

École polytechnique de Louvain

Density and Levelsets: A Hybrid Method for Topology Optimisation

Author: Simon Desmidt

Supervisors: Prof. Jean-François Remacle, Dr. Pierre-Alexandre Beaufort

Reader: Prof. François Glineur

Academic year 2025–2026

Master [120] in Mathematical Engineering

Acknowledgments

I would like to express my sincere gratitude to my supervisors Dr. Pierre-Alexandre Beaufort and Prof. Jean-François Remacle for their guidance, availability, and valuable advice throughout this work. Their feedback helped me overcome the difficulties encountered during the development of this thesis.

I would also like to thank Antoine Quiriny for his precious advice and support. I greatly appreciate the time he dedicated to our meetings.

Finally, I am deeply grateful to my family and friends for their encouragement, patience, and support throughout my studies. Their presence has been invaluable during this journey. I particularly thank my close friends Issambre and Alexandre who supported me during the moments of doubt.

Abstract

Topology optimisation is a design methodology for reducing structural weight while preserving mechanical performance. However, high-resolution topology optimisation problems remain computationally demanding, and the resulting designs often require post-processing before they can be manufactured, because of grey regions and mesh dependencies.

This thesis presents a novel framework combining density- and levelset-based topology optimisation with the eXtreme Mesh deformation method (X-Mesh) developed at UCLouvain. The proposed approach modifies the positions of the mesh nodes after the density-based optimisation to sharpen the material boundaries without remeshing the domain. The goal is to produce sharp and well-defined boundaries.

The density approach is analysed in details to clearly define its limitations and which problems the new framework addresses. All experiments are conducted on the Messerschmitt-Bölkow-Blohm (MBB) beam, a classical benchmark problem in topology optimisation. The framework defined in this thesis shows great potential if further researched. It is also a very powerful tool in solving the problem of manufacturability, as it allows some complex constraints and flexible objective choices. Moreover, the framework creates a connection between the two main methods used in topology optimisation today. Future work on the subject should concentrate on the convergence of the method, as no clear criterion has been identified as relevant.

Contents

List of Figures	v
1 State of the art	1
1.1 What is topology optimisation?	2
1.2 A brief history and the central paradigm	3
1.3 Density-based methods and the SIMP framework	3
1.4 Levelset and other geometric representations	4
1.5 Manufacturing constraints and additive manufacturing	5
1.6 Numerical scaling and high-performance computing	6
1.7 Topology optimisation beyond linear elasticity	6
1.8 Summary	6
1.9 Contributions	7
2 Finite element methods for linear elasticity	8
2.1 Continuum Mechanics preliminaries	8
2.2 Strong Formulation	9
2.3 Weak Formulation	9
2.4 Finite element discretisation	10
3 Density approach in topology optimisation	13
3.1 Finite-element methods in linear elasticity	13
3.2 Topology optimisation	17
3.3 Adjoint method for sensitivities	20
3.4 Implementation	23
3.5 Results	25
4 Using X-Mesh to capture the interface	32
4.1 X-Mesh	32
4.2 Connecting X-Mesh to topology optimisation	34
4.3 Results and limitations	35
5 Hybrid levelset approach	38
5.1 Levelset approach	38
5.2 Mixing the two methods	40
5.3 Numerical developments	41
5.4 Preprocessing of the velocity	45

5.5	Limitations and perspectives	49
6	Conclusion	51

List of Figures

1.1	Examples of topology-optimised components.	1
1.2	Cantilever beam design from topology optimisation, figure from [14].	3
1.3	Levelset: $\phi > 0$ where there is matter and $\phi < 0$ where there is none. The boundary is the isocurve $\phi = 0$. Figure from [30].	4
1.4	Full MBB beam (left figure) and half beam with rollers to apply the symmetry condition (right figure).	7
3.1	SIMP curves for different values of p compared to the Hashin-Shtrikman bound.	15
3.2	Parent triangle for numerical integration.	16
3.3	Error between finite differences ($\epsilon = 10^{-3}$) and analytical adjoint method on a mesh of 114 nodes.	25
3.4	Time complexity of the adjoint method and the finite differences with respect to the number of nodes in the mesh.	26
3.5	Isocurves of the solution of the topology optimisation using the density approach with a Helmholtz filter.	26
3.6	Evolution of the design throughout the optimisation process.	27
3.7	Evolution of the compliance throughout the optimisation process.	28
3.8	Time complexity of the density approach topology optimisation algorithm using a Helmholtz filter.	29
3.9	Evolution of the design for different values of filter radius.	30
3.10	Sharpness of the boundary with the refinement of the mesh for a fixed filter radius $r/h = 1/\sqrt{12}$.	31
4.1	Mesh evolution for a linear front that goes forward as time increases.	33
4.2	Oscillatory behaviour of X-Mesh and SIMP.	36
5.1	Topology optimisation of a wheel-shaped bridge using the levelset method, initialised with a single hole. The colors go from purple ($\phi < 0$, void) to orange ($\phi > 0$, matter). Figure from [34].	39
5.2	Topology optimisation of a wheel-shaped bridge using the levelset method, initialised with many holes. Figure from [34].	39
5.3	Hybrid topology optimisation workflow.	41
5.4	Comparison of the adjoint method with the finite differences, using $\epsilon = 10^{-3}$, for the sensitivity with respect to the node positions.	46
5.5	Correctness of the adjoint method for the sensitivity of the volume constraint	46

5.6	Time complexity of the adjoint method for the sensitivity with respect to the position of the nodes versus the finite differences.	47
5.7	Smoothing of the velocity field	47
5.8	Improvements generated by the levelset method.	48
5.9	Evolution of the compliance, the length of the boundary and the volume constraint during the optimisation process, using $g(\boldsymbol{\rho}) = V(\boldsymbol{\rho})/V_{tot} \leq 30\%$.	49
6.1	Examples of topology optimisation on different geometries.	52

CHAPTER 1

State of the art

In motorsport, lightweight vehicles often are the key to victory. Being able to design components that can sustain high constraints while being as light as possible is an incredible asset in motorsport, but also in many other domains. A lighter component reduces the costs as well, as less material is needed to manufacture it; and a lighter and cheaper component is every engineer's dream. To this end, topology optimisation is used in many CAD tools, e.g. SolidWorks or Autodesk Fusion [29]. It can be applied to various components, such as suspension wishbones that sustain mechanical constraints (cf. Figure 1.1a), or rotors in electric motors for which the constraints are rather thermal and magnetic (cf. Figure 1.1b). This thesis aims to mix different existing methods in topology optimisation to leverage the advantages of each of them.



(a) Bugatti Tourbillon suspension wishbone, figure from [9].



(b) Optimal rotor core geometry, figure from [10].

Figure 1.1: Examples of topology-optimised components.

1.1 What is topology optimisation?

Engineering design traditionally proceeds in three successive stages. The first one is sizing, i.e. determining the cross-sectional dimensions of given members. Afterwards comes the shape optimisation, that is adjusting the boundary of a given geometry. The last stage is topology optimisation that decides which members or holes should exist at all. Topology optimisation is the most general of the three, because it does not require a fixed initial connectivity: it can create new holes, merge existing ones, and completely rearrange the load paths within a domain.

Formally, given a design domain Ω , a set of boundary conditions and applied loads, and a budget on the total volume of material, topology optimisation seeks the material distribution $\rho : \Omega \rightarrow [0, 1]$ that maximises a chosen performance criterion Φ subject to the governing physics and the volume constraint:

$$\max_{\rho} \Phi(\mathbf{u}(\rho), \rho) \quad \text{subject to} \quad \begin{cases} \text{governing PDE in } \Omega, \\ V(\rho) \leq \alpha V_{\text{tot}}, \quad \alpha \in [0, 1]. \end{cases} \quad (1.1)$$

The field ρ simultaneously plays the role of the design variable and a material indicator: $\rho = 1$ corresponds to solid material and $\rho = 0$ to void. The governing PDE connects the state variable $\mathbf{u}$ (displacement, temperature, pressure, electric field, etc.) to the design through the constitutive properties, which are interpolated as smooth functions of ρ .

What makes topology optimisation powerful (and challenging) is precisely this generality. The space of admissible designs is vast (for instance, one degree of freedom per mesh node), the problem is non-convex, and the state variable must be recomputed at every iteration of the optimiser. Despite these difficulties, the combination of the SIMP¹ interpolation (cf. [Section 3.1](#) for more details), sensitivity filtering, and gradient-based solvers described in the following sections has made topology optimisation a routine design tool in aerospace, automotive, civil engineering and even biomedical sciences. The key insight shared by all these applications is that the adjoint method (cf. [Section 3.3](#)) reduces the computation of the full design sensitivity to the cost of one additional PDE solve, regardless of the number of design variables, which makes large-scale optimisation tractable.

¹Solid Isotropic Material with Penalisation

1.2 A brief history and the central paradigm

The key practical breakthrough in topology optimisation came with the SIMP model (Solid Isotropic Material with Penalisation), proposed independently by Bendsøe [6] and Zhou and Rozvany [38]. The idea is to assign to each point of the domain a single scalar density $\rho \in [0, 1]$ representing the local amount of material, and to penalise intermediate values through a power law, so that the optimiser naturally pushes toward fully solid or fully void regions. This made topology optimisation a practical tool. Combined with efficient gradient-based solvers (the optimality criteria (OC) method and, later, the Method of Moving Asymptotes (MMA) of Svanberg [25]) SIMP became the dominant paradigm and remains so today. An application of this method is the cantilever beam of Figure 1.2. Its accessibility was further reinforced by the publication of a compact 99-line MATLAB implementation by Sigmund [22], later revised into an even more efficient 88-line version still used as the standard teaching reference for the field; by Andreassen et al. [5]

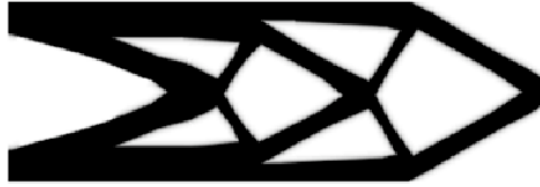


Figure 1.2: Cantilever beam design from topology optimisation, figure from [14].

Over more than three decades, this simple formulation has been extended in virtually every direction: richer physical models, more complex constraints, new ways to represent the geometry, and, more recently, the use of machine learning to accelerate the optimisation loop. The following sections review each of these directions as they stand today.

1.3 Density-based methods and the SIMP framework

The SIMP approach represents the design as a continuous density field and uses penalisation to discourage intermediate values: it interpolates a property of the material as a polynomial of the density. For example, in linear elasticity, it uses the Young modulus:

$$E(\rho) = E_{\text{void}} + (E_{\text{matter}} - E_{\text{void}})\rho^p,$$

for some natural parameter p . In practice, solving the problem directly on the raw density field leads to numerical artefacts, most notably the checkerboard instability, in which adjacent elements alternate between full and empty in a way that is artificially stiff. This is corrected by applying a *filter* that smooths the density field over a neighbourhood, making the design mesh-independent. The Helmholtz-type filter of Lazarov and Sigmund [17], used later in this work, expresses this smoothing as a partial differential equation (PDE), which is convenient

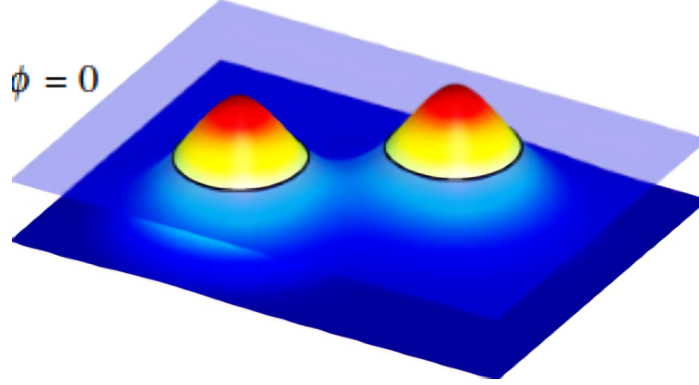


Figure 1.3: Levelset: $\phi > 0$ where there is matter and $\phi < 0$ where there is none. The boundary is the isocurve $\phi = 0$. Figure from [30].

to implement within an existing finite-element code.

On the solver side, the Method of Moving Asymptotes [25] is the de facto standard for multi-constraint problems. At each iteration, it constructs a convex approximation of the problem, solves it, and updates the design accordingly. This makes it flexible enough to handle any smooth objective and set of constraints, which is why it is used as the optimisation engine in this work.

1.4 Levelset and other geometric representations

An important family of alternative methods represents the design geometry explicitly rather than through a density field, which can be advantageous when clean, sharp boundaries are required.

Levelset methods

In the *levelset* approach, the structural boundary is defined implicitly as the zero-contour of an auxiliary scalar function $\phi : \Omega \rightarrow \mathbb{R}$; the solid ($\{\phi > 0\}$) and the void ($\{\phi < 0\}$) (as shown in Figure 1.3). During the optimisation, the boundary evolves by advecting ϕ according to a Hamilton–Jacobi equation driven by shape sensitivities. Introduced independently in this context by Allaire et al. [3] and Wang et al. [33], levelset methods naturally produce designs with smooth, crisp boundaries and no intermediate-density grey material, which makes them attractive both for visualisation and for direct interfacing with computer aided design (CAD) tools.

Their most significant practical limitation concerns topological changes: the boundary can deform continuously, but nucleating a new hole inside a solid region is not possible unless the

initial levelset function already contains it, or unless a topological derivative strategy is used to trigger the insertion of new holes during the process [23]. Various hybridisations with evolutionary approaches have been proposed to mitigate this [35]. Despite this limitation, the framework has been successfully extended to multiphysics problems [11]. Furthermore, its sensitivity analysis has been made more tractable through automatic differentiation [20], keeping it an active and competitive alternative to SIMP, especially when geometric precision matters.

Moving Morphable Components

The *Moving Morphable Components* (MMC) framework of Guo et al. [12] takes an even more geometrically explicit view: the structure is described as the union of a finite number of parametric bar-like shapes that move, rotate, and resize during the optimisation. Because the design variables are directly the positions and dimensions of these components, the output is inherently manufacturable and compatible with CAD tools. The main difficulty is that the number of components must be fixed in advance, and the non-convex parameter space can make convergence sensitive to the starting configuration.

1.5 Manufacturing constraints and additive manufacturing

A well-known limitation of classical topology optimisation is that the designs it produces, while optimal on paper, are often difficult or impossible to manufacture. Minimum member size, maximum void size, casting draw directions, and extrusion symmetry can all be enforced through appropriate modifications of the density filter or the update rule [17].

The rise of *additive manufacturing* (AM, or 3D printing) has profoundly changed this situation by making many traditional manufacturing constraints irrelevant: lattice infill, and organic geometries that would be impossible to machine can be printed layer by layer. This has made topology optimisation a natural companion to AM. That said, 3D printing introduces its own geometric constraints. Overhanging surfaces require support structures that must be removed after printing; this can be avoided by constraining the design so that no surface leans more than a critical angle from the vertical axis [36]. The mechanical properties of printed parts are also anisotropic (stronger along the print layers than across them) which must be reflected in the material model [39]. More advanced two-scale approaches simultaneously optimise the global shape of the part, and the geometry of its internal lattice microstructure, which AM can realise with high fidelity [37].

1.6 Numerical scaling and high-performance computing

For two-dimensional or coarsely meshed problems, a standard desktop computer can complete a topology optimisation run in a few minutes. However, high-resolution three-dimensional designs can involve tens of millions of unknowns, and the finite element solve at each iteration then dominates the computational budget. Direct factorisation methods become prohibitively expensive at this scale, and multigrid-preconditioned iterative solvers are necessary. Strategies that reuse information from one iteration to the next, such as reanalysis methods [16], further reduce the number of full solves required. The landmark study of Aage et al. [1] demonstrated that a wing-like structure discretised on a mesh of one billion elements could be optimised using a massively parallel solver, producing a result that closely resembles the internal architecture of bird bones. Open-source frameworks have since brought large-scale topology optimisation within reach of a much wider community.

1.7 Topology optimisation beyond linear elasticity

The framework described in the previous sections is not specific to structural mechanics. The adjoint method for computing design sensitivities applies to any problem governed by a PDE, and the density-based interpolation of material properties can be adapted to any physical medium. The field has therefore expanded well beyond its structural origins, and topology optimisation is now routinely applied to thermal, fluidic, acoustic, and electromagnetic problems. All of those applications share the same mathematical skeleton. The design variable ρ appears in the governing PDE only through the material coefficients (conductivity, viscosity, permittivity, stiffness). The adjoint method reduces the sensitivity computation to two PDE solves per objective (one forward and one adjoint) regardless of the number of design variables. The density filter and the SIMP-like penalisation transfer directly to any scalar material coefficient. This universality is precisely why the tools developed for structural compliance minimisation are the natural starting point for understanding topology optimisation in any physical domain [2].

1.8 Summary

The landscape of topology optimisation nowadays can be organised around three axes.

- **Method:** density-based SIMP remains the dominant approach thanks to its simplicity, solid mathematical foundations, and compatibility with standard FEM codes. Levelset methods are preferred when crisp, smooth boundaries matter; MMC when explicit geometric primitives are required;
- **Manufacturing:** additive manufacturing has removed many classical geometric constraints while introducing new ones (overhang, anisotropy, two-scale microstructure), making the SIMP-AM pairing one of the most active current research directions;

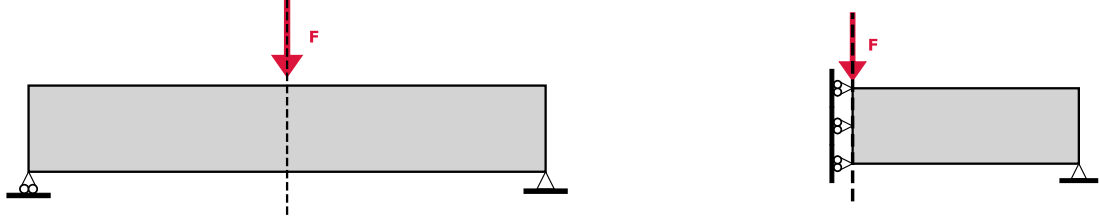


Figure 1.4: Full MBB beam (left figure) and half beam with rollers to apply the symmetry condition (right figure).

- **Computation:** classical gradient-based optimisation with adjoint sensitivities remains the gold standard. Machine-learning models offer speedups for well-defined problem families but do not yet replace gradient-based methods for general structural design.

The present thesis aims to couple the density-based SIMP framework with the levelset methods, focusing on compliance minimisation. This setting is representative of the current state of the art for a broad class of structural design problems.

1.9 Contributions

The goal of this thesis is to create a hybrid topology optimisation method; more particularly, to connect both the density and levelset approaches to leverage their advantages. The main advantage of the density approach is the spontaneous nucleation, i.e. the creation of holes in the design, while the levelset approach creates crisp boundaries where no post processing of the design boundary is required before manufacturing. To compare the different methods and analyse the results, we use a well-researched problem in topology optimisation: the Messerschmitt-Bölkow-Blohm beam (MBB, cf. [Figure 1.4](#)). This beam has a length-to-height ratio of 6 and is fixed on its lower right and left corners. The left corner is a roller so that horizontal deformation can happen. A vertical downward force is applied in the middle of the beam. Thanks to the symmetry of the setup, we will only work on the right half of the beam to reduce the computations. On that half-beam, the lower right corner is fixed and the left vertical edge can move vertically but not horizontally, hence the rollers on the whole edge. The force is applied to the upper left corner, which corresponds to the middle of the full beam.

CHAPTER 2

Finite element methods for linear elasticity

2.1 Continuum Mechanics preliminaries

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3 in general) be an open, bounded domain with Lipschitz boundary $\partial\Omega$, i.e. sufficiently smooth. The boundary is decomposed into two disjoint parts:

$$\partial\Omega = \Gamma_D \cup \Gamma_N, \quad \Gamma_D \cap \Gamma_N = \emptyset,$$

where the Dirichlet boundary conditions are applied on Γ_D and the Neumann boundary condition are applied on Γ_N .

The strain tensor

For a displacement field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$, the *linearised strain tensor* (symmetric gradient) is

$$\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (2.1)$$

or equivalently, $\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$. The linearisation hypothesis, which underpins the entire theory, states that the displacements and their gradients are small compared to the characteristic length of Ω .

Constitutive law: Hooke's theorem

A linear elastic material is characterised by a fourth-order elasticity tensor $\mathbf{C} = (C_{ijkl})$ satisfying the symmetry and coercivity conditions:

$$C_{ijkl} = C_{jikl} = C_{klij}, \quad C_{ijkl} \xi_{ij} \xi_{kl} \geq \alpha_0 |\xi|^2 \quad \forall \xi \in \mathbb{R}_{\text{sym}}^{d \times d}, \alpha_0 > 0. \quad (2.2)$$

The Cauchy stress tensor is then given by

$$\sigma_{ij}(\mathbf{u}) = \sum_{k,l} C_{ijkl} \varepsilon_{kl}(\mathbf{u}) \quad (2.3)$$

For an isotropic material, the components of the elasticity tensor are

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}).$$

where $\lambda, \mu > 0$ are the Lamé parameters, related to Young's modulus E and Poisson's ratio ν , of the considered material:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}.$$

2.2 Strong Formulation

Let $\mathbf{f}$ be a body-force density and $\mathbf{t}$ a prescribed traction. The strong form of the linear elasticity problem reads is to find $\mathbf{u} : \bar{\Omega} \rightarrow \mathbb{R}^d$ such that

$$-\operatorname{div} \sigma(\mathbf{u}) = \mathbf{f} \quad \text{in } \Omega, \quad (2.4a)$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_D, \quad (2.4b)$$

$$\sigma(\mathbf{u}) \mathbf{n} = \mathbf{t} \quad \text{on } \Gamma_N. \quad (2.4c)$$

Equation 2.4a expresses conservation of momentum and the other two express the boundary conditions.

2.3 Weak Formulation

Functional-analytic framework

The natural function space for the displacement is defined as

$$V = \{ \mathbf{v} \in H^1(\Omega) \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D \}, \quad (2.5)$$

endowed with the H^1 semi-norm $|\cdot|_{H^1}$, and H^1 the first-order Sobolev space: $H^1(\Omega) = \{ \mathbf{v} : \Omega \rightarrow \mathbb{R}^d : \int_{\Omega} (\mathbf{v}^2 + \|\nabla \mathbf{v}\|^2) d\omega < \infty \}$.

Multiplying equation (2.4a) by a *test function* $\mathbf{v} \in V$, integrating over Ω , and applying the divergence theorem yields the following:

$$\begin{aligned} - \int_{\Omega} (\partial_j \sigma_{ij}) v_i dx &= \int_{\Omega} \sigma_{ij} \partial_j v_i dx - \int_{\partial\Omega} \sigma_{ij} n_j v_i ds \\ &= \int_{\Omega} \sigma(\mathbf{u}) : \epsilon(\mathbf{v}) dx - \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} ds, \end{aligned} \quad (2.6)$$

where the symmetry of σ and the fact that $\mathbf{v} = \mathbf{0}$ on Γ_D were used. This yields the bilinear and linear forms

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \sigma(\mathbf{u}) : \epsilon(\mathbf{v}) dx = \int_{\Omega} C_{ijkl} \varepsilon_{kl}(\mathbf{u}) \varepsilon_{ij}(\mathbf{v}) dx, \quad (2.7)$$

$$\ell(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx + \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} \, ds. \quad (2.8)$$

A function $\mathbf{u} \in V$ is a *weak (variational) solution* of problem (2.4) if

$$a(\mathbf{u}, \mathbf{v}) = \ell(\mathbf{v}), \quad \forall \mathbf{v} \in V. \quad (2.9)$$

Well-posedness: Lax–Milgram theorem

Well-posedness of (2.9) rests on two conditions: continuity and coercivity.

Using the ellipticity condition (2.2), we verify that $a(\cdot, \cdot)$ is:

- Continuous: $|a(\mathbf{u}, \mathbf{v})| \leq M \|\mathbf{u}\|_V \|\mathbf{v}\|_V$ for some $M > 0$;
- Coercive: $a(\mathbf{v}, \mathbf{v}) \geq \alpha \|\mathbf{v}\|_V^2$ for some $\alpha > 0$.

It remains to check that the right-hand side defines a continuous linear functional on V . Linearity of ℓ follows directly from the linearity of the integral. To prove continuity, we estimate each term separately. By the Cauchy–Schwarz inequality,

$$\left| \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx \right| \leq \|\mathbf{f}\|_{L^2(\Omega)} \|\mathbf{v}\|_{L^2(\Omega)}. \quad (2.10)$$

Since Γ_D has positive measure, Poincaré’s inequality gives

$$\|\mathbf{v}\|_{L^2(\Omega)} \leq C_P \|\mathbf{v}\|_V. \quad (2.11)$$

Similarly, for the boundary term, the trace theorem implies that there exists a constant $C_T > 0$ such that

$$\|\mathbf{v}\|_{L^2(\Gamma_N)} \leq C_T \|\mathbf{v}\|_V. \quad (2.12)$$

Therefore,

$$\left| \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} \, ds \right| \leq \|\mathbf{t}\|_{L^2(\Gamma_N)} \|\mathbf{v}\|_{L^2(\Gamma_N)} \leq C_T \|\mathbf{t}\|_{L^2(\Gamma_N)} \|\mathbf{v}\|_V. \quad (2.13)$$

Combining the two estimates, we obtain

$$|\ell(\mathbf{v})| \leq \left(C_P \|\mathbf{f}\|_{L^2(\Omega)} + C_T \|\mathbf{t}\|_{L^2(\Gamma_N)} \right) \|\mathbf{v}\|_V. \quad (2.14)$$

Hence ℓ is continuous on V .

2.4 Finite element discretisation

Galerkin approximation

Let $\mathcal{T}_h$ be a triangulation of Ω , with mesh size h . For $p \geq 1$, we define the finite element space

$$V_h = \{\mathbf{v}_h \in V : \mathbf{v}_h|_K \in \mathcal{P}_p(K) \text{ for all elements } K \in \mathcal{T}_h\}. \quad (2.15)$$

The Galerkin approximation of (2.9) is then: find $\mathbf{u}_h \in V_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = \ell(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h. \quad (2.16)$$

Since V_h is a subspace of V , the continuity and coercivity of $a(\cdot, \cdot)$ remain valid on V_h . Hence the discrete problem admits a unique solution by the Lax–Milgram theorem.

Basis functions and stiffness matrix

Let $\{\varphi_i\}_{i=1}^N$ be a nodal basis of V_h . Writing $\mathbf{u}_h = \sum_{j=1}^N U_j \varphi_j$ and testing against each φ_i yields the linear system

$$K\mathbf{u} = \mathbf{f}, \quad (2.17)$$

where the global stiffness matrix K and load vector $\mathbf{f}$ are assembled from element contributions:

$$K_{ij} = a(\varphi_j, \varphi_i) \quad f_i = \ell(\varphi_i). \quad (2.18)$$

The symmetry and positive-definiteness of $\mathbf{K}$ follow directly from those of $a(\cdot, \cdot)$.

A priori error analysis

The Galerkin orthogonality relation

$$a(\mathbf{u} - \mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in V_h, \quad (2.19)$$

combined with coercivity and the best-approximation property of V_h , gives *Céa's lemma*:

$$\|\mathbf{u} - \mathbf{u}_h\|_V \leq \frac{M}{\alpha} \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_V. \quad (2.20)$$

Standard interpolation theory then yields, for $\mathbf{u} \in H^{p+1}(\Omega)$,

$$\|\mathbf{u} - \mathbf{u}_h\|_{H^1(\Omega)} \leq C h^p |\mathbf{u}|_{H^{p+1}(\Omega)}, \quad (2.21)$$

where $C > 0$ is independent of h and $\mathbf{u}$. This equation shows that the finite element error converges to zero at rate h^p in the energy norm, meaning that refining the mesh significantly improves the quality of the solution.

Numerical integration

In practice, the element integrals in Equation 2.17 are evaluated by a Gaussian quadrature rule of sufficiently high order. On each element E we introduce a reference element $\hat{E}$ via an affine map $F_E : \hat{E} \rightarrow E$, so that

$$\int_E f \, dx = \int_{\hat{E}} f \circ F_E |\det J_E| \, d\hat{x} \approx \sum_{i=1}^n w_i (f \circ F_E)(\hat{x}_i) |\det J_E(\hat{x}_i)|, \quad (2.22)$$

where $\{(\hat{x}_i, w_i)\}$ are the quadrature points and weights and J_E is the Jacobian of F_E . Exact integration of the stiffness integrand (a polynomial of degree $2(p - 1)$) is achieved with $n \geq \lceil p \rceil$ Gauss points per direction.

CHAPTER 3

Density approach in topology optimisation

The density approach in topology optimisation consists in defining a density field $\rho : \Omega \rightarrow [0, 1]$, where Ω is the domain of the optimisation problem. It represents the absence or presence of matter at a given point of the domain. The goal is to have values that are (almost) binary so that the boundary of the final design of the object is well-defined and manufacturable.

3.1 Finite-element methods in linear elasticity

From the strong formulation to the discrete formulation

Let $\Omega \subset \mathbb{R}^2$ be the design domain with Lipschitz boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, where Γ_D carries Dirichlet (displacement) boundary conditions and Γ_N carries Neumann (traction) boundary conditions. As derived previously in [Section 2.2](#), the strong form of the planar linear elasticity problem is to find the displacement field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ such that

$$\begin{cases} -\operatorname{div}(C(\rho)\varepsilon(\mathbf{u})) = \mathbf{f}(\rho) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma_D, \\ (C(\rho)\varepsilon(\mathbf{u}))\mathbf{n} = \mathbf{t} & \text{on } \Gamma_N, \end{cases} \quad (3.1)$$

where $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ is the linearised strain tensor, $C(\rho)$ is the (density-dependent) elasticity matrix, $\mathbf{f}(\rho)$ is the body-force vector and $\mathbf{t}$ is the imposed traction.

Multiplying the equilibrium equation by a test function $\mathbf{v} \in V = \{\mathbf{v} \in H^1(\Omega) \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$ and integrating over Ω , then applying the divergence theorem gives the *weak formulation*: find $\mathbf{u} \in V$ such that

$$\int_{\Omega} \varepsilon(\mathbf{v}) : C(\rho)\varepsilon(\mathbf{u}) dS = \int_{\Omega} \mathbf{v} \cdot \mathbf{f}(\rho) dS + \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{t} dc \quad \forall \mathbf{v} \in V. \quad (3.2)$$

The boundary term on Γ_D vanishes because $\mathbf{v} = \mathbf{0}$, and the boundary term on Γ_N gives the traction integral.

To discretise this, we introduce a conforming triangulation $\mathcal{T}_h = \{e\}$ of Ω with mesh size h and consider the finite-dimensional subspace $V_h \subset V$ of piecewise-linear functions on $\mathcal{T}_h$.

The *Galerkin problem* then reads: find $\mathbf{u}_h \in V_h$ such that Equation 3.2 holds for all $\mathbf{v}_h \in V_h$. Substituting the nodal expansion $\mathbf{u}_h = \sum_j \mathbf{u}_j N_j(x, y)$ and choosing $\mathbf{v}_h = N_i(x, y) \mathbf{e}_\alpha$ for each node i and direction $\alpha \in \{x, y\}$, we recover the global linear system

$$K(\boldsymbol{\rho}) \mathbf{u} = \mathbf{f}(\boldsymbol{\rho}), \quad (3.3)$$

where the global stiffness matrix $K \in \mathbb{R}^{2n \times 2n}$ and load vector $\mathbf{f} \in \mathbb{R}^{2n}$ are assembled from their element-wise contributions $K^{(e)}$ and $\mathbf{f}^{(e)}$.

Both the stiffness matrix and the load vector are the assembling of local matrices and vectors respectively. Those local quantities are denoted $K^{(e)}$ and $\mathbf{f}^{(e)}$. Their expressions are:

$$\begin{cases} K^{(e)}(\boldsymbol{\rho}) = \int_{\Omega^{(e)}} B^{(e)T} C(\boldsymbol{\rho}) B^{(e)} d\Omega \\ \mathbf{f}^{(e)}(\boldsymbol{\rho}) = \int_{\Omega^{(e)}} N^{(e)T}(x, y) \begin{pmatrix} f_x(\boldsymbol{\rho}) \\ f_y(\boldsymbol{\rho}) \end{pmatrix} d\Omega + \int_{\partial\Omega^{(e)}} N^{(e)T}(x, y) \begin{pmatrix} t_x \\ t_y \end{pmatrix} dc \end{cases} \quad (3.4)$$

where $N^{(e)}$ contains the shape functions and $B^{(e)}$ contains its derivatives with respect to the position of the nodes. $C(\boldsymbol{\rho})$ is the matrix of physical properties, (f_x, f_y) the volume forces, and (t_x, t_y) the stress applied to the object. The expressions of the matrices are the following:

$$N^{(e)} = \begin{pmatrix} N_1(x, y) & 0 & N_2(x, y) & 0 & N_3(x, y) & 0 \\ 0 & N_1(x, y) & 0 & N_2(x, y) & 0 & N_3(x, y) \end{pmatrix} \in \mathbb{R}^{2 \times 6}, \quad (3.5)$$

$$B^{(e)} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{pmatrix} \in \mathbb{R}^{3 \times 6}, \quad (3.6)$$

and with $C(\boldsymbol{\rho})$ the material property matrix in planar stress:

$$C(\boldsymbol{\rho}) = E(\boldsymbol{\rho}) \frac{1}{1 - \nu^2} \underbrace{\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix}}_{=: A(\nu)}, \quad (3.7)$$

for $E(\boldsymbol{\rho})$ the Young modulus defined by the SIMP, and ν the Poisson coefficient.

The only volume force that applies here is gravity: $(f_x(\boldsymbol{\rho}), f_y(\boldsymbol{\rho})) = (0, -\rho_{\text{matter}} g \rho(x, y))$. Moreover, the term (t_x, t_y) is given by the Neumann boundary conditions and does not depend on the density variables.

For simplicity, we consider linear shape functions, meaning that the matrix $B^{(e)}$ is constant with the density field. Moreover, the density field is a linear interpolation of the nodal values:

$$\rho(x, y) = \sum_{j=1}^3 \rho_j N_j(x, y). \quad (3.8)$$

The density variable can take any value between 0 and 1. However, we want to be as close as possible to those bounds, to decide clearly where there is matter and where there is not. With that objective in mind, the SIMP (Solid Isotropic Material with Penalisation) model is used for the expression of the Young modulus:

$$E(\rho) = E_{\text{void}} + (E_{\text{matter}} - E_{\text{void}})\rho^p, \quad (3.9)$$

where the exponent p must be greater than or equal to 1. When this method started being used in topology optimisation, the SIMP was purely mathematical and did not correspond to any known physical model for the Young modulus. However, there now exists a theoretical upper bound for the effective elastic properties of heterogeneous composite materials, which is the Hashin-Shtrikman bound [13](cf. Figure 3.1). The SIMP for exponents $p = 1$ and $p = 2$ are above this upper bound, meaning that they are not physically accurate, while the curves corresponding to higher values of p stay below the bound. Therefore, $p = 3$ is generally used, and this has also proven empirically to be the best value [28].

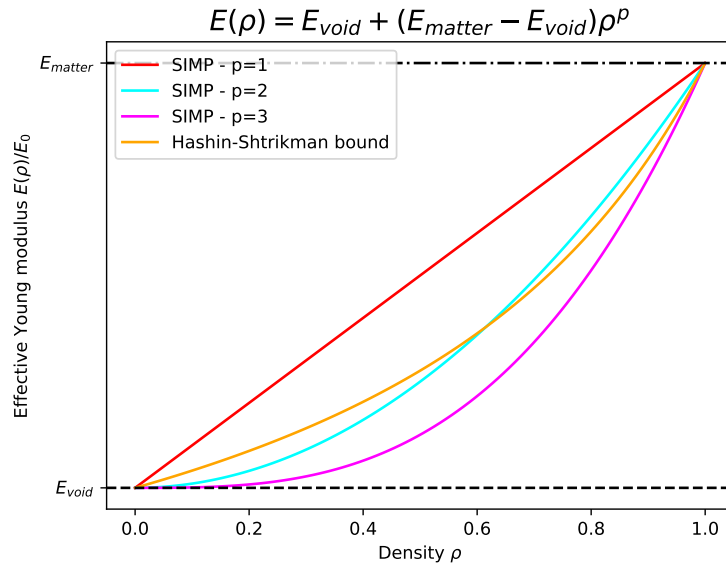


Figure 3.1: SIMP curves for different values of p compared to the Hashin-Shtrikman bound.

For the integration, we will always consider a parent triangle (cf. Figure 3.2) and a linear

change of variables to get to any triangle. On the parent triangle $\hat{\Omega}$, the shape functions are

$$\hat{N}_1(\xi, \eta) = 1 - \xi - \eta, \quad \hat{N}_2(\xi, \eta) = \xi, \quad \hat{N}_3(\xi, \eta) = \eta \quad (3.10)$$

On such triangles, the Jacobian determinant is the ratio of the areas of both triangles:

$$J_{\Omega^{(e)}} = \frac{\text{Area}(\Omega^{(e)})}{\text{Area}(\hat{\Omega})} = \frac{\text{Area}(\Omega^{(e)})}{1/2} = 2\text{Area}(\Omega^{(e)}) \quad (3.11)$$

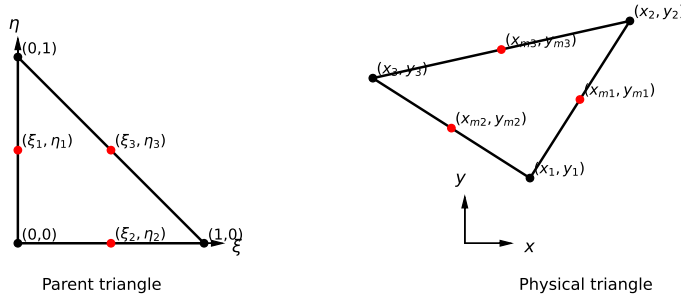


Figure 3.2: Parent triangle for numerical integration.

We can now compute the final expressions for $K^{(e)}$ and $\mathbf{f}^{(e)}$.

$$K^{(e)}(\boldsymbol{\rho}) = \int_{\Omega^{(e)}} B^{(e)T} C(\boldsymbol{\rho}) B^{(e)} d\Omega = B^{(e)T} A(\nu) B^{(e)} \int_{\Omega^{(e)}} E(\boldsymbol{\rho}) d\Omega \quad (3.12)$$

To compute the integral of the Young modulus, we can use a barycentric formulation: the function is defined by its values at the nodes of the triangle.

$$\int_{\Omega^{(e)}} E(\boldsymbol{\rho}) d\Omega = 2\text{Area}(\Omega^{(e)}) \left(E_{\text{void}} + (E_{\text{matter}} - E_{\text{void}}) \int_{\hat{\Omega}} \rho^p d\Omega \right) \quad (3.13)$$

$$\int_{\hat{\Omega}} \rho^p d\Omega = \frac{1}{(p+1)(p+2)} \sum_{i+j+\ell=p} \rho_1^i \rho_2^j \rho_3^\ell \quad (3.14)$$

$$\Rightarrow K^{(e)}(\boldsymbol{\rho}) = B^{(e)T} A(\nu) B^{(e)} \times 2\text{Area}(\Omega^{(e)}) \left(E_{\text{void}} + \frac{(E_{\text{matter}} - E_{\text{void}})}{(p+1)(p+2)} \sum_{i+j+\ell=p} \rho_1^i \rho_2^j \rho_3^\ell \right) \quad (3.15)$$

The force vector yields

$$\mathbf{f}^{(e)} = \int_{\Omega^{(e)}} N^{(e)T}(x, y) \begin{pmatrix} 0 \\ -\rho_{\text{matter}} g \rho(x, y) \end{pmatrix} d\Omega = -\rho_{\text{matter}} g \int_{\Omega^{(e)}} \begin{pmatrix} 0 \\ N_1(x, y) \rho(x, y) \\ 0 \\ N_2(x, y) \rho(x, y) \\ 0 \\ N_3(x, y) \rho(x, y) \end{pmatrix} d\Omega \quad (3.16)$$

The x -components are all 0, so only $\mathbf{f}_y^{(e)}$ is considered for the sake of clarity. By the linearity assumption of Equation 3.8, this becomes

$$\mathbf{f}_y^{(e)} = -\rho_{\text{matter}} g \int_{\Omega^{(e)}} \begin{pmatrix} \sum_{j=1}^3 \rho_j N_j(x, y) N_1(x, y) \\ \sum_{j=1}^3 \rho_j N_j(x, y) N_2(x, y) \\ \sum_{j=1}^3 \rho_j N_j(x, y) N_3(x, y) \end{pmatrix} d\Omega \quad (3.17)$$

The local mass matrix is defined as follows

$$M_{ij}^{(e)} = \int_{\Omega^{(e)}} N_i(x, y) N_j(x, y) d\Omega, \quad (3.18)$$

and the force vector simplifies to

$$\boxed{\mathbf{f}_y^{(e)} = -\rho_{\text{matter}} g M^{(e)} \boldsymbol{\rho}^{(e)}} \quad (3.19)$$

3.2 Topology optimisation

Topology optimisation consists in solving the following problem:

$$\begin{aligned} & \min_{\boldsymbol{\rho}} \Phi(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) \\ & \text{subject to } \mathbf{r}(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) = \mathbf{K}(\boldsymbol{\rho}) \mathbf{u} - \mathbf{f}(\boldsymbol{\rho}) = 0 \\ & \quad g_i(\boldsymbol{\rho}) = 0 \quad i = 1, 2, \dots \end{aligned} \quad (3.20)$$

where $\Phi(\cdot)$ is a quantity to minimise, $\mathbf{r}(\cdot)$ is the finite-element residual of the linear elasticity problem, and $g_i(\cdot)$ are design constraints. We take as objective the *compliance* of the design, i.e. the reciprocal of the stiffness:

$$\Phi(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) = c(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) = \mathbf{u}^T \mathbf{f}(\boldsymbol{\rho}) = \mathbf{u}^T \mathbf{K}(\boldsymbol{\rho}) \mathbf{u} \quad (3.21)$$

and we add a constraint on the proportion of matter used in the design:

$$g(\boldsymbol{\rho}) = V(\boldsymbol{\rho}) - \alpha V_{\text{tot}} \leq 0 \quad (3.22)$$

where $V(\cdot)$ is the volume for a given density field and $\alpha \in [0, 1]$ is the maximum allowed volume fraction.

Helmholtz filter

A standard difficulty in density-based topology optimisation is the appearance of mesh-dependent patterns and numerical instabilities such as checkerboards. To overcome these issues, the design field is regularised through a *Helmholtz filter*. Instead of directly optimising the physical density field ρ , we introduce an auxiliary control variable k^1 . The filtered density ρ is then defined as the solution of the Helmholtz-type partial differential equation

$$\begin{cases} -r^2 \Delta \rho + \rho = k & \text{in } \Omega, \\ \nabla \rho \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (3.23)$$

It can be seen as a diffusion equation solved by an implicit time step:

$$\frac{\partial \rho}{\partial t} = \beta \Delta \rho \implies \frac{\rho^{\Delta t} - \rho^0}{\Delta t} = \beta \Delta \rho^{\Delta t} \quad (3.24)$$

which yields

$$-\Delta t \beta \Delta \rho^{\Delta t} + \rho^{\Delta t} = \rho^0 \quad (3.25)$$

In this formulation, the control variable is the initial field $k := \rho^0$ while the physical density is the diffused field $\rho := \rho^{\Delta t}$. This filter smooths the raw density field. By setting $r^2 = \beta \Delta t$, this equation becomes

$$-r^2 \Delta \rho + \rho = k, \quad (3.26)$$

which is the Helmholtz filtering equation used in this work.

The parameter $r \geq 0$ defines the characteristic filter radius and controls the amount of smoothing applied to the density field. In particular, the diffusion term $-r^2 \Delta \rho$ penalises rapid spatial oscillations of the density, thereby introducing a minimum length scale in the final design. When $r \rightarrow 0$, the diffusion contribution vanishes and we recover $\rho = k$, meaning that no filtering is applied. Conversely, increasing r produces smoother density distributions and suppresses fine-scale features.

The homogeneous Neumann boundary condition ensures that no artificial flux of density crosses the boundary $\partial\Omega$. This choice preserves the total amount of material during the filtering process and avoids boundary artefacts that could otherwise bias the optimisation.

¹This field has no physical interpretation and is introduced solely for regularisation purposes.

Multiplying the strong form in Equation 3.23 by a test function w , and integrating over the domain yields the weak formulation: find ρ such that

$$\int_{\Omega} r^2 \nabla \rho \cdot \nabla w d\Omega + \int_{\Omega} \rho w d\Omega = \int_{\Omega} k w d\Omega \quad \forall w. \quad (3.27)$$

After integration by parts, the boundary integral vanishes because of the homogeneous Neumann condition. The resulting weak form can be interpreted as the variational formulation of a reaction-diffusion problem, where the diffusion operator smooths the density field while the reaction term maintains proximity between ρ and the control field k .

Using a standard finite-element approximation, both the filtered density and the control field are interpolated with the same shape functions:

$$\rho(x, y) = \sum_j N_j(x, y) \rho_j, \quad k(x, y) = \sum_j N_j(x, y) k_j. \quad (3.28)$$

Choosing the test function according to the Galerkin procedure, $w_i = N_i$, leads to the discrete linear system

$$(r^2 K_f + M) \boldsymbol{\rho} = M \mathbf{k}, \quad (3.29)$$

where K_f denotes the diffusion stiffness matrix associated with the Laplacian operator and M is the mass matrix. The filter therefore acts as a linear smoothing operator mapping the optimisation variable $\mathbf{k}$ onto the physical density field $\boldsymbol{\rho}$ used in the mechanical analysis.

With this formulation, the optimisation problem is no longer posed directly on the physical density field but on the control variable $\mathbf{k}$. This guarantees that the resulting density distribution possesses a controlled spatial regularity which significantly reduces the dependence of the final topology on the underlying mesh.

Diffusion matrix

The entries of the diffusion matrix are defined as follows:

$$K_{f,ij} = \int_{\Omega} \nabla N_i \cdot \nabla N_j d\Omega = \int_{\Omega} \frac{\partial N_i}{\partial \mathbf{x}} \cdot \frac{\partial N_j}{\partial \mathbf{x}} d\Omega = 2\text{Area}(\Omega) \frac{\partial \boldsymbol{\xi}^T}{\partial \mathbf{x}} \int_{\hat{\Omega}} \frac{\partial \hat{N}_i}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \hat{N}_j}{\partial \boldsymbol{\xi}} d\hat{\Omega} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} \quad (3.30)$$

On the parent element, the linear basis functions are

$$\hat{N}_1(\xi, \eta) = 1 - \xi - \eta, \quad \hat{N}_2(\xi, \eta) = \xi, \quad \hat{N}_3(\xi, \eta) = \eta, \quad (3.31)$$

and their gradients are thus constant:

$$\nabla \hat{N}_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \nabla \hat{N}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \nabla \hat{N}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.32)$$

and the local matrix is

$$K_f^{(e)} = \frac{\mathbf{b}\mathbf{b}^T + \mathbf{c}\mathbf{c}^T}{4\text{Area}(\Omega^{(e)})}, \quad \mathbf{b} = \begin{pmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} x_3 - x_2 \\ x_1 - x_3 \\ x_2 - x_1 \end{pmatrix} \quad (3.33)$$

Additionally, by Equations 3.18 and 3.31, the local mass matrix is

$$M^{(e)} = \frac{\text{Area}(\Omega^{(e)})}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad (3.34)$$

The global matrices K_f and M are obtained by standard finite-element assembly of their local counterparts.

Full optimisation problem

The final formulation of our problem is then

$\begin{aligned} & \min_{\mathbf{k}} \quad c(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) \\ & \text{subject to} \quad \mathbf{r}(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) = \mathbf{0}, \\ & \quad \quad \mathbf{h}(\boldsymbol{\rho}, \mathbf{k}) = \mathbf{0}, \\ & \quad \quad g(\boldsymbol{\rho}) \leq 0. \end{aligned}$	$\begin{aligned} & \min_{\mathbf{k}} \quad \mathbf{u}^T \mathbf{f} \\ & \text{subject to} \quad K(\boldsymbol{\rho})\mathbf{u} - \mathbf{f}(\boldsymbol{\rho}) = \mathbf{0}, \\ & \quad \quad (r^2 K_f + M)\boldsymbol{\rho} - M\mathbf{k} = \mathbf{0}, \\ & \quad \quad V(\boldsymbol{\rho}) - \alpha V_{\text{tot}} \leq 0. \end{aligned}$
--	---

3.3 Adjoint method for sensitivities

To perform gradient-based optimisation, expressions for the gradients of the objective function and constraints with respect to the control variable $\mathbf{k}$ need to be obtained. These involve implicit dependencies (through $\mathbf{u}$ and $\boldsymbol{\rho}$) on the control field, which are handled by means of the *adjoint method*. This method is used when the objective function depends both explicitly and implicitly on the variable on which we optimise. It leverages the equality constraints of the problem to cancel the implicit derivatives and speed up significantly the computations of the sensitivities: the adjoint method requires only one linear solve, compared to the finite differences that require as many linear solves as there are degrees of freedom.

Sensitivity of the compliance

Let us define the Lagrangian of the compliance, with Lagrangian parameter $\lambda \in \mathbb{R}^{2n}$:

$$\mathcal{L}_c(\mathbf{k}) = c(\mathbf{u}(\rho), \rho) + \lambda^T \mathbf{r}(\mathbf{u}(\rho), \rho) \quad (3.36)$$

A feasible point is a point that verifies the constraints, i.e. the linear elasticity equation and the filter. Because $\mathbf{r}(\mathbf{u}(\rho), \rho) = 0$ for any feasible point, $\mathcal{L}_c = c$ and we may compute the gradient of c through that of $\mathcal{L}_c$ and determine the value of λ that suits us best. Applying the chain rule:

$$\begin{aligned} \frac{d\mathcal{L}_c(\mathbf{k})}{d\mathbf{k}} &= \left[\frac{\partial c}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \rho} + \frac{\partial c}{\partial \rho} + \lambda^T \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \rho} + \frac{\partial \mathbf{r}}{\partial \rho} \right) \right] \frac{\partial \rho}{\partial \mathbf{k}} \\ &= \left[\frac{\partial c}{\partial \rho} + \lambda^T \frac{\partial \mathbf{r}}{\partial \rho} + \left(\frac{\partial c}{\partial \mathbf{u}} + \lambda^T \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right) \frac{\partial \mathbf{u}}{\partial \rho} \right] \frac{\partial \rho}{\partial \mathbf{k}} \end{aligned} \quad (3.37)$$

The term $\frac{\partial \mathbf{u}}{\partial \rho}$ highlighted in green is difficult to compute, as there is no direct explicit expression connecting the displacement field to the density field. We therefore choose λ such that these terms cancel:

$$\left[\left(\frac{\partial c}{\partial \mathbf{u}} \right)^T + \lambda^T \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right] \frac{\partial \mathbf{u}}{\partial \rho} = 0 \quad \forall \frac{\partial \mathbf{u}}{\partial \rho} \iff \left(\frac{\partial c}{\partial \mathbf{u}} \right)^T + \lambda^T \frac{\partial \mathbf{r}}{\partial \mathbf{u}} = 0 \quad (3.38)$$

Replacing the derivatives by their explicit forms, we obtain the *adjoint equation*:

$$K^T(\rho) \lambda = -\mathbf{f}(\rho) \quad (3.39)$$

Remark 3.3.1. *The transposes are kept so that the formula stays general, even though the stiffness matrix is often symmetric depending on how the Dirichlet conditions are handled.*

The gradient of the compliance then reduces to

$$\frac{dc}{d\mathbf{k}} = \frac{d\mathcal{L}_c(\mathbf{k})}{d\mathbf{k}} = \left[\mathbf{u}^T \frac{\partial \mathbf{f}}{\partial \rho} + \lambda^T \left(\frac{\partial K}{\partial \rho} \mathbf{u} - \frac{\partial \mathbf{f}}{\partial \rho} \right) \right] \frac{\partial \rho}{\partial \mathbf{k}} \quad (3.40)$$

Derivations

Three tensors remain to be computed: $\frac{\partial \mathbf{f}}{\partial \rho}$, $\frac{\partial K}{\partial \rho}$, and $\frac{\partial \rho}{\partial \mathbf{k}}$.

- $\frac{\partial \mathbf{f}}{\partial \rho}$: since $\mathbf{f}(\rho)$ is assembled from the local vectors, its derivative is the assembly of the local derivatives. From [Equation 3.19](#):

$$\frac{\partial \mathbf{f}^{(e)}}{\partial \rho^{(e)}} = \frac{\partial}{\partial \rho^{(e)}} \left(-\rho_{\text{matter}} g M^{(e)} \rho^{(e)} \right) = -\rho_{\text{matter}} g M^{(e)} \quad (3.41)$$

- $\frac{\partial K}{\partial \rho}$: by the same argument, we compute the local derivatives. From [Equation 3.15](#):

$$\begin{aligned}
\frac{\partial K^{(e)}}{\partial \boldsymbol{\rho}^{(e)}} &= B^{(e)T} A(\nu) B^{(e)} \times 2 \text{Area}(\Omega^{(e)}) \frac{E_{\text{matter}} - E_{\text{void}}}{(p+1)(p+2)} \left[\frac{\partial}{\partial \boldsymbol{\rho}^{(e)}} \sum_{i+j+\ell=p} \rho_1^i \rho_2^j \rho_3^\ell \right] \\
&= B^{(e)T} A(\nu) B^{(e)} \times 2 \text{Area}(\Omega^{(e)}) \frac{E_{\text{matter}} - E_{\text{void}}}{(p+1)(p+2)} \begin{pmatrix} \sum_{i+j+\ell=p} i \rho_1^{i-1} \rho_2^j \rho_3^\ell \\ \sum_{i+j+\ell=p} j \rho_1^i \rho_2^{j-1} \rho_3^\ell \\ \sum_{i+j+\ell=p} \ell \rho_1^i \rho_2^j \rho_3^{\ell-1} \end{pmatrix} \quad (3.42)
\end{aligned}$$

- $\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{k}}$: from the discrete filter [Equation 3.29](#), $\boldsymbol{\rho}$ is an affine function of $\mathbf{k}$, hence:

$$\boldsymbol{\rho}(\mathbf{k}) = \left(r^2 K_f + M \right)^{-1} M \mathbf{k} \implies \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{k}} = \left(r^2 K_f + M \right)^{-1} M \quad (3.43)$$

Gradient of the volume constraint

As there is no implicit dependence in the density field in the volume constraint, its gradient is straightforward to compute and does not require to use the adjoint method.

$$g(\boldsymbol{\rho}(\mathbf{k})) = V(\boldsymbol{\rho}(\mathbf{k})) - \alpha V_{\text{tot}} \leq 0 \quad (3.44)$$

By definition, the volume is

$$V(\boldsymbol{\rho}) = \sum_e \int_{\Omega^{(e)}} \rho(x, y) dS \quad (3.45)$$

Using the linear approximation shown in [Equation 3.9](#) and the barycentric quadrature formula, we have for each element

$$\int_{\Omega^{(e)}} \rho(x, y) dS = 2 \text{Area}(\Omega^{(e)}) \int_{\hat{\Omega}} \sum_{j=1}^3 \rho_j N_j(\xi, \eta) d\omega = \mathbb{1}^T M^{(e)} \boldsymbol{\rho}^{(e)} \quad (3.46)$$

where $\mathbb{1} = (1, 1, 1)^T$.

Therefore, by the chain rule,

$$\frac{dg}{d\mathbf{k}} = \frac{\partial g}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{k}} \quad (3.47)$$

and

$$\left[\frac{\partial g}{\partial \boldsymbol{\rho}} \right]_i = \sum_{e \ni i} \left[\mathbb{1}^T M^{(e)} \right]_i \implies \frac{\partial g}{\partial \boldsymbol{\rho}} = \mathbb{1}_n^T M \quad (3.48)$$

Moreover, from the Helmholtz filter,

$$\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{k}} = \left(r^2 K_f + M \right)^{-1} M \quad (3.49)$$

This yields the final expression of the gradient of the volume constraint:

$$\frac{dg}{d\mathbf{k}} = \mathbb{1}_n^T M \left(r^2 K_f + M \right)^{-1} M \quad (3.50)$$

3.4 Implementation

The algorithm (from [22]) used for the implementation of the topology optimisation is the following:

Algorithm 3.1: Topology optimisation

Input: Mesh, boundary conditions, α

```

1  Define  $\rho_0 = \alpha \mathbb{1}_n$ , and  $i = 0$ ;
2  while  $\|\rho_i - \rho_{i-1}\| \geq \epsilon$  do
3       $\rho_i = \text{Helmholtz}(\mathbf{k}_i)$ ;
4      Solve  $K(\rho_i)\mathbf{u}_i = f(\rho_i)$ ;
5      Compute compliance  $c_i = \mathbf{u}_i^T K(\rho_i)\mathbf{u}_i$ ;
6      Compute gradient of the compliance and volume constraint;
7       $l_1 = 0; l_2 = 10^9, m = 0.2$ 
8      while  $(l_2 - l_1)/(l_2 + l_1) > \epsilon$  do
9           $l_{mid} = (l_1 + l_2)/2$ ;
10          $\mathbf{k}_{i+1} =$ 
             $\max \left\{ 0; \max \left[ \mathbf{k}_i - m \min \left( 1 \min \left\{ \mathbf{k}_i + m; \mathbf{k}_i \cdot \sqrt{\frac{\partial c}{\partial \mathbf{k}}(\mathbf{k}_i) / l_{mid} \frac{\partial g}{\partial \mathbf{k}}(\mathbf{k}_i)} \right\} \right) \right] \right\}$ 
11         if  $\langle \mathbf{k}_{i+1} - \mathbf{k}_i, \frac{\partial g}{\partial \mathbf{k}}(\mathbf{k}_i) \rangle > 0$  then
12              $l_1 = l_{mid}$ ;
13         else
14              $l_2 = l_{mid}$ ;
15         end
16     end
17      $i \leftarrow i + 1$ 
18 end
19 return  $\rho$ 
```

Starting from an initial uniform material distribution satisfying the prescribed volume fraction, the algorithm iteratively updates the design variables until convergence.

At iteration i , the optimisation variables are first filtered through the Helmholtz operator (line 3). The Helmholtz filter introduces a spatial regularisation of the density field and prevents mesh-dependent artefacts such as checkerboard patterns and excessively fine structural features.

The filtered density field is then used to assemble the finite element system (line 4). The material interpolation follows the SIMP formulation introduced previously, such that intermediate densities are penalised and the optimisation progressively fosters nearly binary material distributions.

Once the equilibrium problem has been solved, the structural compliance is evaluated (line 5), which corresponds to the elastic work of the external forces and therefore measures the global flexibility of the structure.

The sensitivities of both the compliance objective and the volume constraint with respect to the optimisation variables k are then computed (line 6). These gradients are required to determine a descent direction and update the material distribution. The optimisation step itself is performed using an optimality criteria (OC) strategy (lines 7 to 16). This update can be interpreted as a fixed-point approximation of the first-order *Karush–Kuhn–Tucker* (KKT) optimality conditions:

$$\nabla_k c + \lambda \nabla_k g = 0 \quad (3.51)$$

At each iteration, the density field is updated according to the relative sensitivities of the compliance objective and the volume constraint. In particular, the classical OC update at iteration q is

$$k_i^{(q+1)} = k_i^{(q)} \sqrt{-\frac{\partial c}{\partial k} / \lambda \frac{\partial g}{\partial k_i}} \quad (3.52)$$

where λ is the Lagrange multiplier associated with the volume constraint. The multiplier λ is adjusted, typically by bisection, until the updated design satisfies the prescribed volume fraction. When the KKT optimality conditions are exactly satisfied, $k_i^{(q+1)} = k_i^{(q)}$.

This update increases the density in regions where the compliance sensitivity indicates that adding material is beneficial, and decreases it in regions where material contributes little to the stiffness. The move limit (specified by the parameter $m = 0.2$ in the algorithm) stabilises the process by restricting excessively large density variations between two successive iterations. It therefore prevents abrupt design changes and reduces the risk of numerical oscillations, while still allowing the material layout to evolve progressively toward an optimal distribution.

Finally, the iterative process is repeated until the variation between two successive density fields satisfies $\|\rho_{i+1} - \rho_i\| / \|\rho_i\| \leq \epsilon$, where ϵ is a prescribed convergence tolerance. The resulting density field defines the optimised structural topology.

3.5 Results

Sensitivities

The adjoint method is a very powerful tool. It computes the gradient analytically, meaning that all results are exact up to machine errors. A reliable method to check that the results of the adjoint method are correct is to compare with the results of finite differences. The partial derivative $\frac{\partial f_i}{\partial x_j}$ of a multivariate function f with respect to a variable x_j is approximated by

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(x_1, \dots, x_j + \epsilon, \dots, x_n) - f_i(x_1, \dots, x_j - \epsilon, \dots, x_n)}{2\epsilon} \quad (3.53)$$

The accuracy of the different terms is displayed in [Figure 3.3](#), where $\epsilon = 10^{-3}$. This figure shows that the derivations done previously in this section indeed correspond to the actual sensitivities.

The second advantage of the adjoint method is the efficiency: it is linear with respect to the number of nodes, while the finite differences have a quadratic complexity, as shown in [Figure 3.4](#).

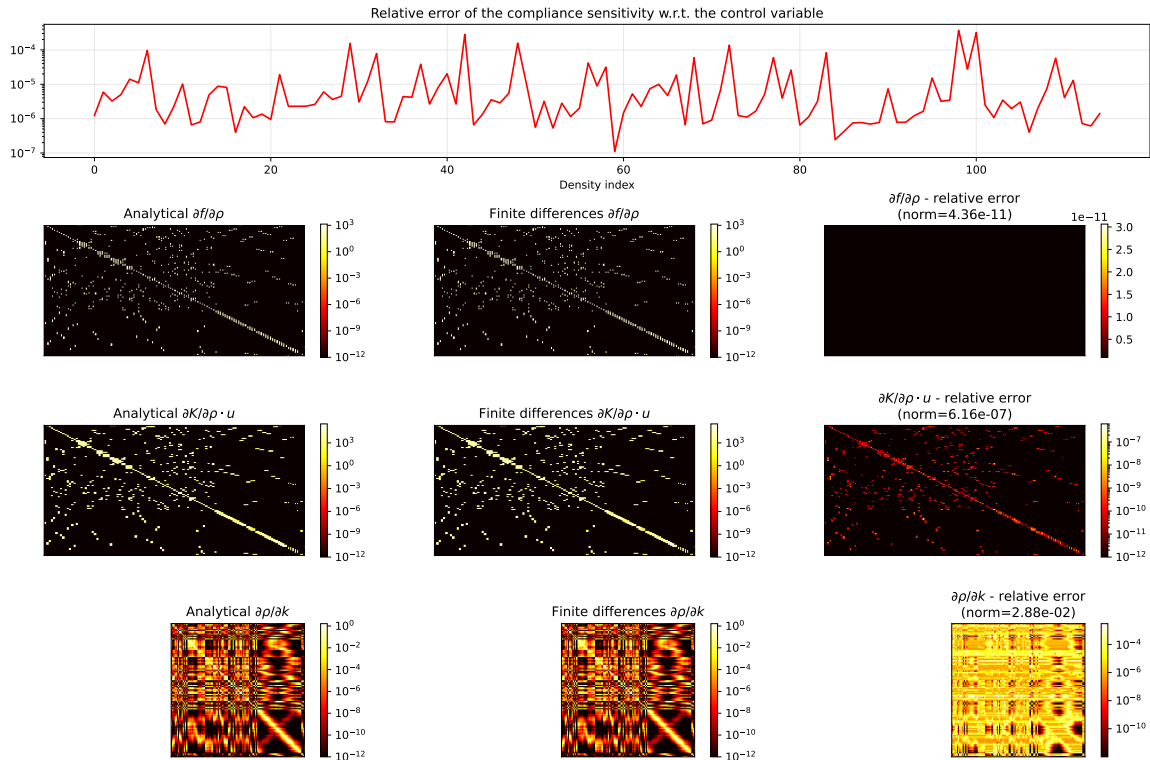


Figure 3.3: Error between finite differences ($\epsilon = 10^{-3}$) and analytical adjoint method on a mesh of 114 nodes.

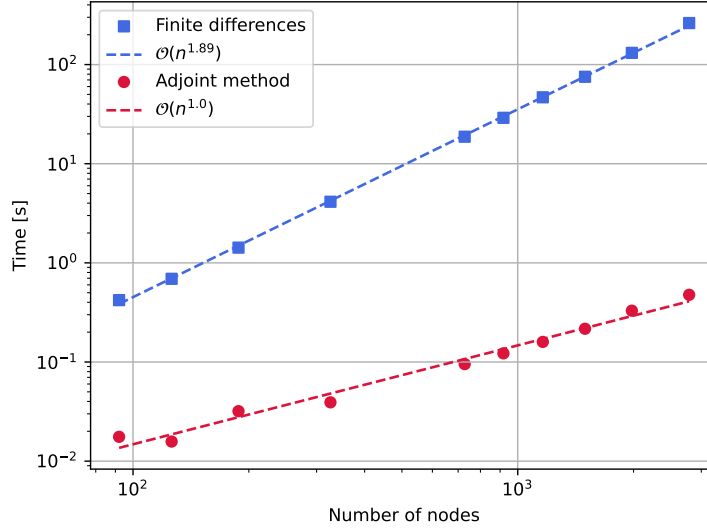


Figure 3.4: Time complexity of the adjoint method and the finite differences with respect to the number of nodes in the mesh.

Convergence and complexity

Let us apply the density approach topology optimisation algorithm to the MBB beam.

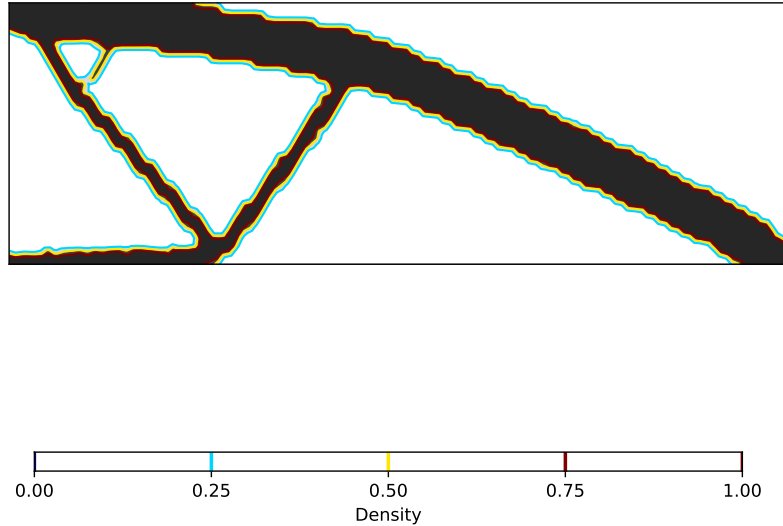


Figure 3.5: Isocurves of the solution of the topology optimisation using the density approach with a Helmholtz filter.

Figure 3.5 displays some iso-curves of the density field. As $\rho : \Omega \rightarrow [0, 1]$, the iso-curve $\rho = 0.5$ is taken as the boundary of the final design, i.e. the interface between matter and void.

This design contains a main structural part that goes from the lower right support point to

the upper left corner, where the vertical downwards force (Neumann condition) is applied. An arm goes down from the middle of that part to the second support point. It is linked to the upper left corner to ensure maximal strength. With the refinement of the mesh, more details may appear, but the core components of the beam are already present.

During the optimisation process, these core components are present very early on, after only a few iterations (cf. Figure 3.6). Most of the iterations are only used to refine the details of the boundaries and smaller structures. This means that, as shown in Figure 3.7, the compliance decreases very fast at the beginning of the process. The relative design variation, i.e. $\|\rho_{i+1} - \rho_i\|/\|\rho_i\| \leq \epsilon$, which corresponds to the stopping criterion, decreases during the process until reaching its threshold value ϵ . Moreover, the volume proportion stays below its maximum value at every iteration of the process, meaning that the validity of the constraint does not need to be checked before ending the algorithm. Hence, the threshold value ϵ can be changed solely depending on the quality of the solution that the user requires. This is a strong result and a main advantage of the method.

Remark 3.5.1. *When the initial iterate has a volume that does not verify the constraint, the compliance first increases to find a feasible point and decreases afterwards at each iteration.*

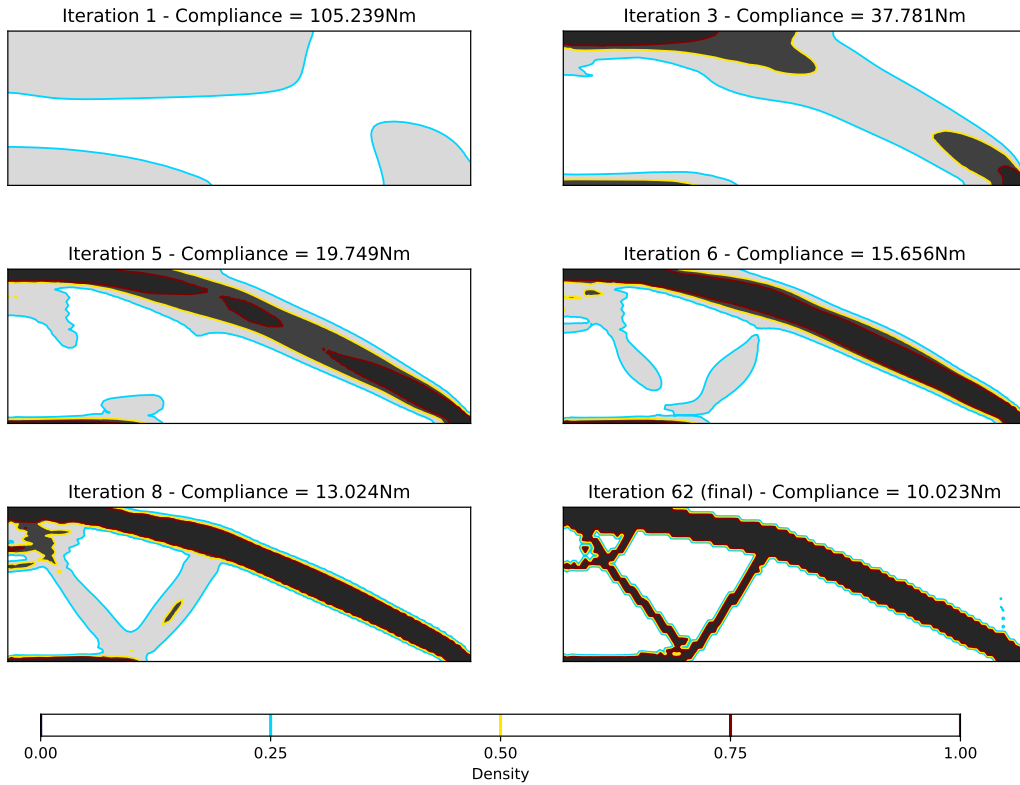


Figure 3.6: Evolution of the design throughout the optimisation process.

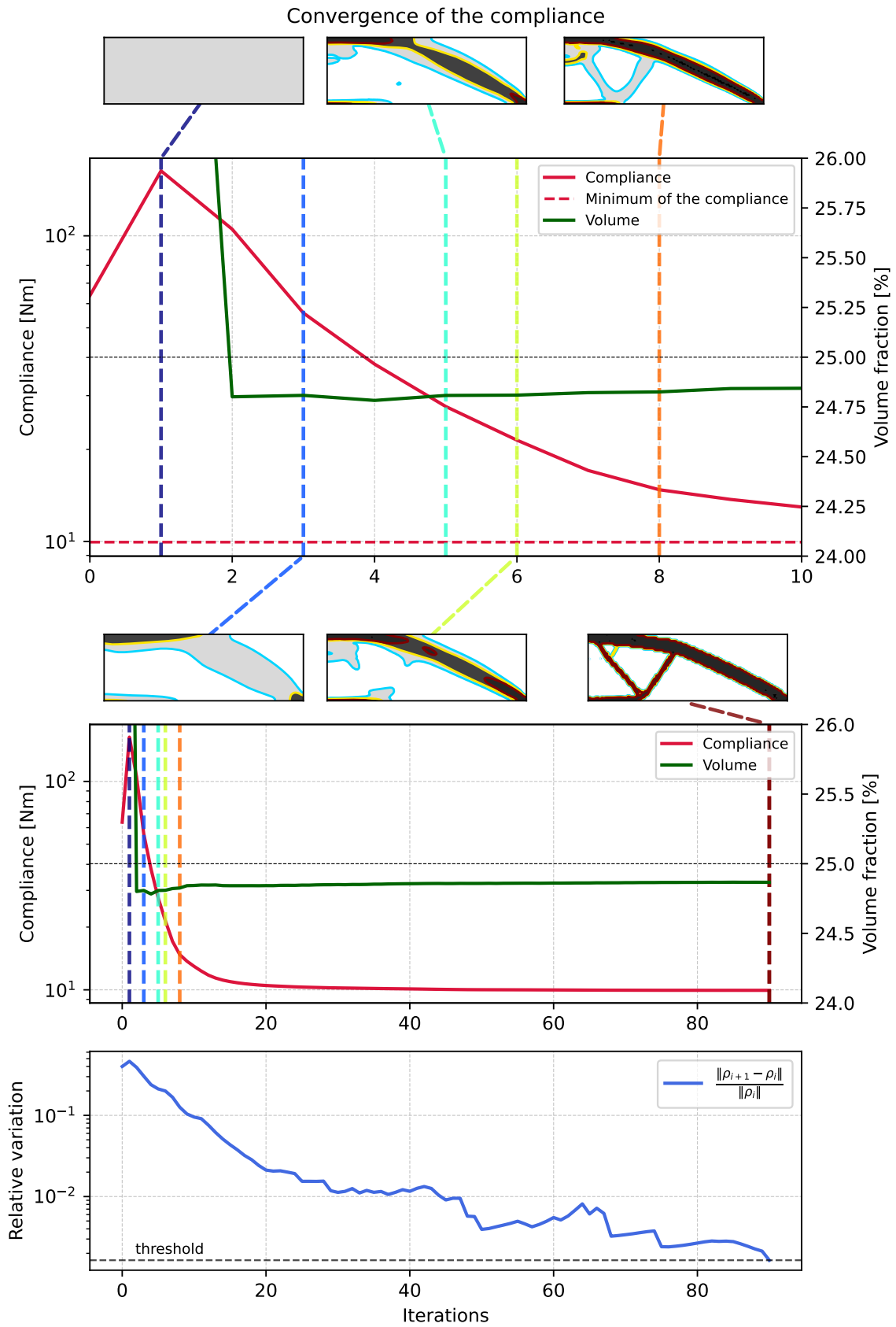


Figure 3.7: Evolution of the compliance throughout the optimisation process.

The complexity of the algorithm is displayed in Figure 3.8. One iteration of the algorithm requires a time that evolves linearly with the number of nodes: the bottleneck is the computation of the gradients, which has been shown previously to be linear. On the other hand, the number of iterations depends more on the complexity of the geometry of the domain and the mesh than on the number of nodes itself. This low complexity makes the algorithm scalable to very large meshes. It also allows to add some more constraints like geometric symmetries.

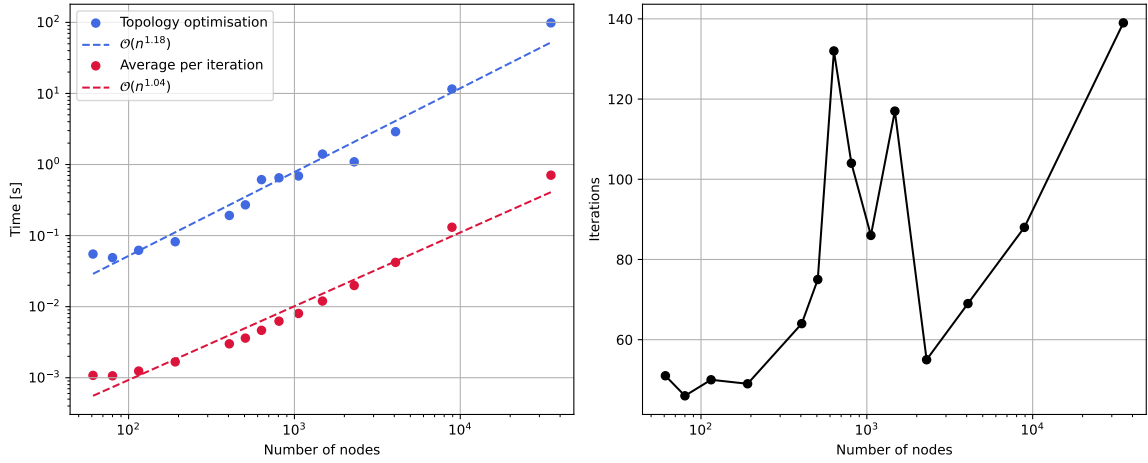


Figure 3.8: Time complexity of the density approach topology optimisation algorithm using a Helmholtz filter.

Filter parameter

As stated previously, the value of the filter radius r has a significant impact on the final design. If this parameter tends to zero, then the control variable corresponds exactly to the density variable resulting to no smoothing. As r grows, the boundary becomes smoother. However, this may delete many details and substructures of the design for high values of r . Figure 3.9 illustrates the evolution of the design for different radii.

Remark 3.5.2. *The parameter h corresponds to the maximum value of the length of the elements. The radius r is usually defined as a proportion of this value.*

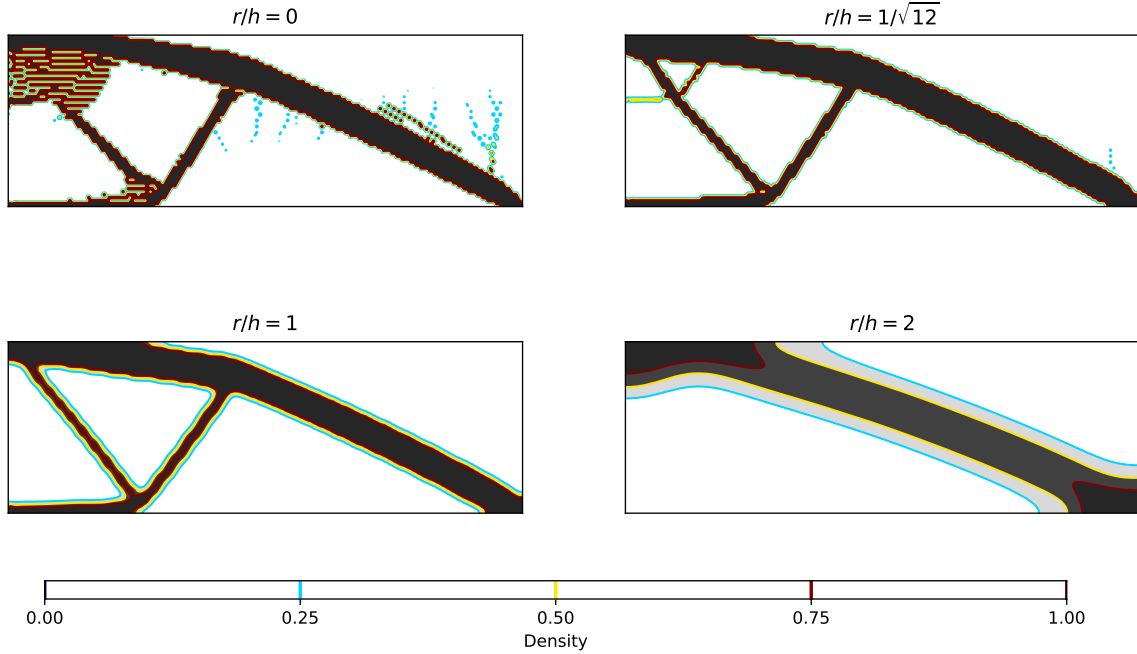


Figure 3.9: Evolution of the design for different values of filter radius.

Summary and outlook

The density-based approach is computationally efficient and offers a high degree of flexibility regarding the incorporation of additional constraints into the optimisation problem. However, it also presents several drawbacks, the most significant being the poor sharpness of the interface between solid and void regions. Since the density field is interpolated linearly between nodes, the resulting boundary is not smooth and exhibits oscillatory features composed of peaks and valleys. Refining the mesh does not fundamentally eliminate this issue: although the oscillations become smaller in absolute size, they persist at the scale of the finite elements. Consequently, mesh refinement merely shifts these artefacts to a finer scale rather than truly resolving the lack of interface regularity. This problem is inherent to the finite element methods and different meshes of the same domain would give different designs, at least at low scales.

Adapting the mesh geometry to better represent the interface between solid and void regions would help reduce these oscillations and produce a smoother boundary. The use of first-order interpolation for the density field inherently limits the regularity of the interface, leading to piecewise-linear boundaries. A possible remedy consists in reconstructing the interface using higher-order interpolation schemes, which can smooth the geometry without significantly altering the total volume of the design. Such techniques are generally introduced as post-processing procedures and are therefore not further investigated in this, which focuses primarily on the numerical methods involved in the optimisation process itself. Although only first-order finite elements have been considered in the previous sections, the methodology

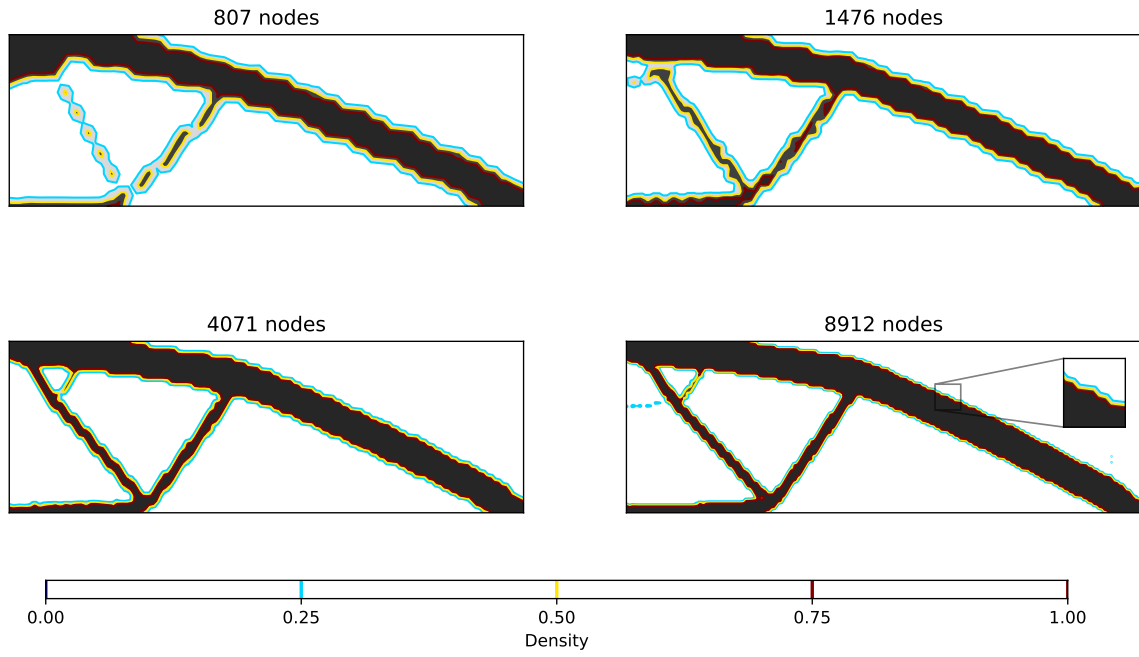


Figure 3.10: Sharpness of the boundary with the refinement of the mesh for a fixed filter radius $r/h = 1/\sqrt{12}$.

naturally extends to higher-order elements, allowing the resulting interfaces to exhibit higher regularity and smoother geometric transitions.

CHAPTER 4

Using X-Mesh to capture the interface

This chapter aims to solve the problem of smooth interface explained at the end of the previous section, using the X-Mesh method (*eXtreme Mesh* deformation method [4]). This method is currently in development at UCLouvain and aims to *continuously* deform meshes to better track physical interfaces in finite element simulations. The main advantage is that it does not require remeshing when the nodes are moved, hence the word *continuous*. It moves the nodes along the interface and relaxes the position of the nodes to their original position when the interface changes, so that the mesh is only locally modified [19]. The goal here is to capture a smoother interface by moving the nodes during the optimisation process.

This chapter begins with an introduction to X-Mesh, and its advantages and disadvantages. The section that follows is the implementation of two different methods to connect X-Mesh to the optimisation algorithm derived previously. Finally, the chapter ends with the results of those methods, and the limitations of the algorithm.

4.1 X-Mesh

The main idea behind the development of X-Mesh is to heavily deform a mesh to follow some sharp physical interfaces, such as cracks or phase changes. X-Mesh currently achieves this well with low computational cost and high robustness [4]. It can handle topological changes, which is why this method is particularly interesting to research.

Instead of remeshing, X-Mesh aims to allow the elements to be distorted to very high extends in a continuous manner so that the interface stays sharp. This idea of a continuous deformation is what makes the algorithm computationally cheap: the connectivity is fixed and the nodes are continuously moved along the edges of the mesh. When a node is moved along an edge and gets to the other end of that edge, i.e. the nodes are superposed and the element has a zero area, then the element is not flipped: the other node continues the deformation along an edge of the next element. The name *eXtreme* comes from the fact that elements are allowed to reach zero measure: those elements are considered as intermediate

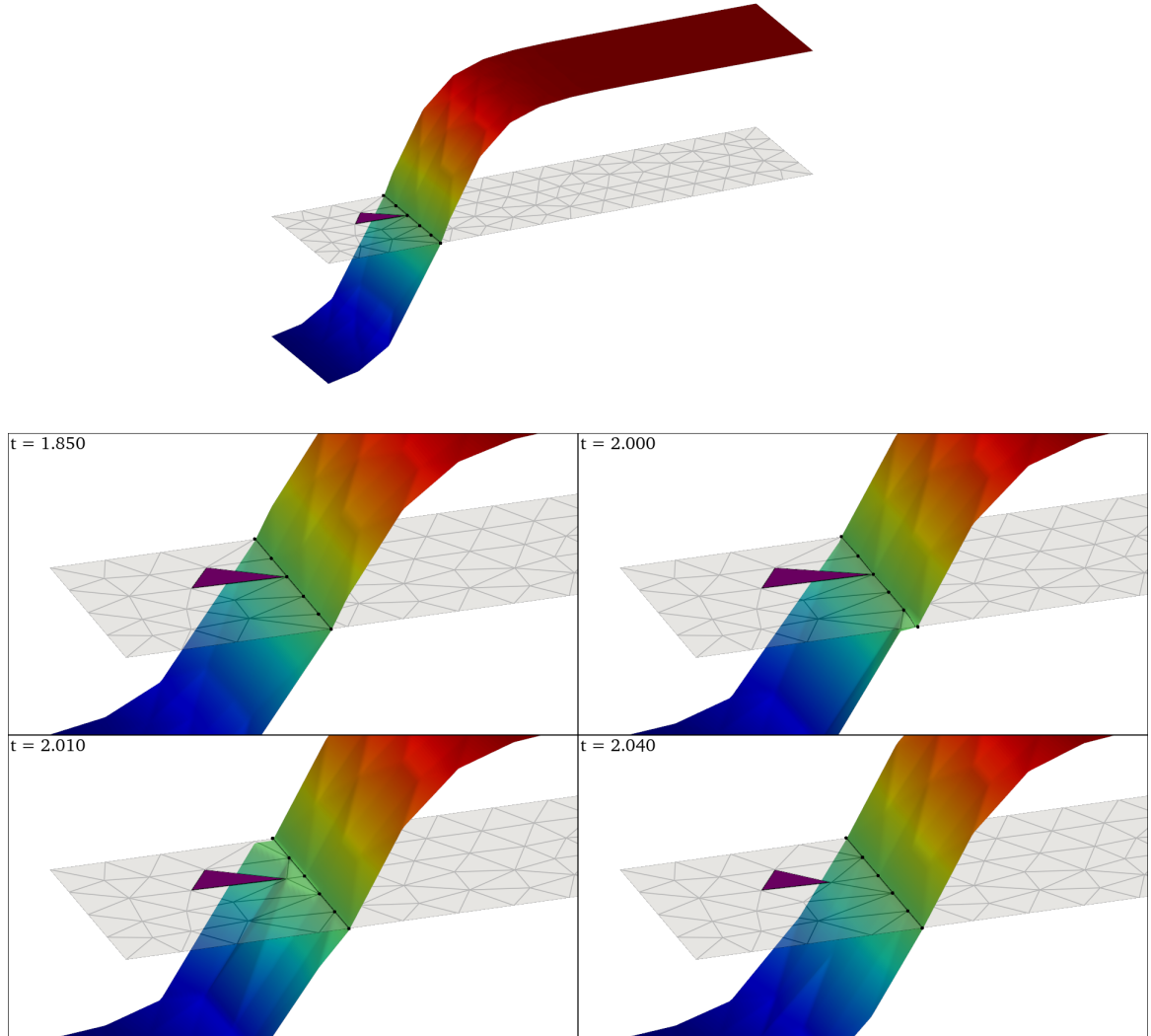


Figure 4.1: Mesh evolution for a linear front that goes forward as time increases.

states during the topological change and the singularities of the stiffness matrix are handled.

Let us consider the example of [Figure 4.1](#). The computational domain is a 4×1 rectangle where the levelset is defined by $\phi : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R} : (x, y, t) \rightarrow \phi(x, y, t) = \tanh(x - \frac{t}{v})$, where v is the speed. As time increases, the levelset $\phi = 0$ propagates in the positive x -direction, and the mesh nodes are displaced accordingly. Let us focus on the evolution of the purple triangle. As the front propagates to the right (from upper-left figure to upper-right), the triangle is stretched in that same direction until the moving front node reaches a neighboring node. At that instant, the original node ceases to belong to the front, while the neighboring node becomes part of the interface instead (from upper right figure to lower left). The front is therefore transferred from one node to another through a local relay

mechanism. Once none of the vertices of the purple triangle belong to the front (from lower left figure to lower right), the triangle progressively relaxes toward its initial configuration while the deformation induced by the propagating interface is transferred to adjacent elements.

Remark 4.1.1. *X-Mesh requires a value for the iso-curve of the interface. For example, it is $\phi = 0$ for the tanh example above, and $\rho = 0.5$ in the case of topology optimisation on the density field.*

4.2 Connecting X-Mesh to topology optimisation

Let us consider once again Algorithm 3.1. Two approaches are possible to add X-Mesh in this topology optimisation algorithm.

Move the nodes after the optimisation process

The first and simplest idea is to treat both steps separately: we first optimise on the density and then move the nodes on the interface that is found. The algorithm is the following:

Algorithm 4.1: Use density optimisation and X-Mesh separately

```

Input: mesh, boundary conditions,  $\alpha$ ;
1   $\rho_0 = \alpha \mathbb{1}_n$ ,  $x_0 = \text{nodes}(\text{mesh})$ ,  $i = 0$ ;
2  while nodes and density have not converged do
3       $\rho_{i+1} = \text{DensityOptimisation}(x_i, \rho_i)$ ;
4       $x_{i+1} = \text{XMesh}(x_i)$ ;
5       $i \leftarrow i + 1$ ;
6  end
7  return  $\rho, x$ 
```

This is computationally heavy as it requires a complete optimisation of the density at every iteration. As seen in Figure 3.7, most of the iterations only shape the details of the design, and most of the work is done at the beginning of the algorithm. As moving the nodes will change those details, the last few iterations do not impact much the final design. Therefore, interconnecting the density and node changes can be more efficient, to avoid having useless iterations of the density optimisation.

Move the nodes during the optimisation process

To implement this interconnectivity, the node displacement can be applied directly inside the optimisation algorithm. Based on Algorithm 3.1, we get Algorithm 4.2.

Algorithm 4.2: Connect X-Mesh to the density optimisation

Input: Inputs: mesh, boundary conditions, α ;

```

1   $\rho_0 = \alpha \mathbb{1}_n$ ,  $x_0 = \text{nodes}(\text{mesh})$ ,  $i = 0$ ;
2  while density and nodes have not converged do
3      Compute the gradient of the compliance and volume constraints;
4      Update the density variable according to the OC criterion;
5      Apply the X-Mesh framework to the density field;
6  end
7  return  $\rho, x$ ;
```

To avoid moving the nodes too often, especially at the beginning where updating the density field is more important, the X-Mesh step can be done every m iterations, typically 10. This gives time to the density field to settle into a quasi-optimal setup before moving. This algorithm is much more efficient thanks to the interconnection of the two main points.

4.3 Results and limitations

The problem of those approaches is that the X-Mesh framework is not compatible with the SIMP: X-Mesh aims to put some nodes on the front, i.e. converge to nodes with a density $\rho = 0.5$, while the SIMP method is used to avoid intermediate values and rather have $\rho \in \{0, 1\}$. They have very opposite goals that clash and create some oscillations of the solution instead of converging.

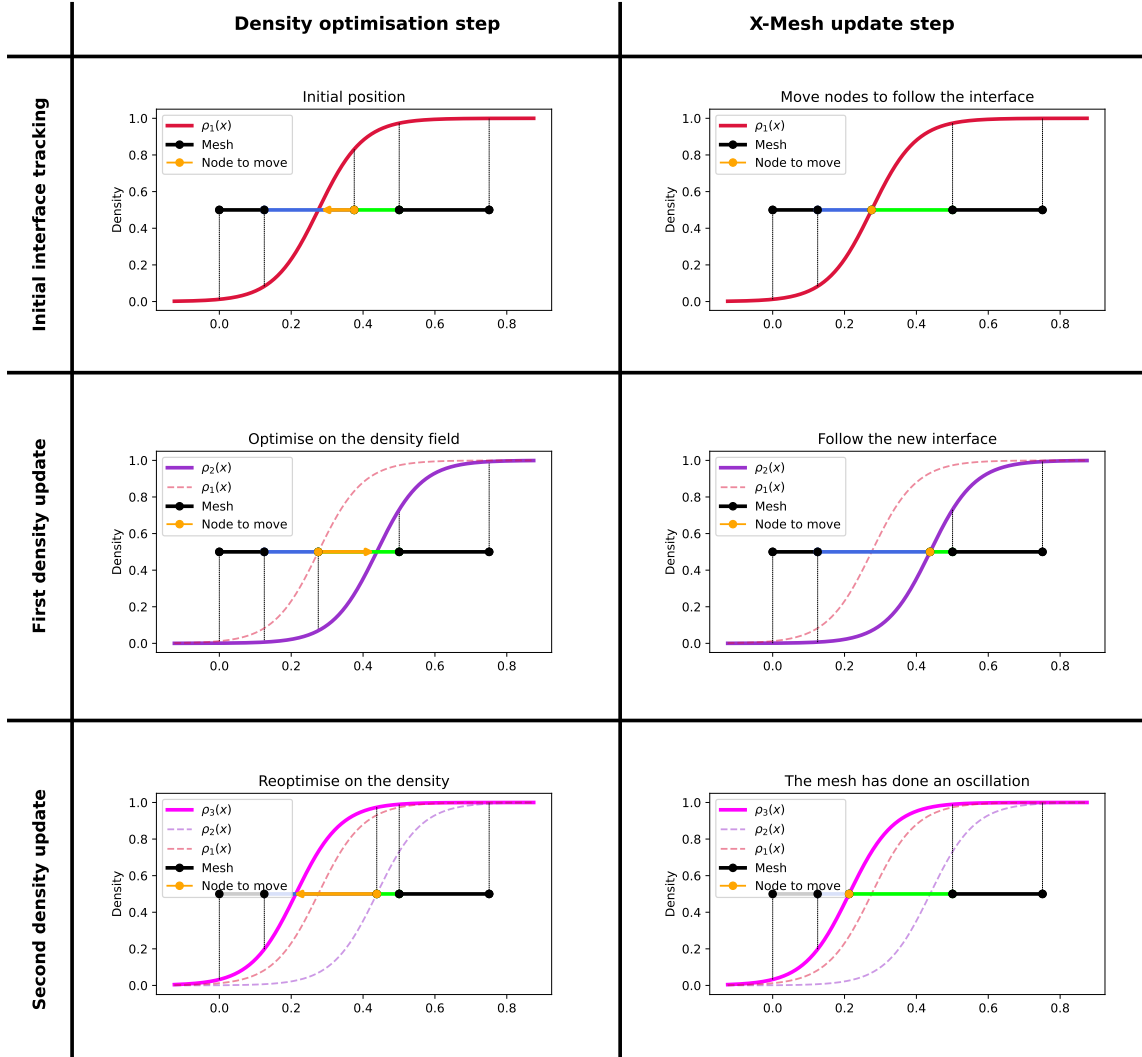


Figure 4.2: Oscillatory behaviour of X-Mesh and SIMP.

Let us take an example of a 1-dimensional mesh to illustrate this phenomenon. Figure 4.2 is a close-up on five nodes of the unidimensional domain, numbered from 1 to 5 from the left. On the initial position, X-Mesh considers that the third node is the one to move and shifts it to the left, to the interface of the field $\rho_1(x)$ between void ($\rho(x) < 0.5$) and matter ($\rho(x) > 0.5$). Then, one or several iterations of the density optimisation process are done until a new density field $\rho_2(x)$ is found. Once again, the node to move is the third one¹. Then, node 3 is moved to the right, and is almost back to its original position! With one more optimisation of the density field, it almost returns to its initial configuration, even though the nodes have moved.

The fact that the SIMP pushes the values to their bounds is incompatible with the objective

¹Note that it could be node 4 as well, but the phenomenon is more visually striking using node 3.

of putting nodes on the interface. This prevents convergence and does not capture the interface better than the algorithm that does not move the nodes. Therefore, to be able to connect X-Mesh to the density optimisation process, we need to define another levelset, that depends on the density but is not the density itself. This idea is explored in the next chapter.

CHAPTER 5

Hybrid levelset approach

The previous chapters used a density approach to compute the optimal design of the domain. This chapter aims to mix the density approach to another one: the levelset approach. This method uses a levelset to track the interface instead of a density field. This allows for a smooth interface which the density approach cannot do on its own, but cannot nucleate, i.e. change the topology by creating holes. Both approaches have an advantage that balances the main disadvantage of the other. Therefore, the two methods seem very interesting to couple, to get a better hybrid method that can nucleate and create a smooth boundary.

This chapter begins with an introduction to the levelset approach in topology optimisation, explaining its pros and cons. It continues with the algorithm developed to mix the density and levelset approaches, and the numerical developments needed to implement this algorithm. Finally, a preprocessing step is explained to stabilize the gradients before developping the results, limitations and perspectives of the method.

5.1 Levelset approach

The levelset approach defines a levelset on the domain, and the boundary between matter ($\phi(x) > 0$) and void ($\phi(x) < 0$) is defined as the set of points $\{x \in \Omega : \phi(x) = 0\}$. The optimisation is done on the domain itself instead of a field defined on that domain. The problem reads

$$\begin{aligned} \min_{\Omega_s \subseteq \Omega} c(\Omega_s), \quad \text{subject to} \quad & K(\Omega_s) \mathbf{u} = \mathbf{f}, \\ & V(\Omega_s) \leq \alpha V_{tot}. \end{aligned} \tag{5.1}$$

The goal is thus to find the stiffest material domain $\Omega_s \subseteq \Omega$ with a limited proportion of the volume of Ω , and this domain Ω_s is defined by the levelset: $\Omega_s = \{x \in \Omega : \phi(x) \geq 0\}$. To update the solution during the iterative process of the method, the shape sensitivity is computed. This measures how the objective changes if the boundary is moved. The levelset function ϕ is updated using an advection equation:

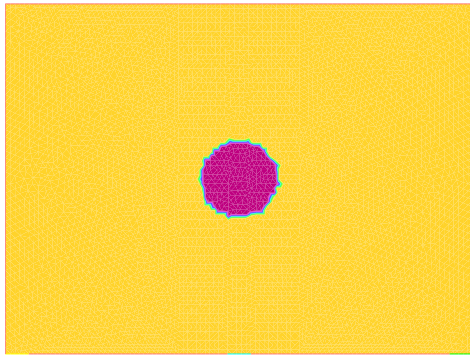
$$\frac{\partial \phi}{\partial \tau} + \mathbf{v} \cdot \nabla \phi = 0, \tag{5.2}$$

where $\mathbf{v}$ constitutes the normal velocity and τ is a pseudo-time. This equation moves the boundary according to the computed sensitivity.

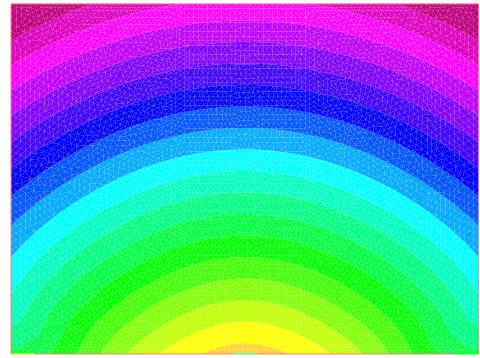
Remark 5.1.1. *To avoid numerical instability, the levelset can be reinitialised without changing the boundary. This allows to keep a well-conditioned field during the whole optimisation process.*

The main drawback of the levelset method is that a boundary must be defined for the first iterate. Once done, the number of cavities in the design cannot increase: we say that the method cannot *nucleate*.

As shown on Figures 5.1 and 5.2, the levelset method cannot create any hole. To have

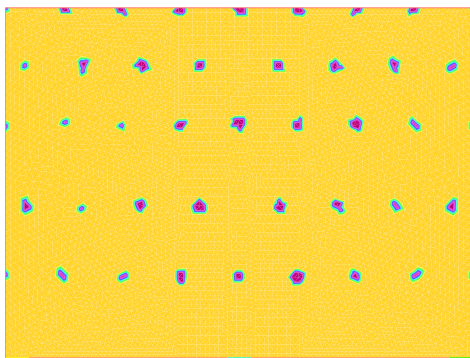


(a) Initial configuration.

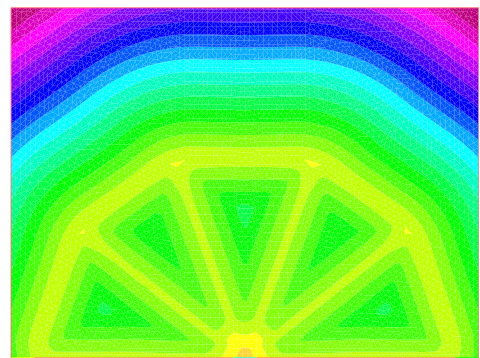


(b) Final design, not representing anything useful.

Figure 5.1: Topology optimisation of a wheel-shaped bridge using the levelset method, initialised with a single hole. The colors go from purple ($\phi < 0$, void) to orange ($\phi > 0$, matter). Figure from [34].



(a) Initial configuration.



(b) Final design.

Figure 5.2: Topology optimisation of a wheel-shaped bridge using the levelset method, initialised with many holes. Figure from [34].

cavities in the final design, they must be added to the initial domain, before the optimisation. As the method can delete some holes, it is not a problem to have too many holes in the

initial domain. As seen in the figures, if the initial number of holes is too low, the final design will not be optimal: the output of the method highly depends on its input.

The main advantage of this method is that the boundary is well-defined and crisp. This method creates designs easier to manufacture than the density method. The idea of this chapter is then to mix the two methods to have nucleation in the method itself, without any user input, and to get the crisp boundary that the levelset method creates.

5.2 Mixing the two methods

Mixing the two methods should bypass the problem of incompatibility between the density method and X-Mesh, by defining a different levelset. The algorithm is illustrated on the diagram in [Figure 5.3](#).

A first design is obtained through the density method, and a levelset field is constructed based on that solution. The levelset used is the signed distance, that is positive in the matter and negative in the void. Afterwards, a velocity field is computed based on the sensitivity of the compliance with respect to the position of the nodes. The levelset is then advected using this velocity and the nodes are moved using X-Mesh (see previous chapter). This process is done until convergence of the nodal positions is reached. To improve stability, the algorithm can recreate the levelset from the density field to stay close to the initial design. Even more rarely, the density field can also be recomputed from the first step of the algorithm to continue the nucleation if the advected design allows holes that were restrained by the shape of the mesh.

Defining an intermediate levelset helps avoid the problem of directly applying X-Mesh to the density field, that is oscillations of the boundary because of the incompatibility of the SIMP and X-Mesh.

The discretised advection equation is solved by using the time-implicit method of Crank-Nicholson. This helps with numerical stability and allows to use any time step value. The time step used later in the results is $\Delta t = h/(2 \max_i \|\mathbf{v}(\mathbf{x}_i)\|)$, where h is the mesh size.

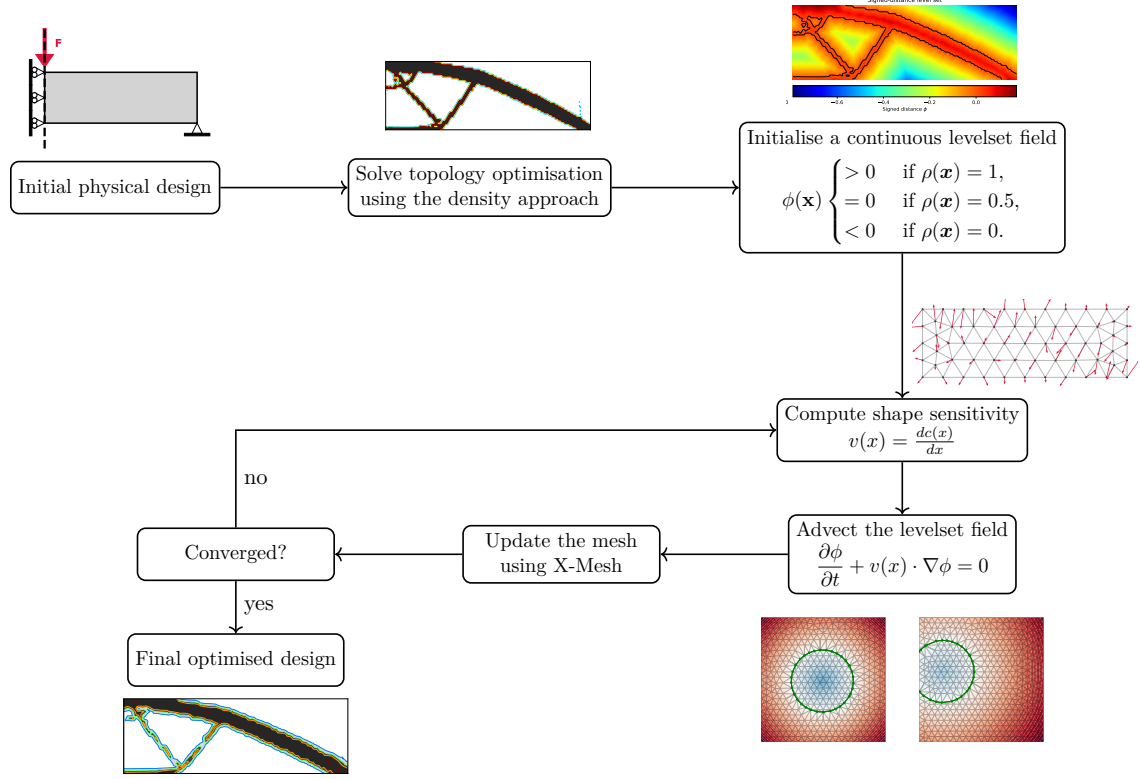


Figure 5.3: Hybrid topology optimisation workflow.

5.3 Numerical developments

To compute the sensitivity of the objective function with respect to the position of the nodes, the adjoint method is once again used. This method uses the equality constraints of the optimisation problem to delete the implicit derivatives in the computation of the gradient. As a reminder, the topology optimisation problem in discrete form is

$$\begin{aligned}
 \min_{\boldsymbol{\rho}} c(\mathbf{x}, \mathbf{u}(\mathbf{x}), \boldsymbol{\rho}(\mathbf{x})), \quad \text{subject to} \quad & \mathbf{r}(\mathbf{x}, \mathbf{u}(\mathbf{x}), \boldsymbol{\rho}(\mathbf{x})) = 0, \\
 & \mathbf{h}(\mathbf{x}, \boldsymbol{\rho}(\mathbf{x})) = 0, \\
 & g(\mathbf{x}, \boldsymbol{\rho}(\mathbf{x})) \leq 0,
 \end{aligned} \tag{5.3}$$

where $c : \Omega \rightarrow \mathbb{R}$ is the objective function, namely the compliance of the domain; $\mathbf{r} : \Omega \rightarrow \mathbb{R}^n$ is the physical constraint, i.e. the discretised linear elasticity equation; $\mathbf{h} : \Omega \rightarrow \mathbb{R}^n$ is the Helmholtz filter; and $g : \Omega \rightarrow \mathbb{R}$ is the volume constraint.

The adjoint method defines a Lagrangian $\mathcal{L}_c = c + \boldsymbol{\lambda}^T \mathbf{r} + \boldsymbol{\mu}^T \mathbf{h}$, where the Lagrangian variables $\boldsymbol{\lambda}, \boldsymbol{\mu}$ are chosen to suppress the implicit derivatives. The sensibility of the compliance with

respect to the position of the nodes is thus

$$\begin{aligned}
\frac{dc}{dx} &= \frac{d\mathcal{L}_c}{dx} = \frac{\partial c}{\partial x} + \frac{\partial c}{\partial \mathbf{u}} \left[\frac{\partial \mathbf{u}}{\partial x} \right] + \frac{\partial c}{\partial \boldsymbol{\rho}} \left[\frac{\partial \boldsymbol{\rho}}{\partial x} \right] + \boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{r}}{\partial x} + \frac{\partial \mathbf{r}}{\partial \boldsymbol{\rho}} \left[\frac{\partial \boldsymbol{\rho}}{\partial x} \right] + \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \left[\frac{\partial \mathbf{u}}{\partial x} \right] \right) \\
&\quad + \boldsymbol{\mu}^T \left(\frac{\partial \mathbf{h}}{\partial x} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\rho}} \left[\frac{\partial \boldsymbol{\rho}}{\partial x} \right] \right) \\
&= \frac{\partial c}{\partial x} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{r}}{\partial x} + \boldsymbol{\mu}^T \frac{\partial \mathbf{h}}{\partial x} + \left(\frac{\partial c}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right) \left[\frac{\partial \mathbf{u}}{\partial x} \right] \\
&\quad + \left(\frac{\partial c}{\partial \boldsymbol{\rho}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{r}}{\partial \boldsymbol{\rho}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{h}}{\partial \boldsymbol{\rho}} \right) \left[\frac{\partial \boldsymbol{\rho}}{\partial x} \right]
\end{aligned} \tag{5.4}$$

The adjoint method simplifies this expression by choosing the Lagrangian variables such that the colored terms do not appear in the final expression of the sensitivity, by cancelling out the factors in front:

$$\begin{cases} \frac{\partial c}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{r}}{\partial \mathbf{u}} = 0 \\ \frac{\partial c}{\partial \boldsymbol{\rho}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{r}}{\partial \boldsymbol{\rho}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{h}}{\partial \boldsymbol{\rho}} = 0 \end{cases} \tag{5.5}$$

By the definition of each of these quantities (cf. [Section 3.3](#)), we get

$$\begin{cases} \mathbf{u}(K + K^T) + \boldsymbol{\lambda}^T K = 0 \iff K^T \boldsymbol{\lambda} = -(\mathbf{K} + K^T) \mathbf{u} \\ \mathbf{u}^T \frac{\partial K}{\partial \boldsymbol{\rho}} \mathbf{u} + \boldsymbol{\lambda}^T \frac{\partial K}{\partial \boldsymbol{\rho}} \mathbf{u} + \boldsymbol{\mu}^T H = 0 \iff H^T \boldsymbol{\mu} = -\left(\mathbf{u}^T \frac{\partial K^T}{\partial \boldsymbol{\rho}} \mathbf{u} + \mathbf{u}^T \frac{\partial K^T}{\partial \boldsymbol{\rho}} \boldsymbol{\lambda} \right) \end{cases} \tag{5.6}$$

This gives two linear system to solve to get the Lagrangian variables, and the sensitivity of the compliance becomes

$$\frac{dc}{dx} = (\mathbf{u} + \boldsymbol{\lambda})^T \frac{\partial K}{\partial x} \mathbf{u} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial x} + \boldsymbol{\mu}^T \left(\frac{\partial H}{\partial x} \boldsymbol{\rho} - \frac{\partial M}{\partial x} \mathbf{k} \right) \tag{5.7}$$

The derivatives with respect to the density variable $\boldsymbol{\rho}$ have been computed earlier, so let us focus on the $\frac{\partial}{\partial x}$ terms. We consider the terms defined on each element, and the global derivatives are the assembly of those local terms.

$$\begin{aligned}
\frac{\partial K^{(e)}}{\partial \mathbf{x}^{(e)}} &= \frac{\partial}{\partial \mathbf{x}^{(e)}} \left(\int_{\Omega} B^{(e)T} C B^{(e)} dS \right) = \frac{\partial}{\partial \mathbf{x}^{(e)}} \left(B^{(e)T} A(\nu) B^{(e)} \int_{\Omega} E(\rho) dS \right) \\
&= \frac{\partial}{\partial \mathbf{x}^{(e)}} \left(B^{(e)T} A(\nu) B^{(e)} 2\text{Area}(\Omega^{(e)}) \Gamma(\rho) \right) \\
&= \frac{\partial B^{(e)T}}{\partial \mathbf{x}^{(e)}} A(\nu) B^{(e)} 2\text{Area}(\Omega^{(e)}) \Gamma(\rho) + B^{(e)T} A(\nu) \frac{\partial B^{(e)}}{\partial \mathbf{x}^{(e)}} 2\text{Area}(\Omega^{(e)}) \Gamma(\rho) \\
&\quad + B^{(e)T} A(\nu) B^{(e)} \frac{\partial 2\text{Area}(\Omega^{(e)})}{\partial \mathbf{x}^{(e)}} \Gamma(\rho)
\end{aligned} \tag{5.8}$$

As we consider P1 elements, shape functions are linear and thus B is linear in the position of the nodes, therefore the computation of $\frac{\partial B}{\partial x}$ is straightforward and this tensor is of shape $(6 \times 3 \times 6)$.

Recall that the area of an element is given by the following formula:

$$\text{Area}(\Omega) = \frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)| \quad (5.9)$$

Then, its gradient is

$$\frac{\partial \text{Area}(\Omega)}{\partial \mathbf{x}} = \begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \\ y_3 - y_1 \\ x_1 - x_3 \\ y_1 - y_2 \\ x_2 - x_1 \end{pmatrix} \quad (5.10)$$

Because of the definition of the meshes, where the nodes are browsed counter-clockwise, the absolute value is not necessary and therefore is not handled in the derivative.

For the force vector, the sensitivity is

$$\begin{aligned} \frac{\partial \mathbf{f}_y}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} (-\rho_{\text{matter}} g M \boldsymbol{\rho}) = \rho_{\text{matter}} g \frac{\partial M}{\partial \mathbf{x}} \boldsymbol{\rho} \\ &= \rho_{\text{matter}} g \frac{\partial}{\partial \mathbf{x}} \left(\frac{\text{Area}(\Omega)}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \right) \boldsymbol{\rho} \\ &= \frac{\rho_{\text{matter}} g}{12} \frac{\partial (\text{Area}(\Omega))}{\partial \mathbf{x}} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \boldsymbol{\rho} \end{aligned} \quad (5.11)$$

This gives a (6×3) matrix, to which should be concatenated the contributions of $\frac{\partial \mathbf{f}_x}{\partial \mathbf{x}}$, but as $\mathbf{f}_x = 0$, this term is 0.

For the sensitivities in the expression of the filter, $\frac{\partial M}{\partial \mathbf{x}}$ has been computed for $\mathbf{f}_y$ above, thus only $\frac{\partial H}{\partial \mathbf{x}}$ is yet to be derived.

$$\frac{\partial H}{\partial \mathbf{x}} = \frac{\partial (r^2 K_f + M)}{\partial \mathbf{x}} = r^2 \frac{\partial K_f}{\partial \mathbf{x}} + \frac{\partial M}{\partial \mathbf{x}} \quad (5.12)$$

And the only term to compute is $\frac{\partial K_f}{\partial \mathbf{x}}$: the expression in Equation 3.33 can also be written as

$$K_f = \frac{G}{4\text{Area}(\Omega)} = \frac{\mathbf{b}\mathbf{b}^T + \mathbf{c}\mathbf{c}^T}{4\text{Area}(\Omega)}, \quad \mathbf{b} = \begin{pmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} x_3 - x_2 \\ x_1 - x_3 \\ x_2 - x_1 \end{pmatrix} \quad (5.13)$$

From this, its derivative is

$$\frac{\partial K_f}{\partial \mathbf{x}} = \frac{\partial G}{\partial \mathbf{x}} \frac{1}{4\text{Area}(\Omega)} - \frac{\partial \text{Area}(\Omega)}{\partial \mathbf{x}} \frac{G}{\text{Area}^2(\Omega)} \quad (5.14)$$

We get

$$\begin{aligned} \frac{\partial G}{\partial \mathbf{x}} &= \frac{\partial \mathbf{b}}{\partial \mathbf{x}} \mathbf{b}^T + \mathbf{b} \frac{\partial \mathbf{b}^T}{\partial \mathbf{x}} + \frac{\partial \mathbf{c}}{\partial \mathbf{x}} \mathbf{c}^T + \mathbf{c} \frac{\partial \mathbf{c}^T}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{b}}{\partial \mathbf{x}} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \end{pmatrix} \quad \frac{\partial \mathbf{c}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (5.15)$$

We can use all those derivations for Equation 5.12 to get a $(6 \times 3 \times 3)$ local tensor.

Sensitivity of the constraint In the case of nodal positions, the volume constraint also requires to use the adjoint method, because of the implicit derivative $\frac{\partial \rho}{\partial \mathbf{x}}$:

$$\mathcal{L}_g(\mathbf{x}) = g(\boldsymbol{\rho}(\mathbf{x}), \mathbf{x}) + \boldsymbol{\zeta}^T \mathbf{h}(\boldsymbol{\rho}(\mathbf{x}), \mathbf{x}) \quad (5.16)$$

Then, the sensitivity is

$$\begin{aligned} \frac{dg}{d\mathbf{x}} &= \frac{d\mathcal{L}_g}{d\mathbf{x}} = \frac{\partial g}{\partial \mathbf{x}} + \frac{\partial g}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}} + \boldsymbol{\zeta}^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}} \right) \\ &= \frac{\partial g}{\partial \mathbf{x}} + \boldsymbol{\zeta}^T \frac{\partial \mathbf{h}}{\partial \mathbf{x}} + \left[\frac{\partial g}{\partial \boldsymbol{\rho}} + \boldsymbol{\zeta}^T \frac{\partial \mathbf{h}}{\partial \boldsymbol{\rho}} \right] \boxed{\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}}} \end{aligned} \quad (5.17)$$

The derivatives with respect to the density variable are computed in Section 3.3. This gives for the Lagrangian parameter:

$$H^T \boldsymbol{\zeta} = -\frac{\partial g}{\partial \boldsymbol{\rho}} \implies \frac{dg}{d\mathbf{x}} = \frac{\partial g}{\partial \mathbf{x}} + \boldsymbol{\zeta}^T \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \quad (5.18)$$

Let us compute the $\frac{\partial}{\partial \mathbf{x}}$ terms. As a reminder,

$$g(\boldsymbol{\rho}(\mathbf{x}), \mathbf{x}) = \sum_e \int_{\Omega^{(e)}} \rho(x, y) dS - \alpha V_{tot} \quad (5.19)$$

Its sensitivity is thus

$$\frac{\partial g}{\partial \mathbf{x}^{(e)}} = \left(\frac{\rho_1^{(e)} + \rho_2^{(e)} + \rho_3^{(e)}}{3} - \alpha \right) \frac{\partial \text{Area}(\Omega^{(e)})}{\partial \mathbf{x}^{(e)}} \quad (5.20)$$

Moreover, the derivative of the area has been computed in Equation 5.10, and the derivative of H in Equation 5.12.

Numerical sanity check As done in Section 3.5, let us compare those analytical results to the finite differences. All previous computations indeed correspond to the finite difference approximations, for the compliance sensitivity as well as the one of the volume constraint. Moreover, we get the same results as previously, i.e. the complexity of the adjoint method (cf. Figure 5.6) is far better than the one of the finite differences, and the values are exact up to machine errors as the derivations are purely analytical. Therefore, this method is better on all fronts, even though it is more complex to implement.

5.4 Preprocessing of the velocity

Two improvements have been added to the computation of the shape sensitivity in Figure 5.3. The first one concerns stability: the velocity field is smoothed using a Helmholtz filter (cf. Figure 5.7), just like for the density approach, with homogeneous Dirichlet conditions on the boundary of the domain:

$$\begin{cases} -r^2 \Delta \mathbf{w} + \mathbf{w} = \mathbf{v} & \text{in } \Omega, \\ \mathbf{w} \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (5.21)$$

where the field $\mathbf{v}$ corresponds to the velocity and $\mathbf{w}$ is the smoothed field used for the advection. The boundary conditions prevent the final design from exiting the initial domain Ω . In Figure 5.7, we can clearly see the shape of the beam in the velocity field. Naturally, the sensitivity to a change in the nodal position is thus much larger at the boundaries of the domain than in the large empty spaces. Therefore, the nodes at the boundary will move much more than the other ones during the advection. The second improvement is to take into account the volume constraint: if the volume is not limited during the advection, then the nodes will move in such a way that they improve the compliance by expanding the thickness of the beam. To do that, the velocity field is upgraded to a Lagrangian:

$$\mathbf{v}(\mathbf{x}) = \frac{dc}{d\mathbf{x}} + \ell \frac{dg}{d\mathbf{x}} \quad (5.22)$$

where both $\frac{dc}{d\mathbf{x}}$ and $\frac{dg}{d\mathbf{x}}$ have been smoothed through the Helmholtz filter.

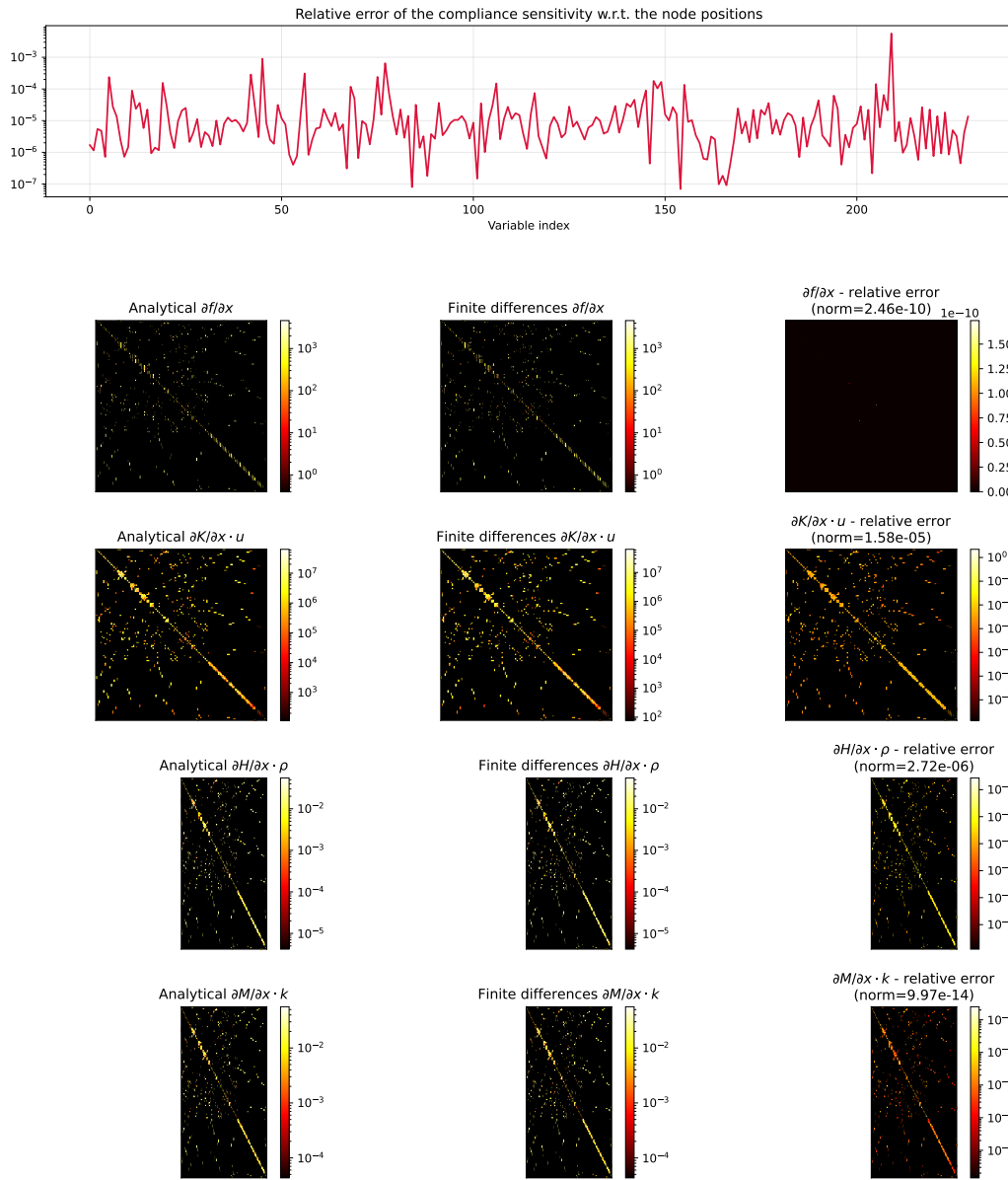


Figure 5.4: Comparison of the adjoint method with the finite differences, using $\varepsilon = 10^{-3}$, for the sensitivity with respect to the node positions.

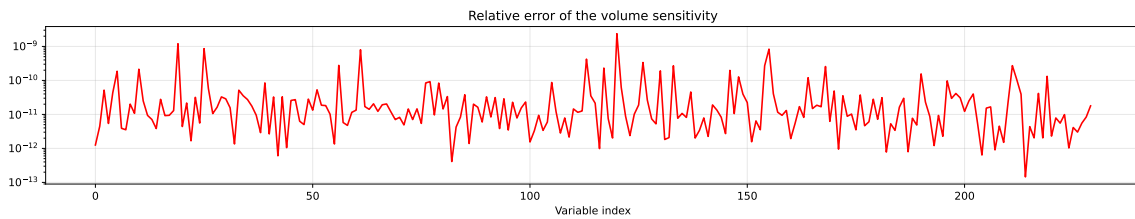


Figure 5.5: Correctness of the adjoint method for the sensitivity of the volume constraint

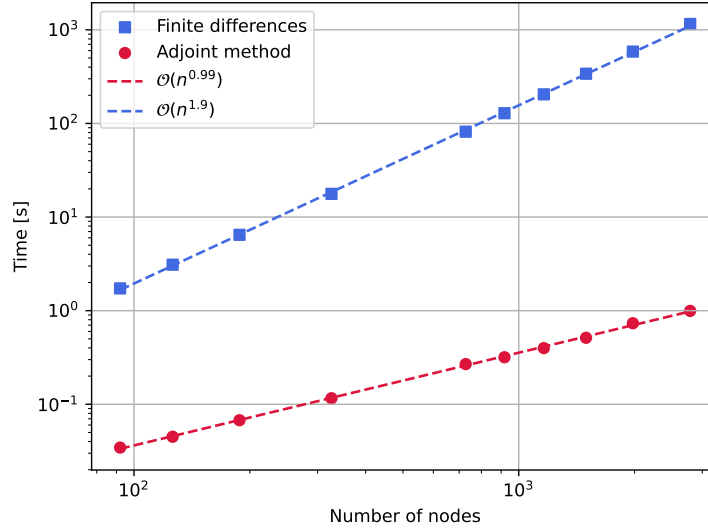


Figure 5.6: Time complexity of the adjoint method for the sensitivity with respect to the position of the nodes versus the finite differences.

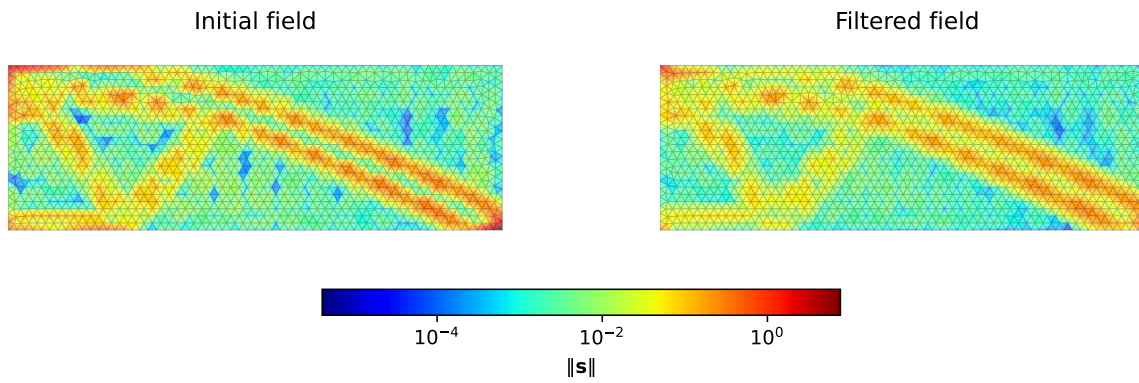


Figure 5.7: Smoothing of the velocity field

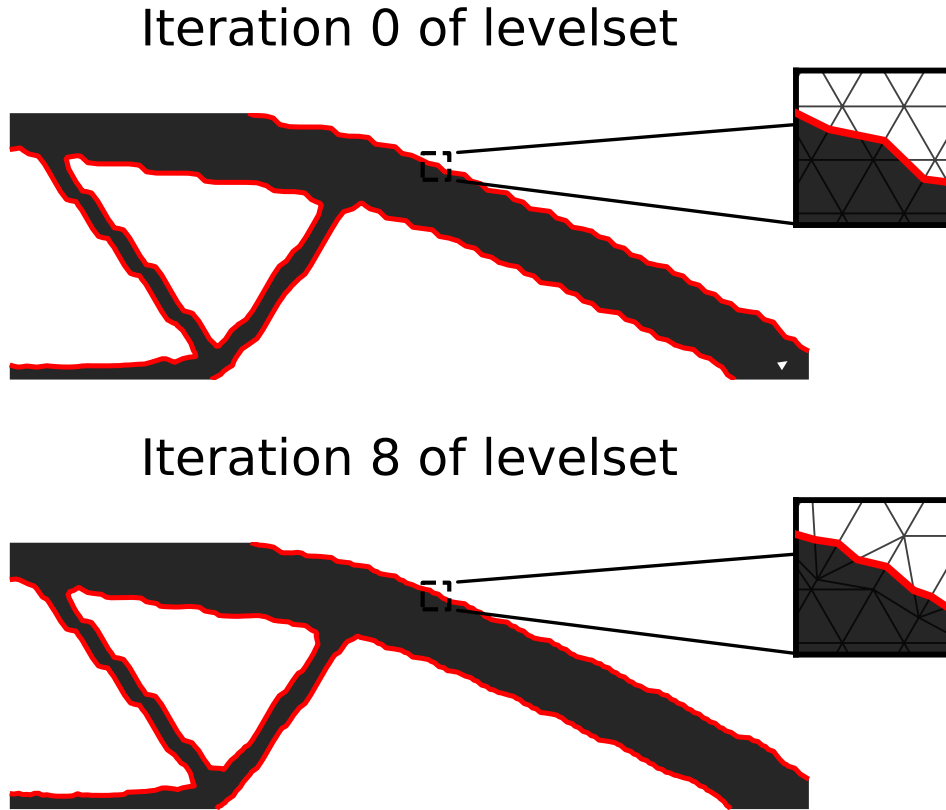


Figure 5.8: Improvements generated by the levelset method.

Remark 5.4.1. *By linearity of both the Helmholtz equation and the Lagrangian, applying them in any order yields the same velocity values.*

The Lagrangian parameter ℓ is determined by a bisection method.

The results displayed in Figure 5.8 show that the algorithm improves the sharpness of the boundary even after a small number of iterations. The Figure 5.9 displays some metrics about those results. Surprisingly, moving the nodes increases the compliance. Moreover, as expected, the method decreases the total length of the boundary because it straightens it, as less oscillations imply a shorter boundary. Moreover, the volume proportion increases during some iterations, but the choice of Lagrangian parameter ℓ eventually brings the volume back down to its upper bound.

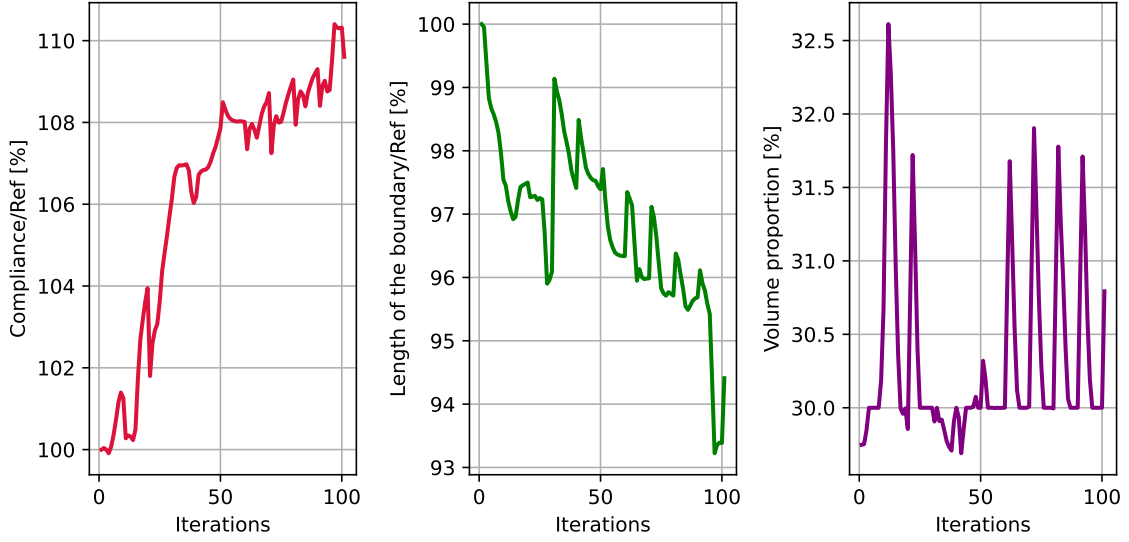


Figure 5.9: Evolution of the compliance, the length of the boundary and the volume constraint during the optimisation process, using $g(\boldsymbol{\rho}) = V(\boldsymbol{\rho})/V_{tot} \leq 30\%$.

5.5 Limitations and perspectives

The hybrid method shows promising results for the regularisation of the material boundary. In particular, the oscillatory behaviour observed when the boundary is directly fitted to the density field is reduced after only a few iterations. The introduction of a Lagrangian term in the velocity field also provides a practical way to control the volume constraint during the levelset evolution. In addition, the smoothing of the velocity field and the use of a Crank–Nicholson scheme for the advection equation improve the numerical robustness of the method.

However, several limitations remain. First, the bisection procedure used to determine the Lagrange multiplier is computationally expensive, since each trial value requires the levelset advection and mesh update to be recomputed. A more efficient strategy should therefore be investigated, to reduce the number of repeated advection solves while still enforcing the volume constraint approximately.

Second, the convergence of the hybrid method remains difficult to assess. In the numerical experiments considered here, the nodal positions did not converge within a reasonable number of iterations. This suggests that the current stopping criterion, based on the variation of the mesh nodes, may not be sufficient on its own. Future work should therefore investigate more appropriate convergence indicators, combining for instance the relative variation of the compliance, the volume constraint violation, the interface length, and the displacement of the boundary nodes.

Finally, the choice of numerical parameters remains important. The time step, the velocity-filtering radius, the reinitialisation frequency, and the relaxation parameter used in the mesh updates all influence the stability and efficiency of the method. A systematic study of these parameters would be necessary to improve the robustness of the algorithm. Another possible improvement would be to periodically recompute the density field during the levelset evolution. This could reintroduce some topological flexibility, and in particular allow new holes to nucleate during the optimisation process, which is not naturally achieved by a purely boundary-evolution levelset method.

CHAPTER 6

Conclusion

This work is the continuation of a previous master thesis that explored a method to sharpen the boundary of topology optimisation designs, therefore needing less postprocessing before manufacturing. It used the density approach as seen in [Chapter 3](#) without the filter and used X-Mesh to modify the mesh. Here, a novel hybrid method has been investigated. The goal was to mix the two main methods used in this domain: the density approach and the levelset approach.

The density approach has been developed in details in [Chapter 3](#). It defines a density field $\rho : \Omega \rightarrow [0, 1]$ on the domain. The goal is to find the optimal field that minimises a certain quantity, generally the compliance, i.e. the reciprocal of the stiffness. The SIMP, a penalization on the Young modulus of the material, was introduced to force the density to binary values and have a clear distinction between matter and void. Furthermore, the method used a filter to regularise the density field and to reduce mesh-dependent artefacts, such as checkerboard patterns. This thesis considered the Helmholtz filter that corresponds to a diffusion equation in pseudo-time. The density approach is very powerful and works well because of its flexibility: the objective function can easily be modified and constraints can be added to the model. It also allows nucleation, and this is the reason why it has been studied in this work. It creates designs such as the ones displayed in [Figure 6.1](#): it adapts to any sort of domain and is highly generalised, making it a very strong basis for the rest of the research.

The scalability of the method is due to the use of the adjoint method for the computation of the gradients. This method allows to compute the sensitivities analytically at very low cost ($\mathcal{O}(n)$ for n the number of nodes), while easily enforcing the constraints of linear elasticity and filter.

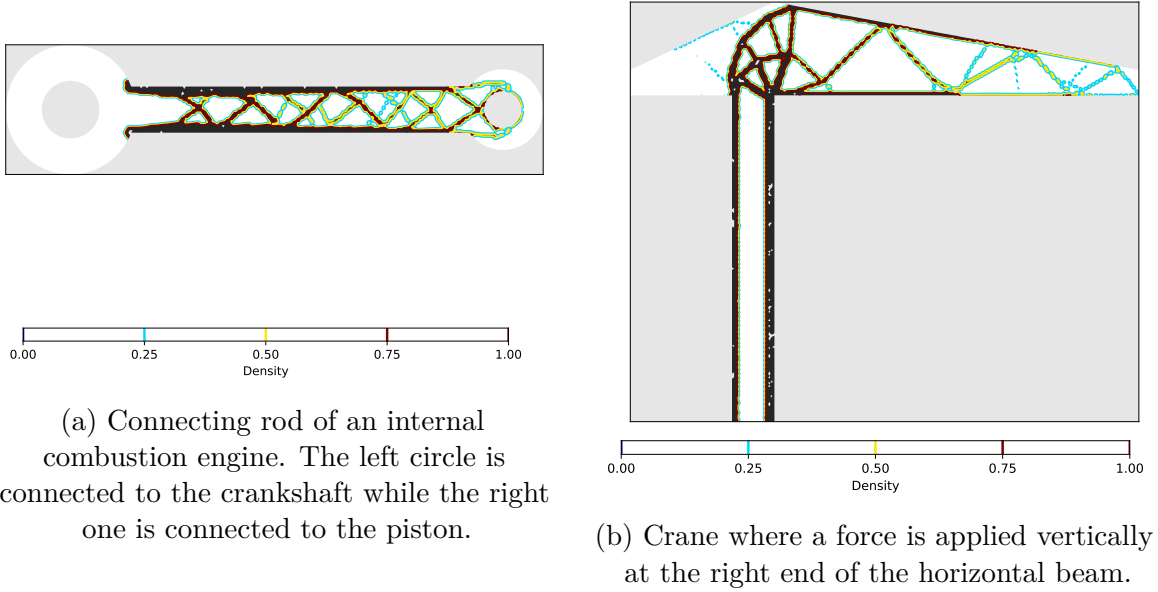


Figure 6.1: Examples of topology optimisation on different geometries.

The second method is the levelset approach, developed more extensively in [Chapter 5](#). It defines a levelset $\phi : \Omega \rightarrow \mathbb{R}$ that is positive to define matter and negative for the void. The isocurve $\{x \in \Omega : \phi(x) = 0\}$ defines the boundary of the design. The method advects the levelset in the direction of steepest descent of the objective function until the boundary stops moving. In that case, the velocity is zero and the gradient of the objective as well, meaning that an optimum has been found. This method creates sharp boundaries thanks to the levelset definition, but cannot nucleate.

To join those methods, the *eXtreme Mesh deformation* method (X-Mesh) was used. Given an interface (or boundary), the method moves the nodes of the mesh to capture it as well as possible. It has been shown in [Chapter 4](#) that using X-Mesh with only the density approach was not possible due to the incompatibility between the definition of the interface and the SIMP. Indeed, the SIMP drives the density field towards binary values, whereas X-Mesh requires some nodes to lie on the interface where $\rho = 0.5$. This is why the levelset method has been added to the algorithm: to define an intermediate levelset, based on the density design, and use X-Mesh to move the nodes based on that levelset.

This hybrid method has shown some promising results concerning the smoothing of the boundary. It improves the design by decreasing the length of the boundary, meaning that it is smoother and contains less oscillations coming from the definition of the mesh. Additionally, the volume constraint is taken into account in a Lagrangian formulation of the velocity, meaning that no more explicit constraint remains when working with the levelset. The results of this new method are interesting, but not yet perfect. Future works should improve the algorithm by defining the convergence criterion more rigorously, as the method does not

converge yet and only the number of iterations stops the algorithm. One could be interested in researching a criterion enforces the volume constraint and also verifies that an optimum has been found, using the norm of the gradient. Other criteria could include compliance variation, boundary displacement, or interface length. It also requires a better tuning of the hyperparameters, e.g. the method is very sensitive to the time step of the advection of the levelset.

Bibliography

- [1] Niels Aage et al. “Giga-Voxel Computational Morphogenesis for Structural Design”. In: *Nature* 550.7674 (Oct. 2017), pp. 84–86. ISSN: 1476-4687. DOI: [10.1038/nature23911](https://doi.org/10.1038/nature23911). URL: <https://www.nature.com/articles/nature23911> (visited on 05/15/2026).
- [2] Joe Alexandersen and Casper Schousboe Andreasen. “A Review of Topology Optimisation for Fluid-Based Problems”. In: *Fluids* 5.1 (2020). ISSN: 2311-5521. DOI: [10.3390/fluids5010029](https://doi.org/10.3390/fluids5010029).
- [3] Grégoire Allaire, François Jouve, and Anca-Maria Toader. “Structural Optimization Using Sensitivity Analysis and a Level-Set Method”. In: *Journal of Computational Physics* 194.1 (Feb. 10, 2004), pp. 363–393. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2003.09.032](https://doi.org/10.1016/j.jcp.2003.09.032). URL: <https://www.sciencedirect.com/science/article/pii/S002199910300487X> (visited on 05/15/2026).
- [4] *An eXtreme Mesh Deformation Method to Follow Sharp Physical Interfaces*. URL: <https://www.x-mesh.eu/> (visited on 05/15/2026).
- [5] Erik Andreassen et al. “Efficient Topology Optimization in MATLAB Using 88 Lines of Code”. In: *Structural and Multidisciplinary Optimization* 43.1 (Jan. 2011), pp. 1–16. ISSN: 1615-147X, 1615-1488. DOI: [10.1007/s00158-010-0594-7](https://doi.org/10.1007/s00158-010-0594-7). URL: <http://link.springer.com/10.1007/s00158-010-0594-7> (visited on 05/15/2026).
- [6] M. P. Bendsøe. “Optimal Shape Design as a Material Distribution Problem”. In: *Structural optimization* 1.4 (Dec. 1, 1989), pp. 193–202. ISSN: 1615-1488. DOI: [10.1007/BF01650949](https://doi.org/10.1007/BF01650949). URL: <https://doi.org/10.1007/BF01650949> (visited on 05/15/2026).
- [7] Martin P. Bendsøe and Ole Sigmund. *Topology Optimization: Theory, Methods and Applications*. Berlin, Heidelberg: Springer, 2004. ISBN: 978-3-642-07698-5 978-3-662-05086-6. DOI: [10.1007/978-3-662-05086-6](https://doi.org/10.1007/978-3-662-05086-6). URL: <http://link.springer.com/10.1007/978-3-662-05086-6> (visited on 05/15/2026).
- [8] Matteo Bruggi. “On an Alternative Approach to Stress Constraints Relaxation in Topology Optimization”. In: *Structural and Multidisciplinary Optimization* 36.2 (Aug. 1, 2008), pp. 125–141. ISSN: 1615-1488. DOI: [10.1007/s00158-007-0203-6](https://doi.org/10.1007/s00158-007-0203-6). URL: <https://doi.org/10.1007/s00158-007-0203-6> (visited on 05/15/2026).
- [9] *Bugatti Tourbillon - Page 2 - F1technical.Net*. URL: <https://www.f1technical.net/forum/viewtopic.php?t=31793&start=15> (visited on 05/18/2026).
- [10] Nishanth Gadiyar and Bingnan Wang. “Topology Optimization of Electric Machines: A Review”. In: *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2022 IEEE Energy Conversion Congress and Exposition (ECCE). Detroit, MI, USA: IEEE, Oct. 9, 2022, pp. 1–8. ISBN: 978-1-7281-9387-8. DOI: [10.1109/ECCE50734.2022](https://doi.org/10.1109/ECCE50734.2022).

9948073. URL: <https://ieeexplore.ieee.org/document/9948073/> (visited on 05/18/2026).
- [11] Alexandre T. R. Guibert et al. “Facilitating Multidisciplinary Collaboration through a Versatile Level-Set Topology Optimization Framework via COMSOL Multiphysics”. In: *Structural and Multidisciplinary Optimization* 67.9 (Sept. 12, 2024), p. 159. ISSN: 1615-1488. DOI: [10.1007/s00158-024-03877-w](https://doi.org/10.1007/s00158-024-03877-w). URL: <https://doi.org/10.1007/s00158-024-03877-w> (visited on 05/15/2026).
- [12] Xu Guo, Weisheng Zhang, and Wenliang Zhong. “Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework”. In: *Journal of Applied Mechanics* 81 (May 1, 2014), p. 081009. DOI: [10.1115/1.4027609](https://ui.adsabs.harvard.edu/abs/2014JAM...81h1009G). URL: <https://ui.adsabs.harvard.edu/abs/2014JAM...81h1009G> (visited on 05/15/2026).
- [13] Z. Hashin and S. Shtrikman. “A Variational Approach to the Theory of the Elastic Behaviour of Multiphase Materials”. In: *Journal of the Mechanics and Physics of Solids* 11.2 (Mar. 1, 1963), pp. 127–140. ISSN: 0022-5096. DOI: [10.1016/0022-5096\(63\)90060-7](https://www.sciencedirect.com/science/article/pii/002250966390060-7). URL: <https://www.sciencedirect.com/science/article/pii/002250966390060-7> (visited on 05/19/2026).
- [14] Van-Nam Hoang. “An Explicit Topology Optimization Method Using Moving Polygonal Morphable Voids (MPMVs)”. In: *Science and Technology Development Journal* 23 (June 27, 2020), pp. 536–540. DOI: [10.32508/stdj.v23i2.2067](https://doi.org/10.32508/stdj.v23i2.2067).
- [15] Issam El Khadiri et al. “Topology Optimization Methods for Additive Manufacturing: A Review”. In: *International Journal for Simulation and Multidisciplinary Design Optimization* 14 (2023), p. 12. ISSN: 1779-6288. DOI: [10.1051/smdo/2023015](https://www.ijsmdo.org/articles/smdo/abs/2023/01/smdo230032/smdo230032.html). URL: <https://www.ijsmdo.org/articles/smdo/abs/2023/01/smdo230032/smdo230032.html> (visited on 05/15/2026).
- [16] U. Kirsch. “A Unified Reanalysis Approach for Structural Analysis, Design, and Optimization”. In: *Structural and Multidisciplinary Optimization* 25.2 (July 1, 2003), pp. 67–85. ISSN: 1615-1488. DOI: [10.1007/s00158-002-0269-0](https://doi.org/10.1007/s00158-002-0269-0). URL: <https://doi.org/10.1007/s00158-002-0269-0> (visited on 05/15/2026).
- [17] B. S. Lazarov and O. Sigmund. “Filters in Topology Optimization Based on Helmholtz-type Differential Equations”. In: *International Journal for Numerical Methods in Engineering* 86.6 (2011), pp. 765–781. ISSN: 1097-0207. DOI: [10.1002/nme.3072](https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3072). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3072> (visited on 05/15/2026).
- [18] Chau Le et al. “Stress-Based Topology Optimization for Continua”. In: *Structural and Multidisciplinary Optimization* 41.4 (Apr. 1, 2010), pp. 605–620. ISSN: 1615-1488. DOI: [10.1007/s00158-009-0440-y](https://doi.org/10.1007/s00158-009-0440-y). URL: <https://doi.org/10.1007/s00158-009-0440-y> (visited on 05/15/2026).
- [19] Nicolas Moës et al. “The eXtreme Mesh Deformation Approach (X-MESH) for the Stefan Phase Change Model”. In: *Journal of Computational Physics* 477 (Mar. 15,

- 2023), p. 111878. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2022.111878](https://doi.org/10.1016/j.jcp.2022.111878). URL: <https://www.sciencedirect.com/science/article/pii/S002199912200941X> (visited on 05/15/2026).
- [20] Andreas Neofytou et al. “Level Set Topology Optimization with Sparse Automatic Differentiation”. In: *Structural and Multidisciplinary Optimization* 67.10 (Oct. 14, 2024), p. 178. ISSN: 1615-1488. DOI: [10.1007/s00158-024-03894-9](https://doi.org/10.1007/s00158-024-03894-9). URL: <https://doi.org/10.1007/s00158-024-03894-9> (visited on 05/15/2026).
- [21] Zhenguo Nie et al. *TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain*. Mar. 11, 2020. DOI: [10.48550/arXiv.2003.04685](https://doi.org/10.48550/arXiv.2003.04685). arXiv: [2003.04685](https://arxiv.org/abs/2003.04685) [cs.CE]. URL: <http://arxiv.org/abs/2003.04685> (visited on 05/15/2026). Pre-published.
- [22] O. Sigmund. “A 99 Line Topology Optimization Code Written in Matlab”. In: *Structural and Multidisciplinary Optimization* 21.2 (Apr. 1, 2001), pp. 120–127. ISSN: 1615-1488. DOI: [10.1007/s001580050176](https://doi.org/10.1007/s001580050176). URL: <https://doi.org/10.1007/s001580050176> (visited on 05/15/2026).
- [23] J. Sokolowski and A. Zochowski. “On the Topological Derivative in Shape Optimization”. In: *SIAM Journal on Control and Optimization* (). URL: <https://epubs.siam.org/doi/10.1137/S0363012997323230> (visited on 05/15/2026).
- [24] Ivan Sosnovik and Ivan Oseledets. *Neural Networks for Topology Optimization*. Sept. 27, 2017. DOI: [10.48550/arXiv.1709.09578](https://doi.org/10.48550/arXiv.1709.09578). arXiv: [1709.09578](https://arxiv.org/abs/1709.09578) [cs.LG]. URL: <http://arxiv.org/abs/1709.09578> (visited on 05/15/2026). Pre-published.
- [25] Krister Svanberg. “The Method of Moving Asymptotes—a New Method for Structural Optimization”. In: *International Journal for Numerical Methods in Engineering* 24.2 (Feb. 1987), pp. 359–373. ISSN: 0029-5981, 1097-0207. DOI: [10.1002/nme.1620240207](https://doi.org/10.1002/nme.1620240207). URL: <https://onlinelibrary.wiley.com/doi/10.1002/nme.1620240207> (visited on 05/15/2026).
- [26] Alireza Tabarraei and Saquib Ahmad Bhuiyan. *Graph Neural Network-Based Topology Optimization for Self-Supporting Structures in Additive Manufacturing*. Aug. 26, 2025. DOI: [10.48550/arXiv.2508.19169](https://doi.org/10.48550/arXiv.2508.19169). arXiv: [2508.19169](https://arxiv.org/abs/2508.19169) [cs.CE]. URL: <http://arxiv.org/abs/2508.19169> (visited on 05/15/2026). Pre-published.
- [27] “ ϵ -Relaxed Approach in Structural Topology Optimization | Request PDF”. In: *ResearchGate* (). DOI: [10.1007/BF01197454](https://doi.org/10.1007/BF01197454). URL: https://www.researchgate.net/publication/227098981_epsilon-Relaxed_approach_in_structural_topology_optimization (visited on 05/15/2026).
- [28] *TOP Webinar Tutorials: Density Methods 1 - Introduction and Basics*. Jan. 22, 2026. URL: <https://www.youtube.com/watch?v=DWo2v7QLH2M> (visited on 05/19/2026).
- [29] *Topology Optimization Software & Modeling Tools | SOLIDWORKS*. URL: <https://www.solidworks.com/media/topology-optimization> (visited on 05/18/2026).
- [30] N. P. vanDijk et al. “Level-Set Methods for Structural Topology Optimization: A Review”. In: *Structural and Multidisciplinary Optimization* 48.3 (Sept. 1, 2013), pp. 437–

472. ISSN: 1615-1488. DOI: [10.1007/s00158-013-0912-y](https://doi.org/10.1007/s00158-013-0912-y). URL: <https://doi.org/10.1007/s00158-013-0912-y> (visited on 05/20/2026).
- [31] Cunfu Wang and Xiaoping Qian. “Heaviside Projection–Based Aggregation in Stress-constrained Topology Optimization”. In: *International Journal for Numerical Methods in Engineering* 115.7 (Aug. 17, 2018), pp. 849–871. ISSN: 0029-5981, 1097-0207. DOI: [10.1002/nme.5828](https://onlinelibrary.wiley.com/doi/10.1002/nme.5828). URL: <https://onlinelibrary.wiley.com/doi/10.1002/nme.5828> (visited on 05/15/2026).
- [32] Fengwen Wang, Boyan Stefanov Lazarov, and Ole Sigmund. “On Projection Methods, Convergence and Robust Formulations in Topology Optimization”. In: *Structural and Multidisciplinary Optimization* 43.6 (June 1, 2011), pp. 767–784. ISSN: 1615-1488. DOI: [10.1007/s00158-010-0602-y](https://doi.org/10.1007/s00158-010-0602-y). URL: <https://doi.org/10.1007/s00158-010-0602-y> (visited on 05/15/2026).
- [33] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. “A Level Set Method for Structural Topology Optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 192.1 (Jan. 3, 2003), pp. 227–246. ISSN: 0045-7825. DOI: [10.1016/S0045-7825\(02\)00559-5](https://www.sciencedirect.com/science/article/pii/S004578250200559-5). URL: <https://www.sciencedirect.com/science/article/pii/S004578250200559-5> (visited on 05/15/2026).
- [34] *Web Page of Charles Dapogny*. URL: <https://dapogny.org/coursoptim.html> (visited on 05/19/2026).
- [35] Qi Xia, Tielin Shi, and Liang Xia. “Topology Optimization for Heat Conduction by Combining Level Set Method and BESO Method”. In: *International Journal of Heat and Mass Transfer* 127 (Dec. 1, 2018), pp. 200–209. ISSN: 0017-9310. DOI: [10.1016/j.ijheatmasstransfer.2018.08.036](https://www.sciencedirect.com/science/article/pii/S0017931018320052). URL: <https://www.sciencedirect.com/science/article/pii/S0017931018320052> (visited on 05/15/2026).
- [36] Kaiqing Zhang, Gengdong Cheng, and Liang Xu. “Topology Optimization Considering Overhang Constraint in Additive Manufacturing”. In: *Computers & Structures* 212 (Feb. 1, 2019), pp. 86–100. ISSN: 0045-7949. DOI: [10.1016/j.compstruc.2018.10.011](https://www.sciencedirect.com/science/article/pii/S0045794918303997). URL: <https://www.sciencedirect.com/science/article/pii/S0045794918303997> (visited on 05/15/2026).
- [37] Xiaopeng Zhang et al. “Microscopic Stress-Constrained Two-Scale Topology Optimisation for Additive Manufacturing”. In: *Virtual and Physical Prototyping* 20.1 (Jan. 20, 2025), e2450276. ISSN: 1745-2759. DOI: [10.1080/17452759.2025.2450276](https://doi.org/10.1080/17452759.2025.2450276). URL: <https://doi.org/10.1080/17452759.2025.2450276> (visited on 05/15/2026).
- [38] M. Zhou and G. I. N. Rozvany. “The COC Algorithm, Part II: Topological, Geometrical and Generalized Shape Optimization”. In: *Computer Methods in Applied Mechanics and Engineering*. Second World Congress on Computational Mechanics 89.1 (Aug. 1, 1991), pp. 309–336. ISSN: 0045-7825. DOI: [10.1016/0045-7825\(91\)90046-9](https://www.sciencedirect.com/science/article/pii/0045782591900469). URL: <https://www.sciencedirect.com/science/article/pii/0045782591900469> (visited on 05/15/2026).

-
- [39] Jun Zou and Xiaoyu Xia. “Topology Optimization for Additive Manufacturing with Strength Constraints Considering Anisotropy”. In: *Journal of Computational Design and Engineering* 10.2 (Apr. 1, 2023), pp. 892–904. ISSN: 2288-5048. DOI: [10.1093/jcde/qwad028](https://doi.org/10.1093/jcde/qwad028). URL: <https://doi.org/10.1093/jcde/qwad028> (visited on 05/15/2026).

Declaration

I hereby declare the use of Artificial Intelligence tools such as Claude Opus and ChatGPT-5.5. They were used to reformulate selected sentences and generate some figures. All AI-assisted suggestions were reviewed, edited and validated by me, and I take full responsibility for every aspect of this work.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl