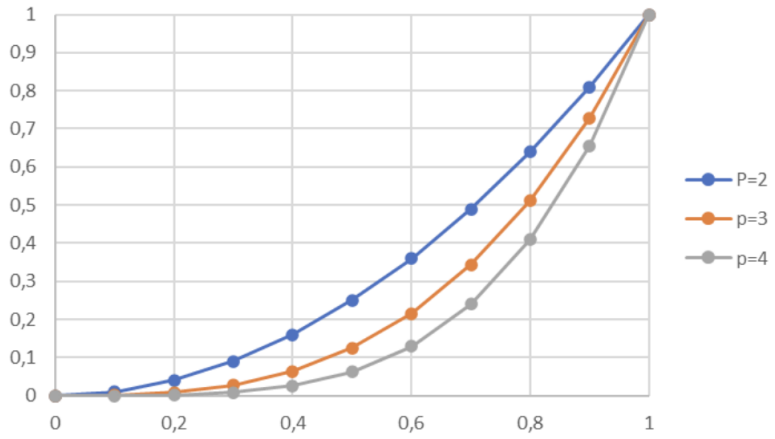


Figure 2.1: SIMP Interpolation Law

2.2 Numerical instability in topology optimization

Numerical instability, like checkerboard patterns, mesh dependency, and porous material inclusion, frequently arise in topology optimization outcomes, affecting both SIMP and homogenization methods. These issues often yield impractical results as structures with such flaws are challenging to fabricate.

2.2.1 Porous material

Porous material refers to numerous elements with intermediate density, such outputs featuring porous elements lack manufacturability and hold little engineering significance.

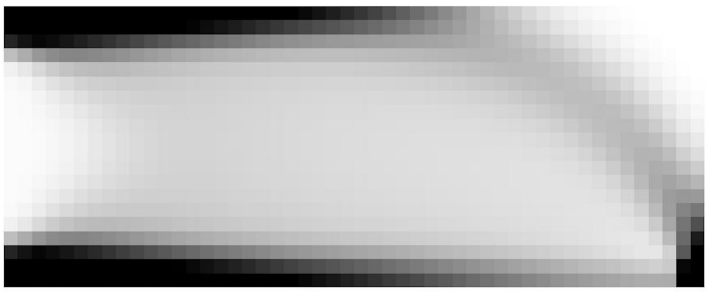


Figure 2.2: Porous Pattern on Cantilever Beam Optimization Problem with p=1

2.2.2 Checkerboard patterns

Checkerboard pattern, shown in Figure 2.3, indicates that the elemental densities of either 0 or 1 periodically exist in the results. This type of pattern is unrealistic and difficult to manufacture. As its name indicates, the checkerboard pattern is related to the inconsistency between displacement and density fields which generates alternating density values.

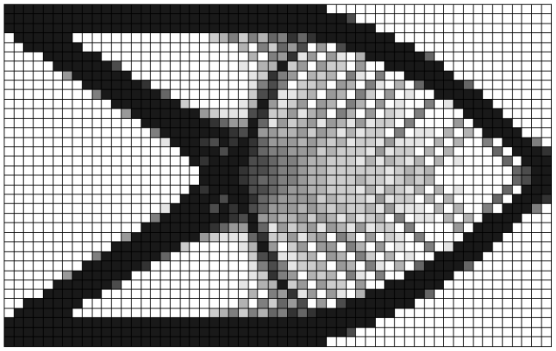


Figure 2.3: Checkerboard Effect on Cantilever Beam Optimization Problem

2.2.3 Mesh dependency

Mesh dependency means that the optimization results are correlated to the mesh density. A different finite mesh density may lead to different topology distribution. It can be observed in Figure 2.4, the finer the mesh, the greater the number of small structures included in the results. For the validity of the results, the solution of topology optimization should be unique and valid for any discretization of the design domain and the refinement of the mesh should only influence the result's accurary in terms of resolution while maintaing a common overall structure.

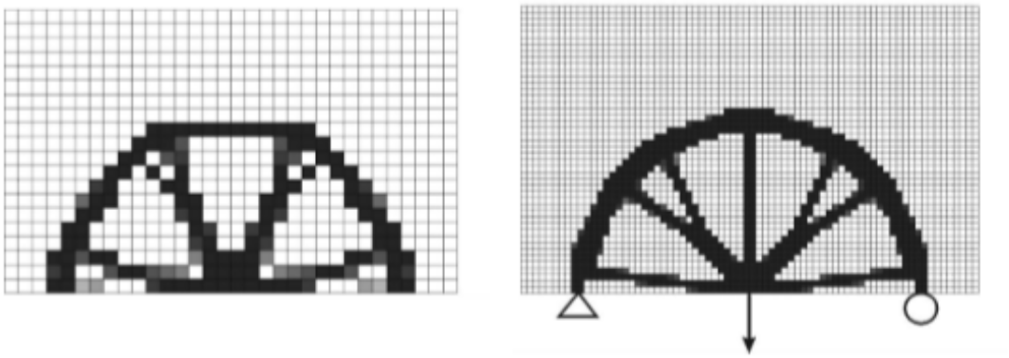
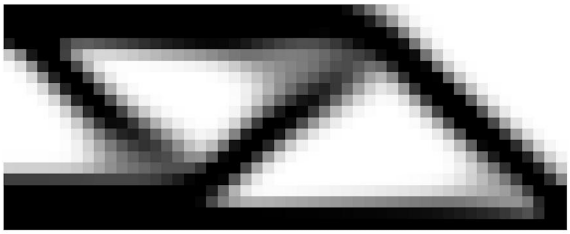


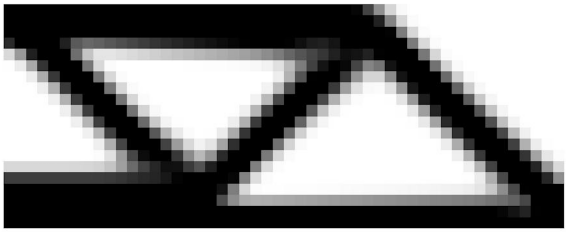
Figure 2.4: Mesh Dependency Instability

2.3 Methods to eliminate numerical instability

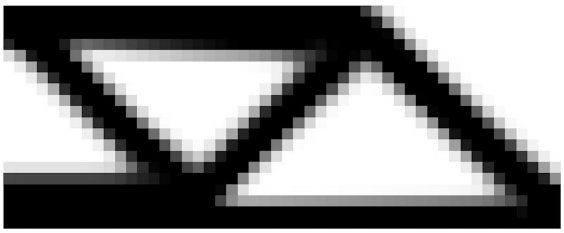
Several strategies have been suggested to eliminate numerical instability. Porous density can be prohibited by introducing a penalty factor p into the SIMP method.



(a) $p = 2$



(b) $p = 3$



(c) $p = 4$

Figure 2.5: Cantilever Beam Optimization Results with Different p Values

2.3.1 Sensitivity and density filtering methods

One commonly employed technique to mitigate issues related to mesh dependency and checkerboard instabilities is the application of a filtering method. This approach involves constraining the solution's design space by prohibiting the occurrence of high-frequency variations in the density field. The filtering method operates similar to image processing, where the act of blurring an image or region results in the removal of fine details, leading to a smoother and more uniform representation of the colour profile.

Historically Ole Sigmund (1994,1997)(7) introduced a filter of the sensitivities based on the averaging of the sensitivities of the objective function within a circle radius, r_{fil}. For uniform meshes, the filtered sensitivity field is obtained using the following equation:

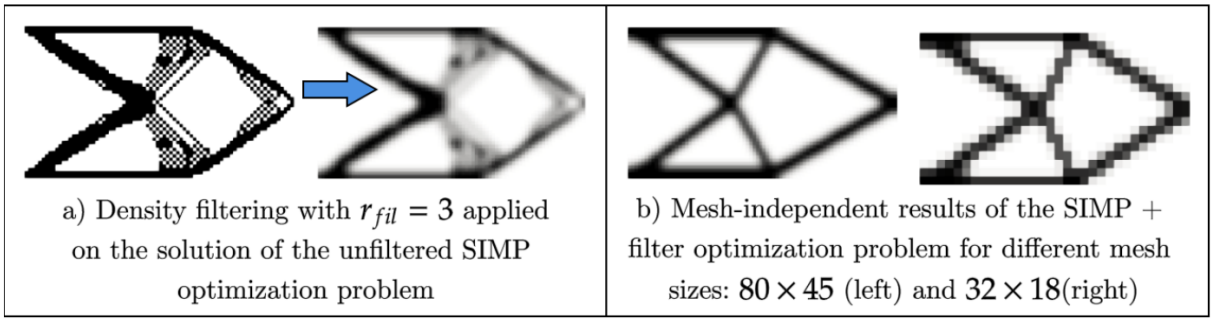
$$\frac{\partial \widetilde{f}}{\partial x_e} = \frac{\sum_{i \in N_e} w_i \left(X_i\right) x_i \frac{\partial f}{\partial x_i}}{x_e \sum_{i \in N_e} w \left(X_i\right)} \tag{2.6}$$

with X_i denoting the coordinates of the element i's centroid, N_e the set of elements within the filtering region $N_e = \{i | \|X_i - X_e\| \leq r_{fil}\}$, x_e the design variable (relative density) of the element and $w_i(X_i)$ the weight of element. The weight function is determined by calculating the Euclidean distances between element e, located at the center of the filtering circle, and element i as follows:

$$w_i(X_i) = \begin{cases} r_{fil} - \|X_i - X_e\| & i \in N_e \\ 0 & i \notin N_e \end{cases} \tag{2.7}$$

Subsequently, the filter was employed in the context of design variables denoted as x, representing relative densities. This utilization of the filter was suggested by Bruns and Tortorelli (5) and subsequently confirmed by Bourdin in 2001 (6). The density filtering method produces a novel set of design variables denoted as x_F, aimed at eliminating the undesirable checkerboard effect, as illustrated in Figure 2.6. In the case of a uniform and regular mesh, such as the one employed in this study, the filtered density field can be calculated using the following equation, utilizing the same filter area and weight functions discussed earlier:

$$x_{F,i} = \frac{\sum_{j \in N_i} w(X_i, X_j) x_j}{\sum_{j \in N_i} w(X_i, X_j)} \tag{2.8}$$



a) Density filtering with $r_{fil} = 3$ applied on the solution of the unfiltered SIMP optimization problem

b) Mesh-independent results of the SIMP + filter optimization problem for different mesh sizes: 80×45 (left) and 32×18(right)

Figure 2.6: Effects of Density Filtering

The key characteristic of the density filter is its ability to eliminate all structural intricacies that are smaller in scale than the filter radius, denoted as "r_{fil}," as demonstrated in Figure 2.6. Consequently, this filtering method enables the establishment of an effective minimum solid size when integrated into the topology optimization problem. This imposition of a minimum solid size contributes to enhancing the manufacturability of the solution. Thus, the mesh independence result becomes achievable by consistently selecting the filter radius "r_{fil}" based on physical dimensions relative to the dimensions of the design domain, as depicted in Figure 2.6.

Since the filter radius is determined in relation to the elements, it becomes necessary to readjust its value when conducting topology optimization iterations that involve remeshing. This adjustment ensures that "r_{fil}" remains consistent in terms of its physical dimensions and enforces a uniform minimum solid size in the final result. However, one drawback of the density filter is the emergence of "blurred" surfaces with

intermediate densities over a length equivalent to 2 times the "r_{fil}". This blurring effect results from the averaging process.

While this may appear as a regression, considering that the SIMP (Solid Isotropic Material with Penalization) law originally aimed to diminish the presence of intermediate elements by penalizing them, the density filter has emerged as an effective tool for addressing issues related to mesh dependency and checkerboard instabilities in the topology optimization process. As a result, it has gained popularity rapidly.

2.3.2 Heaviside filter

Several researchers have explored methods to once again reduce the prevalence of intermediate densities and enhance surface sharpness, aiding in the interpretation of designs. Notable approaches include the concept of progressively reducing the filter radius, as proposed by Sigmund in 1997 (7), and the continued use of mathematical projection techniques. For the achievement of sharp, nearly binary solutions, Guest et al. (8) introduced a Heaviside projection function that alters the density-filtered field. This modification ensures that all element densities (x_e) greater than 0 are projected to a solid physical density ($\rho_e = 1$), while void elements ($x_e = 0$) remain void:

$$\rho_e = \begin{cases} 1 & if \quad x_{F,e} > 0 \\ 0 & if \quad x_{F,e} = 0 \end{cases} \tag{2.9}$$

where, ρ represents the physical density field resulting from the Heaviside projection, and x_F corresponds to the field of design variables with density filtering. To better suit the requirements of gradient-based topology optimization problems, Guest et al.(8) introduced a more efficient and differentiable function, which is a smoothed version of the Heaviside function.

$$\rho_e = 1 - e^{-\beta x_e} + x_e e^{-\beta} \tag{2.10}$$

where, β is the parameter that regulates the steepness of the Heaviside function. As demonstrated in Figure **??**, the use of this Heaviside operator has the visual effect of expanding the optimal designs, resulting in more substantial and bulkier structures. However, this outcome can be circumvented by employing the Heaviside function with respect to a specific cut-off density value. This approach, as introduced by Wang et al. (9), involves utilizing the extended Heaviside function:

$$\rho_e = \frac{tanh(\beta\eta) + tanh[\beta(x_{F,e} - \eta)]}{tanh(\beta\eta) + tanh[\beta(1 - \eta)]} \tag{2.11}$$

where, η, in the range [0,1], represents the chosen cut-off density, while β, the same steepness parameter mentioned earlier, maintains its role. The extended Heaviside function is visually depicted in Figure 2.7, highlighting the impact of the β and η parameters. It's evident that for $\beta = 0$, the Heaviside projection preserves the original filtered field without any alterations. In contrast, higher values of β yield progressively sharper results, although with the potential risk of introducing oscillatory instabilities due to the rapid changes in projection behavior around $x_{F,e} = \eta$. As a general guideline, it is advisable to start the optimization process with a low β value and incrementally raise it during the course of optimization. Notably, it's worth mentioning that the

original Heaviside dilation formulation can be obtained by setting the density cut-off, η, to 0 in the extended function. This extended Heaviside projection, with the progressive adjustment of the steepness factor β, has been integrated into the filter-based method developed in this scope. It provides the final field of physical densities (ρ) at each iteration, as further discussed in the following section.

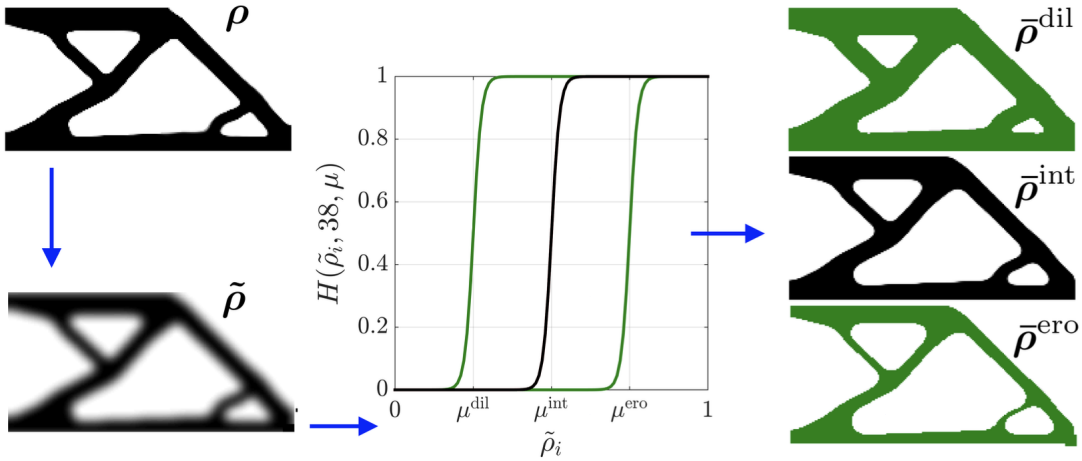


Figure 2.7: Heaviside Projection of Optimal Cantilever Beam

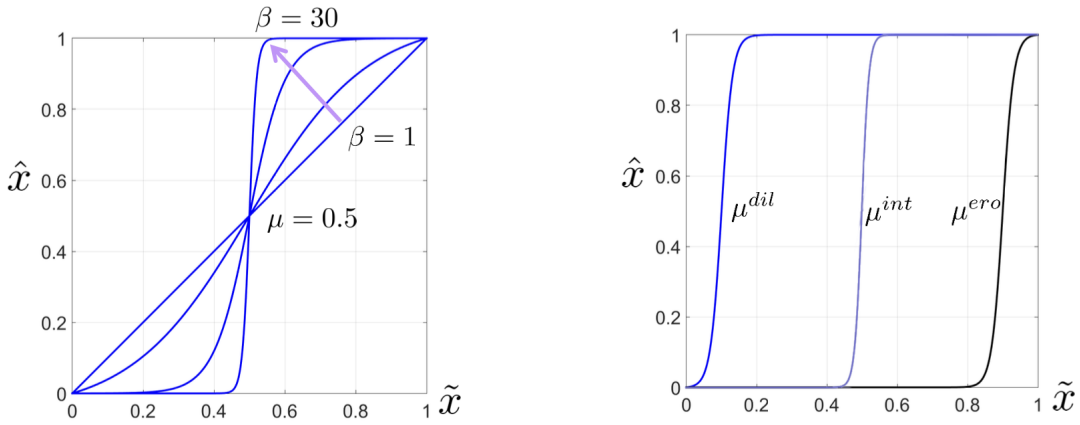


Figure 2.8: Extended Heaviside Function

The implementations of filter and projection techniques do not impose any additional constraints in the topology optimization problem. Instead, they modify the relative density fields to achieve the desired properties. Consequently, the fundamental formulation of the topology optimization problem as in Equation: 2.1remains unchanged when employing these density-based methods. The primary distinction lies in the density field used for computing objective and constraint functions. Specifically, the density field from the previous iteration (i) is consistently utilized to calculate compliance and total volume constraints for the subsequent iteration (i + 1). The mathematical representation of the optimization problem in Equation: 2.1 can be further presented

as below:

$$\begin{aligned} \underset{x}{min} \quad & c(x_F) \\ s.t.: \quad & v^T x_F \leq V^* \\ & \underbrace{0 \leq x_e \leq 1}_{SIMP + Filter} \end{aligned} \tag{2.12}$$

$$\begin{aligned} \underset{x}{min} \quad & c(\rho) \\ s.t.: \quad & v^T \rho \leq V^* \\ & \underbrace{0 \leq x_e \leq 1}_{SIMP + Filter + Heaviside} \end{aligned} \tag{2.13}$$

2.4 Sensitivity analysis

When solving the optimization problem, the derivatives of the objective function and constraints concerning the design variables are essential. These derivatives, termed sensitivities, form the basis of sensitivity analysis. The process involves determining these sensitivities, and there are two predominant methods for the computation; numerical approaches or analytical methods.

2.4.1 Numerical methods

Within numerical sensitivity analysis, sensitivities are computed using finite differences. Various methodologies exist for this purpose, such as forward differences or central differences to determine these sensitivities. The sensitives using forward differences can be computed as:

$$\frac{\partial f(x_P)}{\partial x_{P,j}} \approx D_f = \frac{f(x_P + he_j) - f(x_P)}{h} \tag{2.14}$$

where $e_j = [0, ..., 0, 1, 0, ..., 0]^T$, e_j is 1 on index j and 0 otherwise.
A main issue with forward differences lies in determining an appropriate value for h. Ideally, a smaller $'h'$ is desirable, yet reducing it substantially can introduce numerical errors stemming from cancellation issues. Conversely, using a larger $'h'$ results in a less accurate approximation. Despite the ease of implementation, numerical methods become approximations and tend to be more computationally demanding compared to analytical methods. The primary advantage of numerical methods lies in their straightforward implementation. In this work, numerical differentiation has only been employed to validate the accuracy of the implemented analytical method.

2.4.2 Analytical methods

The direct and adjoint methods stand as the primary analytical approaches, each carrying distinct advantages. In scenarios where the number of design variables exceeds that of constraint functions, the adjoint method is more efficient. Conversely, when constraint functions outnumber design variables, the direct method proves more efficient. The adjoint method is applied to compute the sensitivities in this work.

The preceding sections within this chapter explored filter and projection techniques

applied to the design variable field, resulting in what is referred to as the multi-field optimization approach. At the end of each iteration, a physical design x_P is attained through Heaviside projection, which is in a ready to manufacture state. With the SIMP power law penalization and employing the weak formulation of the finite element model within this work, the objective function $f(x_P)$is defined as the sum of strain energies of all elements:

$$f(x_P) = \sum_{e=1}^{N} (x_{P,e})^p \psi_0(u_e) \tag{2.15}$$

Likewise, the volume constraint is defined as the sum of all element densities of the projected field x_P:

$$\begin{aligned} g(x_P) \quad &= \frac{\sum_{e=1}^{N} x_{P,e}}{N} - V^* \leq 0 \\ &= \sum_{e=1}^{N} x_{P,e} - NV^* \leq 0 \end{aligned} \tag{2.16}$$

The sensitivity of the objective and constraint functions $\frac{df(x_P)}{dx_e}$ and $\frac{dg(x_P)}{dx_e}$ with respect to the design variables x is also required for the Method of Moving Asymptotes (MMA) used in this work. However, this derivative is not straightforward, as both values are calculated concerning the physical density field x_P, attained after numerous filtering and projection steps. In the context of the current multi-field optimization process, the sensitivity analysis with respect to the design variables is obtained after application of chain rule for all filtering and projection steps applied on the relative densities:

$$\frac{df}{dx} = \frac{df}{dx_P}\frac{dx_P}{dx_I}\left(\frac{dx_I}{dx_{C1}}\frac{dx_{C1}}{dx_F} + \frac{dx_I}{dx_{C2}}\frac{dx_{C2}}{dx_F}\right)\frac{dx_F}{dx} \tag{2.17}$$

2.4.3 Adjoint approach for computing sensitivities

The adjoint method stands out as an efficient technique that avoids the need to solve the PDEs repeatedly for each gradient component computation. To demonstrate the effectiveness of this approach, a classical optimization problem in the following form is taken into account:

$$\begin{cases} min & J(U,x) \\ s.t. & R(U,x) = 0 \end{cases} \tag{2.18}$$

with $J(U,x)$ $(R^{n_U} \times R^{n_x} \rightarrow R)$ is an objective function to be minimized, x corresponds to the design variables, U is the set of state variables with the relationship $R(U,x) = 0$ for a function $(R^{n_U} \times R^{n_x} \rightarrow R)$ to be satisfied (32).

The objective is to determine the total variation or sensitivity of the objective function concerning the design variables. Solving the relationship $R(U,x) = 0$ is generally complex. For a given set of design variable values, obtaining the state variables U involves resolving $R(U,x) = 0$. When utilizing FD or DD methods, this requires computing n_x finite differences, implying that solving $R(U,x) = 0$ must occur n_x times. In topology optimization, this quickly becomes expensive due to the abundance of design variables, leading to a high value of n_x. However, with the adjoint approach, computational expenses decrease significantly compared to the FD (or DD) approach. The

adjoint method introduces a set of equations solved only once alongside the $R(U, x) = 0$ relationship, which is also solved only once within a single optimization cycle.

In a straightforward scenario where the objective function directly relies on the state variables, $J = J(U)$, and these state variables U depend on the design variables x. The objective is to efficiently compute $\frac{dJ}{dx}$ defined as:

$$\frac{dJ}{dx} = \frac{dJ(U(x))}{dx} = \frac{\partial J}{\partial U}\frac{\partial U}{\partial x} \tag{2.19}$$

and

$$R(U, x) = 0 \tag{2.20}$$

must be satisfied everywhere. Equation: 2.20 implies that $\frac{dR}{dx} = 0$ because $R = 0$ for each of the design variables.

The adjoint approach involves utilizing adjoint variables in constructing the Lagrangian, expressed by the following equation:

$$L(U, x, \xi) = J(U) + \xi^T R(U, x) \tag{2.21}$$

where ξ is the vector of Lagrange multipliers (or adjoint variables).

As $R(U, x) = 0$ for each x, $L(U, x, \xi \equiv J(U)$ and ξ can be chosen freely. With this new form of objective function $J(U)$, Equation: 2.19 becomes:

$$\begin{aligned} \frac{dJ}{dx} = \frac{dL}{dx} \quad &= \frac{\partial J(U)}{\partial U}\frac{dU}{dx} + \frac{d\xi^T}{dx}R + \xi^T\left(\frac{\partial R}{\partial U}\frac{dU}{dx} + \frac{\partial R}{\partial U}\right) \\ &= \frac{\partial J}{\partial U}\frac{dU}{dx} + \xi^T\left(\frac{\partial R}{\partial U}\frac{dU}{dx} + \frac{\partial R}{\partial U}\right) \\ &= \left(\frac{\partial J}{\partial U} + \xi^T\frac{\partial R}{\partial U}\right)\frac{dU}{dx} + \xi^T\frac{\partial R}{\partial U} \end{aligned} \tag{2.22}$$

If ξ is chosen, such that:

$$\frac{\partial R^T}{\partial U}\xi = -\frac{\partial J^T}{\partial U} \tag{2.23}$$

then, the expensive computation of $\frac{dU}{dx}$ is avoided.

The final expression of the sensitivity of the objective function with respect to the design variables is given by:

$$\frac{dJ}{dx} = \xi^T\frac{\partial R}{\partial U} \tag{2.24}$$

In order to obtain the final sensitivity of J, the adjoint equation represented in Equation: 2.23 has to be solved only once. Once the adjoint variables are obtained, $\frac{dJ}{dx}$ is achieved with the vector dot product between the adjoint variables, ξ and $\frac{\partial R}{\partial U}$ at a negligible cost compared to solving the equation $R(U, x) = 0$ n_x times.

2.5 Optimization algorithm

The focus of the topology optimization algorithm is to generate design configurations x that progressively approach the desired objective while ensuring adherence to imposed constraints. In the case of minimizing compliance with volume restrictions, the ultimate design should maximize stiffness regardless of the material quantity used. During the process, the optimization algorithm initiates each iteration. Firstly, it evaluates the performance of the previous iteration's final design, denoted as x_P, by computing its compliance $f(x_P)$ and total volume $g(x_P)$ which serves as the constraint function. Subsequently, small adjustments are made to refine the new design field (x), bringing it closer to optimizing the objective. This work considers MMA method to solve the optimization problem.

2.5.1 Method of Moving Asymptotes

Method of Moving Asymptotes (MMA), developed by Krister Svanberg (33), was adapted as the preferred topology optimization algorithm within the multi-field approach for increased flexibility concerning potential constraint violation.

The problem formulation considered in the MMA method is given below:

$$\begin{array}{cc} \overset{min}{x} & f(x_P) + a_0 z + \sum_{i=1}^{n}(c_i y_i + \frac{1}{2} d_i y_i^2) \\ s.t.: & g_i(x_P) - a_i z - y_i \leq 0, \quad i = 1, ..., m \\ & 0 \leq x_e \leq 1 \quad with \;\; e = 1, ..., n \\ & y \geq 0, \quad z \geq 0 \end{array} \tag{2.25}$$

where $f(x_P)$ is the defined objective function, m is the number of constraints, where $m = 1$ for the case of a single constraint on the volume defined by the function $g(x_P)$.

The coefficients need to be satisfied in a way that:

- a_0,a_i,c_i,d_i are real numbers with $a_0 > 0$, $a_i \geq 0$,$c_i \geq 0$, $d_i \geq 0$ and $c_i + d_i > 0$ for all i;
- $a_i c_i > a_0$ for all i with $a_i > 0$.

The problems will be based on Equation: 2.25 because it consistently offers feasible solutions and, crucially, obtain at least one optimal solution. The demonstration of this is provided in (33).

To align this problem closely with the original formulation, Equation: 2.15,2.16, the following parameters should be chosen for Equation 2.25.

- a_0 can be chosen equal to 1 and $a_i = 0$ for all $i > 0$, leading to $z = 0$ for any optimal solution.

The optimization variable y is then used as a compliance variable with respect to the violation of the original volume constraint $g(x_P)$, such that the new constraint $g(x_P) - y \leq 0$ is respected. To encourage however the validation of the original constraint $g(x_P) \leq 0$, the variable y should become "expensive" by heavily penalizing

the objective, so that $y = 0$ at the optimal solution. Therefore, the coefficients of the optimization variable y can be taken as $d = 1$ and $c =$"a large number", such that the objective is greatly increased for low y values, penalizing the problem and encouraging strictly verifying the constraint at the optimal solution. However, the optimization problem has an increased degree of freedom, being allowed to temporarily violate the constraints in the research of the optimum. It should be noted that the problem 2.25 always has feasible solutions and at least one optimal solution, even though the original problem does not have any feasible solutions, in which case some $y_i > 0$ (33).

MMA functions as an optimization method solving problems defined in Equation 2.25. However, for this topology optimization problem, it demands a significant number of design variables and involves highly nonlinear functions implicitly relying on these variables. Consequently, it become expensive to directly solve the problem , requiring an efficient mathematical programming scheme. This scheme substitutes the original problem with a local approximation in each iteration; a fundamental principle underlying the MMA approach, which resolves the optimization problem through iterative steps. Each iteration centers around the current point that comprises all design variables $(x^{(k)}, y^{(k)}, z^{(k)})$, around which is for defining an approximate subproblem. Within this subproblem, the original objective function $f(x)$ and constraint functions $g_i(x)$ are substituted with certain convex counterparts $\widetilde{f}(x)$ and $\widetilde{g}_i(x)$. These convex functions are determined based on the current sensitivities, computed similarly to the previously discussed method. They also depend on parameters $U_j^{(k)}$ and $L_j^{(k)}$ that describe moving asymptotes, and these parameters are updated in each iteration.

The process involves solving the approximate convex subproblem to obtain a unique optimal solution $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})$, which then becomes the subsequent iteration point. Around this solution, a new convex approximation subproblem is defined. Reaching the solution of the original problem occurs when the design variables between consecutive iterations converge arbitrarily close to zero, meeting a specified convergence tolerance.

Figure 2.9 illustrates the MMA approximation technique around a specific design point, showcasing the vertical asymptotes determined by the $U_j^{(k)}$ and $L_j^{(k)}$ parameters. Within the context of this work, the MMA method emerges as the preferred optimization algorithm within the multi-field optimization approach. This preference is attributed to its good performance, high adaptability for highly filtered material distributions, and its facilitation of implementing various objective and constraint functions easily.

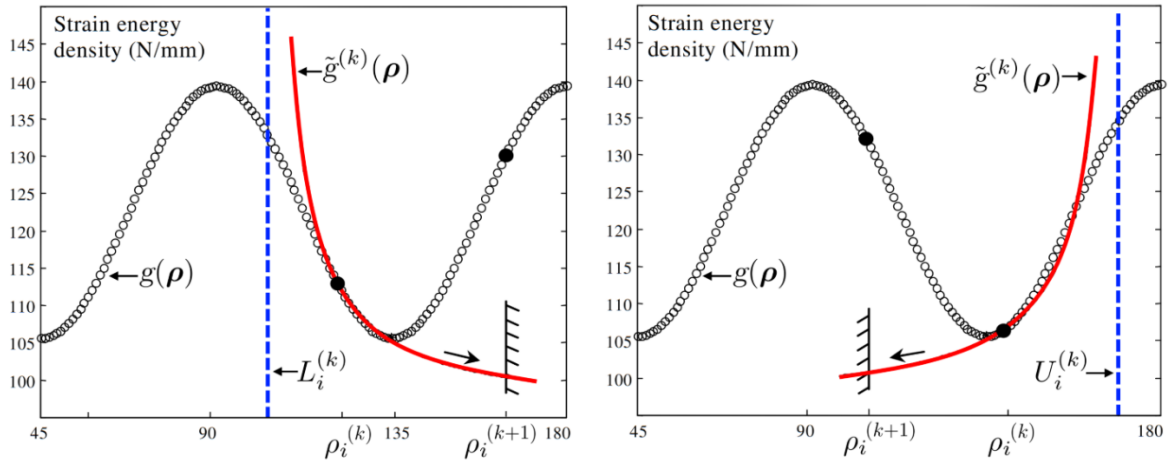


Figure 2.9: MMA Approximation Method(34)

3 Casting

In automotive manufacturing, casting is a pivotal process for creating various components. Molten metal, like aluminum or iron, is poured into a custom-designed mold. This molten metal solidifies within the mold cavity, taking its shape. Once cooled and solidified, the newly formed piece is removed from the mold. The casting process for automotive components includes various techniques such as gravity casting, die casting, high and low-pressure casting, and investment casting. These methods ensure the production of intricate and durable parts crucial for vehicles, ranging from engine blocks and transmission cases to wheels and other structural elements.

In order to optimize cast parts, careful attention to specific geometric requirements is essential. This consideration is crucial in mitigating casting defects, primarily influenced by fluid flow, heat transfer dynamics, and thermal stresses within the manufacturing process.

3.1 Manufacturing constraints

3.1.1 Internal cavities and undercut

Primarily, it is important to ensure that cast part designs are free from internal cavities or undercuts in the parting direction. While closed internal cavities cannot be achieved through metal casting, various techniques exist for creating undercuts, such as using side-action cores in the mold. However, employing such methods significantly impacts mold complexity and the overall manufacturing costs of the part. Figure 3.1 illustrates an undercut on a cast part produced in a mold. To ensure cost-effective outcomes, it shall be clear of both undercuts and internal cavities in the design phase.

3.1.2 Draft angle

Draft angles represent the angled taper integrated into the mold's side walls and cores, facilitating the easy removal of a part by minimizing friction between the part and the mold walls during ejection. Parts lacking vertical walls without draft angles risk being challenging to strip, prone to denting, scratching, or becoming lodged in the tool. However, if external walls serve a functional purpose, they may remain without draft angles.

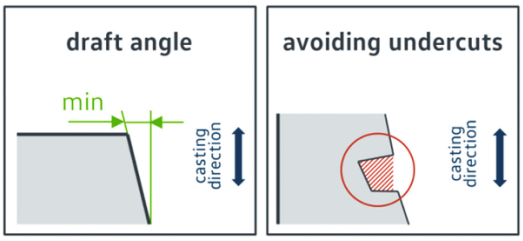


Figure 3.1: Geometrical Constraints of Undercut and Draft Angle

3.1.3 Minimal size constraints

The geometry of the part should adhere to a minimum member size restriction, taking into account the flow properties of the molten metal utilized in casting. This constraint prevents the formation of delicate structures that could solidify rapidly, potentially blocking the flow of molten metal to subsequent sections. Moreover, it is essential to enforce constraints on minimal pocket and hole sizes to guarantee the mold's feasibility, as small holes in the part demand a detailed mold structure. Figure 3.2 provides an illustration highlighting these minimum size limitations.

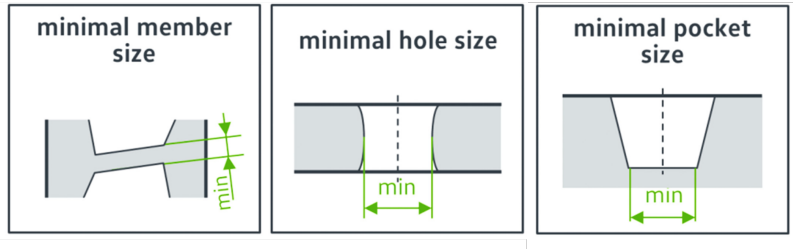


Figure 3.2: Geometrical Size Constraints

3.1.4 Directional solidification

The quality of cast parts is significantly influenced by the solidification process. Typically, a cast part solidifies from its outer regions towards the inner sections. As the material solidifies, temperature-induced shrinkage occurs, leading to a contraction in volume. This reduction in volume needs to be compensated by liquid material. In cases where there is not enough liquid material available, casting defects, known as shrinkage porosity, arise (29). Figure 3.3 provides an overview of potential defects that can occur.

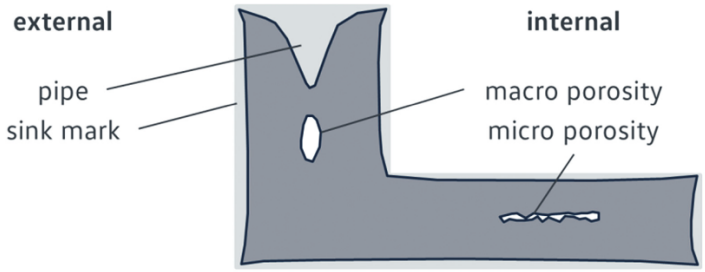


Figure 3.3: General Overview on Possible Shrinkage Porosity

The objective of the part design process is to create a casting that's internally free of porosity. This is accomplished by feeding extra material that supplies the cast part with liquid material, subsequently removed after solidification. Furthermore, the design of the cast part must allow for directional solidification towards the feeder. The method relies on Chvorinov's rule (31) and Heuvers' circle method (30). Chvorinov's rule illustrates that the solidification time is proportional to the square of the modulus M, which represents the ratio between a region's volume V and its heat-transferring

surface A.

$$t_{solidification} \sim (M_{region})^2 = ((\frac{V}{A})_{region})^2 \tag{3.1}$$

Heuvers leverages this principle, asserting that regions farther from a feeder should solidify before others. When solidification time diminishes with increased distance from the feeder, directional solidification is achieved. The shrinkage of the melt's volume can be compensated by liquid material from nearer regions that haven not solidified yet. Feeder areas solidify last due to having the largest modulus or longest solidification time, respectively. Shrinkage porosity only remains in the feeders, not impacting the overall cast part quality. Figure 3.4 demonstrates this principle, adhering to Heuvers' circle method is assessed by the diameter progression in the cast part, which should expand toward the feeder.

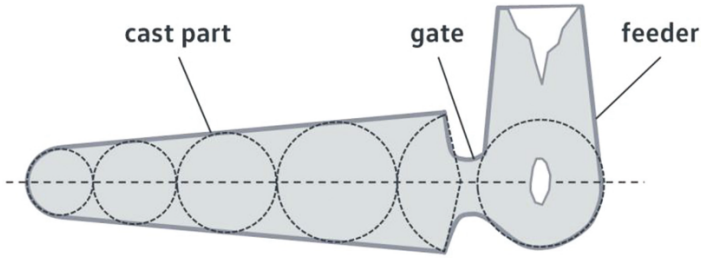


Figure 3.4: Principle of Heuvers'

3.2 A filter based method for casting constraints

In order to implement the previously described manufacturing constraints in a topology optimization framework, these constrains are developed based on filter and projection-based approach. Langelaar (35) introduced a filter-based technique for incorporating multi-axis machining restrictions into optimization, yielding manufacturable designs in both 2D and 3D scenarios. Given the concept between casting and 2.5D machining, where the tool has a single attack direction, a similar approach can be adapted for specific casting needs. Filters efficiently modify design variables, narrowing the solution space suitable for casting without explicitly defining new constraints. This extension of the three-field optimization method incorporates casting-related constraints via filters and projections, maintaining the initial problem formulation with a single constraint on solid fraction (V^* from Equation 2.15,2.16). Details on casting constraints' integration into the multi-field optimization, alongside explanations of basic filters and projections, will be briefly covered in later sections.

3.2.1 Minimum member size constraint

As discussed previously, there exists the design variable field's instability in SIMP-based topology optimization, like the checkerboard pattern and mesh dependency. These issues hinder physical interpretation and manufacturability. To address these problems and achieve a refined physical model, the solution involved expanding the three-field optimization model. This extension called for incorporating multiple fields,

introducing supplementary filters and projection functions specifically designed for casting constraints.

The initial phase of the multi-field method involves the density filter. Here, the design variables (x) within a circular zone of radius "r_{fil}" around an element (i) are averaged using a linear weight function based on proximity. This process establishes the density of element (i) in the newly created filtered density field (x_F). The filtering process can be precisely described by a distance-weight matrix. This matrix relies on the filter radius r_{fil} and calculates the Euclidean distances between elements to define the filtering operation:

$$D_{ij} = [r_{fil} - \|X_j - X_i\|]_+ \qquad i, j = 1, ..., N \tag{3.2}$$

where the operator$[]_+$ gives 0 for all negative arguments and X_i are the coordinates of element's i centroid. Then, the filtered density field x_F can be obtained as:

$$x_{F,i} = \sum_j \left(\frac{D_{ij}}{\sum_g D_{ig}}\right) x_i \quad \Leftrightarrow \quad x_F = Hx \tag{3.3}$$

with H being the density filtering matrix.

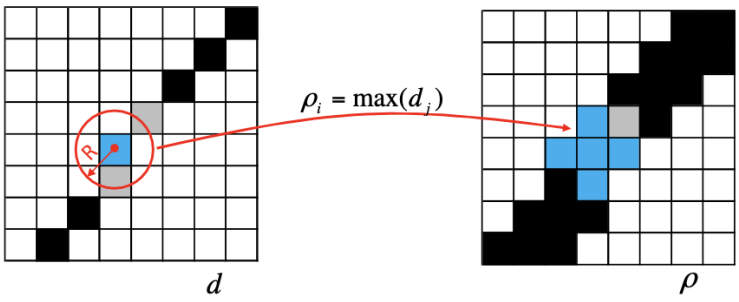
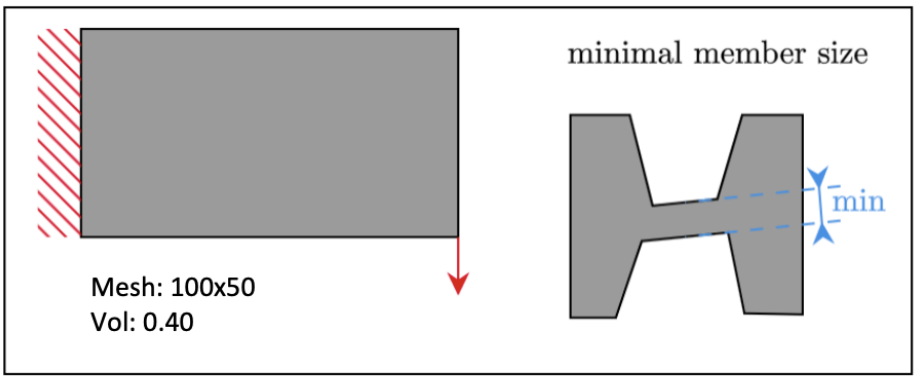


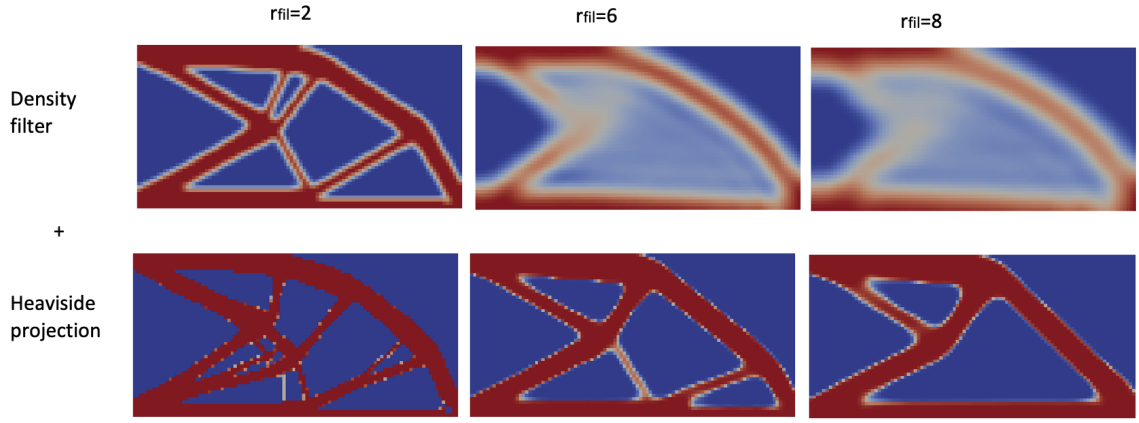
Figure 3.5: Projection Scheme for Minimum Member Size Constraint

This filter effectively eliminates all checkerboard patterns and structural details smaller than the filter radius. When coupled with the Heaviside projection in the final phase of the multi-field method, the density filter becomes a practical means of imposing a minimum member size constraint associated with 'r_{fil}', illustrated in Figure 3.6.

The study focused on the 2D cantilever beam optimization, Figure 3.6, integrating the density filter into the optimization process for various filter radii (r_{fil}). Figure 3.6 highlights that increasing r_{fil} in the optimization solely with the density filter removes finer structures quickly but introduces blurry zones with intermediate elements, making defining the model's contour challenging. Incorporating the Heaviside projection in a complete three-field approach generates sharp designs where the minimum member size is tied to r_{fil} and material availability. The complete three-field method yields thicker structures as r_{fil} increases. However, in cases where material availability becomes limiting, despite a larger filter radius, the design may remain unchanged.



(a) Topology optimization problem with minimum member size constraint



(b) Three-field topology optimization result for simple cantilever beam

Figure 3.6: Minimum Member Size Constraint Enforced with the Density Filter, Based on Filter Radius r_{fil}

3.2.2 Directional molding constraint

Ensuring that the design aligns with the molding and parting process stands as the crucial step in achieving an ideal design ready for casting. It has been observed that the most challenging aspects towards casting manufacturing are the presence of undercuts and internal cavities. Ensuring the compatibility between the part's design and the mold's removal post-casting is the initial step in achieving a design optimized for casting topology. Selecting a draw direction requires the elimination of all undercuts and internal cavities present in the filtered density field x_F, ensuring the final iteration's design aligns with the directional molding limitations.

From a mathematical standpoint and considering the regular finite element mesh employed in this work, directional molding implies that the density field should consistently decrease along the mold's draw direction. Consequently, this condition ensures that only external cavities remain at the design domain's periphery, precisely at the mold interface. To ensure the elimination of internal cavities and undercuts, a new field is created from the filtered density field x_F. This is achieved by employing a cumulative summation filter on elements aligned against the drawing direction. The result is a pseudo-density field x_C that exhibits monotonic increase in the opposite direction of drawing. In this field, elements may surpass the value of 1, as illustrate in Figure 3.7. While areas feasible for molding remain low values close to 0, internal cavities are filled with material, although the specific pseudo-density value (beyond 1) does not hold a direct physical significance.

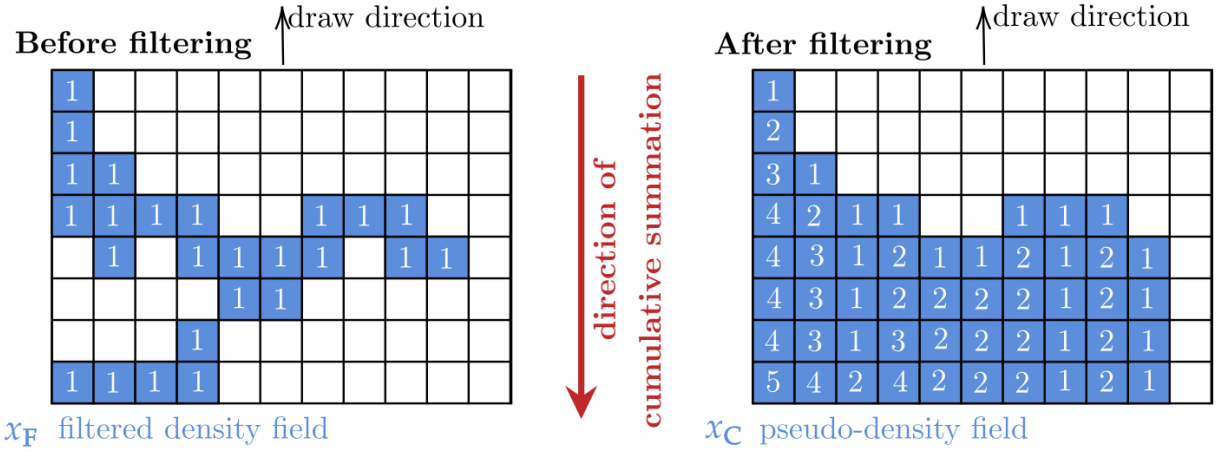


Figure 3.7: Cumulative Summation Filter for Monotonically Decreasing Pseudo-density Field in the Draw Direction

The cumulative summation filter, characterized by a matrix operation, is a computationally efficient linear process with low complexity:

$$x_C = Cx_F \tag{3.4}$$

where x_C is the new vector of pseudo-densities respecting the directional molding constraint, x_F is the filtered density filed obtained previously and C is a matrix of

cumulative summation. Matrix C, with dimensions $[n \times n]$ (n representing the finite elements in the mesh), defines the contribution of the filtered element $x_{F,j}$ to the castable pseudo-density field element $x_{C,i}$. This contribution is determined by the summation coefficients C_{ij}, where $C_{ij} = 1$ signifies contribution and $C_{ij} = 0$ denotes non-contribution.

As illustrated in Figure 3.8, the directional molding constraint in the problem formulation often results in bulky designs that exceed the user-imposed limit V^* for the final solid fraction. The optimization algorithm starts with an initial iteration using a uniform density field where all elements adhere to the volume limit V^*. Despite the optimization algorithm solving Equation 2.15 and generating a design variable field x that complies with the volume constraint, the subsequent application of the cumulative summation filter results in a physical design x_P (post-Heaviside projection) predominantly comprising solid elements. Consequently, the solid volume increases from V^* in iteration 0 (initial design domain) to nearly 1 in iteration 1.

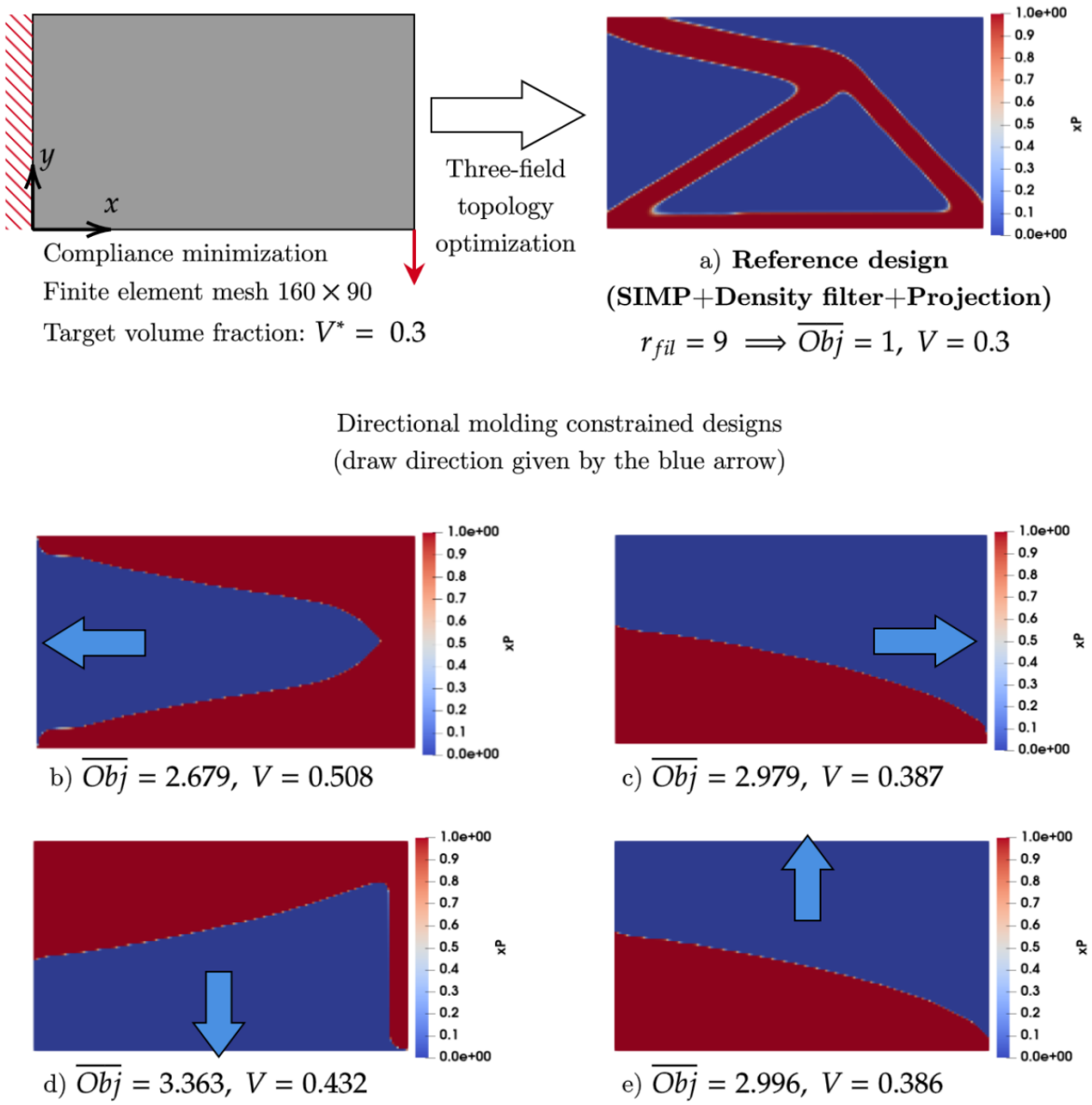


Figure 3.8: Topology Optimization Results Subject to Directional Molding Constraint for Different Draw Directions

3.2.3 Split drawing

The application of a single drawing constraint, achieved by employing the cumulative summation filter exclusively in one direction (specifically, opposite to the draw direction), results in optimization problems that are excessively restrictive. This approach yields sub-optimal designs that may violate the solid volume constraint set by the user. Figure 3.8 illustrates that certain material regions, dictated by the direction of cumulative summation, do not contribute to the structural rigidity of the beam. For instance, applying cumulative summation in the x-direction leads to solid material in the upper right corner of the design domain that has no impact on structural rigidity under given boundary and load conditions. Furthermore, these single draw designs lack consideration for the potential to create cast parts through split drawing within a two-part mold, such as between two closed dies.

For a split drawing compatible design, the optimization algorithm must create and expand cavities from two opposing directions along the draw axis. This simulates the modification of each die in the casting process. Enabling this optimization process introduces two distinct draw directions independently. This results in the generation of two separate castable pseudo-density fields, x_{C1} and x_{C2}, obtained by performing cumulative summation in the positive and negative directions of the draw axis, respectively:

$$x_{C1} = Cx_F \quad and \quad x_{C2} = C^T x_F \tag{3.5}$$

The optimization algorithm is allowed to create cavities on two pseudo-density fields from opposing draw axis directions, considering both directional molding scenarios. To consolidate contributions from both cumulative summation directions into a unified design, the intersection of the two pseudo-density fields x_{C1} and x_{C2} is necessary to preserve generated cavities from both directions. Essentially, a unique pseudo-density field x_I is derived where $x_{I,i}$ equals 0 if either $x_{C1,i}$ equals 0 or $x_{C2,i}$ equals 0. This intersection operation can be accomplished mathematically by employing an elementwise smooth minimum operation with the Kreisselmeier-Steinhauser (KS) function over the two castable pseudo-density fields, resulting in the formulation of the pseudo-density field x_I as follows:

$$x_{I,i} = f(x_{C1,i}, x_{C2,i}) = \frac{1}{p} ln(\frac{1}{2} e^{p x_{C1,i^+}} + \frac{1}{2} e^{p x_{C2,i^+}}) \tag{3.6}$$

The variable "p" represents a significantly negative number. A greater absolute value of "p" brings the KS function closer to a genuine minimum. In this work, "p" is set at -20, which generates values very proximate to the actual minimum, maintaining stability without introducing numerical instabilities. It shall be noted that a true minimum could not be integrated into the optimization loop due to the necessity for smooth functions in gradient-based optimization procedures, crucial for sensitivity analysis.

Integrating the combined bi-directional cumulative summation and the intersection of the resultant pseudo-density fields within the optimization process, succeeded by the Heaviside projection, yields designs suitable for split-drawing applications. Figure 3.9 represents the outcomes of minimizing compliance in a 2-dimensional cantilever beam scenario using a 160×90 mesh, with split drawing constraints imposed separately in the x and y-directions.

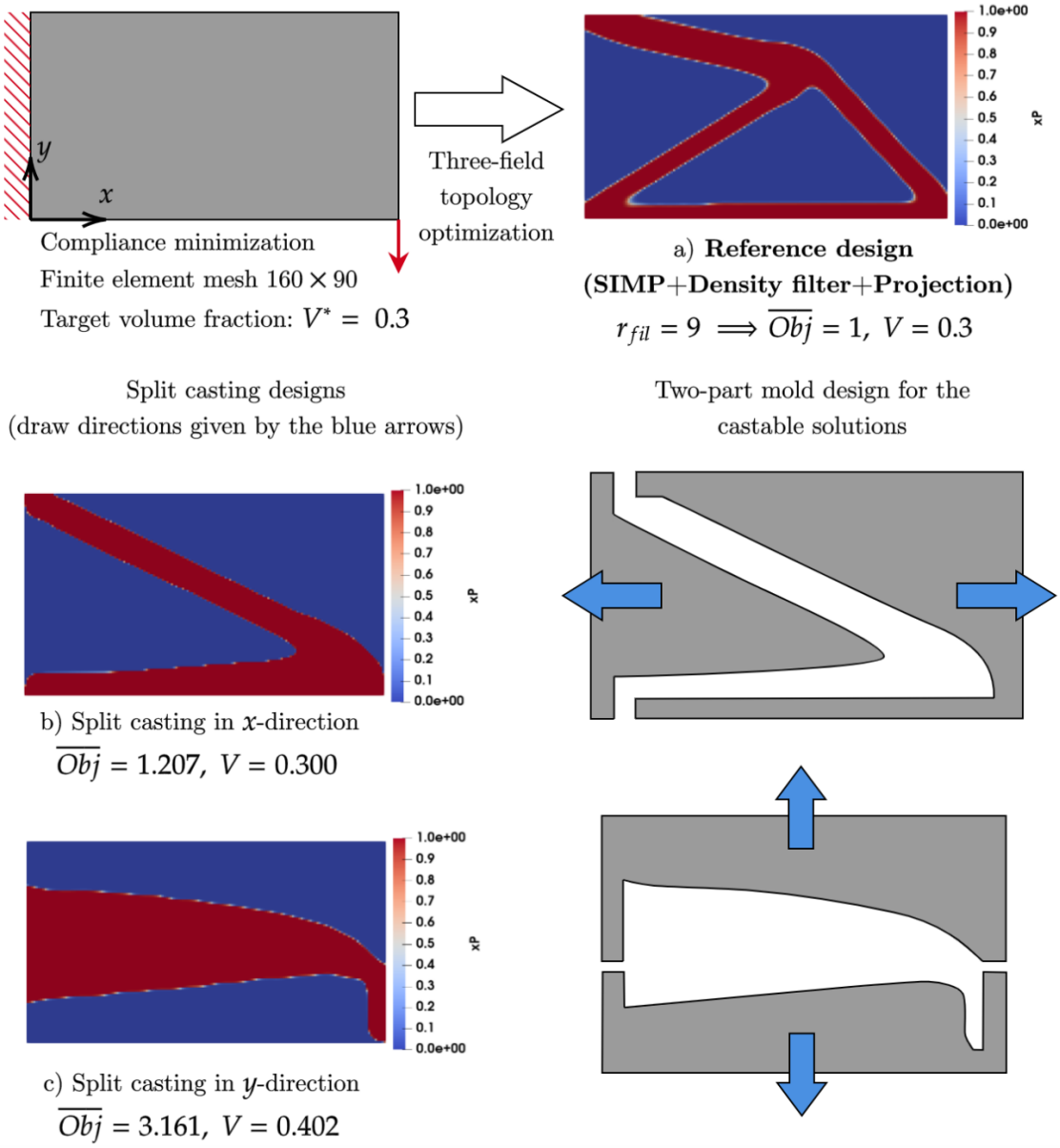


Figure 3.9: Topology Optimization Results with Split Drawing Constraints for x and y Directions

3.2.4 Draft angle constraint

An another important consideration for cast parts is the draft angle. This feature is essential to facilitate easy unmolding and prevent surface defects. A draft angle is applied to the manufacturable design in relation to the draw axis, ensuring compliance with a user-defined taper for all internal and external surfaces within the design domain. It shall be noted that external walls situated on the boundary of the design domain should not undergo drafting. This exception is due to their susceptibility to boundary or load conditions, subjecting the maintenance of their original shape. With the multi-field optimization approach, the draft angle constraint can be enforced by applying a filter based on the smooth minimum KS function to each castable pseudo-density field, whether it is x_{C1} and x_{C2} for the split draw problem or the unique x_C for the single draw case.

The concept of this methodology lies in creating cone-shaped void regions based on each void element within the castable pseudo-density fields, treating them as positions for the cone tips. This process essentially "carves out" material, establishing a new pseudo-density field where all voids are bordered by surfaces conforming to a taper angle at least as specified by the draft angle. However, processing the entire cone-shaped region for each void element demands a considerable number of elements, prompting a more efficient approach. Instead, focusing solely on projecting the frontal elements of the cone suffices. This is feasible due to the monotonically decreasing nature of the pseudo-density field in the draw direction. By addressing only the frontal elements of the cone-shaped area, similar to sweeping through the mesh, it achieves comparable outcomes to projecting the entire region.

In practice, an equivalent element-based method can be established utilizing the smooth minimum KS function. This function is applied to the mirrored pattern of cone-shaped elements oriented in the draw direction, as presented in Figure 3.10.

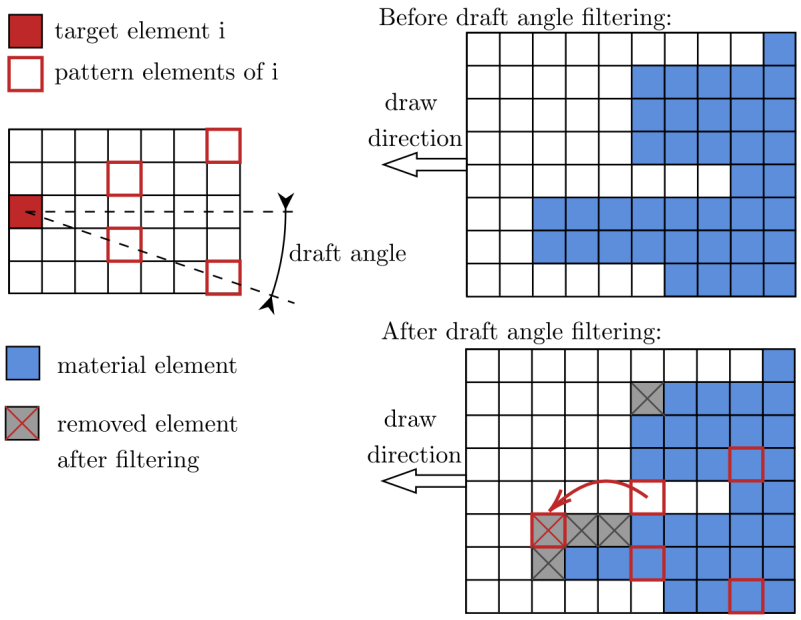


Figure 3.10: Draft Angle Imposed via a Cone-shaped Pattern

In this method, the element at the cone pattern's tip, labeled as i, determines its pseudo-density based on surrounding elements, serving as void detectors. Setting the pseudo-density $x_{S,i}$ as the smooth minimum of all elements within its cone-shaped pattern makes it void if any pattern element is void. This ensures that surfaces, excluding those at the design domain boundaries, respect to the prescribed draft angle:

$$x_{S,i} = \frac{1}{p} ln(\frac{1}{n_s} \sum_j e^{p x_{C,j}}) \quad j \in P_i \tag{3.7}$$

P_i represents the cone-shaped pattern with element i positioned at its tip, comprising a total of n_s elements. This pattern's configuration is determined by the finite element mesh and the user-defined draft angle, ensuring that the discretized shape's taper meets or exceeds the specified draft angle.

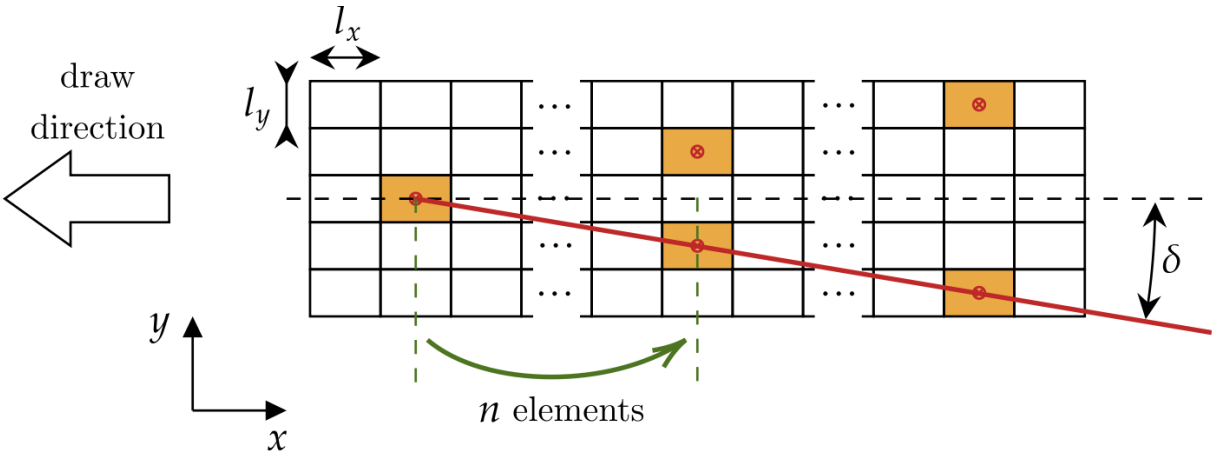


Figure 3.11: Draft Angle Pattern for a Monotonically Decreasing Field in x -draw Direction

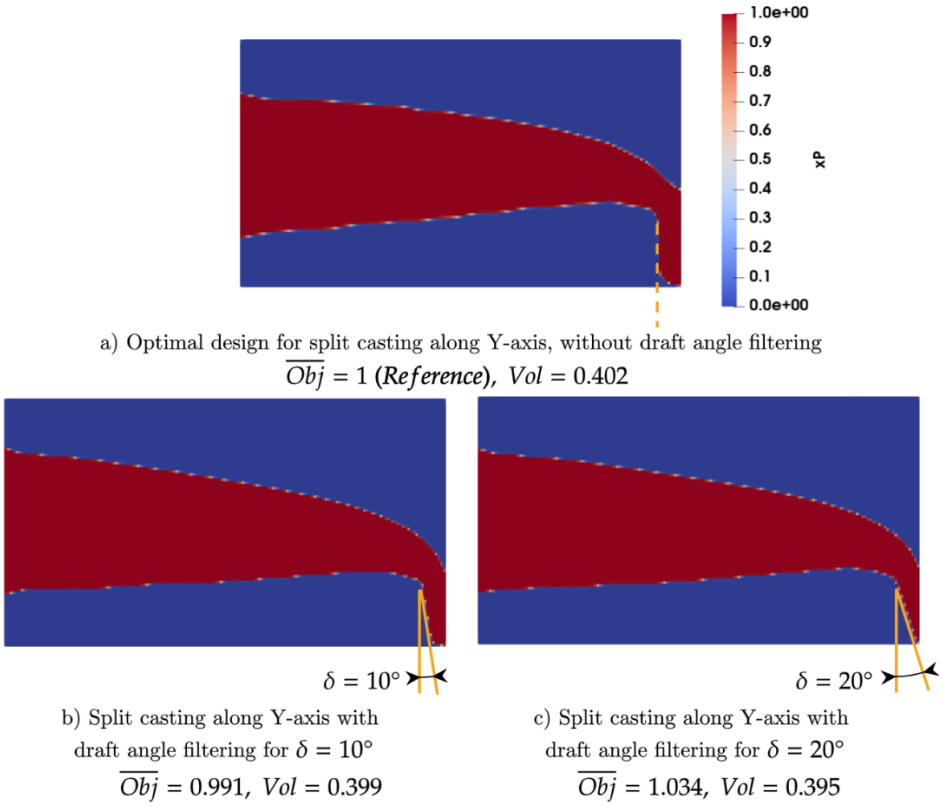
In 2-dimensional problems, the determination of the cone-shaped pattern was achieved for both positive and negative orientations of the principal x and y-axes, employing the subsequent strategy. Assume δ represents the user-defined draft angle, while l_x and l_y denote the dimensions of a single finite element in the x and y directions, respectively. In the case of a castable pseudo-density field resulting from cumulative summation in the x-direction, the pattern enforcing the draft angle is presented in Figure 3.11. This pattern is defined by the parameter n, indicating the number of elements in the x-direction between two consecutive elements within the pattern. The value of n, determining the pattern's aperture, is computed based on the draft angle using the following formula:

$$n_x = \frac{l_y}{l_x tan(\delta)} \tag{3.8}$$

Defining the parameter n for a draft angle pattern in the y direction is similar:

$$n_y = \frac{l_x}{l_y tan(\delta)} \tag{3.9}$$

Introducing draft angle erosion into the optimization process results in a more robust design that adheres to the prescribed draft angle, presented in Figure 3.12. For comparison, the reference design is the undrafted y-axis split draw design depicted in Figure 3.12. The designs shown in Figures 3.12 were achieved by solving the same topology optimization problem using a 160 $\times$ 90 mesh and targeting a volume of $V^* = 0.3$. These designs integrate the draft angle erosion pattern for δ values of 10° and 20°, respectively.



a) Optimal design for split casting along Y-axis, without draft angle filtering
$\overline{Obj}$ = 1 (*Reference*), *Vol* = 0.402

b) Split casting along Y-axis with draft angle filtering for $\delta = 10°$
$\overline{Obj}$ = 0.991, *Vol* = 0.399

c) Split casting along Y-axis with draft angle filtering for $\delta = 20°$
$\overline{Obj}$ = 1.034, *Vol* = 0.395

Figure 3.12: Effect of Draft Angle Erosion Filter in Topology Optimization Problem

3.3 Casting simulations

As discussed previously, casting is a fundamental manufacturing process that plays an important role in the automotive industry, contributing significantly to the production of various parts and components. This method involves pouring molten material into a mold, allowing it to solidify and take the shape of the mold cavity. The automotive sector extensively relies on casting for manufacturing components with complex geometries, designs, and high precision. One of the key advantages of casting in the automotive industry is its ability to produce parts in large quantities with consistent quality. This is crucial for meeting the high-volume demands of the automotive market. Different types of casting processes are employed, which are gravity fed (e.g. die, sand and investment) and pressure fed methods, each tailored to the specific requirements of the components being produced.

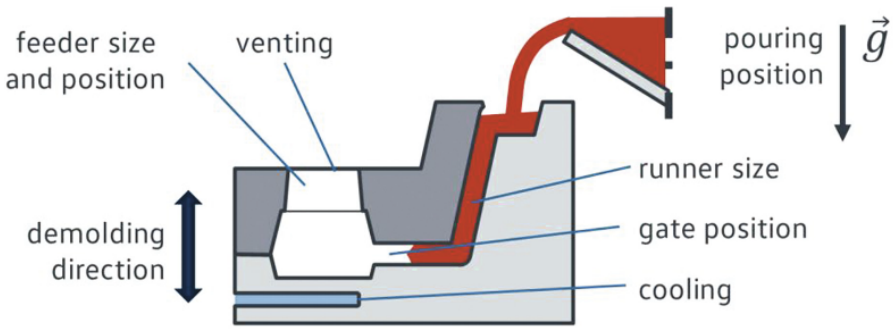


Figure 3.13: General view on casting process parameters (46)

As shown in Figure 3.13, the casting system consists of several components. The mold comprises two halves—the cope (top) and the drag (bottom). The liquid metal fills the space between these halves, forming the mold cavity. Channels called runners guide the molten metal from the sprue (the channel that receives the poured metal) to the mold cavity. The point where the runner connects with the cavity is called the ingate. Extra cavities, known as risers or feeders, are created on the top surface of the mold. These reservoirs hold excess metal. As the metal cools and solidifies in the cavity, it contracts, and the additional metal from the risers flows back to prevent any voids or defects in the final cast. Vents, small openings in the mold, allow gases and air within the cavity to escape to the atmosphere during the casting process. This helps in preventing defects caused by trapped gases or air pockets in the final product.

Numerical simulations of manufacturing processes can minimize the need for trial and error typically associated with new product development, leading to time and cost savings. Specifically within casting processes, numerical simulations enable the analysis of fluid flow, heat transfer, solidification, and defect behavior. Most common numerical techniques for process simulation include the Finite-Element Method, applied, for example, in (36) to enhance the gate system in high pressure die casting

(HPDC), and the Finite-Volume Method (FVM), widely employed in computational fluid dynamics (CFD) (37).

Simulations for the casting process provide valuable insights into defect formation and help ensure a consistent casting procedure. Given the diverse range of casting methods, assessing simulation effectiveness requires analyzing various simulation outcomes. The majority of casting simulations available commercially employ the Eulerian method within the Finite Volume Method (FVM) to assess metal flow throughout the filling phase. The interface between metal and air is depicted using the Volume of Fluid (VOF) method (38). However, these simulations are time-consuming and demand significant computational resources. Alternative methods, such as the Smoothed Particle Hydrodynamics (SPH) utilized in (39) for High-Pressure Die Casting (HPDC) simulations, have not effectively reduced calculation times. As a result, employing casting simulations within computational optimization processes involving multiple iterations becomes highly computationally demanding. Optimization algorithms have typically been limited to adjusting a small set of parameters (40),(41). Hence, there is a need for simplified models to cut down on calculation time and facilitate autonomous optimizations.

Due to the high complexity and computationally demanding to perform the full simulation of the casting system, the current work focuses specifically on the heat transfer dynamics within the mold cavity during the casting process. Rather than simulating the entire casting process, the emphasis lies in understanding and optimizing the flow of heat within this critical space. The objective is to understand the thermal aspects of the casting process, concentrating on how heat distributes within the mold cavity as the molten metal is poured. By narrowing the scope to heat transfer dynamics, this work aims to optimize the design and structure within the mold cavity to enhance casting efficiency and minimize defects.

3.3.1 Niyama's criterion

To anticipate potential issues like shrinkage, hot tears, or mold erosion in specific regions of a casting, various numerical criteria have been developed. Among these, an essential aspect of assessing feeding efficiency involves solidification analysis. The initial criterion proposed by Pellini in 1953 (53) aimed to predict centerline shrinkage. Subsequently, Niyama (54) enhanced this criterion by establishing correlations between solidification simulations and observed centerline shrinkage in steel castings. Niyama's criterion relies on the relationship between centerline shrinkage and the temperature gradient concerning the solidus temperature and cooling rate within the solidifying casting. This relationship is expressed as follows:

$$Niyama = \frac{G_s}{\sqrt{\dot{T}}} \tag{3.10}$$

Where G_s is the temperature gradient and $\dot{T}$ is the first derivative of the temperature or cooling rate [K/s]. From Equation 3.10, it was found that for values larger than 1, the metal is free from porosities, and when the Niyama criterion gives a value less than 1, shrinkage problems are observed regardless of local solidification time (55).

3.3.2 Combining topology optimization and casting simulation

Subsequently, it also aims to integrate process simulation into the structural topology optimization framework of cast components. In the literature, there have been various methods for simulating processes involved in designing and optimizing casting systems, such as molds or ingate configurations for specific component geometries. Given the substantial influence of mold design on casting quality, process simulation becomes pivotal in optimizing either the ingate system or mold design, aiming to refine process efficiency or the final part's quality. For instance, Jadhav and Hujare (42) uses process simulation to refine cooling system placement, thereby reducing stresses within the mold to extend its lifespan. Similarly, Pinto and Silva (43) employs CFD simulations to analyze cavity filling, quantifying porosity issues. Using these findings, adjustments to molten metal flow systems are made, leading to process enhancements and a deeper understanding of flow phenomena in casting procedures. Moreover, (44) introduces a method involving machine learning coupled with process simulations to perform multi-objective optimization for casting solidification, offering a comprehensive approach that considers various factors simultaneously.

Nevertheless, the approach in this work diverges as the component's geometry is not known beforehand, instead, it is generated iteratively from an initial design space. This iterative process combines topology optimization and parallel process simulations to derive optimize component's design. While existing commercial topology optimization accounts for process knowledge through manufacturing restrictions Liu and Ma (45), (46) introduces an innovative method leveraging casting simulations to enhance the manufacturability of topologically optimized components. This approach determines flow-induced defect sources, areas with high vorticity, regions prone to delayed filling, and identifies defect locations unfeasible for feeding due to premature solidification of the melt (47). (48) proposes the utilization of Dijkstra's shortest-path algorithm (49) specifically for investigating High Pressure Die Casting parts. Moreover, these findings have been merged with casting simulations to filter out poorly manufacturable areas within a topology optimization step. Furthermore, another research endeavor focuses on optimizing parametric components using both structural and process simulations.

In the literature, the emphasis from a process perspective revolves around refining mold designs and ingate systems to enhance process quality for established geometries. Meanwhile, significant attention has been directed toward evolving manufacturing constraints from a structural standpoint. However, there is a limited work deeper into integrating process knowledge into the topology optimization (TO) process beyond simple manufacturing constraints. The studies detailed in (46)-(48) integrate comprehensive casting process simulations into toplogy optimization procedures, although with prolonged iteration times due to the high cost associated with acquiring results for parameters like vorticity.

In this work, a new workflow for automatically merging topology optimization (TO) with process simulation outcomes is introduced. Using a casting process-driven standard, the structurally enhanced design undergoes adjustments through implicit modeling, producing an optimized design proposal. This proposal demonstrates improved

manufacturability, making it a structurally and process-optimized solution.

3.3.3 Methods and software for casting simulation

For the simulation of casting process, various commercial software could be made use, such as, Flow-3D, ProCAST, MAGMA, COMSOL Multiphysics, Click2Cast, which can offer a range of capabilities for simulating casting processes, including detailed modeling of filling to solidification, thermal and stress analysis, predicting casting defects and optimizing casting processes.

In this work, an open-source computational fluid dynamics (CFD) software package, OpenFOAM is used to perform the casting related studies for its parallel processing capabilities allow it to efficiently handle large-scale simulations and being cost effective compared to commercial software licenses.

4 Mathematical Formulation of Thermal-fluid Systems

In the field of thermal-fluid systems, optimizing heat transfer is pivotal for achieving efficient operations and desired product properties. Systems such as casting processes rely on complicated thermal interactions to mold materials effectively. Addressing optimization challenges involves determining a balance between complicated objectives and constraints, a task increasingly handled through the integration of computational fluid dynamics (CFD) for simulating flows and numerical optimization techniques.

The optimal configurations within thermal-fluid systems often include minimizing an objective function while conforming to specific thermal and fluidic constraints. Among the current methodologies, the adjoint strategy emerges as a powerful tool for addressing challenges, particularly beneficial when dealing with numerous design variables, the adjoint approach revolves around computing a sensitivity map across the system. This map outlines how alterations to these variables can yield an optimal heat transfer configuration, empowering automatic geometric modifications to enhance thermal-fluid characteristics.

While adjoint optimization has found substantial application in reducing pressure losses and optimizing shapes, its exploration within thermal-fluid systems, especially in casting processes, remains a promising field. Employing adjoint methods to achieve specific thermal objectives, such as regulating temperature gradients or optimizing fluid flow within casting molds, shows great potential.

This study aims to employ adjoint optimization method in simulating thermal-fluid systems, specifically in casting processes to compute the sensitivities of heat transfer and fluid dynamics within these processes. Detailed discussions on the governing equations governing the system are presented in following sections.

4.1 Governing equations and problem formulation

The equations describing thermal-fluid problem consist of a group of non-linear partial differential equations. These equations govern in phenomena involving Newtonian fluids which involve three fundamental equations; the continuity equation, represents mass conservation within a domain; the momentum or Navier-Stokes equation, applying Newton's second law to fluids; and the energy equation, enabling analysis of heat transfer within the fluid.

$$\frac{\partial \rho}{\partial t} + \bigtriangledown.(\rho v) = 0 \tag{4.1}$$

$$\frac{\partial \rho v}{\partial t} + \rho(v.\bigtriangledown)v - \eta \bigtriangledown .(\bigtriangledown v + \bigtriangledown v^T) + \bigtriangledown p - F_b = 0 \tag{4.2}$$

$$\rho c \frac{\partial T}{\partial t} + \rho c(v. \bigtriangledown T) = \bigtriangledown.(k \bigtriangledown T) + Q \tag{4.3}$$

where ρ is the density of the fluid, t represents the temporal variable, p is the pressure, F_b represents body forces, $(\bigtriangledown v + \bigtriangledown v^T)$ is the shear-stress tensor, η is the dynamic viscosity, k represents the conductivity of the fluid, T the temperature, c the specific heat capacity and Q represents the heat source term.

A general optimization problem includes an objective function to be minimized or maximized subject to different constraints. In the case of a general cost function, depending on flow variables v, p, T and the design variable α, subject to the constraint of fulfilling the incompressible, Navier-Stokes equations, the problem can be written as:

$$\begin{array}{cc} min & J(\alpha, U) \\ s.t. & g_i(\alpha) \leq 0 \quad i = 1, ..., m \\ & 0 \leq \alpha_j \leq 1 \quad j = 1, ..., N \\ & R(\alpha, v, p, T) = 0 \end{array} \tag{4.4}$$

In this case, the design variable α is pseudo-density values $[0, 1]$, and the constraints R are given by governing equations defined in Equation 4.1-4.3. Therefore, the incompressible Navier-Stokes equations now become:

$$R_v = \frac{\partial \rho v}{\partial t} + \rho(v.\bigtriangledown)v - \eta \bigtriangledown .(\bigtriangledown v + \bigtriangledown v^T) + \bigtriangledown p + \alpha v = 0 \tag{4.5}$$

$$R_p = \frac{\partial \rho}{\partial t} - \bigtriangledown.(\rho v) = 0 \tag{4.6}$$

$$R_T = \rho c \frac{\partial T}{\partial t} + \rho c(v. \bigtriangledown T) = \bigtriangledown.(k(\alpha) \bigtriangledown T) + Q = 0 \tag{4.7}$$

where R_v, R_p R_T are used as compact notation to refer to the momentum, mass conservation and energy equations respectively such that the constraints abstractly defined in Equation 4.4 can be written as:

$$R = \left\{R_v, R_p, R_T\right\} \tag{4.8}$$

4.2 Thermophysical material properties

With the adapted density-based topology optimization method in this study, the concept of pseudo-density field (α) in the [0,1] is used to describe the distribution of materials between solid and fluid. As per Borrvall and Petersson's research from 2003 (56), an additional friction force linked to the fluid velocity is incorporated into the Navier-Stokes equations:

$$F_b = -\alpha v \tag{4.9}$$

where α is the local permeability and it is related to the pseudo-density field.

For the energy conservation equation, the temperature is controlled by the field $k(\alpha)$, whereas, $k = k_s\alpha$ and k_s is the thermal conductivity of solid which means there is no conduction at the void region, $(\alpha = 0)$:

$$k = k_s\alpha \tag{4.10}$$

The thermal-flow simulation of the conceptual design is performed in OpenFOAM to understand the thermal behaviour of the design problem. The material used for the simulation is Aluminium and the material properties are given in Table 1.

Parameter	**Value**	**Unit**
Density	2710	kgm^{-3}
Thermal conductivity	237	$Wm^{-1}K^{-1}$
Specific heat capacity	903	$Jkg^{-1}K^{-1}$

Table 1: Material Properties in Thermal Simulation

4.3 Continuous adjoint formulation

The objective involves understanding the sensitivity of the objective function J concerning the design field α. This function relies on the state variables. As discussed previously, computing sensitivity directly or using a finite difference approach is extremely costly, especially considering the high number of design variables. To avoid solving the primal equations repeatedly, which scales proportionally with the problem's dimension in each optimization cycle, the adjoint formulation provides a robust method to compute $\frac{\delta J}{\delta \alpha}$. This approach requires solving the flow equations just once per optimization cycle. Additionally, a separate set of equations, known as the adjoint equations, is solved only once within the cycle. By employing this technique, obtaining the sensitivity $\frac{\delta J}{\delta \alpha}$ in each optimization cycle comes at a cost independent of design variables. This highlights the strength and efficiency of the adjoint formulation.

The problem described in Equation 2.15, considering Equation 4.8, can be represented as a Lagrange problem introducing a functional L to minimize as an effective cost function:

$$L := J + \int_{\Omega} u.R_v d\Omega + \int_{\Omega} q.R_p d\Omega + \int_{\Omega} T_a.R_T d\Omega \tag{4.11}$$

where the Lagrange multipliers corresponding to the adjoint velocity u, adjoint pressure q and adjoint temperature T_a are introduced.

Then, it is essential to compute the complete variation of the Lagrangian L to derive the sensitivity of the objective function concerning the design variable α. It is important to note that the Lagrangian functional does not solely rely on α; it also involves contributions from v, p and T. To isolate the dependence on these variables, the chain rule needs to be applied, resulting in:

$$\delta L = \frac{\partial L}{\partial \alpha}\delta\alpha + \frac{\partial L}{\partial v}\delta v + \frac{\partial L}{\partial p}\delta p + \frac{\partial L}{\partial T}\delta T \tag{4.12}$$

The key advantage of the adjoint method lies in formulating optimality equations that isolate the variables of interest. This isolation substantially eliminates the calculation of sensitivity from the complexities of the state equation, making it much simpler.

To establish the adjoint problem and determine the sensitivity of the objective function J with respect to the design variable α, first-order optimality conditions are formulated. The initial condition addressed is the sensitivity equation:

$$\frac{\partial L}{\partial \alpha} = 0 \tag{4.13}$$

Then, the adjoint equations are yielded:

$$\frac{\partial L}{\partial v} = 0 \tag{4.14}$$

$$\frac{\partial L}{\partial p} = 0 \tag{4.15}$$

$$\frac{\partial L}{\partial T} = 0 \tag{4.16}$$

Equation 4.14-4.16 simply recover the original constraints R_v, R_p and R_T:

$$\frac{\partial L}{\partial u} = 0 \tag{4.17}$$

$$\frac{\partial L}{\partial q} = 0 \tag{4.18}$$

$$\frac{\partial L}{\partial T_a} = 0 \tag{4.19}$$

Using the above equations, it is then arrived at the expressions for calculating the sensitivity:

$$\delta L = \frac{\partial L}{\partial \alpha}\delta\alpha \tag{4.20}$$

$$\Leftrightarrow \delta L = \frac{\partial J}{\partial \alpha}\delta\alpha + \int_\Omega u.\frac{\partial R_v}{\partial \alpha}\delta\alpha d\Omega + \int_\Omega q.\frac{\partial R_q}{\partial \alpha}\delta\alpha d\Omega + \int_\Omega T_a.\frac{\partial R_T}{\partial \alpha}\delta\alpha d\Omega \tag{4.21}$$

This highlights the core aspect of the adjoint method, where the equation above specifically depends on reasonably low-cost derivatives related to the porosity term. The efficiency of this method remains intact since the computational expense of sensitivity analysis stays nearly unaffected by the quantity of design variables.

4.4 Calculating the sensitivity

Considering the problem in the discretized framework, the sensitivity of the cost function with respect to the porosity in cell i can be expressed as:

$$\frac{\partial L}{\partial \alpha_i} = \frac{\partial J}{\partial \alpha_i} + \int_\Omega u.\frac{\partial R_v}{\partial \alpha_i} d\Omega + \int_\Omega q.\frac{\partial R_q}{\partial \alpha_i} d\Omega + \int_\Omega T_a.\frac{\partial R_T}{\partial \alpha_i} d\Omega \tag{4.22}$$

In this framework, there is typically no dependence of the objective function J on the porosity. The porosity describes the continuous transition from fluid to solid and is considered an auxiliary variable, thus, $\frac{\partial J}{\partial \alpha_i} = 0$. In addition, the only dependence on α in the constraints R is in the Darcy term in R_v; accordingly, from Equation 4.5-4.7:

$$\frac{\partial R_v}{\partial \alpha_i} = v_i; \quad \frac{\partial R_p}{\partial \alpha_i} = 0; \quad \frac{\partial R_T}{\partial \alpha_i} = - \bigtriangledown .(\frac{k}{\rho c} \bigtriangledown T_i); \tag{4.23}$$

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

$$\frac{\partial L}{\partial \alpha_i} = \int^\Omega (u_i.v_i - T_{a_i} \bigtriangledown .(\frac{k}{\rho c} \bigtriangledown T_i)) \ d\Omega \tag{4.24}$$

Therefore, the sensitivity for each cell, using a single integration point per cell, equals as;

$$\frac{\partial L}{\partial \alpha_i} = (u_i.v_i)V_i - T_{a_i} \bigtriangledown .(\frac{k}{\rho c} \bigtriangledown T_i)V_i \tag{4.25}$$

Hence, to compute the sensitivity, it requires the flow velocity $v_i$and temperature T_i, obtained by solving the state equations R, the volume of each cell V_i, easily derived from the mesh, the adjoint velocity u_i and the adjoint temperature T_{a_i}. This u_i, T_{a_i} needed for the sensitivity equation, are determined using the adjoint equations and the respective boundary conditions.

4.5 Boundary conditions

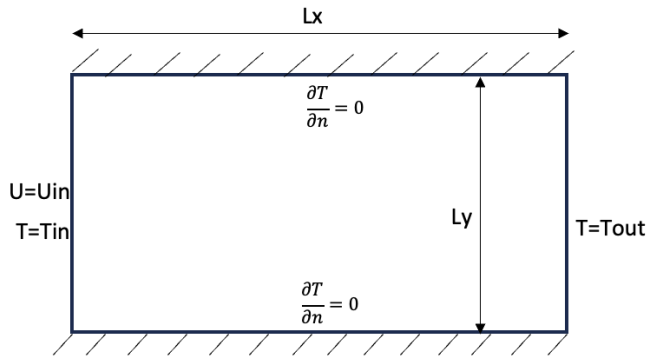


Figure 4.1: 2D Cantilever Beam Detail

Figure 4.1 displays the typical layout of a simple 2D-beam problem. The closed domain $\Omega(x, y) = (0, L_x) \times (0, L_y) \in \mathbb{R}^2$ represents the 2D-beam of lengths L_x,L_y. The walls are thermally isolated and the sides are kept at different temperatures, T_{in} and T_{out} with $T_{in} > T_{out}$. The primal boundary conditions can be seen in Table 2.

	Wall	Inlet	Outlet
v	No-slip	Prescribed value	Zero gradient
p	Zero gradient	Zero gradient	Zero p=0
T	Zero gradient	Prescribed value	Prescribed value

Table 2: Boundary conditions for the primal quantities

The reference temperature T_{ref} is defined as:

$$T_{ref} = \frac{T_{in} - T_{out}}{2} \tag{4.26}$$

4.6 Cost functions formulation

In the optimization of casting processes, achieving optimal thermal performance is important, which involves balancing two key objectives: ensuring a uniform temperature distribution across the casting, crucial for material integrity, and minimizing power dissipation to enhance cooling efficiency. These objectives are focused in the simulations, with one focusing on the mean temperature distribution within the flow and the other addressing the efficient dissipation of power for cooling purposes.

The uniform distribution of temperature throughout the casting is fundamental to prevent thermal stress and distortion, ensuring consistent material properties. Conversely, minimizing power dissipation during cooling is essential for energy efficiency and cost-effectiveness in the casting process.

The cost function for temperature distribution can be written as:

$$J = \frac{1}{2}\int_{\Omega}(T - T^*)^2 \tag{4.27}$$

The cost function for dissipated power is presented as:

$$J = -\int_{\Omega}(p + \frac{1}{2}u.u)u.nd\Omega \tag{4.28}$$

4.7 Cost function 1

In this section, the derivations of adjoint system and sensitivity based on the temperature distribution cost function are briefly described.

4.7.1 Adjoint equations

In this section, the derivation of the adjoint system and the sensitivity of the first objective function is briefly presented. The state equations $R(v, p, T, \alpha) = 0$ can be recalled firstly as Equations 4.5-4.7.

The weak form of the adjoint equations requiring the derivative of L w.r.t. the state variables to be zero can be deduced, as follows:

$$\partial_x L[\delta y] = \partial_v L[\delta v] + \partial_p L[\delta p] + \partial_T L[\delta T] = 0 \tag{4.29}$$

Considering the cost function defined as in Equation 4.27, the adjoint equations are:

$$\frac{\partial u}{\partial t} - (\bigtriangledown u)v - (v.\bigtriangledown)u - \bigtriangledown(2\nu D(u)) + \alpha u + \bigtriangledown q + T_a \bigtriangledown T = 0 \tag{4.30}$$

$$\bigtriangledown.u = 0 \tag{4.31}$$

$$\frac{\partial T_a}{\partial t} - v.\bigtriangledown T_a - \bigtriangledown.(k(\alpha)\bigtriangledown T_a) + T - T^* = 0 \tag{4.32}$$

4.7.2 Adjoint boundary conditions

The boundary conditions for adjoint velocity, pressure and temperature are also derived from the weak form of the adjoint problem. Considering the boundary integrals:

$$\int_\Gamma ((u.v)n + u(v.n) + 2\nu n.D(u) - qn).\delta v - \int_\Gamma 2\nu n.D(\delta v)u = 0 \tag{4.33}$$

$$\int_\Gamma (u.n)\delta p = 0 \tag{4.34}$$

$$\int_\Gamma (T_a v.n + k(\alpha)n.\bigtriangledown T_a)\delta T + \int_\Gamma k(\alpha)n.\bigtriangledown \delta T T_a) = 0 \tag{4.35}$$

From these equations, the following adjoint boundary conditions for the study problem yield:

$$u = 0 \quad on\ \Gamma \tag{4.36}$$

$$T_a = 0 \quad on\ \Gamma_T \tag{4.37}$$

$$\frac{\partial T_a}{\partial n} = 0 \quad on\ \Gamma_A \tag{4.38}$$

4.7.3 Gradient of cost function

The sensitivity of the objection function with respect to the control variable can be computed as:

$$J^{'}(\alpha)[\delta\alpha] = \partial_\alpha L[\delta\alpha] = \partial_\alpha J + \int_\Omega (u, q, T_a)\partial_\alpha R \tag{4.39}$$

in which, it includes a boundary integral, however, considering the case of a null boundary contribution on $J^{'}(\alpha)$, the discrete formulation of the sensitivity in a typical finite volume approximation becomes:

$$J_h^{'}(\alpha_h)[\delta\alpha_h] = \sum_i ((u_i.v_i(\alpha_i)) + \bigtriangledown T_{a_i}.\bigtriangledown T_i k(\alpha_i))V_i\delta\alpha_i \tag{4.40}$$

5 Solver Implementation in OpenFOAM

5.1 OpenFOAM General Aspects

OpenFOAM (Open-source Field Operation And Manipulation) serves as the software employed within this thesis to perform topology optimization of heat transfer related scenarios. It is an open-source C++ program extensively utilized in Computational Fluid Dynamics (CFD). It operates on the finite-volume method, capable of managing diverse physics, ranging from straightforward laminar flows to complex flow dynamics encompassing heat transfer, acoustics, electromagnetism, and more.

OpenFOAM is distributed with a large number of built-in solvers, covering different aspects of simulations which can be done. These solvers can be referenced as examples and new applications can be developed to fit the needs of the one's simulation and study. Its related documentation, such as, user guide (13) and tutorial guide (14) are available online in details and a learner can gain understanding of how the cases are assembled and evaluated within the software environment.

5.1.1 Finite volume method

Fluid equations usually take the form of non-linear partial differential equations and so, most of time, no analytical solution can be derived from them. In that context, different numerical techniques are employed to reach an approximation of the solution to these problems. These methods require a discretization of the domain in which the solution is going to be calculated. OpenFOAM employs the finite volume method, an approach in Computational Fluid Dynamics (CFD) that discretizes equations by dividing the computational domain into smaller control volumes. This method involves the integration of equations over these volumes, focusing on the conservation laws of mass, momentum, and energy. It facilitates the accurate simulation of fluid flow by calculating fluxes across the faces of these volumes, enabling the analysis of various physical phenomena within the fluid domain.

Similarly to the finite element method, the FVM also needs a discretization of the geometric domain but in this numeri- cal method, the elements used to integrate the algebraic equations represent- ing the conservation partial differential equations are finite volumes. Some of the terms in the conservation equation are converted into face fluxes and evaluated in the discretized finite volumes. These face fluxes are strictly conservative. This is that the flux entering the volume is equal to the flux leaving the adjacent volume.

Geometric domain discretization

The finite volume method relies on discretizing the computational domain into control volumes (CV), each with a computational point or centroid where the solution is calculated. OpenFOAM employs a cell-centered approach, defining unknowns at these cell centers and computing their values as averages within the cells. Control volumes are determined by neighboring cells; internal faces separate volumes with adjacent neighbors, while faces without neighboring volumes are considered boundaries.

Discretization of the fluid dynamic's equations

The continuity equation, Navier-Stokes equations, and the heat equation can be expressed more generally using the Reynolds transport theorem formulation:

$$\int_{V_p} \frac{\partial \rho \Phi}{\partial t} dV + \int_{V_p} \bigtriangledown .(\rho \vec{u} \Phi) dV = \int_{V_p} \bigtriangledown .(\rho \Gamma_\Phi \bigtriangledown \Phi) dV + \int_{V_p} S_\Phi dV \tag{5.1}$$

where V_P is the control volume cell, Φ may be any scalar or vectorial variable of the continuum, Γ_Φ is the diffusivity of the variable and S_Φ is a source term.

In order to recover the continuity, momentum and energy equations, the parameters shown in Table 3 need to be shaped in the transport equation.

Equation	Φ	Γ_Φ	S_Φ
Continuity	1	0	0
Momentum	$\vec{u}u$	v	$- \bigtriangledown p$
Energy	C_pT	κ	0

Table 3: Parameters to recover continuity, momentum and energy equations

The fluid variable is determined by integrating itself across the volume cell, resulting in the following form:

$$\Phi = \Phi_P = \frac{1}{V_P} \int_{V_P} \Phi(x) dV \tag{5.2}$$

Therefore, a complete discretization of the previous terms is needed to solve the physics regarding a general fluid dynamics problem.

5.1.2 Structure of a case in OpenFOAM

In this section, a brief explanation on the functioning of OpenFOAM software is presented, including the details of solver extension and implementation of the case in the current thesis.

To conduct a simulation, OpenFOAM relies on a series of files dedicated to configuring specific case parameters such as geometry, boundary conditions, initial states, discretization schemes, etc. These files collectively define the necessary settings for running simulations tailored to different scenarios. The minimum set of files required to run an application is shown in Figure 5.1.

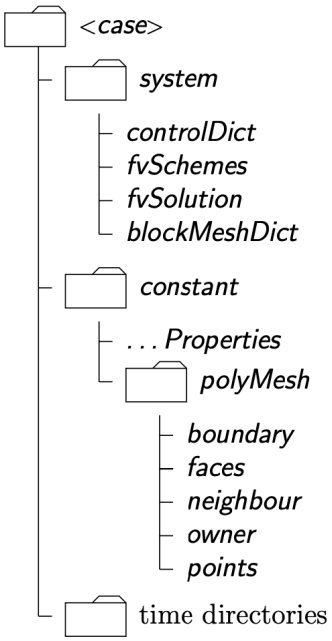


Figure 5.1: Case Directory Structure

The constant directory

It contains fields like gravity and other unchanging physical parameters for the specific simulation. Additionally, upon creating the mesh using the 'blockMesh' command, the 'polyMesh' directory emerges, consisting crucial mesh-related data such as points, faces, boundary and associated details.

The system directory

Within this directory, it includes all the files defining the solution's parameters and its control schemes. A brief overview of the files utilized in this project is described below:

- **controlDict:** sets input parameters essential for the creation of the database. The compulsory entries are the computational time, time step control, data reading and writing, etc.
- **fvSchemes:** sets the numerical schemes for terms, such as derivatives in equations, that appear in applications being run. A scheme should be set for each of the following terms:
  - Time schemes, ddtSchemes, discretize the temporal derivatives $\frac{d}{dt}, \frac{d^2}{dt^2}$.
  - Gradient schemes, gradSchemes, discretize the gradient terms $\bigtriangledown\phi$.
  - Divergent schemes, divSchemes, discretize the convective terms $\bigtriangledown.(U\phi)$.
  - Laplacian schemes, laplacianSchemes, discretize the conductive terms $\bigtriangledown.(DT\bigtriangledown \phi)$.
  - Interpolation schemes, interpolationSchemes, contains the infor mation of values at the faces are obtained from the values at the centres. The most common is the linear interpolation.

- snGradSchemes, discretize the surface normal gradients at the faces

- **fvSolution:** contains several subdictionaries which control the solver parameters. There are three main subdictionaries:
  - Linear solver control: it is outlined within the 'solvers' subdictionary, enabling control over these settings. It is crucial to differentiate between the linear solver, responsible for the numerical approach in addressing transport equations, and the application solver, which encompasses all equations and algorithms tackling the problem at hand. This section allows selection of parameters like tolerance, preconditioner, and more.
  - Relaxation factors: Within this sub-dictionary, the underrelaxation technique is managed, which enhances solution stability, allowing the selection of varied relaxation factors for different solution variables.
  - Algorithm selection: In this section, the method for coupling transport equations is chosen from options like SIMPLE, PIMPLE, and PISO. Additionally, other solution control parameters can be specified within this context. Within the fvSolution dictionary, various parameters extending beyond transport equation resolution can be specified to regulate solvers and algorithms. For instance, in this thesis, the application solver will extended to temperature equation, addressing the pure conduction heat equation or conjugate heat equation.

- **blockMeshDict:** Within this file, configurations for the computational domain and associated parameters are established. This comprises defining the size and shape of the computational domain and mesh, as well as specifying boundary conditions. Once the blockMeshDict dictionary is appropriately configured, executing the command blockMesh generates the mesh, stored in the polyMesh directory within the constant folder.

The time directory

It comprises individual data files for specific fields. These data files contain either the initial values and boundary conditions, which the user needs to define for outlining the problem, or the outcomes generated by OpenFOAM and stored in files. It shall be noted that OpenFOAM fields always need initialization, even in scenarios where the solution might not strictly require it, such as in steady-state problems.

5.2 Description of the solver

In this thesis, a solver applying a density based method, using the adjoint approach to compute the sensitivities with respect to the design variables, is studied. The porosity of each cell, represented by the design variable α, is utilized to penalize unfavorable cells. The update occurs in a single-step process, employing partially converged fields to calculate sensitivity.

In this section, it describes the basic algorithm of "adjointShapeOptimizationFoam" and how extensions are made to solve the simulation problem for cost function related to heat transfer case. In order to be able to validate the results to some extent, the implementation of cost function and sensitivity of each cell is presented.

5.2.1 HeatTransferAdjointFoam

In order to compute sensitivities using an adjoint method, it has been established in the theory section that solving two sets of governing equations is necessary: the primal and the adjoint. Both sets contain comparable terms, allowing for the utilization of similar algorithms to solve them. The solver employs a "one-shot" strategy to solve both the primal and adjoint systems, calculating sensitivities using partially converged quantities. The adjoint system is solved in a manner similar to the primal system, the built in solver, "adjointShapeOptimizationFoam" employing a SIMPLE-type algorithm to couple pressure and velocity. In this scope of thesis, it requires solving an additional equation, therefore, temperature equation is implemented and a new solver named "HeatTransferAdjointFoam" is complied. Figure 5.2 below graphically illustrates the solution procedure.

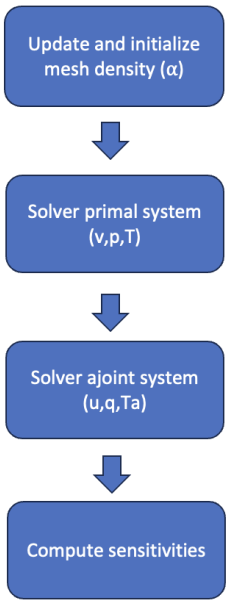


Figure 5.2: Solution Procedure in HeatTransferAdjointFoam.C

5.2.2 HeatTransferAdjointFoam.C

The solver is initialized by including a number of files:

```
int main(int argc, char *argv[])
{
        #include "setRootCase.H"
        #include "createTime.H"
        #include "createMesh.H"
        #include "createFields.H"
        #include "initContinuityErrs.H"
        #include "initAdjointContinuityErrs.H"
```

Figure 5.3: Listing:1 File HeatTransferAdjointFoam.C

The initial header file creates an argList object, named args, comprising the arguments passed into the main function. It verifies both the case path and root path alongside the argument list. Additional details regarding the case, such as processor count and directory specifics, are stored. The files createTime.H and createMesh.H initiate an object of the Time class, runTime, and an object of the fvMesh class, mesh, respectively. The runTime object contains time-related data, essential counters, and values extracted from the control dictionary. On the other hand, the mesh object includes finite volume mesh details, such as, node positions, face areas, cell volumes, and information related to interpolation schemes to be employed.

In createFields.H, the initialization process involves setting up the variables utilized by HeatTransferAdjointFoam. This contains the primal and adjoint variables, alongside mesh density, other dimensioned scalars, and the turbulence model. As for initAdjointContinuityErrs.H, it sets the cumulative adjoint continuity error to zero during initialization.

```
// Pressure-velocity SIMPLE corrector
{
        // Momentum predictor

        tmp<fvVectorMatrix> UEqn
        (
        fvm::div(phi, U)
        + turbulence->divDevReff(U)
        + fvm::Sp(alpha, U)
        );

        UEqn().relax();

        solve(UEqn() == -fvc::grad(p));
```

Figure 5.4: Listing:2 File HeatTransferAdjointFoam.C

At the start of the main loop of the solver, the mesh density field is updated. As for momentum predictor, the different terms of the constraints in Equation 4.2 can be easily identified, including the turbulence term, a pointer to the RASModel used, which is constructed in createFields.H. The sink term due to porosity is treated like other source terms and is implemented implicitly using $fvm :: Sp(alpha, U)$ as shown

in Figure 5.4. The pressure and velocity of the primal system are then solved by using the value of the velocity from the momentum predictor to correct the pressure and then correcting the velocity with the new value of p in a momentum corrector, in a SIMPLE-type algorithm. As an extension to the built-in solver, the energy equation corresponding to temperature distribution is included by reading thermal coefficients from the properties under createFields.H, then the primal temperature is solved.

The adjoint system is solved in a very similar approach to the primal, using the same viscosity (frozen turbulence) and calculated values of the primal quantities needed.

5.2.3 createFields.H

As mentioned previously, it has been discussed how the createFields.H file initializes the variables utilized by the solver. The configuration for writing and reading the adjoint pressure, velocity and temperature is established using the IOobject class, following the approach for the primal quantities. A number of scalar fields used in the solver are also defined in this createFields.H.

```
Info<< "Reading field pa\n" << endl;
volScalarField pa
(
    IOobject
        (
        "pa",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
        ),
    mesh
);
```

Figure 5.5: Listing:2 createFields.H

As shown in Figure 5.5, the code makes the solver write the data to the file “pa” in a directory named after the timestep. The same is done for the adjoint velocity U_a and the adjoint temperature T_a.

5.2.4 Boundary conditions

The existing implemented adjoint sovler of OpenFOAM optimizes for total pressure loss rather than controlling power dissipated or temperature distribution which the thesis focuses on. Therefore, the following files are modified and implemented for those boundary conditions as described in the theory section.
The current setup incorporates additional boundary conditions mainly for the outlet and not for the inlet since the boundary conditions for the inlet patch often align with existing conditions, rendering them non-distinctive for the adjoint method.

adjointOutletPressureHeatFvPatchScalarField.C
adjointOutletVelocityHeatFvPatchVectorField.C

5.2.5 costFunction.H

The intended cost function for minimizing temperature distribution within the domain, as described in the theory section is implemented. In Figure: 5.6, it represent the cost function implemented for power dissipation. More than one cost function can be implemented and user can choose which cost function to be considered for the simulation.

```
for (iLoop=0; iLoop<nObjPatch; iLoop++)
{
        if (objFunction==1) {
                jDissPower =
                        jDissPower - sum(
                        phi.boundaryField()[objPatchList[iLoop]]*
                        (p.boundaryField()[objPatchList[iLoop]]
                        + 0.5*
                        magSqr(U.boundaryField()[objPatchList[iLoop]]))
                        );
        }
```

Figure 5.6: Listing:3 costFunction.H

5.2.6 sensitivity.H

As explained in the theory section of computing sensitivity, the equations are implemented within the solver. In order to be able to obtain the output file for the sensitivity of each cell, an addition to createFields.H has to be made and the values have to be updated in every loop in HeatTransferAdjointFoam.C.

6 Validation of Methods

To validate the solver's progression across various extended stages, creating a benchmark problem for testing proved beneficial. Therefore, 2D cantilever beam has been incorporated as an initial example. This demonstration offers a clear and evident way to check the effectiveness of the implemented solver to analyze the simulation flow in OpenFOAM and continue topology optimization in Python FEniCS in order to design manufacturable and lightweight components.

6.1 Numerical simulations in OpenFOAM

The problem layout consists of 2D domain $\Omega(x, y) = (L_x, L_y) \in \mathbb{R}$ of lengths, $L_x = 50$ [mm], $L_y = 20$ [mm] with prescribed temperature at the boundary, T_h in $(x, y) = (0, 0)$ and T_c in $(x, y) = (L_x, L_y)$, with $T_h > T_c$ and the adiabatic horizontal walls.

6.1.1 Mesh density data transfer between FEniCS solver and OpenFOAM

As the main objective of this work is to integrate OpenFOAM simulation results to the structural topology optimization framework as explained in the flow chart, Figure 1.2 , it is critical to transfer the pseudo-density data of structure's domain obtained from iterations of topological optimization problem to OpenFOAM accurately. This transfer of pseudo-density data to OpenFOAM to further perform the intended simulation and sensitivity analysis shall be done for every iterations or at user's defined intervals. While it ensures that the most recent and updated pseudo-density information is integrated into each iteration, it incurs significant computational overhead due to the repeated data transfers between the models. This may may result the simulation's overall efficiency and escalate computational expenses significantly. On the other hand, opting for user-defined intervals to execute the transfer mitigates the computational burden to a certain extent.

Topology optimization is being done using the previously developed codes in Python with FEniCS finite element solver; FEniCS primarily uses its own mesh format, which is commonly represented in XML-based formats such as XML (.xml) or XDMF (.xdmf). These formats contain information about the mesh topology, geometry, and associated data (such as pseudo-density data). When transferring meshes from FEniCS to other simulation environments, like OpenFOAM, conversion to a format compatible with the target software is often necessary. In OpenFOAM, the mesh is stored in a particular file structure that consists of various files and directories describing the geometry, connectivity, and boundary conditions. OpenFOAM primarily uses ASCII-based formats for readability and ease of manipulation in (.msh) extension. OpenFOAM can also import mesh data in formats like OBJ (.obj), STL (.stl), or VTK (.vtk). Conversion tools or utilities are used to transform these formats into the OpenFOAM-native format.

When transferring a mesh to OpenFOAM from other software or formats (such as FEniCS, Gmsh, etc.), conversion tools or intermediate formats might be necessary to ensure that the mesh data is represented in a manner compatible with OpenFOAM's file structure and formats. After conversion, it's crucial to validate the imported mesh

for accuracy, boundary conformity, and element quality. The methods explored for this conversion are briefly described in the following:

1. **OpenFOAM's mesh conversion utilities:** There are numerous mesh conversion utilities, such as, ansysToFoam, fluentMeshToFoam, gmshToFoam, vtkUnstructuredToFoam, etc. Since there is no direct conversion utilities available from FEniCS mesh (.xdmf), conversion to intermediate format is necessary, such as , Gmsh (.gmsh) or VTK (.vtk). In the scope of this work, ParaView is used to visualize the density fields and it also allows to export the data to other formats, (.csv), (.vtk). Once converted, the mesh can be imported into OpenFOAM using appropriate utilities (vtkUnstructuredToFoam). However, OpenFOAM operates with three-dimensional (3D) meshes by default, therefore, it requires defining the additional dimension and mesh cells appropriately for 2D problems. This can be done by "extrude2DMesh" mesh generation utility for creating a 3D mesh by extruding a 2D mesh with specified thickness.

2. **Mesh generation with the "blockMesh" utility:** The mesh generation utility, blockMesh, is supplied with OpenFOAM, which creates parametric meshes with grading and curved edges. The mesh is generated from a dictionary file named blockMeshDict located in the constant/polyMesh directory of a case. blockMesh reads this dictionary, generates the mesh and writes out the mesh data to points and faces, cells and boundary files in the same directory. For a 2-dimensional geometry, the user has the option to omit block faces lying in the 2D plane, knowing that they will be collected into an empty patch as required.

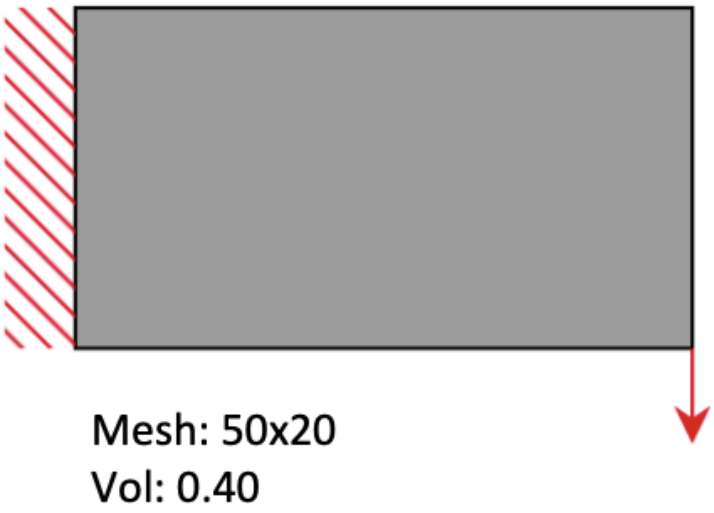
Both methods have been verified to ensure the converted mesh for simulations accurately represents the original geometry and maintains its essential features.

In order to simplify the validation of the method, the solution of the compliance topology optimization problem stated in Equation 2.15, 2.16 is first considered as:

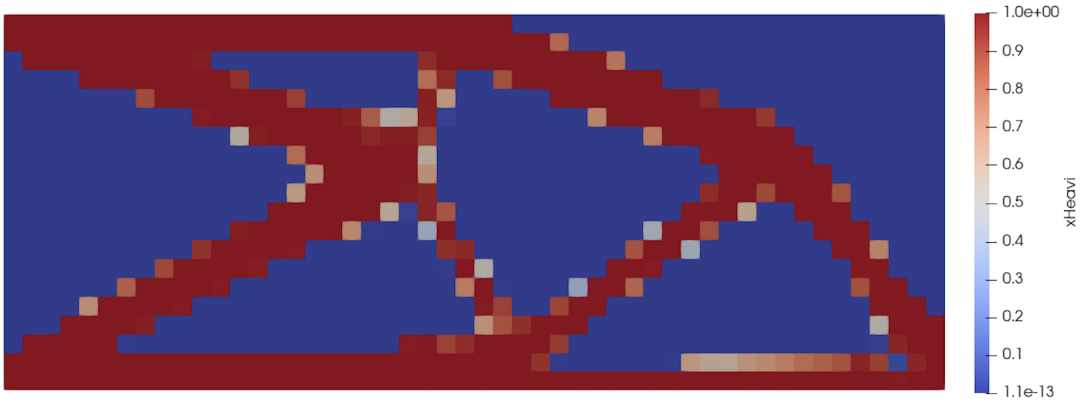
$$f(x_P) = \sum_{e=1}^{N} (x_{P,e})^p \psi_0(u_e)$$

$$\begin{aligned} g(x_P) &= \frac{\sum_{e=1}^{N} x_{P,e}}{N} - V^* \leq 0 \\ &= \sum_{e=1}^{N} x_{P,e} - NV^* \leq 0 \end{aligned}$$

The compliance topology optimization is applied to simple cantilever 2D-beam with boundary conditions and load case, as illustarted in Figure 6.1a and its end result is shown in Figure 6.1b, this pseudo-density data is used to initialize the mesh in the OpenFOAM domain at the start of the simulation before executing the solver equations.



(a) Boundary conditions and load case of 2D cantilever beam



(b) Topology optimization result of 2D cantilever beam

Figure 6.1: Cantilever Beam Topology Optimization Result

6.1.2 2D thermal flow problem

In order to investigate the validity of the solver, a 2D thermal problem is presented. As described in the theory section, the problem is formulated based on Equation 4.4as:

$$\begin{array}{cc} min & J(T) = \frac{1}{2}\int_{\Omega}(T - T_{ref})^2 \\ s.t. & \frac{\partial v}{\partial t} + (v.\bigtriangledown)v - \bigtriangledown.\nu(\bigtriangledown v + \bigtriangledown v^T) + \bigtriangledown p + \alpha v = 0 \\ & \frac{\partial \rho}{\partial t} - \bigtriangledown.(\rho v) = 0 \\ & \frac{\partial T}{\partial t} + (\alpha v.\bigtriangledown T) - \bigtriangledown.(k(\alpha)\bigtriangledown T) = 0 \\ & 0 \leq \alpha \leq 1 \end{array}$$

The problem domain $\Omega(x, y) = (0, L_x) \times (0, L_y) \in \mathbb{R}^2$ is a beam of side $L_x = 50$[mm], $L_y = 20$[mm], with the two vertical sides at fixed temperature, $T(0, L_y) = T_h$ and $T(L_x, L_y) = T_c$, and the adiabatic horizontal sides.

The design variable, α represents the pseudo-density field, $[0, 1]$, which is obtained by solving compliance topology optimization problem using FEniCS solver. Therefore, in the very first step of performing the simulation in OpenFOAM, it is critical that the accurate pseudo-density field to the OpenFOAM mesh has been updated and this has been verified in the earlier section.

Secondly, it has been investigated the validity and reliability of the sensitivity analysis by carrying out a comparison with the finite difference (FD) method. In finite difference method, the difference step is set to 1×10^{-6} as when working with FD, governing equations must be converged fully because sensitivities are computed by subtracting two very close values. A coarse mesh is used to perform the comparison and the detailed values of sensitivities and relative errors are presented in Table 4. The sensitivities computed by the continuous adjoint method in OpenFOAM delivered good agreement with FD values, as illustrated in Figure 6.2. Therefore, the parallel solver is capable of calculating correct sensitivities.

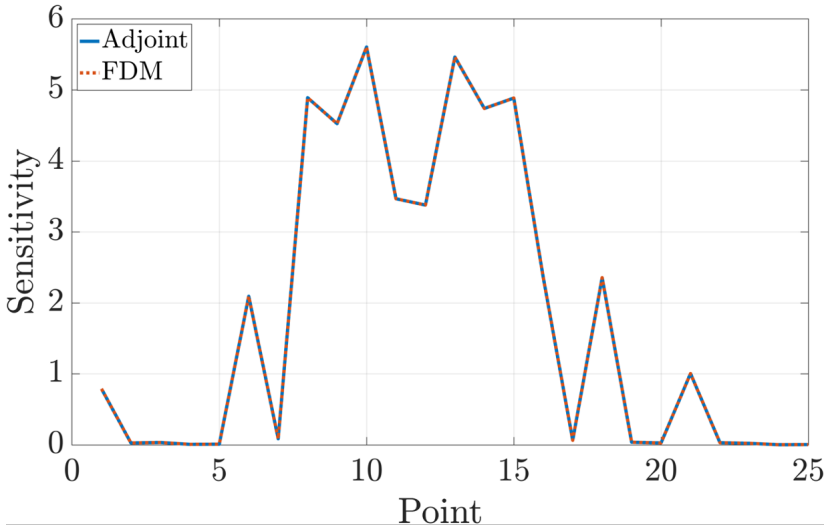


Figure 6.2: Sensitivity Analysis Result

Sensitivities of J							
Cells	Continuous adjoint	FDM	Relative difference (%)	Cells	Continuous adjoint	FDM	Relative difference (%)
1	0.7895	0.7895	0.0063	14	4.7395	4.7394	0.0010
2	0.0263	0.0262	0.1900	15	4.8865	4.8864	0.0010
3	0.0331	0.0330	0.1510	16	2.3587	2.3586	0.0021
4	0.0071	0.0070	0.7083	17	0.0639	0.0638	0.0782
5	0.0091	0.0090	0.5474	18	2.3554	2.3553	0.0021
6	2.0926	2.0925	0.0023	19	0.0366	0.0365	0.1366
7	0.0852	0.0851	0.0587	20	0.0266	0.0265	0.1883
8	4.8910	4.8909	0.0010	21	1.0045	1.0044	0.0050
9	4.5258	4.5257	0.0011	22	0.0297	0.0296	0.1684
10	5.6059	5.6058	0.0008	23	0.0212	0.0211	0.2359
11	3.4673	3.4672	0.0014	24	0.0019	0.0018	0.1588
12	3.3783	3.3782	0.0014	25	0.0041	0.0040	0.4870
13	5.4640	5.4639	0.0009				

Table 4: Sensitivities and relatives error

6.2 Coupling OpenFOAM results into topology optimization framework

In the next step, in order to integrate the sensitivities derived from OpenFOAM simulations into a structural topology optimization framework, transferring these calculated sensitivities to the optimizer of the FEniCS solver becomes essential. The purpose is to kick off the multidisciplinary optimization approach by considering thermal effects alongside structural performance in a single optimization framework. With this combined approach, it aims to create lightweight design solutions that balance structural integrity, thermal requirements and manufacturability optimally.

Figure 6.3 represents the topology optimization solution of the minimized compliance problem with manufacturing constraints through integrated approach. It can be noticed that the solution obtained through this combined approach deliver more feasible solution in terms of manufacturability while taking into account the structural robustness.

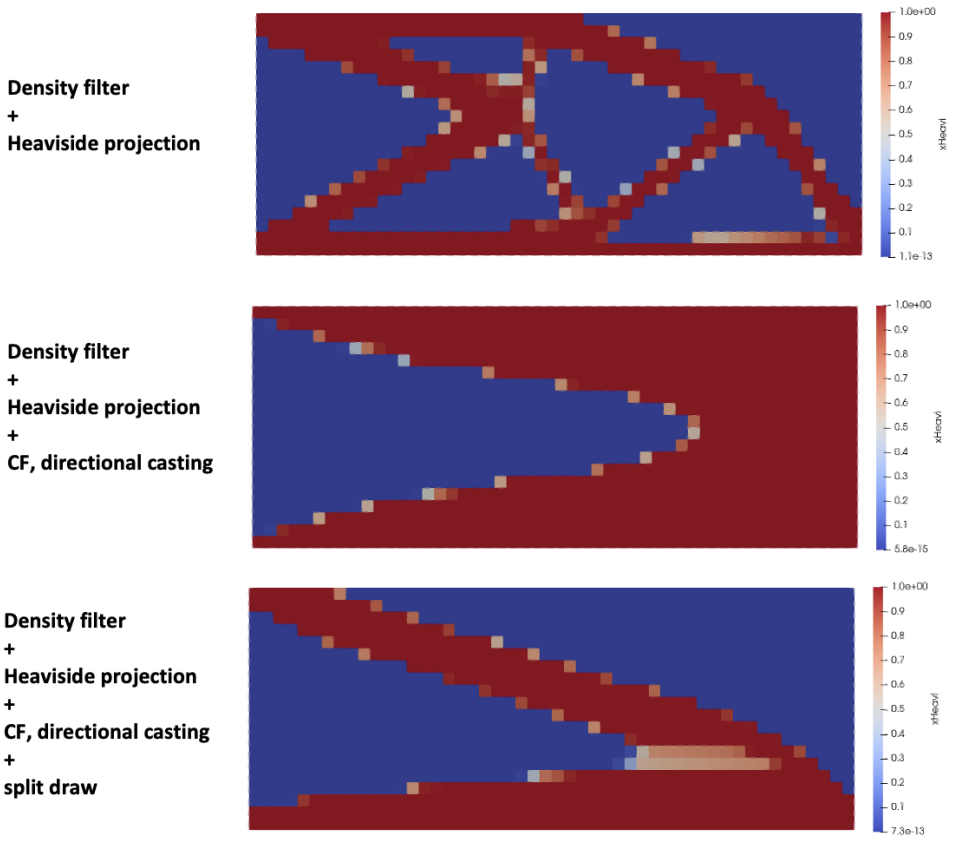


Figure 6.3: Cantilever Beam Topology Optimization Result with Integrated Manufacturing Constraints

7 Conclusion and Perspectives

In this thesis, the first goal was to explore the fundamental principles of topology optimization, initially centered on the widely used SIMP methodology and its integration into a three-field optimization approach. This transformation marked the evolution of the methodology into a multi-field method. This innovative multi-field method integrated several casting manufacturing constraints, implementing filtering and projection techniques onto the design variable field x. By doing so, the aim was to provide engineers with a robust framework capable of accommodating different constraints associated with the casting process.

The ultimate goal was to empower engineers in the creation of cutting-edge, lightweight designs for future cast parts. Through the adoption of this advanced approach, engineers are ready to explore novel design spaces, leveraging the optimization framework to craft innovative solutions that not only meet stringent structural requirements but also adhere to the complexities of casting manufacturing. With this insight, it has been explored to further advance by integrating casting-related simulations into the structural topology optimization framework. To achieve this, the formulation of a simplified heat transfer problem within OpenFOAM has been accomplished and extended the existing solver's capabilities to compute sensitivities through OpenFOAM's continuous adjoint approach. Additionally, ensuring accurate transfer of mesh density data between FEniCS Python and OpenFOAM became a pivotal aspect of this integration. Therefore, the validation process involved confirming accurate update of mesh density data within the software and the accuracy of thermal sensitivities using the finite difference method, strengthening the reliability and robustness of the implemented solver in OpenFOAM.

The integration of OpenFOAM-derived sensitivities into the topology optimization optimizer marks a significant step towards comprehensive optimization. By aligning structural, geometric, and thermal constraints, this approach reaches to integrated optimization, aligning the development of designs that excel in structural integrity, manufacturing feasibility, and thermal management.

While this thesis has laid a foundational work by integrating a OpenFOAM simulation to topology optimization, it only presented the surface of the vast potential within the field of optimizing cast parts. There remains an expansive horizon of possibilities awaiting exploration and refinement beyond the scope of this research. Thus, further improvement on the current method is strongly encouraged.

The following section presents the scope of future work to expand the optimization driven design process of reliable lightweight cast component. The recommendations are based on the experience gained during the progress of the project and are focused on improving the process even further.

1. In this study, the thermal simulation and sensitivity analysis in OpenFOAM is developed based on simplified conductive heat transfer aspect based on design variable $\alpha[0, 1]$, however, this can be further enhanced by conjugate heat transfer

approach, i.e., including the convective heat transfer for the void area within the design domain. This can be done by defining the solid and fluid region based on the mesh density value [0,1] and adding convective term for fluid region in the energy equation.

2. To further challenge the numerical investigation in OpenFOAM for the cast components, particularly to analyze the solidification behaviour, the coupled use of the enthalpy-porosity technique can be established within OpenFOAM. Rösler and Brüggermann (50) introduced a numerical model for a solid-liquid phase change inside a latent heat thermal energy storage. Richter et al. (51), worked out a method for the simultaneous mould filling and solidification process which settles the developing of free surface flow and the liquid-solid phase transition under the volume-of-fluid and enthalpy-porosity methods. By inspiring these work, a new solver based on the coupling of volume of fluid (VOF) and enthalpy-porosity techniques which covers the relevant physical effects during the process of solidification can be developed.

3. Furthermore, for solidification analysis, the Niyama criteria developed can be integrated into solidification solver to evaluate the Niyama values based on the temperature gradient and the cooling rate. This aspect can predict whether the cast part is free from porosities.

4. The current approach can be further extended for practical applications, with complex structural configurations beyond the simple cantilever beam. Such a complex configuration can be determined within the parallelepipedic reference domain by defining three sub-domains: the non-design domain, which can be either set to void or full material and will not be modified by the optimizer, the support region on which boundary conditions are applied and the load region, on which external forces act. This kind of structural configuration can be created using the 3-dimensional finite element mesh generator gmsh developed by C.Geuzaine and J.-F.Remacle (52), which allows the user to simply mark the different sub-domains with numerical labels. The gmsh file can then be converted to an xdmf extension, which FEniCS is able to read and import into the problem, allowing the optimization of mechanical parts of complex shapes. The supports and loads are defined by reading the labels of the imported mesh. Using OpenFOAM's mesh conversion utilities "gmshToFoam", the gmsh file with respective boundary and load conditions can also be converted for simulations with OpenFOAM solvers.

References

[1] M. P. Bendsøe and O. Sigmund, Topology optimization: theory, methods and applications, Springer-Verlag, Berlin, 2003.

[2] R.T. Haftka, Z. Gürdal, and M.P. Kamat, Elements of Structural Optimization, Dordrecht: Kluwer Academic, 1992.

[3] K. Svanberg, The method of moving asymptotes - a new method for structural optimization, International Journal of Numerical Methods in Engineering, 24 (1987) 359-373.

[4] M. P. Bendsøe, Optimal shape design as a material distribution problem, Structural Optimization, 1 (1989) 193-202.

[5] Tyler E. Bruns and Daniel A. Tortorelli. "Topology optimization of non-linear elastic structures and compliant mechanisms". In: Computer Methods in Applied Mechanics and Engineering 190.26-27 (Mar. 2001), pp. 3443–3459 (cit. on pp. 20, 93).

[6] Blaise Bourdin. "Filters in topology optimization". en. In: International Journal for Numerical Methods in Engineering 50.9 (2001), pp. 2143–2158 (cit. on p. 20).

[7] Ole Sigmund. "On the design of compliant mechanisms using topology optimization". In: Mechanics of structures and machines (Print) 25.4 (1997), pp. 493–524 (cit. on p. 21).

[8] James Guest, Jean Prevost, and T. Belytschko. "Achieving minimum length scale in topology optimization using nodal design variable and projection functions". In: International Journal for Numerical Methods in Engineering 61 (Sept. 2004), pp. 238–254 (cit. on p. 21).

[9] Fengwen Wang, Boyan Stefanov Lazarov, and Ole Sigmund. "On projection methods, convergence and robust formulations in topology optimization". en. In: Structural and Multidisciplinary Optimization 43.6 (June 2011), pp. 767–784 (cit. on pp. 22, 23, 93).

[10] Vidan Andrei-Robert, "Topology Optimization of Automotive Components Subject to Casting Constraints". https://matheo.uliege.be/handle/2268.2/14378.

[11] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber. "Topology optimization of channel flow problems". In: Structural and Multidisciplinary Optimization 30.3 (2005), pp. 181–192 (cit. on p. 14).

[12] G. Marck, M. Nemer, and J.-L. Harion. "Topology optimization of heat and mass transfer problems: laminar flow". In: Numerical Heat Transfer, Part B: Fundamentals 63.6 (2013), pp. 508–539 (cit. on pp. 14, 19–21, 25, 29, 35, 69).

[13] OpenFOAM User Guide; https://www.openfoam.com/documentation/user-guide

[14] OpenFOAM Tutorial Guide; https://www.openfoam.com/documentation/tutorial-guide

[15] Ulf Nilsson, Daniel Lindblad, Oliver Petit, "Description of adjointShapeOptimizationFoam and how to implement new objective functions".

[16] Othmer, C. & de Villiers, E. & Weller, H.G. (2007). Implementation of a continuous adjoint for topology optimization of ducted flows. American Institute of Aeronautics and Astronautics, AIAA- 3947.

[17] Othmer, C. (2008). A continuous adjoint formulation for the computation of topological and surface sensitives of ducted flows. Int. J. Numer. Meth. Fluid, 58, 861-877.

[18] Schramm, U., Thomas, H., Zhou, M.: Manufacturing Considerations and Structural Optimization for Automotive Components, SAE 2002 World Congress, Detroit (2002)

[19] Zhou, M., Fleury, R., Shyy, Y.K., Thomas, H., Brennan, J.M.: Progress in topology optimization with manufacturing constraints. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta (2002)

[20] Harzheim, L., Graf, G.: A review of optimization of cast parts using topology optimization II-topology optimization with manufacturing constraints. Struct. Multi. Optim. 31, 388–399 (2006)

[21] Marx, S., Kristensen, A.S.: Implementation of molding constraints in topology optimization. In: Twenty Second Nordic Seminar on Computational Mechanics (2009)

[22] Almeida, S.R.M., Paulino, G.H., Silva, E.C.N.: A simple and effective inverse projection scheme for void distribution control in topology optimization. Struct. Multi. Optim. 39, 359–371 (2009)

[23] Saxena, P., Agrawal, P., Jain, H.: Casting simulation & optimization for T casting junctions using response surface methodology. Int. J. Eng. Res. Technol. 2(5), 2076–2081 (2013)

[24] Li, Q., Steven, G.R., Querin, O.M., Xie, Y.M.: Shape and topology design for heat conduction by evolutionary structural optimization. Int. J. Heat Mass Transf. 42, 3361–3371 (1999)

[25] Michailidis, G.: Manufacturing constraints and multi-phase shape and topology optimization via a level-set method, École Polytechnique (2014)

[26] Teichmann, E.: Efficient structural update for three-dimensional topology optimization problems using level set functions, Universität Kaiserslautern (2008)

[27] Durde, N.C., Meier, L., Hoffmann, H., Scheurle, J.: Model based strategies for an optimised ribbing design of large forming tools. Prod. Eng. Res. Dev. 3, 435–440 (2009)

[28] Schumacher, A., Wagner, A., Smarsly, W., Fischer, R., Bartsch, M., Scholz, A.: Process chain simulation integrated in automatic optimization loops for developing cast parts founded by validated material models. In: 4th European Conference on Materials and Structures in Aerospace, 7–8 February 2012, Hamburg (2012)

[29] Campbell, J.: Castings Practice: The Ten Rules of Castings. Elsevier Butterworth-Heinemann, Oxford (2004)

[30] Campbell, J.: Castings Practice: The Ten Rules of Castings. Elsevier Butterworth-Heinemann, Oxford (2004)

[31] Chvorinov, N.: Theory of casting solidification. Giesserei 27, 177–186 (1940)

[32] A. M. Bradley. Pde-constrained optimization and the adjoint method. https://cs.stanford.edu/ ambrad/adjointtutorial.pdf, November 2010

[33] K. Svanberg. MMA and GCMMA – two methods for nonlinear optimization. en. 2014 (cit. on pp. 46, 47)

[34] Michaël Bruyneel. "Optimization of laminated composite structures: problems, solution procedures and applications". English. In: (2008). Publisher: Nova Science Publishers (cit. on p. 48).

[35] Matthijs Langelaar. "Topology optimization for multi-axis machining". en. In: Computer Methods in Applied Mechanics and Engineering 351 (July 2019), pp. 226–252 (cit. on pp. 5, 10, 50, 56, 93).

[36] Mehtedi, M.E.; Mancia, T.; Buonadonna, P.; Guzzini, L.; Santini, E.; Forcellese, A. Design Optimization of Gate System on High Pressure Die Casting of AlSi13Fe Alloy by Means of Finite Element Simulations. Procedia CIRP 2020, 88, 509–514.

[37] Shahane, S.; Aluru, N.; Ferreira, P.; Kapoor, S.G.; Vanka, S.P. Finite Volume Simulation Framework for Die Casting with Uncertainty Quantification. Appl. Math. Model. 2019, 74, 132–150.

[38] Hirt, C.W.; Nichols, B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. 1981, 39, 201–225.

[39] Cleary, P.; Ha, J.; Alguine, V.; Nguyen, T. Flow modelling in casting processes. Appl. Math. Model. 2002, 26, 171–190.

[40] Dabade, U.A.; Bhedasgaonkar, R.C. Casting Defect Analysis using Design of Experiments (DoE) and Computer Aided Casting Simulation Technique. Procedia CIRP 2013, 7, 616–621.

[41] Hahn, I.; Sturm, J. Autonomous optimization of casting processes and designs. In Proceedings of the World Foundry Congress, Hangzhou, China, 16–20 October 2010; pp. 16–20.

[42] Jadhav, A.R.; Hujare, D.P.; Hujare, P.P. Design and Optimization of Gating System, Modification of Cooling System Position and Flow Simulation for Cold Chamber High Pressure Die Casting Machine. Mater. Today Proc. 2021, S2214785320389598.

[43] Pinto, H.A.; Silva, F.J.G.; Martinho, R.P.; Campilho, R.D.S.G.; Pinto, A.G. Improvement and Validation of Zamak Die Casting Moulds. Procedia Manuf. 2019, 38, 1547–1557.

[44] Shahane, S.; Aluru, N.; Ferreira, P.; Kapoor, S.G.; Vanka, S.P. Optimization of Solidification in Die Casting Using Numerical Simulations and Machine Learning. J. Manuf. Process. 2020, 51, 130–141.

[45] Liu, J.; Ma, Y. A Survey of Manufacturing Oriented Topology Optimization Methods. Adv. Eng. Softw. 2016, 100, 161–175.

[46] Franke, T.; Fiebig, S.; Paul, K.; Vietor, T.; Sellschopp, J. Topology optimization with integrated casting simulation and parallel manufacturing process improvement. In Advances in Structural and Multidisciplinary Optimization; Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.-U., Maute, K., Eds.; Springer International Publishing: Basel, Switzerland, 2018; ISBN 978-3-319-67987-7.

[47] Franke, T.; Fiebig, S.; Bartz, R.; Vietor, T.; Hage, J.; vom Hofe, A. Adaptive Topology and Shape Optimization with Integrated Casting Simulation. In EngOpt 2018, Proceedings of the 6th International Conference on Engineering Optimization, Braunschweig, Germany, 5–9 June 2017; Rodrigues, H.C., Herskovits, J., Mota Soares, C.M., Araújo, A.L., Guedes, J.M., Folgado, J.O., Moleiro, F., Madeira, J.F.A., Eds.; Springer International Publishing: Basel, Switzerland, 2019; ISBN 978-3-319-97772-0.

[48] Heilmeier, F.; Goller, D.; Opritescu, D.; Thoma, C.; Rieg, F.; Volk, W. Support for Ingate Design by Analysing the Geometry of High Pressure Die Cast Geometries Using Dijkstra's Shortest Path Algorithm. Adv. Mater. Res. 2016, 1140, 400–407.

[49] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1959, 1, 269–271.

[50] Fabian Rösler and Dieter Brüggemann. "Shell-and-tube type latent heat thermal energy storage: numerical analysis and comparison with ex- periments". In: Heat and Mass Transfer 47.8 (2011), pp. 1027–1033. DOI:

[51] Ole Richter et al. "Numerical simulation of casting processes: coupled mould filling and solidification using VOF and enthalpy-porosity method". In: Heat and Mass Transfer 53.6 (2016), pp. 1957–1969.

[52] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". en. In: International Journal for Numerical Methods in Engineering 79.11 (2009).

[53] W.S. Pellini. Factors which determine riser adequacy and feeding range. AFS Transactions, 61: 61 81, 1953.

[54] E. Niyama, T. Uchida, M. Morikawa, and S. Saito. A method of shrinkage prediction and its application to steel casting practice. AFS Cast Metals Research J., 1982, 7, 52, 1982.

[55] Hattel, J., 2005. Fundamentals of Numerical Modelling of Casting Processes. I: Jesper Hattel. s.l.:Polyteknisk, pp. 198-200

[56] Borrvall T, Petersson J (2003) Topology optimization of fluids in stokes flow. Int J Numer Methods Fluids 41:77–107. https://doi.org/10. 1002/fld.426