TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Topological Optimization:
Using Level Set Algorithms and Newton's Method

MASTERARBEIT

zur Erlangung des akademischen Grades
Master of Science
(bei Lehramt: Magister der Naturwissenschaften)

im Rahmen des Studiums
Interdisiplinary Mathematics

eingereicht von
Amr Khaled Nagy Mohamed
Matrikelnummer 12243288

ausgeführt am Institut für Analysis und Scientific Computing
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuung:
Betreuer: Prof. Kevin Sturm

Wien, 29.08.2024

(Unterschrift Verfasser) (Unterschrift Betreuer)

TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

MERKBLATT FÜR VERFASSER/INNEN VON HOCHSCHULSCHRIFTEN

(Diplom-/Masterarbeiten und Dissertationen)

Der/die unterzeichnete Verfasser/Verfasserin der nachstehend angeführten Hochschulschrift:

_____

nimmt im Sinne der §§ 42 und 42a Urheberrechtsgesetz 1936 in der jeweils gültigen Fassung zur Kenntnis:

1. Die Universitätsbibliothek darf, solange die Hochschulschrift veröffentlicht, aber nicht erschienen oder vergriffen ist, ohne Zustimmung des Verfassers/der Verfasserin für eigenen Gebrauch einzelne Vervielfältigungsstücke herstellen.
2. Ist die Hochschulschrift vergriffen (d.h. durch Druck oder eine Vervielfältigungsmethode herausgegeben und ohne Zutun des Verfassers/der Verfasserin nicht mehr im Handel erhältlich), darf die Universitätsbibliothek ohne Zustimmung des Verfassers/der Verfasserin für eigenen Gebrauch und Teile der Hochschulschrift in Verkehr bringen.
3. Die Universitätsleitung hat in der Richtlinie des Vizerektors für Lehre die elektronische Abgabepflicht von Hochschulschriften beschlossen, zusätzlich zum gedruckten Exemplar ein elektronisches Exemplar abzugeben.
4. Die Hochschulschrift muss selbstständig verfasst sein, andere als die angegebenen Quellen und Hilfsmittel dürfen nicht benutzt werden.

Ich versichere, dass ich diese Hochschulschrift bisher weder im In- oder Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Datum: Unterschrift:

Dekanat: Ablage im Prüfungsamt

Abstract

The thesis investigates numerical and theoretical approaches to topology optimization with applications to for instance linear elasticity. The primary objective is to explore different algorithms that can find a local minimum and compare their efficiency. The study begins with an exploration of topological derivatives and their integration with the level set method. The first-order topological derivative is employed in gradient descent and projected gradient descent algorithms, while the second-order derivative, as well as the novel topological state derivative, are utilized in the formulation of Newton and Lagrange-Newton algorithms. Finally several applications are examined and the performance as well as the shortcomings of each algorithm are evaluated.

Contents

Chapter 1

Introduction

Shape and topology optimization have gained significant attention due to their broad range of applications across various fields. Traditional approaches often focused solely on shape optimization, which limited their ability to accommodate topological changes within the design space [3]. Over time, more advanced methods have been developed to overcome these limitations, enabling topological transformations. Among these methods is the level set method. Originally developed for numerically tracking fronts and free boundaries, it stands out as a powerful tool, offering flexibility in handling complex geometries and allowing for continuous topological changes during the optimization process.

Applying the level set method in topological optimization is built upon the utilization of foundational concepts like the topological derivative, which provide a robust sensitivity analysis tool that enables precise identification of regions where material should be added or removed. By leveraging this sensitivity information, optimization algorithms can be formulated to follow well-established approaches, such as gradient descent and projected gradient descent, ensuring both accuracy and efficiency in navigating the design space [5][3].

Expanding the topological derivative to higher-order derivatives, such as the second-order topological derivative, or incorporating the topological state derivative [4], opens the door to more sophisticated optimization approaches, including Newton's method and the Lagrange-Newton method. These methods, known for their faster convergence rates compared to first-order methods like gradient descent [1], offer significant advantages, particularly in large-scale problems where efficiency and accuracy are important.

The complexity of calculating higher-order derivatives, is due to the fact that they require solving additional partial differential equations (PDEs),which presents a challenge for the Newton and Lagrange-Newton methods. This step can be particularly resource-intensive, especially for large-scale problems.

Chapter 2

Topological Derivative and Level Set

2.1 Topological Derivative

The topological derivative has been devised to measure the sensitivity of a domain dependent criterion $J(\Omega)$ with respect to the size of a small hole ω_ε of parameter ε created around a given point $x_0 \in \Omega$ of the domain [3]. More precisely, it is defined by the following asymptotic expansion:

$$J(\Omega \setminus \omega_\varepsilon(x_0)) - J(\Omega) = g(\varepsilon)\, DJ(\Omega)(x_0) + o(g(\varepsilon)), \tag{2.1}$$

where $g : \mathbb{R}^+ \to \mathbb{R}^+$ is a smooth function that describes the measure of the hole, with $g(\varepsilon) \to 0$ as $\varepsilon \to 0$, where $\mathbb{R}^+$ is the space of positive real numbers.

The criterion $J(\Omega)$, which quantifies the sensitivity, can be understood through the framework of shape functionals. Let $D \subset \mathbb{R}^d$ ($d = 2$ or 3) be a smooth domain which will henceforth be referred to as the hold-all domain, meaning that it will contain in it all possible shapes Ω, where $\Omega \subset D$ is assumed to be a open and bounded set compactly contained in D.

Definition 2.1 (Shape Functional)**.** *Given a set $D \subset \mathbb{R}^d$, let $P(D)$ be the powerset of D, i.e. $P(D) := \{\Omega \subset \mathbb{R}^d : \Omega \subset D\ \}$. A mapping $J : \Xi \subset P(D) \to \mathbb{R}$ assigning to each shape $\Omega \in \Xi$ a number $J(\Omega) \in \mathbb{R}$ is called shape functional. The elements in Ξ are called shapes. The set $\Xi := \{\Omega \subset D : \Omega$ measurable$\}$ is also referred to as admissible set.*

Let $\omega \subset \mathbb{R}^d$ be a connected set with $0 \in \omega$. The perturbation of the set Ω at $x_0 \in D \setminus \partial\Omega$ is defined by:

$$\Omega_\varepsilon := \Omega_\varepsilon(x_0) := \begin{cases} \Omega \cup \omega_\varepsilon & \text{for } x_0 \in D \setminus \overline{\Omega}, \\ \Omega \setminus \omega_\varepsilon & \text{for } x_0 \in \Omega, \end{cases} \tag{2.2}$$

where the hole at x_0 is represented by the set:

$$\omega_\varepsilon(x_0) := x_0 + \varepsilon\omega := \{x_0 + \varepsilon x : x \in \omega\}. \tag{2.3}$$

Henceforth, it is assumed that ω is a ball of radius one centered at the origin, namely $\omega = B_1(0) = \{x \in \mathbb{R}^d : \|x\| < 1\} = \{x \in \mathbb{R}^d : \|x - x_0\| < \varepsilon\}$, where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^d$. Hence we have $\omega_\varepsilon = B_\varepsilon(x_0)$.

Definition 2.2 (Topological Derivative)**.** *For a shape functional $J(\Omega)$ and $\Omega \subset D$ as described above, then the topological derivative at the point $x_0 \in D \setminus \partial\Omega$ according to the asymptotic expansion* (2.1) *with $g(\varepsilon) := |\omega_\varepsilon|$ is given by:*

$$DJ(\Omega)(x_0) := \lim_{\varepsilon \to 0} \frac{J(\Omega_\varepsilon(x_0)) - J(\Omega)}{|\omega_\varepsilon|}.$$

Here $|\omega_\varepsilon|$ denotes the d-volume of the set ω_ε.

Example 2.1. *For the shape functional:*

$$J(\Omega) = \int_\Omega f \, dx, \tag{2.4}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a continuous function, assumed to be in $C(\mathbb{R}^d)$ for simplicity, meaning it continuous on $\mathbb{R}^d$ and hence uniformly continuous on any bounded subset, and $\Omega \subset D \subset \mathbb{R}^d$ is a bounded and open domain of finite measure, the topological derivative for $x_0 \in \Omega$ and $\Omega_\varepsilon := \Omega \setminus \omega_\varepsilon$ can be calculated as follows:

$$J(\Omega_\varepsilon) - J(\Omega) = \int_{\Omega_\varepsilon} f \, dx - \int_\Omega f \, dx = -\int_{\omega_\varepsilon} f \, dx \tag{2.5}$$

$$= -\varepsilon^d \int_\omega f(x_0 + \varepsilon x) \, dx, \tag{2.6}$$

where in the last step the change of variables $x \to \varepsilon\, x + x_0$ is invoked, then division by $|\omega_\varepsilon| = |\omega|\varepsilon^d$ yields:

$$\frac{J(\Omega_\varepsilon) - J(\Omega)}{|\omega_\varepsilon|} = \frac{-1}{|\omega|} \int_\omega f(x_0 + \varepsilon x) \, dx. \tag{2.7}$$

Taking the limit as $\varepsilon \to 0$ and by the uniform continuity of f one gets:

$$DJ(\Omega)(x_0) = -f(x_0). \tag{2.8}$$

A similar computation shows for $x_0 \in D \setminus \overline{\Omega}$ and $\Omega_\varepsilon := \Omega \cup \omega_\varepsilon$

$$DJ(\Omega)(x_0) = f(x_0). \tag{2.9}$$

Summarizing we obtain:

$$DJ(\Omega)(x_0) = sgn_\Omega(x_0) f(x_0), \tag{2.10}$$

where,

$$sgn_\Omega(x_0) = \begin{cases} -1 & \text{for } x_0 \in \Omega, \\ 1 & \text{for } x_0 \in D \setminus \overline{\Omega}. \end{cases} \tag{2.11}$$

In the context of topological optimization, the aim is to find a minimizer for the criterion, i.e. finding a domain Ω^* solving:

$$\min_{\Omega \subset \Xi} J(\Omega), \tag{2.12}$$

which requires defining optimality conditions using the topological derivative. Moreover, the existence of the solution is not dealt with as it is assumed that there exists a minimizer for (2.12).

Definition 2.3. *We say $\Omega^* \in \Xi$ is a stationary point of J if:*

$$DJ(\Omega^*)(x_0) \geq 0 \quad \textit{for all } x_0 \in D \setminus \partial\Omega^*.$$

Definition 2.4. *We say Ω^* is a minimizer of J with respect to topological perturbations if there is $\varepsilon^* > 0$ for each $x_0 \in D \setminus \partial\Omega^*$, such that*

$$J(\Omega_\varepsilon(x_0)) \geq J(\Omega^*) \quad \textit{for all } 0 \leq \varepsilon \leq \varepsilon^*.$$

Definition 2.5. *Let $J : \Xi \subset \mathcal{P}(D) \to \mathbb{R}$ be a shape functional. The second order topological derivative [6] at $(x_0, y_0) \in D \setminus \partial\Omega$ for $x_0 \neq y_0$ is defined by:*

$$D^2J(\Omega)(x_0, y_0) := \begin{cases} \lim_{\varepsilon\to 0} \frac{DJ(\Omega\setminus B_\varepsilon(y_0))(x_0) - DJ(\Omega)(x_0)}{|\omega_\varepsilon(y_0)|} & \textit{for } y_0 \in \Omega, \\ \lim_{\varepsilon\to 0} \frac{DJ(\Omega\cup B_\varepsilon(y_0))(x_0) - DJ(\Omega)(x_0)}{|\omega_\varepsilon(y_0)|} & \textit{for } y_0 \in D \setminus \overline{\Omega}. \end{cases}$$

For if $x_0 = y_0$, namely the "diagonal" elements of the second order topological derivative are given by:

$$D^2J(\Omega)(x_0, x_0) := \lim_{\substack{x\to x_0 \\ x\neq x_0}} D^2J(\Omega)(x_0, x).$$

- The shape functional J is said to be topologically differentiable [6] at Ω, if $DJ(\Omega)(x)$ exists for all x in $D \setminus \partial\Omega$ and said to be twice topologically differentiable at Ω, if $D^2J(\Omega)(x, y)$ exists for all x, y in $D \setminus \partial\Omega$.
- J is said to be topologically differentiable if it's topologically differentiable for all domains Ω in Ξ.

2.2 PDE Constrained Topological Optimization: Source Control Problem

In this section, a topological optimization problem constrained by a partial differential equation (PDE) is considered. The objective is to modify the topology and shape of the domain $\Omega \subset D$ such that a certain state u_Ω closely matches a predefined reference state u_{ref}. The optimization criterion, or shape functional, is formulated as the distance between the two states:

$$J(\Omega) = \int_D (u_\Omega - u_{\text{ref}})^2 \, dx, \tag{2.13}$$

This functional measures the L^2-norm of the difference between u_Ω and u_{ref} over the domain D. The state $u_\Omega \in H_0^1(D)$ is required to satisfy the following PDE(in weak form):

$$\int_D \nabla u_\Omega \cdot \nabla v + \alpha_\Omega uv \, dx = \int_D f_\Omega v \, dx \quad \forall v \in H_0^1(D), \tag{2.14}$$

where α_Ω and f_Ω are piece wise constant functions defined below. Here, $H_0^1(D)$ is the Sobolev space of functions that have square-integrable first derivatives and vanish on the boundary of D. This space ensures the appropriate regularity for weak solutions of the PDE.

Definition 2.6 (Sobolev Spaces [10]). *For $k \in \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of non negative integers, and $D \subset \mathbb{R}^d$ bounded set, we define the Sobolev spaces via:*

$$H^k(D) = \{u \in L^1_{loc}(D) : \|u\|_{H^k(D)} < \infty\},$$

where the Sobolev norms are given as: (here D^α is the weak derivative)

$$\|u\|_{H^k(D)} := \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|^2_{L^2(D)} \right)^{1/2},$$

and the space $W_0^{1,q}(D)$ with $1 \leq q < \infty$ is defined as:

$$W_0^{1,q}(D) = \{u \in W^{1,q}(D) \mid u|_{\partial D} = 0\},$$

where $W^{1,q}(D) = \{u \in L^q(D) \mid \nabla u \in L^q(D)\}$ is the space of functions with integrable derivatives up to order 1 and the functions themselves are in $L^q(D)$,and its dual space is $W^{1,q'}(D)$, where $1 \leq q' \leq \infty$ with $\frac{1}{q} + \frac{1}{q'} = 1$. Note that $H_0^1(D) = W_0^{1,2}(D)$.

Definition of the Coefficients

The source term $f_\Omega(x)$ and the coefficient $\alpha_\Omega(x)$ are defined piece-wise over the domain D, that is:

$$f_\Omega(x) := f_1\, \chi_\Omega(x) + f_2\, \chi_{D\setminus\Omega}(x) \quad \text{for} \quad f_1, f_2 \in \mathbb{R}, \tag{2.15}$$

$$\alpha_\Omega(x) := \alpha_1\, \chi_\Omega(x) + \alpha_2\, \chi_{D\setminus\Omega}(x) \quad \text{for} \quad \alpha_1, \alpha_2 \in \mathbb{R}, \tag{2.16}$$

where $\chi_A(x)$ denotes the characteristic function associated with a set $A \subset \mathbb{R}^d$: $\chi_A(x)$ is equal to 1 if $x \in A$ and 0 otherwise. These definitions allow for different material properties or source terms within and outside the region Ω.

The PDE constraint in (2.14) can be expressed as:

$$\langle E_\Omega(u_\Omega), \phi \rangle = 0 \qquad \forall \phi \in X(D), \tag{2.17}$$

where u_Ω is the solution to (2.14) and $X(D)$ is defined as the space of functions mapping $D \to \mathbb{R}$, $X'(D)$ is the dual space of $X(D)$, consisting of all continuous linear functionals on $X(D)$, and $E_\Omega : X(D) \to X'(D)$ is a possibly non-linear operator.

Similar to the unconstrained case as in example 2.1, we want to calculate the topological derivative of the criterion $J(\Omega)$. For simplicity, we henceforth assume that $\omega = B_1(0)$. Note that $|\omega_\varepsilon| = \varepsilon^d |B_1(0)|$.

Let $u_\varepsilon \in H_0^1(D)$ be the solution to the weak formulation of the weak form of the Laplace equation (2.14) with $\Omega := \Omega_\varepsilon$, that is, $u_\varepsilon := u_{\Omega_\varepsilon}$ solves:

$$\int_D \nabla u_\varepsilon \cdot \nabla v + \alpha_\varepsilon u_\varepsilon v \, dx = \int_D f_\varepsilon v \, dx \quad \forall v \in H_0^1(D), \tag{2.18}$$

with $f_\varepsilon := f_1 \chi_{\Omega_\varepsilon} + f_2 \chi_{D\setminus\Omega_\varepsilon}$ and $\alpha_\varepsilon := \alpha_1 \chi_{\Omega_\varepsilon} + \alpha_2 \chi_{D\setminus\Omega_\varepsilon}$.

Definition 2.7 (Topological State Derivative)**.** *For $\varepsilon > 0$ and a hole $\omega_\varepsilon(x_0) = B_\varepsilon(x_0)$ one can define:*

$$U_\varepsilon := \frac{u_\varepsilon - u_0}{|\omega_\varepsilon|}.$$

Then the limit:

$$\dot{u}_{x_0} := \lim_{\varepsilon \to 0} U_\varepsilon,$$

is called the topological state derivative of $u_\Omega = u_0$ at Ω at the point x_0.

Definition 2.8 (Shape to State Operator)**.** *The mapping $S : \Xi \subset \mathcal{P}(D) \to X(D)$ with $S(\Omega) = u_\Omega$ where u_Ω solves (2.17) and the derivative of the shape to state operator S in direction (ω, x_0) can be as: $S'(\Omega)(\omega, x_0) := \lim_{\varepsilon \to 0} U_\varepsilon = \dot{u}_{x_0}$.*

Using the previous definition, the topological derivative of a function $\mathscr{J}(\Omega) = J(u_\Omega)$ with $\mathscr{J} : X(D) \to \mathbb{R}$ can be calculated by the chain rule, that is:

$$D\mathscr{J}(\Omega)(x_0) = J'(S(\Omega))(S'(\Omega)(\omega, x_0)) \qquad \forall x_0 \in D \setminus \partial\Omega. \tag{2.19}$$

Lemma 2.1. *For U_ε as defined in (2.7), it is proven in [4, Theorem 2.1] that it converges weakly the Sobolev space $W_0^{1,q}(D)$ with $1 \leq q < \frac{d}{d-1}$ to the unique $\dot{u} \in W_0^{1,q}(D)$ satisfying:*

$$\begin{aligned}\int_D \nabla \dot{u}_{x_0} \cdot \nabla v \, dx + \int_D \alpha_\Omega \dot{u}_{x_0} v \, dx = &- sgn_\Omega(x_0)(f_1 - f_2)v(x_0) \\ &+ sgn_\Omega(x_0)(\alpha_1 - \alpha_2)u(x_0)\, v(x_0).\end{aligned}$$

for all $v \in W_0^{1,q'}(D)$.

Proof. Let $x_0 \in \Omega$ and let Ω_ε be defined as in (2.2). Subtracting equation (2.18) at Ω from the same equation at Ω_ε yields:

$$\int_D \nabla(u_\varepsilon - u_\Omega) \cdot \nabla v \, dx + (\alpha_\varepsilon \, u_\varepsilon - \alpha_\Omega \, u_\Omega) v \, dx = \int_D (f_\varepsilon - f_\Omega) \, v \, dx \qquad \forall v \in W_0^{1,q'}(D), \tag{2.20}$$

using the definitions of coefficients in (2.15) and (2.16) one gets:

$$\begin{aligned}\int_D \nabla(u_\varepsilon - u_0) \cdot \nabla v \, dx + \int_D \alpha_1(\chi_{\Omega_\varepsilon} u_\varepsilon - \chi_\Omega u_\Omega) v \, dx + \int_D \alpha_2(\chi_{\Omega_\varepsilon^c} u_\varepsilon - \chi_{\Omega^c} u_\Omega) v \, dx = \\ \int_D f_1(\chi_{\Omega_\varepsilon} - \chi_\Omega) v \, dx + \int_D f_2(\chi_{\Omega_\varepsilon^c} - \chi_{\Omega^c}) v \, dx,\end{aligned} \tag{2.21}$$

adding constructive zeros $\pm\alpha_1 \chi_\Omega u_\varepsilon$ and $\pm\alpha_2 \chi_{\Omega^c} u_\varepsilon$ yields:

$$\begin{aligned}\int_D \nabla(u_\varepsilon - u_0) \cdot \nabla v \, dx + \int_D \alpha_\Omega(u_\varepsilon - u_\Omega) v \, dx + \int_D (\alpha_\varepsilon - \alpha_\Omega) u_\varepsilon v, dx = \\ \int_D f_1(\chi_{\Omega_\varepsilon} - \chi_\Omega) v \, dx + \int_D f_2(\chi_{\Omega_\varepsilon^c} - \chi_{\Omega^c}) v \, dx,\end{aligned} \tag{2.22}$$

since $\Omega_\varepsilon = \Omega \setminus \omega_\varepsilon$ then the integrals with the term that has the difference in characteristic functions, will yield and integral over the ball ω_ε, that is:

$$\int_D \nabla(u_\varepsilon - u_0) \cdot \nabla v \, dx + \int_D \alpha_\Omega(u_\varepsilon - u_\Omega) v \, dx = -(\alpha_2 - \alpha_1) \int_{\omega_\varepsilon} u_\varepsilon v, dx + (f_2 - f_1) \int_{\omega_\varepsilon} v \, dx, \tag{2.23}$$

dividing by the measure of the ball $|\omega_\varepsilon|$ and using Definition 2.7 results in the following:

$$\begin{aligned}\int_D \nabla U_\varepsilon \cdot \nabla v \, dx + \int_D \alpha_\Omega U_\varepsilon v \, dx =& \frac{-1}{|\omega_\varepsilon|} \int_{\omega_\varepsilon} (f_1 - f_2) v \, dx \\ &+ \frac{1}{|\omega_\varepsilon|} \int_{\omega_\varepsilon} (\alpha_1 - \alpha_2) uv \, dx\end{aligned} \tag{2.24}$$

for all $v \in W_0^{1,q'}(D)$. A similar calculation shows that for $x_0 \in D \setminus \overline{\Omega}$ we have

$$\begin{aligned}\int_D \nabla U_\varepsilon \cdot \nabla v \, dx + \int_D \alpha_\Omega U_\varepsilon v \, dx =& \frac{1}{|\omega_\varepsilon|} \int_{\omega_\varepsilon} (f_1 - f_2) v \, dx \\ &- \frac{1}{|\omega_\varepsilon|} \int_{\omega_\varepsilon} (\alpha_1 - \alpha_2) uv \, dx\end{aligned} \tag{2.25}$$

for all $v \in W_0^{1,q'}(D)$. So taking the limit of the previous equations as $\varepsilon \to 0$ yields:

$$\begin{aligned}\int_D \nabla \dot{u}_{x_0} \cdot \nabla v \, dx + \int_D \alpha_\Omega \dot{u}_{x_0} v \, dx =& \operatorname{sgn}_\Omega(x_0)(f_1 - f_2) v(x_0) \\ &- \operatorname{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2) u(x_0) v(x_0)\end{aligned} \tag{2.26}$$

for all $v \in W_0^{1,q'}(D)$. This concludes the proof. □

Theorem 2.1. *The topological derivative of the criterion:*

$$J(\Omega) = \int_D (u_\Omega - u_{ref})^2 \, dx, \tag{2.27}$$

subject to the PDE constraint, $u_\Omega \in H_0^1(D)$ solving:

$$\int_D \nabla u_\Omega \cdot \nabla v + \alpha_\Omega uv \, dx = \int_D f_\Omega v \, dx \quad \text{for all } v \in H_0^1(D), \tag{2.28}$$

is given by:

$$DJ(\Omega)(x_0) = \int_D 2\dot{u}_{x_0}(u_\Omega - u_{ref}) \, dx, \tag{2.29}$$

where $\dot{u}_{x_0} \in W_0^{1,q}(D)$ solves:

$$-\Delta \dot{u}_{x_0} + \alpha_\Omega \dot{u}_{x_0} = sgn_\Omega(x_0)[(f_1 - f_2) - (\alpha_1 - \alpha_2) u_\Omega(x_0)]\delta_{x_0}, \tag{2.30}$$

with

$$\delta_{x_0}(v) = v(x_0) \qquad \forall v \in C(D).$$

Moreover, we have:

$$\mathcal{D}J(\Omega)(x_0) = sgn_\Omega(x_0)((\alpha_1 - \alpha_2)\, u_\Omega(x_0) - (f_1 - f_2)) p_\Omega(x_0), \tag{2.31}$$

where $p_\Omega \in H_0^1(D)$ is the adjoint variable solving:

$$-\Delta p_\Omega + \alpha_\Omega \, p = -2(u_\Omega - u_{ref}) \quad \text{in } D. \tag{2.32}$$

Proof. Considering the difference in the criterion: .

$$J(\Omega_\varepsilon) - J(\Omega) = \int_D (u_\varepsilon - u_0)(u_\varepsilon + u_0 - 2u_{\text{ref}})\,dx, \tag{2.33}$$

dividing this difference by the measure of the hole $|\omega_\varepsilon|$ and using Definition 2.7 one gets:

$$\frac{J(\Omega_\varepsilon) - J(\Omega)}{|\omega_\varepsilon|} = \int_D U_\varepsilon(u_\varepsilon + u_0 - 2u_{\text{ref}})\,dx. \tag{2.34}$$

Taking the limit as $\varepsilon \to 0$ and using Definitions 2.7 and 2.2 and Lemma 2.1 yields:

$$DJ(\Omega)(x_0) = 2\int_D \dot{u}_{x_0}(u_0 - u_{\text{ref}})\,dx. \tag{2.35}$$

The weak form of equation (2.32) reads:

$$\int_D \nabla p_\Omega \cdot \nabla v\,dx + \int_D \alpha_\Omega p_\Omega v\,dx = -\int_D 2(u_\Omega - u_{\text{ref}})v\,dx \quad \forall v \in H_0^1(D), \tag{2.36}$$

and that of equation (2.30) reads:

$$\int_D \nabla \dot{u}_{x_0} \cdot \nabla v\,dx + \int_D \alpha_\Omega \dot{u}_{x_0} v\,dx = \text{sgn}_\Omega(x_0)(-(\alpha_1 - \alpha_2)u_\Omega(x_0) + (f_1 - f_2))v(x_0) \tag{2.37}$$
$$\text{for all } v \in W_0^{1,q'}(D).$$

By testing with $v = \dot{u}_{x_0}$ in (2.36) and $v = p_\Omega$ in (2.37) one gets:

$$\int_D \nabla p_\Omega \cdot \nabla \dot{u}_{x_0}\,dx + \int_D \alpha_\Omega p_\Omega \dot{u}_{x_0}\,dx = -\int_D 2(u_\Omega - u_{\text{ref}})\dot{u}_{x_0}\,dx, \tag{2.38}$$

$$\int_D \nabla \dot{u}_{x_0} \cdot \nabla p_\Omega\,dx + \int_D \alpha_\Omega \dot{u}_{x_0} p_\Omega\,dx = \text{sgn}_\Omega(x_0)(-(\alpha_1 - \alpha_2) + (f_1 - f_2))p_\Omega(x_0), \tag{2.39}$$

then subtracting those two equations yields:

$$\int_D 2(u_\Omega - u_{\text{ref}})\dot{u}_{x_0}\,dx = \text{sgn}_\Omega(x_0)(-(\alpha_1 - \alpha_2) + (f_1 - f_2))p_\Omega(x_0), \tag{2.40}$$

by substituting this in the right hand side of (2.29), (2.31) is proven. □

2.3 Topological Derivative via The Averaged Adjoint Equation

To calculate the topological derivative for the setting in Theorem 2.1, an alternative method could be employed that avoids the need to compute the topological state derivative, making the process essentially equivalent to the previous calculation while simplifying certain steps. This is achieved using the Lagrangian approach.

Definition 2.9 (Parameterised Lagrangian)**.** *For $\varepsilon \geq 0$, we introduce the parameterised Lagrangian:*

$$\mathcal{L}(\varepsilon, u, p) := \int_D (u - u_{ref})^2\,dx + \int_D \nabla u \cdot \nabla p\,dx + \int_D \alpha_\varepsilon up\,dx - \int_D f_\varepsilon p\,dx, \quad u, p \in H_0^1(D).$$

Definition 2.10 (Averaged Adjoint Equation)**.** *Associated with the state $u_\varepsilon = u_{\Omega_\varepsilon}$ and $u_0 = u_\Omega$, we define the averaged adjoint equation by: find $p_\varepsilon \in H_0^1(D)$, such that*

$$\int_0^1 \partial_u \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)(v)\, ds = 0 \quad \textit{for all } v \in H_0^1(D).$$

One can reformulate the previous Definition starting with the Lagrangian in Definition 2.9 at $u = su_\varepsilon + (1-s)u_0$, which is given by:

$$\begin{aligned}\mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon) = \\ &\int_D (su_\varepsilon + (1-s)u_0 - u_{\text{ref}})^2\, dx + \int_D \nabla(su_\varepsilon + (1-s)u_0)\cdot\nabla p_\varepsilon\, dx \\ &+ \int_D \alpha_\varepsilon(su_\varepsilon + (1-s)u_0)v\, dx - \int_D f_\varepsilon p_\varepsilon\, dx, \end{aligned} \tag{2.41}$$

and using the definition of directional derivative:

Definition 2.11. *(Directional derivative) Let E be a Banach space and $U \subset E$ an open subset. The directional derivative of $h : U \to \mathbb{R}$ at $x \in U$ in the direction of $v \in E$ is defined as:*

$$\mathrm{d}h(x)(v) = \lim_{t\to 0} \frac{h(x+tv) - h(x)}{t}.$$

Similarly for a function $h : U_1 \times \cdots \times U_r \to \mathbb{R}$ defined on open subsets $U_\ell \subset E_\ell$ for $\ell = 1, \ldots, r$ the partial derivative with respect to the sth argument at $(x_1, \ldots, x_r)$ in direction $v \in E_s$ for $s \in \{1, \ldots, r\}$ is defined by:

$$\partial_{x_s} h(x_1, \ldots, x_r)(v) := \lim_{t\to 0} \frac{h(x_1, \ldots, x_s + tv, \ldots, x_r) - h(x_1, \ldots, x_r)}{t}.$$

By this Definition, the derivative of the Lagrangian with respect to u in direction v is given by:

$$\partial_u \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)(v) = \lim_{t\to 0} \frac{\mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0 + tv, p_\varepsilon) - \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)}{t}. \tag{2.42}$$

Let $\overline{u} := (su_\varepsilon + (1-s)u_0)$, then (2.41) reads:

$$\mathcal{L}(\varepsilon, \overline{u}, p_\varepsilon) = \int_D (\overline{u} - u_{\text{ref}})^2\, dx + \int_D \nabla\overline{u}\cdot\nabla p_\varepsilon\, dx + \int_D \alpha_\varepsilon \overline{u} v\, dx - \int_D f_\varepsilon p_\varepsilon\, dx, \tag{2.43}$$

and the perturbed Lagrangian is given by:

$$\begin{aligned}\mathcal{L}(\varepsilon, \overline{u} + tv, p_\varepsilon)\, dx &= \int_D (\overline{u} + tv - u_{\text{ref}})^2\, dx + \int_D \nabla(\overline{u} + tv)\cdot\nabla p_\varepsilon\, dx \\ &\quad + \int_D \alpha_\varepsilon(\overline{u} + tv)p_\varepsilon\, dx - \int_D f_\varepsilon p_\varepsilon\, dx \\ &= \int_D (\overline{u} - u_{\text{ref}})^2 + t^2v^2 + 2tv(\overline{u} - u_{\text{ref}})\, dx + \int_D \nabla\overline{u}\cdot\nabla p_\varepsilon\, dx \\ &\quad + \int_D t\nabla v\cdot\nabla p_\varepsilon\, dx + \int_D \alpha_\varepsilon\overline{u}p_\varepsilon\, dx + \int_D t\alpha_\varepsilon v p_\varepsilon\, dx - \int_D f_\varepsilon p_\varepsilon\, dx. \end{aligned} \tag{2.44}$$

By subtracting, (2.43) from (2.44) and dividing by t one gets:

$$\frac{\mathcal{L}(\varepsilon, \overline{u} + tv, p_\varepsilon) - \mathcal{L}(\varepsilon, \overline{u}, p_\varepsilon)}{t} = \int_D tv^2 + 2v(\overline{u} - u_{\text{ref}}) + \nabla p_\varepsilon \cdot \nabla v + \alpha_\varepsilon v p_\varepsilon \, dx, \tag{2.45}$$

taking the limit as t goes to zero yields:

$$\begin{aligned} \partial_u \mathcal{L}(\varepsilon, \overline{u}, p_\varepsilon)(v) &= \int_D 2v(\overline{u} - u_{\text{ref}}) + \nabla p_\varepsilon \cdot \nabla v + \alpha_\varepsilon v p_\varepsilon \, dx \\ \partial_u \mathcal{L}(\varepsilon, (su_\varepsilon + (1-s)u_0), p_\varepsilon)(v) &= \int_D 2v((su_\varepsilon + (1-s)u_0) - u_{\text{ref}}) + \nabla p_\varepsilon \cdot \nabla v + \alpha_\varepsilon v p_\varepsilon \, dx, \end{aligned} \tag{2.46}$$

then integrating from zero to one with respect to s:

$$\int_0^1 \{\partial_u \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)(v)\} \, ds = \int_D 2v(u_\varepsilon + u_0 - u_{\text{ref}}) + \nabla p_\varepsilon \cdot \nabla v + \alpha_\varepsilon v p_\varepsilon \, dx. \tag{2.47}$$

Using Definition 2.10 one gets:

$$\int_D \nabla p_\varepsilon \cdot \nabla v + \alpha_\varepsilon v p_\varepsilon \, dx = -\int_D (u_\varepsilon + u_0 - 2u_{\text{ref}}) v \, dx. \tag{2.48}$$

Taking the limit $\varepsilon \searrow 0$ yields the weak form of the adjoint equation (2.32) that is $p_0 = p_\Omega$ solves:

$$\int_D \nabla p_\Omega \cdot \nabla v \, dx + \int_D \alpha_\Omega u v \, dx = -\int_D 2(u_\Omega - u_{\text{ref}}) v \, dx \quad \text{for all } v \in H_0^1(D). \tag{2.49}$$

Lemma 2.2. *For the parameterised Lagrangian defined in Definition 2.9 and the criterion $J(\Omega)$ as in (2.27), $\forall \varepsilon \in [0, \tau], \tau > 0$ it holds that:*

$$J(\Omega_\varepsilon(x_0)) = \mathcal{L}(\varepsilon, u_\varepsilon, 0) = \mathcal{L}(\varepsilon, u_0, p_\varepsilon), \tag{2.50}$$

$$J(\Omega) = \mathcal{L}(0, u_0, 0) = \mathcal{L}(0, u_0, p_0). \tag{2.51}$$

Proof. To prove (2.50) we consider the difference:

$$\mathcal{L}(\varepsilon, u_\varepsilon, 0) - \mathcal{L}(\varepsilon, u_0, p_\varepsilon) = \int_D (u_\varepsilon - u_{\text{ref}})^2 \, dx - \int_D (u_0 - u_{\text{ref}})^2 \, dx, \tag{2.52}$$

which is equal to:

$$\mathcal{L}(\varepsilon, u_\varepsilon, 0) - \mathcal{L}(\varepsilon, u_0, p_\varepsilon) = \mathcal{L}(\varepsilon, u_\varepsilon, p_\varepsilon) - \mathcal{L}(0, u_0, p_\varepsilon). \tag{2.53}$$

By using the fundamental theorem of calculus and the Definition 2.10 we get:

$$\mathcal{L}(\varepsilon, u_\varepsilon, p_\varepsilon) - \mathcal{L}(0, u_0, p_\varepsilon) = \int_0^1 \partial_u \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)(u_\varepsilon - u_0) \, ds = 0, \tag{2.54}$$

where in the equality to zero is due to the fact that $(u_\varepsilon - u_0) \in H_0^1(D)$, this proves (2.50), the second identity comes directly from Definition 2.9 as:

$$\mathcal{L}(0, u_0, 0) = \int_D (u_0 - u_{\text{ref}})^2 \, dx, \tag{2.55}$$

$$\mathcal{L}(0, u_0, p_0) = \int_D (u_0 - u_{\text{ref}})^2 \, dx. \tag{2.56}$$

□

So using the previous lemma, the difference between the criterion at the original domain and perturbed domain is given as for $x_0 \in D \setminus \partial\Omega$:

$$J(\Omega_\varepsilon(x_0)) - J(\Omega) = \mathcal{L}(\varepsilon, u_0, p_\varepsilon) - \mathcal{L}(0, u_0, p_0). \tag{2.57}$$

one could further split the right-hand side by adding a constructive zero as follows:

$$J(\Omega_\varepsilon(x_0)) - J(\Omega) = \mathcal{L}(\varepsilon, u_0, p_\varepsilon) - \mathcal{L}(\varepsilon, u_0, p_0) + \mathcal{L}(\varepsilon, u_0, p_0) - \mathcal{L}(0, u, p_0). \tag{2.58}$$

Moreover, it is proven in [11] that the following limit vanishes:

$$\lim_{\varepsilon\to 0} \frac{\mathcal{L}(\varepsilon, u_0, p_\varepsilon) - \mathcal{L}(\varepsilon, u_0, p_0)}{|\omega_\varepsilon|} = 0, \tag{2.59}$$

the topological derivative is then given by:

$$DJ(\Omega)(x_0) = \lim_{\varepsilon\to 0} \frac{\mathcal{L}(\varepsilon, u_0, p_0) - \mathcal{L}(0, u_0, p_0)}{|\omega_\varepsilon|}. \tag{2.60}$$

By Definition 2.9 we have that:

$$\mathcal{L}(\varepsilon, u_0, p_0) = \int_D (u_0 - u_{\text{ref}})^2\, dx + \int_D \nabla u_0 \cdot \nabla p_0\, dx + \int_D \alpha_\varepsilon u_0 p_0\, dx - \int_D f_\varepsilon p_0\, dx, \tag{2.61}$$

$$\mathcal{L}(0, u_0, p_0) = \int_D (u_0 - u_{\text{ref}})^2\, dx + \int_D \nabla u_0 \cdot \nabla p_0\, dx + \int_D \alpha_0 u_0 p_0\, dx - \int_D f_0 p_0\, dx. \tag{2.62}$$

By subtracting the two equations one gets:

$$\mathcal{L}(\varepsilon, u_0, p_0) - \mathcal{L}(0, u_0, p_0) = \int_D (\alpha_\varepsilon - \alpha_0) u_0 p_0\, dx - \int_D (f_\varepsilon - f_0) p_0\, dx. \tag{2.63}$$

Dividing by the measure of the hole ω_ε and using the calculation done in the proof of Lemma 2.1 yields:

$$\begin{aligned}\frac{\mathcal{L}(\varepsilon, u_0, p_0) - \mathcal{L}(0, u_0, p_0)}{|\omega_\varepsilon|} &= \frac{1}{|\omega_\varepsilon|}\left(\int_D (\alpha_\varepsilon - \alpha_0) u_0 p_0\, dx - \int_D (f_\varepsilon - f_0) p_0\, dx\right) = \\ &= \operatorname{sgn}_\Omega(x_0) \frac{1}{|\omega_\varepsilon|}\left((\alpha_1 - \alpha_2)\int_{\omega_\varepsilon} u_0\, p_0\, dx - (f_1 - f_2)\int_{\omega_\varepsilon} p_0\, dx\right).\end{aligned} \tag{2.64}$$

Assuming that u_0 and p_0 are continuous at x_0, the topological derivative will be given by:

$$DJ(\Omega)(x_0) = \lim_{\varepsilon\to 0} \frac{\mathcal{L}(\varepsilon, u_0, p_0) - \mathcal{L}(0, u_0, p_0)}{|\omega_\varepsilon|} \tag{2.65}$$

$$= \operatorname{sgn}_\Omega(x_0)\left((\alpha_1 - \alpha_2) u_0(x_0) - (f_1 - f_2)\right) p_0(x_0). \tag{2.66}$$

Now this derivative could be used in the implementation of the so called level set algorithm, which represents the interface of the domain implicitly, allowing for smooth and natural handling of topological changes such as merging and splitting. By incorporating the concept of topological derivatives, the level set method can measure the sensitivity of the objective functional with respect to infinitesimal topological changes, such as the introduction of small holes. This level set method is discussed in the following section.

2.4 Classical Level Set Algorithm by Amstutz-Andrä and The Generalized Topological Derivative

The level-set method introduced by Osher and Sethian [9], was originally developed for the numerical tracking of fronts and free boundaries, yet it has been a cornerstone in computational topology optimization due to its robust handling of complex geometries and topological changes. The classical level-set algorithm by Amstutz and Andrä [3] further refines this method by incorporating a generalized topological derivative, providing enhanced sensitivity analysis for shape and topology optimization.

The core idea of the level-set method is to represent the evolving boundary of a shape implicitly through a higher-dimensional scalar function, typically denoted as $\psi(x)$. The zero level-set of this function delineates the boundary of the shape of interest. Following this approach, the domain $\Omega \subset D$ is to be represented by the continuous level set function $\psi : D \to \mathbb{R}$ as follows:

$$\begin{aligned} \psi < 0 &\iff x \in \Omega, \\ \psi > 0 &\iff x \notin \Omega, \\ \psi = 0 &\iff x \in \partial\Omega \setminus \partial D. \end{aligned} \tag{2.67}$$

Definition 2.12 (Generalized topological derivative)**.** *For $J : P(D) \to \mathbb{R}$ a shape functional, $\Omega \in P(D)$ open set with Ω compactly contained in D. Assume that the topological derivative $DJ(\Omega)(x)$ exists for all $x \in D \setminus \partial\Omega$. Then, the generalized topological derivative is given by:*

$$\mathcal{D}J(\Omega)(x) := \begin{cases} -DJ(\Omega)(x) & \text{for } x \in \Omega, \\ DJ(\Omega)(x) & \text{for } x \in D \setminus \overline{\Omega}. \end{cases}$$

It follows from Definition 2.4 that a necessary local minimality condition for the problem in terms of the generalized topological derivative becomes:

$$\begin{cases} \mathcal{D}J(\Omega)(x) < 0 & \text{for } x \in \Omega, \\ \mathcal{D}J(\Omega)(x) > 0 & \text{for } x \in D \setminus \overline{\Omega}. \end{cases} \tag{2.68}$$

Lemma 2.3. *For a shape functional $J(\Omega)$ and a shape $\Omega^* \subset D$ described by the level set function $\psi : D \to \mathbb{R}$, such that:*

$$\psi(x) < 0 \implies x \in \Omega^*, \quad \psi(x) > 0 \implies x \in D \setminus \overline{\Omega^*},$$

where there is a constant $c > 0$, such that:

$$c\psi(x) = \mathcal{D}J(\Omega^*)(x),$$

Then the shape Ω^* is a local minimum for the shape functional $J(\Omega)$.

Proof. For a domain Ω expressed by the level set function $\psi(x)$ and assuming that $\psi(x)$ is given by, $\psi(x) = \frac{1}{c}\mathcal{D}J(\Omega)(x)$ with $c > 0$: If $x \in \Omega$ then $\psi(x) < 0$, which means that $\mathcal{D}J(\Omega)(x) < 0$ which satisfies the optimally condition (2.68).
Similarly for if $x \in \Omega^c$ then $\psi(x) > 0$ and $\mathcal{D}J(\Omega)(x) > 0$. □

Now the fictitious time t is introduced and the family of domains $\Omega(t)$ is considered. The idea is now to evolve the shape according to the level set equation solved by:

$$[0,\infty) \ni t \mapsto \Omega(t) := \{x \in D \mid \psi(t,x) > 0\}.$$

Within the domain $\Omega(t)$, it is desirable to increase the level-set function at a point x (to push it towards the direction where holes appear) where $DJ(\Omega)(x) < 0$ i.e. $\mathcal{D}J(\Omega)(x) > 0$. Conversely, in $D\backslash\Omega(t)$, it is desirable to decrease the level set function where $DJ(\Omega)(x) < 0$ i.e. $\mathcal{D}J(\Omega)(x) < 0$. Thus a natural algorithm would be to control the level-set function by the differential equation:

$$\frac{\partial\psi}{\partial t}(x) = \mathcal{D}J(\Omega)(x). \tag{2.69}$$

From Lemma 2.3, multiplying the level-set function by a positive constant does not alter the domain it represents. Hence, to enhance the stability of the algorithm, it can be advantageous to eliminate this unnecessary degree of freedom by, for example, ensuring that a specific norm of the level-set function is maintained, that is achieved by normalizing the vectors $\psi_k(x)$ and $\mathcal{D}J(\Omega_k)(x)$ by their norms for each iteration k.

If the optimality conditions are met at a (local) minimum of the criterion for all $x \in D \setminus \partial\Omega$, the generalized topological gradient can serve as a level-set function to represent the corresponding domain.

Algorithm 1 Convex Combination Algorithm for Topology Optimization.

Require: Initial level-set function ψ_0 and corresponding geometry Ω_0, initial step size λ_0, stopping tolerance $\tau > 0$, maximum number of iterations $k_{\max} \in \mathbb{N}$

for $k = 0, 1, 2, \ldots, k_{\max}$ **do**
  Compute the generalized topological derivative $\mathcal{D}J(\Omega_k)$
  Compute $\theta_k = \arccos\left(\frac{(\psi_k, DJ(\Omega_k))}{\|\psi_k\|_{L^2(D)}\|DJ(\Omega_k)\|_{L^2(D)}}\right)$
  if $\theta_k < \tau$ **OR** $N_{\text{opt}} \geq \text{Int}(0.995 \times N)$ **then**
    Stop with approximate minimizer Ω_k
  end if
  Set $\lambda_k = \min(1, 2\lambda_{k-1})$
  while $J\left(\lambda_k \frac{DJ(\Omega_k)}{\|DJ(\Omega_k)\|_{L^2(D)}} + (1-\lambda_k)\psi_k\right) > Max(J(\psi_k), ..., J(\psi_{k-4}))$ **do**
    Decrease the step size $\lambda_k = 0.8\lambda_k$
  end while
  Update the level-set function: $\psi_{k+1} = \lambda_k \frac{DJ(\Omega_k)}{\|DJ(\Omega_k)\|_{L^2(D)}} + (1-\lambda_k)\psi_k$
end for

However, the generalized topological derivative is not bound to vanish at a minimizer Ω^*, hence this algorithm is generally divergent. As the generalized topological gradient provides sensitivity information about the objective function with respect to infinitesimal perturbations in the domain, said divergence may occur because the topological gradient does not necessarily become zero at a local optimum, which is crucial for ensuring convergence in traditional optimization algorithms.
Which leads to the utilization of the orthogonal projection defined by:

$$P_{\psi(t)^\perp}(v) := v - \frac{1}{\|\psi\|^2_{L_2(D)}}\langle v, \psi\rangle_{L_2(D)}, \tag{2.70}$$

so that the algorithm would be to control the level-set function by the differential equation:

$$\frac{\partial \psi}{\partial t}(x) = P_{\psi(t)^\perp}(\mathcal{D}J(\Omega)(x)). \tag{2.71}$$

This orthogonal projection ensures that the evolution of the level-set function remains within a constrained subspace, improving the stability and convergence properties of the algorithm. The projection operator $P_{\psi(t)^\perp}$ effectively removes any components of the gradient that are aligned with the current level-set function.
So by construction:

$$\forall v \in L_2(D) \quad \langle P_{\psi(t)^\perp}(v), \psi \rangle_{L_2(D)} = 0.$$

If ψ tends to a stationary point (i.e. $\frac{\partial \psi}{\partial t} = 0$) and the topological gradient at said point is non zero, then the sufficient local optimality conditions are satisfied, since the existence of a stationary point implies that $c\psi = \mathcal{D}J(\Omega)$ for if $c > 0$ the point is a local minimum and for $c < 0$ the point is a local maximum, yet the achievement of a local maximum is unlikely due to the fact that the algorithm is constructed to decrease the criterion.

Algorithm 2 Gradient Descent Method for Topology Optimization.

Require: Initial level-set function ψ_0, initial step size $\lambda_0 = 1$, stopping tolerance $\tau > 0$, maximum number of iterations $k_{\max} \in \mathbb{N}$

for $k = 0, 1, 2, \ldots, k_{\max}$ **do**
  Compute the generalized topological derivative $DJ(\Omega_k)$
  Set $g_k = -P_{\psi_k^\perp}(DJ(\Omega_k)) = -\left(DJ(\Omega_k) - \frac{(DJ(\Omega_k), \psi_k)}{\|\psi_k\|^2_{L^2(D)}}\psi_k\right)$
  Compute $\theta_k = \arccos\left(\frac{(\psi_k, DJ(\Omega_k))}{\|\psi_k\|_{L^2(D)}\|DJ(\Omega_k)\|_{L^2(D)}}\right)$
  if $\theta_k < \tau$ **OR** $N_{\text{opt}} \geq \text{Int}(0.995 \times N)$ **then**
    Stop with approximate minimizer Ω_k
  end if
  Set $\lambda_k = \min(1, 2\lambda_{k-1})$
  while $J(\psi_k - \lambda_k g_k) > J(\psi_k)$ **do**
    Decrease the step size $\lambda_k = 0.8\lambda_k$
  end while
  Update the level-set function: $\psi_{k+1} = \psi_k - \lambda_k g_k$
end for

2.4.1 Step size control and stopping criteria

As the algorithm is designed to progress in the direction of cost decrease, a loose step size control is implemented. The cost of each iteration is compared with the maximum of the costs in the last five iterations. The step size, λ_k, is either increased or decreased by 20% based on this comparison. Specifically, if the cost function decreases, the step size is increased to accelerate convergence; if the cost function increases, the step size is reduced to ensure stability and prevent divergence. This adaptive approach helps in balancing the trade-off between convergence speed and algorithm stability.

Stopping criterion θ

In optimization algorithms, a common stopping criterion involves monitoring the change in the objective function or other relevant metrics to determine when to terminate the iterations. One such criterion is based on the angle θ between the gradient of the objective function and the search direction.

The stopping criterion can be defined as follows:

$$\theta_k = \arccos\left(\frac{\nabla J(u_k) \cdot p_k}{\|\nabla J(u_k)\|\|p_k\|}\right), \tag{2.72}$$

where:

- $\nabla J(u_k)$ is the gradient of the objective function J at iteration k.
- p_k is the search direction at iteration k.
- $\|\cdot\|$ denotes the Euclidean norm.

The algorithm terminates when θ_k falls below a predetermined threshold τ, indicating that the gradient and the search direction are sufficiently aligned, and further iterations are unlikely to yield significant improvements. Mathematically, the stopping condition is:

$$\theta_k < \tau, \tag{2.73}$$

where τ is a small positive value chosen based on the desired accuracy and convergence properties of the algorithm.

Number of optimum elements N_{opt}

It is possible that at the optimum domain, the adjoint is equal to zero as in the case of example 3.3 in the next chapter, the theta will be then an undetermined quantity which calls for a Definition of an alternative stopping criteria.

Since the simulation is done in the framework of Finite elements, and the degrees of freedom on each element can be categorized as either optimum or non optimum according to the optimality conditions (2.68),therefore for total numbers of degrees of freedom N, and the number of degrees of freedom satisfying (2.68) N_{opt}, the stopping criterion would be $N_{opt} \geq Int(0.995 \times N)$, where $Int(.)$ denotes the integer part of the number.

Chapter 3

Structural Optimization Using the Level Set Method

3.1 A Case Study for The Level Set Algorithm

In this section, we study the optimization over the the criterion which is a shifted $L^2(D)$ distance, the shift imposes that the topological derivative, at the optimum domain is non zero, this is done for the sake of comparison with the Newton-like methods in chapter 5.

Let $D \subset \mathbb{R}^d$ be the hold-all domain and $\psi(t,\cdot) : D \to \mathbb{R}$ a time-dependent level-set function. It's argued in [3] that for if the generalized topological derivative does not vanish at the optimum, the differential equation:

$$\frac{\partial \psi}{\partial t} = \mathcal{D}J(\Omega_{\psi(t)}), \quad \Omega_{\psi(t)} := \{x \in D : \ \psi(t,x) < 0\}, \tag{3.1}$$

would not be a good fit to control the evolution of the level set, and subsequently algorithm 1 would not converge, and instead one should use algorithm 2 which is governed by the equation:

$$\frac{\partial \psi}{\partial t} = P_{\psi(t)^\perp}(\mathcal{D}J(\Omega_{\psi(t)})). \tag{3.2}$$

However, in the following two cases both algorithms converge.

The target is now to solve the following minimization problem:

$$\min_{\Omega} J(\Omega, u) = \frac{1}{2}\int_D (u - u_{\mathrm{d}} + 10\,xy)^2\,dx, \tag{3.3}$$

subject to (in a weak sense) the elliptic PDE constraint with the right hand side perturbation and with homogeneous Neumann boundary conditions:

$$-\Delta u + u = f_\Omega \quad \text{in } D, \tag{3.4}$$

$$\frac{\partial u}{\partial \nu} = 0 \quad \text{on } \partial D, \tag{3.5}$$

where $f_\Omega(x) = \chi_\Omega(x) f_1 + \chi_{\Omega^c}(x) f_2$ with $f_1, f_2 \in \mathbb{R}$ are constants, u_{d} is given as the solution of the PDE constraint on a predefined domain Ω_{d} and ν is the outward facing unitary normal vector.

The PDE constraint can be formulated as:

$$\int_D \nabla u \cdot \nabla v \, dx + \int_D uv \, dx = \int_D f_\Omega v \, dx \quad \forall v \in H^1(D). \tag{3.6}$$

The adjoint equation is now given by:

$$-\Delta p + p = -2(u - u_d) - 20\,xy \quad \text{in } D, \tag{3.7}$$

and the generalized topological derivative for this case is given as follows:

$$\mathcal{D}J(\Omega)(x) := \begin{cases} -DJ(\Omega)(x) = ((f_2 - f_1)p(x)) & \text{for } x \in \Omega, \\ DJ(\Omega)(x) = -(f_2 - f_1)p(x) & \text{for } x \in D \setminus \overline{\Omega}. \end{cases}$$

A hold-all domain of $D = (-2, 2)^2$ is considered and discretized by maximum edge size of 0.1, which leads to a mesh of 1942 nodes (3722 triangular elements), the relatively coarse mesh size is chosen to compare with cases in chapter 4.
The solution converges in 18 iterations into the following shape (blue) of cost $J = 2240$, the effect of the $10\,xy$ term is clear, because it has the dominant order of magnitude in the criterion $J(\Omega)$.

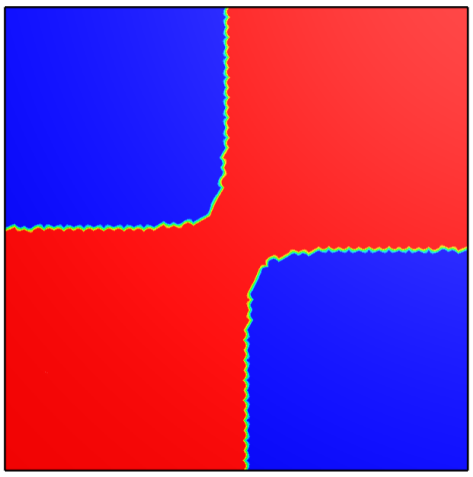


Figure 3.1: Solution Domain for The Shifted Case.

3.2 Jump in Linear Term

For further comparison with cases in the following chapters, it is beneficial to also study a more generalized case where the linear term is also dependent on the domain, we want to solve the problem for the same criterion:

$$\min_{\Omega} J(\Omega, u) = \frac{1}{2}\int_D (u - u_{\mathrm{d}} + 10\,xy)^2\,dx, \tag{3.8}$$

but subject to different constraint, that is, $u \in H^1(D)$ solves (in a weak sense):

$$-\Delta u + \alpha_\Omega u = f_\Omega \quad \text{in } D, \tag{3.9}$$

$$\frac{\partial u}{\partial \nu} = 0 \quad \text{on } \partial D, \tag{3.10}$$

where $\alpha_\Omega(x) = \chi_\Omega(x)\alpha_1 + \chi_{\Omega^c}(x)\alpha_2$, the constants α_1 and α_2 are chosen to be 2 and 7 respectively.

Using the calculation done in Theorem 2.1, the generalized topological derivative for this case will be as follows:

$$\mathcal{D}J(\Omega)(x) := \begin{cases} -DJ(\Omega)(x) = -1\,(-(f_2 - f_1)p(x) - (\alpha_1 - \alpha_2)p(x)) & \text{for } x \in \Omega, \\ DJ(\Omega)(x) = -(f_2 - f_1)p(x) - (\alpha_1 - \alpha_2)p(x) & \text{for } x \in D \setminus \overline{\Omega}. \end{cases}$$

For the same discretization as in the previous example, the solution converges to the following figure in 14 iterations with final cost $J = 2735.3$.

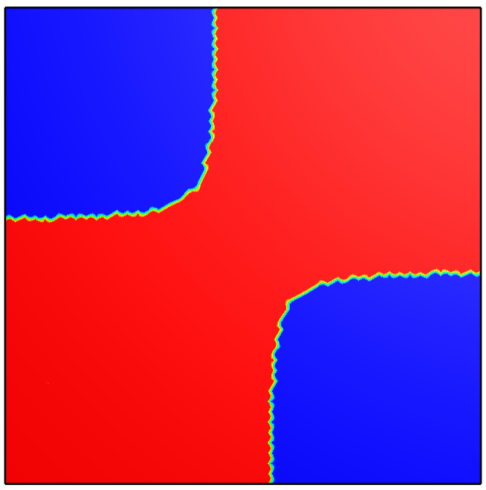


Figure 3.2: Solution for The Case with a Jump in Linear Term.

3.3 Primary Part Perturbation

To pave the way for investigating topological optimization in linear elasticity, it would be useful to have a reference case that have the same fundamental elements as the linear elasticity PDE.

To do so, the case from [7] is considered where the main differences from the two previous cases are in the PDE constraint, as in this case, the primary part of the PDE

also has a dependence on the domain Ω, besides the PDE is subject to homogeneous Dirichlet boundary conditions,which will correspond to fastening in elasticity cases of the next chapter.

Let $D \subset \mathbb{R}^d$ with $0 < d \in \mathbb{N}_0$ be an open and bounded domain with boundary ∂D and let $\Omega \subset D$ be an open subset. We denote by $\Gamma = \partial\Omega \setminus \partial D$ the interior boundary of Ω in D and by $\Omega^c = D \setminus \Omega$ the complement of Ω in D. The following problem is considered:

$$\min_{\Omega} J(\Omega, u) = \frac{1}{2} \int_D (u - u_{\mathrm{d}})^2 \, dx, \tag{3.11}$$

subject to (in a weak sense):

$$-\mathrm{div}(\beta_\Omega \nabla u) = f_\Omega \quad \text{in } D, \tag{3.12}$$

$$u = 0 \quad \text{on } \partial D. \tag{3.13}$$

or

$$\int_D \beta_\Omega \nabla u \cdot \nabla v \, dx = \int_D f_\Omega v \, dx \quad \forall v \in H_0^1(D), \tag{3.14}$$

where $\beta_\Omega(x) = \chi_\Omega(x)\beta_1 + \chi_{\Omega^c}(x)\beta_2$ and $f_\Omega(x) = \chi_\Omega(x)f_1 + \chi_{\Omega^c}(x)f_2$ with $\beta_1, \beta_2 > 0$ as well as $f_1, f_2 \in \mathbb{R}$. In our setting, u_{d} is given as the solution of the PDE constraint on a desired domain Ω_{d}.
The adjoint equation (in a weak sense) is defined as

$$-\mathrm{div}(\beta_\Omega \nabla p) = -2(u - u_d) \quad \text{in } D, \tag{3.15}$$

$$u = 0 \quad \text{on } \partial D. \tag{3.16}$$

Remark 3.1. *If $u^+ := u|_\Omega \in C^2(\overline{\Omega})$ and $u^- := u|_{\Omega^c} \in C^2(\overline{\Omega}^c)$ then the above weak form is equivalent to*

$$-\beta_1 \Delta u = f_1 \quad \text{in } \Omega, \tag{3.17}$$

$$-\beta_2 \Delta u = f_2 \quad \text{in } \Omega^c, \tag{3.18}$$

$$u = 0 \quad \text{on } \partial D, \tag{3.19}$$

$$\beta_1 \nabla u^+ \cdot \nu = \beta_2 \nabla u^- \cdot \nu \quad \text{on } \partial\Omega, \tag{3.20}$$

where ν is the outward facing unitary normal vector from Ω and u^+ and u^- denote the restriction of u to Ω and $\overline{\Omega}^c$. respectively. Similarly for the adjoint variable we get:

$$-\beta_1 \Delta p = -2(u - u_d) \quad \text{in } \Omega, \tag{3.21}$$

$$-\beta_2 \Delta p = -2(u - u_d) \quad \text{in } \Omega^c, \tag{3.22}$$

$$p = 0 \quad \text{on } \partial D, \tag{3.23}$$

$$\beta_1 \nabla p^+ \cdot \nu = \beta_2 \nabla p^- \cdot \nu \quad \text{on } \partial\Omega. \tag{3.24}$$

Proof. By choosing $v \in C_c^\infty(\Omega)$ in the weak form (3.14) one gets:

$$\int_D \beta_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega \beta_1 \nabla u \cdot \nabla v \, dx = \int_\Omega f_1 v \, dx, \tag{3.25}$$

similarly for $v \in C_c^\infty(\Omega^c)$:

$$\int_D \beta_\Omega \nabla u \cdot \nabla v \, dx = \int_{\Omega^c} \beta_2 \nabla u \cdot \nabla v \, dx = \int_{\Omega^c} f_2 v \, dx. \tag{3.26}$$

So that for $u|_\Omega \in C^2(\overline{\Omega})$ and $u|_{\Omega^c} \in C^2(\overline{\Omega}^c)$ it holds that:

$$\begin{aligned} -\beta_1 \Delta u &= f_1 \quad \text{in } \Omega, \\ -\beta_2 \Delta u &= f_2 \quad \text{in } \Omega^c. \end{aligned} \tag{3.27}$$

So by integrating the left hand side of (3.27) by part in each of Ω and Ω^c yields (note that $\nu|_{\partial\Omega} = -\nu|_{\partial\Omega^c}$)

$$-\int_\Omega \beta_1 \Delta u v \, dx = \int_\Omega \beta_1 \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \beta_1 (\nabla u^+ \cdot \nu) v \, ds \qquad \text{in} \quad \Omega, \tag{3.28}$$

$$-\int_{\Omega^c} \beta_2 \Delta u v \, dx = \int_{\Omega^c} \beta_2 \nabla u \cdot \nabla v \, dx + \int_{\partial\Omega} \beta_2 (\nabla u^- \cdot \nu) v \, ds \qquad \text{in} \quad \Omega^c. \tag{3.29}$$

Moreover, the left hand side of (3.14) can be reformulated as:

$$\int_D \beta_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega \beta_1 \nabla u \cdot \nabla v \, dx + \int_{\Omega^c} \beta_2 \nabla u \cdot \nabla v \, dx. \tag{3.30}$$

Adding equations (3.28) and (3.29) yields:

$$\begin{aligned} -\int_\Omega \beta_1 \Delta u v \, dx - \int_{\Omega^c} \beta_2 \Delta u v \, dx = \int_D \beta_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \beta_1 (\nabla u^+ \cdot \nu) v \, ds \\ + \int_{\partial\Omega^c} \beta_2 (\nabla u^- \cdot \nu) v \, ds. \end{aligned} \tag{3.31}$$

Note that $\int_{\partial\Omega^c} \beta_2 (\nabla u^- \cdot \nu) v \, ds = \int_{\partial\Omega} \beta_2 (\nabla u^- \cdot \nu) v \, ds$ because $v = 0$ on ∂D. So using (3.27) yields:

$$\int_\Omega f_1 v \, dx + \int_{\Omega^c} f_2 v \, dx = \int_D \beta_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \beta_1 (\nabla u^+ \cdot \nu) v \, ds + \int_{\partial\Omega} \beta_2 (\nabla u^- \cdot \nu) v \, ds.$$

From the previous equation and since u satisfies (3.14) we get the transmission condition:

$$\int_{\partial\Omega} \beta_2 (\nabla u^- \cdot \nu) v \, ds = \int_{\partial\Omega} \beta_1 (\nabla u^+ \cdot \nu) v \, ds \qquad \forall v \in C_c^\infty(D) \tag{3.32}$$

□

The generalized topological derivative for this case similar to the linear elasticity case in [3] is given as follows:

$$\mathcal{D}J(\Omega)(x) := \begin{cases} -1 \left(2\beta_2 \left(\frac{\beta_2 - \beta_1}{\beta_1 + \beta_2}\right) \nabla u(x) \cdot \nabla p(x) - (f_2 - f_1) p(x)\right) & \text{for } x \in \Omega, \\ 2\beta_1 \left(\frac{\beta_1 - \beta_2}{\beta_1 + \beta_2}\right) \nabla u(x) \cdot \nabla p(x) - (f_2 - f_1) p(x) & \text{for } x \in D \setminus \overline{\Omega}. \end{cases} \tag{3.33}$$

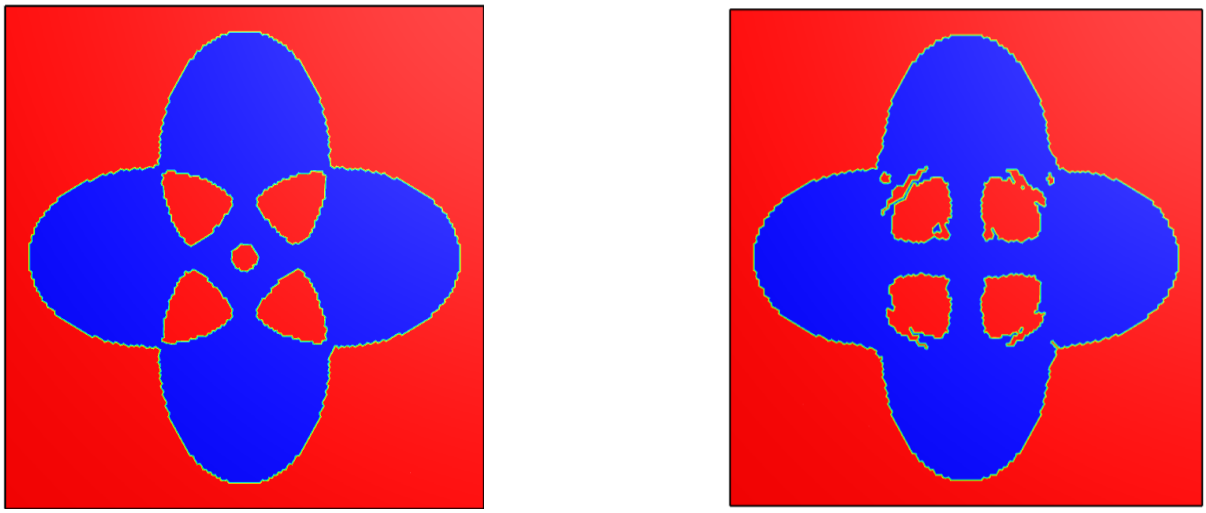
The first case is equal to $-DJ(\Omega)(x)$ while the second one is $DJ(\Omega)(x)$. We set the piece wise constants as follows:

$$\begin{aligned} \beta_1 &= f_1 = 10, \\ \beta_2 &= f_2 = 1. \end{aligned}$$

As the right hand side function $f \in L^2(D)$, the destination sate u_d is the piece wise constant interpolation of the two values f_1 and f_2 according to the clover shape destination level set ψ_{des}, representing the destination domain Ω_d that is:

$$\psi_{des} = \left(0.1 \left((|x| - 1)^2 \cdot (|y| - 1)^2 - 0.001\right)\right). \tag{3.34}$$

We consider a hold-all domain of $D = (-2, 2)^2$ with the discretization of 7571 nodes (14820 triangular elements), the convex combination algorithm 1 is used for this case and it converges to the shape(blue) in the following figure in 1060 iterations.



Destination Domain ψ_{des}. Result after 1060 Iterations.

Figure 3.3: Comparison between The Destination Domain and The Solution Domain.

By construction of this problem the cost at the destination domain is exactly zero, subsequently the topological derivative is also zero, the result of that is when the cost becomes small, the convergence becomes slower as the value of the topological derivative keeps fluctuating around zero, that makes the convergence according to the previously discussed criteria harder, this behaviour can be seen in the following figures:

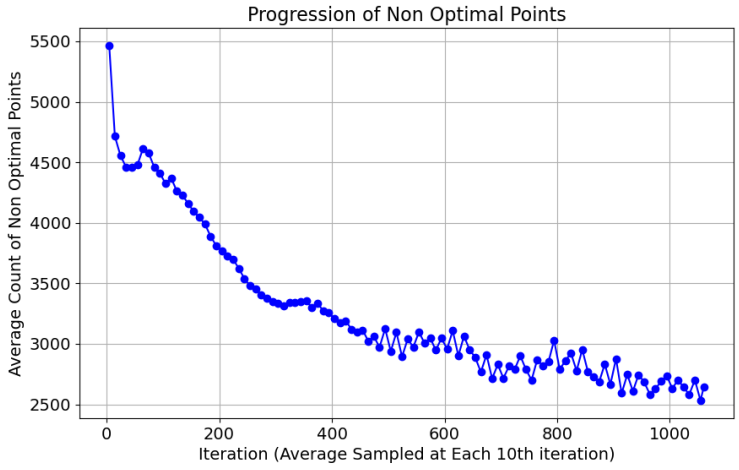


Figure 3.4: Progression of the Number of Active DOFs.

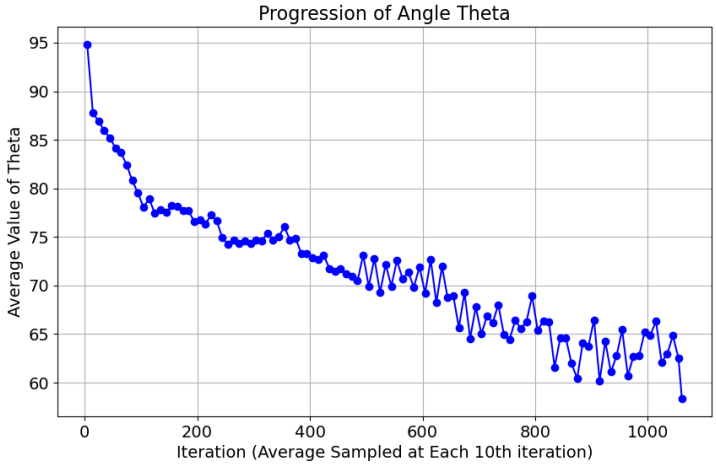


Figure 3.5: Progression of the Angle θ_k.

Moreover, in this case it is clear that algorithm 1 is converging which can be seen from the progression of the criterion $J(\Omega_k)$.

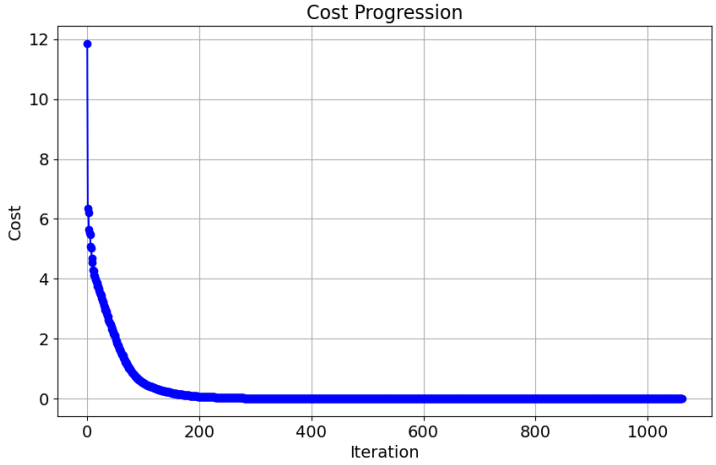


Figure 3.6: Progression of the Cost.

Chapter 4

Level Set Algorithm in Linear Elasticity

Similar to the last example in the previous chapter, the linear elasticity PDE has a domain dependency in the primary part, where the hold-all domain would contain both material and air, the elasticity solution is different for each.

The target is to apply the level set algorithm to minimize the work represented by the product of force with the displacement as well as minimizing the total volume(and subsequently the total weight assuming uniform density) of the structure, the problem from [3] is then formulated as follows:

For D a smooth and bounded domain of $\mathbb{R}^d$ and $\Omega \subset D$ a smooth subdomain containing a linear isotropic elastic material. Here, D is the hold-all domain and Ω is the design domain that will undergo optimization. The boundary ∂D of D is partitioned into three non-overlapping parts:

$$\partial D = \Gamma_D \cup \Gamma_F \cup \Gamma_N,$$

with Γ_D having a non-zero Lebesgue measure. It's assumed that the boundary of Ω satisfies the following:

$$\partial\Omega = (\Gamma_D \cap \partial\Omega) \cup \Gamma_F \cup (\Gamma_N \cap \partial\Omega) \cup \Gamma_0,$$

where $\Gamma_0 = \partial\Omega \cap D$.Given load $F \in (H^{-1/2}(\Gamma_F))^d$, the mixed boundary value problem (BVP) for the displacement u_Ω is formulated as follows:

$$\begin{cases} -\mathrm{div}(Ae(u_\Omega)) = 0 & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \Gamma_D \cap \partial\Omega, \\ (Ae(u_\Omega))\nu = F & \text{on } \Gamma_F, \\ (Ae(u_\Omega))\nu = 0 & \text{on } (\Gamma_N \cap \partial\Omega) \cup \Gamma_0, \end{cases} \tag{4.1}$$

As linear elasticity is assumed, that means that the material's mechanical properties are identical in all directions and follows Hooke's law, which states that the stress is proportional to the strain. So the zero right hand side in the first line of (4.1) represents zero body forces and A is Hooke's tensor and e(u) is the linear strain tensor given by:

$$\begin{aligned} Ae(u) &= \sigma(u), \\ \sigma(u) &= 2\mu e(u) + \lambda\,\mathrm{tr}(e(u))\,I. \end{aligned} \tag{4.2}$$

Now the criterion considered $j(\Omega)$ is the sum of the compliance and a term accounting for the weight of the structure, i.e.,

$$j(\Omega) = J(u_\Omega) + \ell|\Omega|,$$

where the compliance J is defined on $(H^1(\Omega))^d$, can be understood as the total work done due to the application of an external force, which is given by:

$$J(u) = \int_{\Gamma_F} \Phi \cdot u \, ds,$$

and $|\Omega|$ is the Lebesgue measure of Ω. The constant $\ell > 0$ is given and can be interpreted as a Lagrange multiplier. Due to Green's formula, the compliance can also be formulated as

$$J(u_\Omega) = \int_\Omega Ae(u_\Omega) : e(u_\Omega) \, dx, \tag{4.3}$$

where : denotes the Frobenius inner product between two matrices $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{d\times d}$: $A : B := \sum_{i.j=1}^{d} a_{ij}b_{ij}$. However, applying the surface integral form of the criterion is more beneficial numerically.

To avoid re-meshing, the elasticity Problem 4.1 is approximated for $\varepsilon \to 0$ by the following boundary value problem [3]. with variable coefficients, formulated in the fixed domain D:

$$\begin{cases} -\mathrm{div}(\tilde{\alpha} Ae(u_\Omega)) = 0 & \text{in } D, \\ u_\Omega = 0 & \text{on } \Gamma_D, \\ (Ae(u_\Omega))\nu = \Phi & \text{on } \Gamma_F, \\ (Ae(u_\Omega))\nu = 0 & \text{on } \Gamma_N, \end{cases} \tag{4.4}$$

with

$$\tilde{\alpha}_\Omega = \begin{cases} 1 & \text{in } \Omega, \\ \eta & \text{in } D \setminus \Omega. \end{cases}$$

The constant η must be chosen small enough as it is the approximation of the void in (4.1) with a suitable accuracy. In the computations presented, a fixed value $\eta = 10^{-3}$ has been used.

Multiplying (4.4) by a test function $v \in H_0^1(D)$ and integrating over the domain D yields:

$$\int_D -div(\widetilde{\alpha}_\Omega \sigma(u))v \, dx = 0, \tag{4.5}$$

then by using Green's theorem the equation becomes:

$$-\int_{\partial D} \widetilde{\alpha}_\Omega \sigma(u)\nu \cdot v \, ds + \int_D \widetilde{\alpha}_\Omega \sigma(u) : \nabla v \, dx = 0. \tag{4.6}$$

Dividing the boundary ∂D into $\Gamma_F \cup \Gamma_n \cup \Gamma_d$ and using the Neumann boundary condition as well as $v = 0$ *on* Γ_d yields:

$$-\int_{\Gamma_F} \Phi \cdot v \, ds + \int_D \widetilde{\alpha}_\Omega \sigma(u) : \nabla v \, dx = 0. \tag{4.7}$$

Since σ happens to be symmetric, one can equivalently write:

$$-\int_{\Gamma_F} \Phi \cdot v \, ds + \int_D \widetilde{\alpha}_\Omega \sigma(u) : \left(\frac{1}{2}(\nabla v + (\nabla v)^T) \right) dx = 0, \tag{4.8}$$

or

$$-\int_{\Gamma_F} \Phi \cdot v \, ds + \int_D \widetilde{\alpha}_\Omega \sigma(u) : e(v) \, dx = 0, \tag{4.9}$$

which is the weak formulation of linear elasticity.

Remark 4.1. *Under the assumption that the boundary Γ_F is completely included in Ω, for $u^+ := u|_\Omega \in C^2(\overline{\Omega})$ and $u^- := u|_{\Omega^c} \in C^2(\overline{\Omega}^c)$ the previous weak form can be formulated as:*

$$\nabla \cdot \sigma(u) = 0 \quad \text{in } \Omega, \tag{4.10}$$
$$\eta \nabla \cdot \sigma(u) = 0 \quad \text{in } \Omega^c, \tag{4.11}$$
$$-\sigma(u)\nu = \Phi \quad \text{on } \Gamma_F \subset \partial\Omega, \tag{4.12}$$
$$u = 0 \quad \text{on } \Gamma_D, \tag{4.13}$$
$$\sigma(u^+) \cdot \nu + \eta \sigma(u^-) \cdot \nu = 0 \quad \text{on } \partial\Omega. \tag{4.14}$$

Proof. By restricting $v \in C_0^\infty(\Omega)$ in the weak form (4.9) one gets:

$$\int_\Omega \sigma(u) : e(v) \, dx = \int_{\Gamma_F} \Phi v \, ds. \tag{4.15}$$

Similarly for $v \in C_0^\infty(\Omega^c)$:

$$\int_{\Omega^c} \eta \, \sigma(u) : e(v) \, dx = 0, \tag{4.16}$$

so that for $u|_\Omega \in C^2(\overline{\Omega})$ and $u|_{\Omega^c} \in C^2(\overline{\Omega}^c)$ it holds that:

$$\begin{aligned} \nabla \cdot \sigma(u) &= 0 \quad \text{in } \Omega, \\ \eta \nabla \cdot \sigma(u) &= 0 \quad \text{in } \overline{\Omega}^c, \end{aligned} \tag{4.17}$$

Moreover for $v \in C_c^\infty(D)$ and $\nu|_{\partial\Omega} = -\nu|_{\partial\Omega^c}$, so by integrating the left hand side of (4.17) by part in each of Ω and Ω^c and using Einstein's index notation for simplicity one gets:

$$\int_\Omega \sigma(u) : e(v) \, dx = \int_\Omega \sigma_{ij}(u) \frac{\partial v_i}{\partial x_j} \, dx = \int_\Omega \frac{\partial \sigma_{ij}(u)}{\partial x_j} v_i \, dx - \int_{\partial\Omega} \sigma_{ij}(u^+) v_i \cdot \nu \, ds, \tag{4.18}$$

$$\eta \int_{\Omega^c} \sigma(u) : e(v) \, dx = \eta \int_{\Omega^c} \sigma_{ij}(u^-) \frac{\partial v_i}{\partial x_j} \, dx = \eta \int_{\Omega^c} \frac{\partial \sigma_{ij}(u^-)}{\partial x_j} v_i \, dx + \eta \int_{\partial\Omega} \sigma_{ij}(u^-) v_i \cdot \nu \, ds. \tag{4.19}$$

Moreover, the weak form (4.9) can be written as:

$$\int_D \overline{\alpha}_\Omega \sigma(u) : e(v) \, dx = \int_\Omega \sigma(u^+) : e(v) \, dx + \eta \int_{\Omega^c} \sigma(u^-) : e(v) \, dx = \int_{\Gamma_F} \Phi v \, ds. \tag{4.20}$$

Adding equations (4.18) and (4.19) yields:

$$\begin{aligned}\int_\Omega \sigma(u^+) : e(v)\,dx + \eta \int_{\Omega^c} \sigma(u^-) : e(v)\,dx &= \int_D \overline{\alpha}_\Omega \sigma(u) : e(v)\,dx \\ &= \int_\Omega \frac{\partial \sigma_{ij}(u^+)}{\partial x_j} v_i\,dx - \int_{\partial\Omega} \sigma_{ij}(u^+) v_i \cdot \nu\,ds \\ &\quad + \eta \int_{\Omega^c} \frac{\partial \sigma_{ij}(u^-)}{\partial x_j} v_i\,dx + \eta \int_{\partial\Omega} \sigma_{ij}(u^-) v_i \cdot \nu\,ds, \end{aligned} \tag{4.21}$$

from (4.17) we have:

$$\int_\Omega \frac{\partial \sigma_{ij}(u^+)}{\partial x_j} v_i\,dx + \eta \int_{\Omega^c} \frac{\partial \sigma_{ij}(u^-)}{\partial x_j} v_i\,dx = 0, \tag{4.22}$$

then from equation (4.21) the following transmission condition is obtained:

$$-\int_{\partial\Omega} \sigma_{ij}(u) v_i \cdot \nu\,ds + \eta \int_{\partial\Omega} \sigma_{ij}(u) v_i \cdot \nu\,ds = 0. \tag{4.23}$$

□

In this case Definition 2.9 of the parameterised Lagrangian would take the following form:

$$\mathcal{L}(\varepsilon, u, p) = l|\Omega| + \int_{\Gamma_F} \Phi u\,dx + \int_D \widetilde{\alpha}_\Omega \sigma(e(u)) : e(p)\,dx + \int_{\Gamma_F} \Phi p\,dx \quad \forall v \in H_0^1(D),$$

and

$$\mathcal{L}(\varepsilon, u+tv, p) = \int_{\Gamma_F} \Phi(u+tv)\,dx + \int_D \widetilde{\alpha}_\Omega \sigma(e(u+tv)) : e(p)\,dx + \int_{\Gamma_F} \Phi p\,dx \quad \forall v \in H_0^1(D).$$

By linearity of $\sigma(.)$, $e(.)$ and the L^2 dot product $(.,.)$ we get:

$$\begin{aligned}&\lim_{t\to 0} \frac{\mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0 + tv, p_\varepsilon) - \mathcal{L}(\varepsilon, su_\varepsilon + (1-s)u_0, p_\varepsilon)}{t} = \\ &= \int_{\Gamma_F} \Phi v\,dx + \int_D \widetilde{\alpha}_\Omega \sigma(e(v)) : e(p)\,dx = 0,\end{aligned}$$

which proves that the problem is self adjoint, meaning that the adjoint is equal the state multiplied by a negative $p_\Omega = -u_\Omega$.
By using the generalized form of the asymptotic expansion (2.1) where it's been generalized in [2] to the case where the density $\tilde{\alpha}$ inside the ball $B(x,\varepsilon)$ is shifted from its initial value α_0 into the new value α_1. By using the expression of the elastic moment tensor calculated in [1], the topological derivative takes the following form in the linear elasticity 2D plane strain:

$$g(x) = \frac{r-1}{r+1} \frac{\mu+1}{2\mu(1-\nu)} \left[2\sigma(u_\Omega) : e(p_\Omega) + \frac{(r-1)(\kappa-2)}{\kappa+2r-1} \mathrm{tr}\sigma(u_\Omega)\mathrm{tr}e(p_\Omega) \right] + l\delta,$$

with

$$r = \frac{1}{\eta}, \quad \mu = \frac{\lambda+3\mu}{\lambda+\mu},$$

The constants λ and μ are the Lamé coefficients and the stress tensor $\sigma(u_\Omega)$ is computed with the local density at the point x. In plane stress, $\lambda^* = \frac{2\mu\lambda}{\lambda+2\mu}$ must be substituted for λ . The displacement field p_Ω is the adjoint state and the binary variable δ is introduced for convenience to indicate the sense of the variation of surface of Ω:

$$\delta = \begin{cases} -1 & \text{if } |\Omega| \text{ is decreased (creation of a hole),} \\ +1 & \text{if } |\Omega| \text{ is increased (addition of matter).} \end{cases}$$

4.1 Cantilever

The first case in elasticity examples is the long cantilever problem considered in [3]and [5], where the hold-all domain D is a 2×1 rectangle with homogeneous Dirichlet boundary conditions on the left side representing the fixed support, while a vertical point wise unitary load is applied at the middle of the right side of the domain, as shown in the following figure:

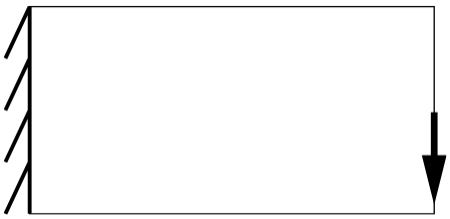


Figure 4.1: Cantilever Setting.

To compare with the literature [3] [5], we consider the same values for l which is $l = 100$ and also the same values for the material properties, which are characterized by a Young's modulus $E = 1$ and a Poisson's ratio $\nu = 0.3$. The optimization aims to distribute the material within D such that the structural compliance is minimized under the specified load and boundary conditions.
The level set is initialized by $\psi_0 = -1$ so that the initial domain of calculation is the entire rectangle. For a coarse mesh of 990 vertices and 1858 elements, the algorithm 1 converges (according to the $\theta_k \leq 1.5°$ convergence criterion) in 54 iterations to the following shape(blue):

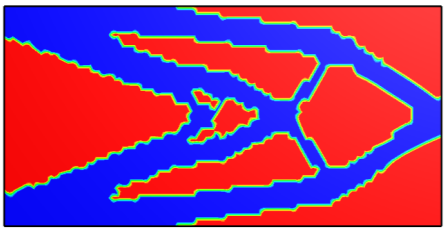


Figure 4.2: Cantilever Solution for Coarse Mesh.

The optimization process for the cantilever problem is monitored through the convergence of the number of active degrees of freedom (DOFs) and the angle θ_k. As shown in Figure 4.3, the number of active DOFs stabilizes as the optimization progresses, indicating that most of the degrees of freedom satisfy the optimality conditions (2.68)

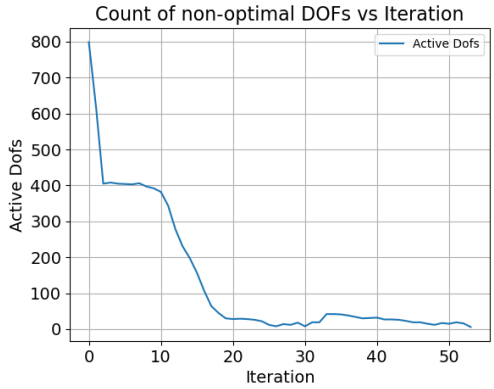


Figure 4.3: Progression of The Count of Non-Optimal Points.

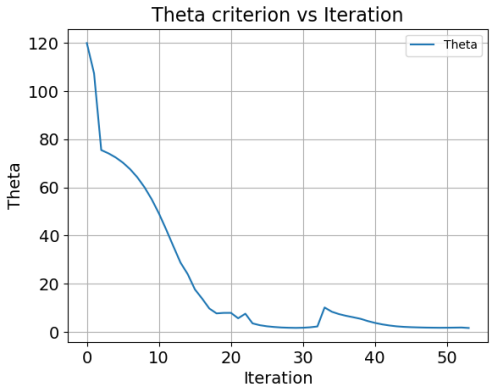


Figure 4.4: Progression of The Angle θ_k.

Figure 4.5: Convergence Behaviour for The Cantilever Case (Coarse Mesh).

Similarly, the angle θ_k convergence, depicted in Figure 4.4, demonstrates that the optimization meets the criterion $\theta_k \leq 1.5°$.
Together, these criteria provide a comprehensive assessment of the optimization's progress and stability, confirming the validity of the solution.

Refining the mesh to use a mesh of 3814 vertices and 7386 elements, the algorithm converges in the same number of iterations and with similar convergence behaviour but to a slightly different shape as shown in the following figure:

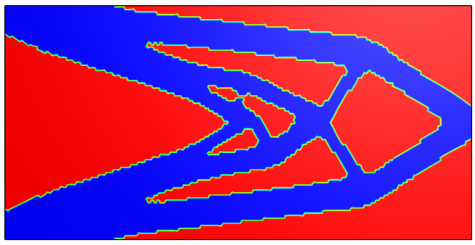


Figure 4.6: Cantilever Solution for Refined Mesh.

It's also worth mentioning that since the algorithm converges to a local minimum, the starting point ψ_0 affects the final solution, for choosing $\psi_0 = (x-1)^2 + (y-\frac{1}{2})^2 - (\frac{1}{4})^2$, the solution converges to a different shape as shown in the following figure:

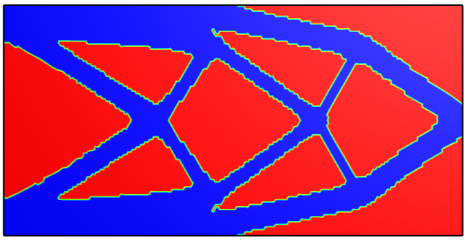


Figure 4.7: Alternative Cantilever Solution for The Refined Mesh.

The cost of this shape is $J = 153$ and the solution converges in 25 iterations, with the convergence behaviour shown in the following figures:

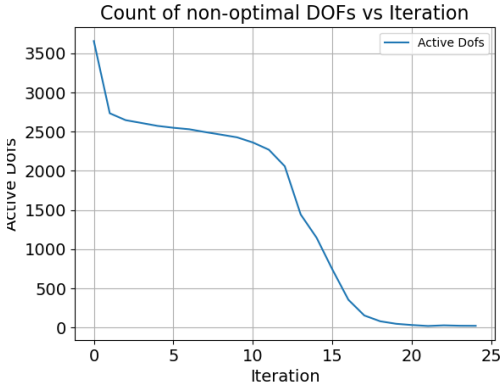


Figure 4.8: Progression of the Count of Non-Optimal Points.

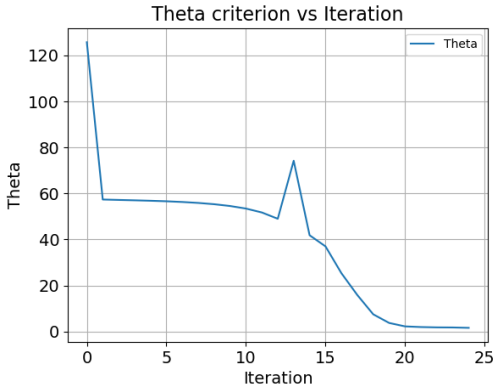


Figure 4.9: Progression of The Angle θ_k.

Figure 4.10: Convergence Behaviour for The Cantilever Case (Refined Mesh).

4.2 Bridge

Here we consider the case of a bridge under different loading conditions, namely single loaded and triple loaded. For this case, we consider a hold-all domain D as a rectangle of dimensions 2×1.2 and the bridge is to be simply supported on both ends, that is translated into homogeneous Dirichlet boundary conditions on said ends, moreover the material properties, are the same as the previous case, that is: $E = 1$ and $\nu = 0.3$, however, the volume multiplier l is different in each loading setting.

4.2.1 Single Loaded Bridge

In this case, we model the loads on the bridge by a single point downward force of magnitude 1 acting on the middle of the structure as shown in the following figure:

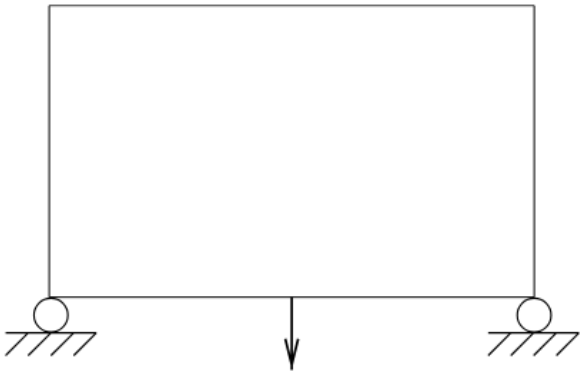


Figure 4.11: Single Loaded Bridge Setting.

Moreover, the multiplier l is chosen $l = 30$ for comparison with literature. The level set is also initialized by $\psi_0 = -1$ so that the initial calculation domain is the entire rectangle.

For a coarse mesh of 1164 vertices and 2198 elements, the Algorithm 1 converges in 37 iterations to the following shape:

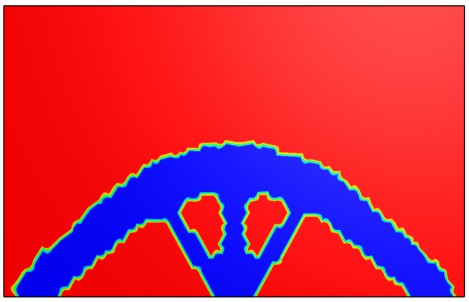


Figure 4.12: Single Loaded Bridge Solution for Coarse Mesh.

Here, the convergence behaviour is quite different from the previous case, as the angle condition is not suitable for determining convergence, that is due to the fact that as the algorithm progresses, the theta remain around 90°. However, the convergence is judged by the number of degrees of freedom that satisfy the optimality condition 2.68.

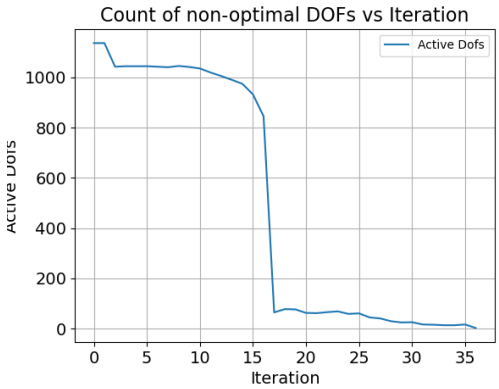


Figure 4.13: Progression of The Count of Non-Optimal Points.

Figure 4.14: Progression of The Angle θ_k.

Figure 4.15: Convergence Behaviour for The Single Loaded bridge Case (Coarse Mesh).

Refining the discretization of the hold-all domain D such that it's divided into 8914 triangular elements with 4586 vertices, leads to convergence in larger number of steps namely 48, with similar convergence behaviour, the shape is slightly different as shown in the following figure:

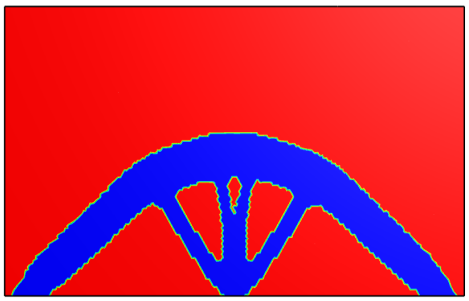


Figure 4.16: Single Loaded Bridge Solution for Refined Mesh.

Similar to previous case, the algorithm converges to a local minimum and the starting point ψ_0 affects the final solution, for choosing $\psi_0 = (x-1)^2+(y-\frac{1}{2})^2-(\frac{1}{4})^2$, the solution converges to a different shape as shown in the following figure:

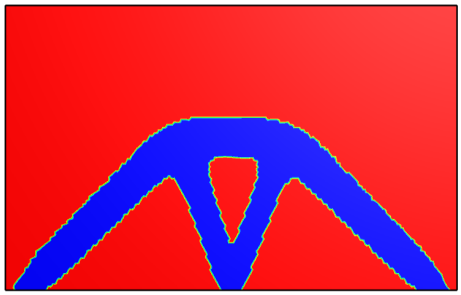


Figure 4.17: Alternative Single Loaded Bridge Solution for Refined Mesh.

For this setting, the cost is $j = 27.5$ and it converges in 31 iterations with the following convergence behaviour,

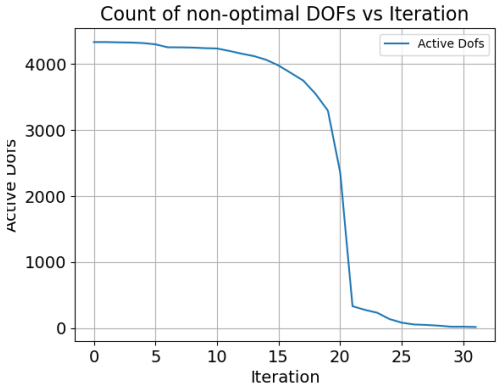


Figure 4.18: Progression of the Count of Non-Optimal Points.

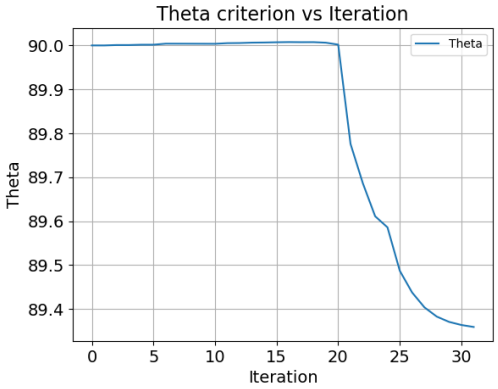


Figure 4.19: Progression of The Angle θ_k.

Figure 4.20: Convergence Behaviour for The Single Loaded Bridge Case (Refined Mesh).

4.2.2 Triple Loaded Bridge

For this case, the loads are modeled by three point forces, each with a magnitude of 1, acting in the downward direction. The points of application of these forces are located

at the center of the bridge and at positions 0.5 length units to either side of the center,as shown in the following figure:

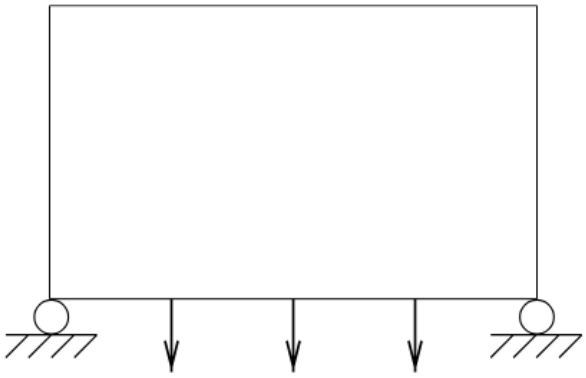


Figure 4.21: Single Loaded Bridge Setting.

Furthermore, the multiplier l is set to 100 and the level set function is initially set to $\psi_0 = -1$, ensuring that the initial computational domain encompasses the entire rectangle. Using a coarse mesh consisting of 1164 vertices and 2198 elements, algorithm 1 converges to the following shape of cost $j = 114.8$ after 37 iterations:

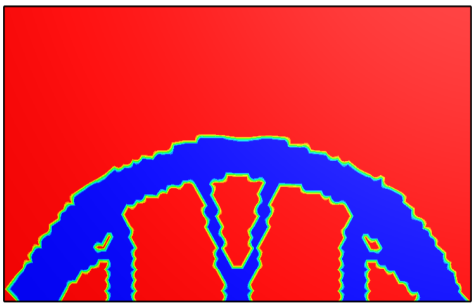


Figure 4.22: Single Loaded Bridge Solution for Coarse Mesh.

The convergence behaviour is similar to the single load case, as the angle theta remains around 90°, and the convergence is judged by the number of degrees of freedom satisfying the optimality condition 2.68

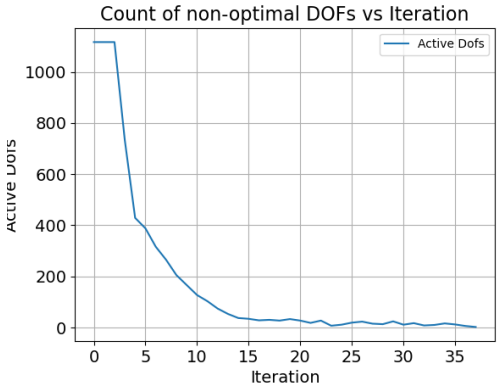


Figure 4.23: Progression of The Count of Non-Optimal Points.

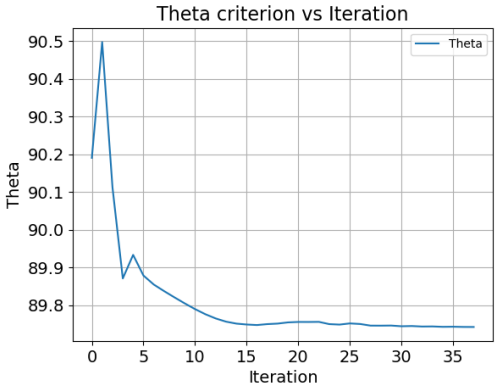


Figure 4.24: Progression of The Angle θ_k.

Figure 4.25: Convergence Behaviour for The Triple Loaded bridge Case (Coarse Mesh).

Refining the discretization of the hold-all domain D to divide it into 8914 triangular elements with 4586 vertices leads to convergence in a smaller number of steps, specifically 18. Despite the different number of steps, the convergence behavior remains similar. The resulting shape, however, is slightly different, as shown in the following figure:

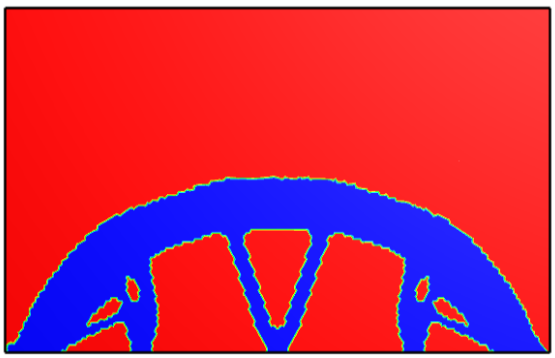


Figure 4.26: Triple Loaded Bridge Solution for Refined Mesh.

As in the previous case, the algorithm converges to a local minimum, with the initial shape ψ_0 influencing the final solution. When choosing $\psi_0 = (x-1)^2 + (y-\frac{1}{2})^2 - (\frac{1}{4})^2$ the solution converges to a different shape, as shown in the following figure:

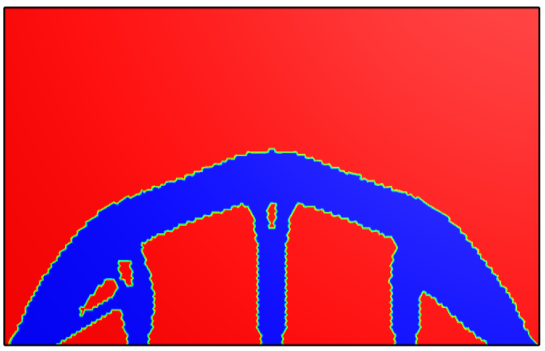


Figure 4.27: Alternative Triple Loaded Bridge Solution for Refined Mesh.

This setting demonstrates a different convergence behavior, as the number of optimum elements fluctuates until it converges after 32 iterations:

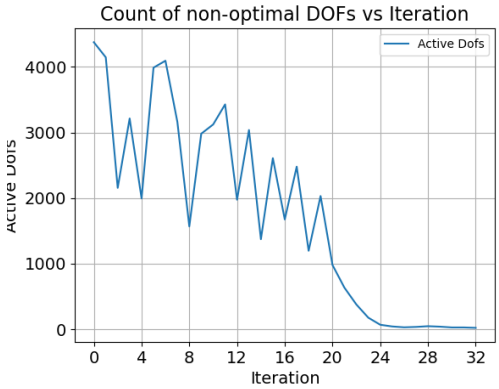


Figure 4.28: Progression of the Count of Non-Optimal Points.

Figure 4.29: Progression of the angle θ_k.

Figure 4.30: Convergence Behaviour for The Triple Loaded Bridge Case (Refined Mesh).

Chapter 5

A Lagrange-Newton Level Set Method

5.1 Newton Method

The Newton method is a widely used iterative technique for finding the roots of a real-valued function[8]. Given a suitable function f and its derivative f', the method starts with an initial guess x_0 for a root of the function. It then iteratively improves this guess for $n \in \mathbb{N}_0$ using the formula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{5.1}$$

The iterations continue until the difference between successive approximations is smaller than a predefined tolerance, indicating convergence to a root. The Newton method is particularly powerful due to its quadratic convergence property[8].

In the context of topological optimization, the Newton method can be applied to find the critical points of a function J. Here, the goal is to find Ω such that the gradient $\nabla J(\Omega) = 0$. The iterative update rule becomes:

$$\Omega_{n+1} = \Omega_n - \left(\nabla^2 J(\Omega_n)\right)^{-1} \nabla J(\Omega_n), \tag{5.2}$$

where $\nabla J(\Omega)$ is the gradient and $\nabla^2 J(\Omega)$ is the Hessian matrix of J at Ω. This method is effective in finding local minima or maxima of J, especially when the initial guess is close to the desired solution.

5.2 A Newton Method Using First and Second Topological Derivatives

Let $J : A \subset \mathcal{P}(D)$ be a twice topologically differentiable shape functional, the following problem is considered:

$$\min_{\Omega} J(\Omega, u) = \frac{1}{2} \int_D (u - u_\mathrm{d})^2 \, dx, \tag{5.3}$$

with u_d as in (3.3) and subject to the linear elliptic PDE constraint to find $u \in H^1(D)$ satisfying:

$$\int_D [\nabla u \cdot \nabla \varphi + \alpha_\Omega u \varphi]\, dx = \int_D f_\Omega \varphi \, dx. \tag{5.4}$$

By the chain rule the first and second order topological derivatives are:

$$DJ(\Omega)(x_0) = \int_D 2(u - u_d)\dot{u}_{x_0}\, dx, \tag{5.5}$$

$$D^2J(\Omega)(x_0, y_0) = \int_D 2\dot{u}_{x_0}\dot{u}_{y_0} + 2(u - u_d)\ddot{u}_{x_0,y_0}\, dx, \tag{5.6}$$

where the 1st and 2nd topological state derivatives $\dot{u}_{x_0}$ and $\ddot{u}_{x_0,y_0}$ are defined as the solution of the differential equations resulting from differentiating the weak form of the PDE (5.4) with respect to the shape Ω ,that is, for $\Omega \subset \mathbb{R}^d$ and $x_0 \in \Omega$, $\dot{u}_{x_0}$ solves:

$$\int_D \nabla \dot{u}_{x_0} \cdot \nabla \varphi + \alpha_\Omega \dot{u}_{x_0} \varphi \, dx = -\mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)u(x_0)\varphi(x_0) + \mathrm{sgn}_\Omega(x_0)(f_1 - f_2)\varphi(x_0) \tag{5.7}$$

for all $\varphi \in W^{1,q}(D)$. Moreover, $\ddot{u}_{x_0,y_0}$ solves:

$$\begin{aligned}\int_D \nabla \ddot{u}_{x_0,y_0} \cdot \nabla \varphi + \alpha_\Omega \ddot{u}_{x_0,y_0} \varphi \, dx = &-\mathrm{sgn}_\Omega(y_0)(\alpha_1 - \alpha_2)\dot{u}_{x_0}(y_0)\varphi(y_0) + \\ &- \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)\dot{u}_{y_0}(x_0)\varphi(x_0),\end{aligned} \tag{5.8}$$

together with the adjoint equation:

$$\int_D \nabla p \cdot \nabla \varphi + \alpha_\Omega p \varphi \, dx = -\int_D 2(u - u_d)\varphi \, dx \tag{5.9}$$

for all $\varphi \in H^1(D)$ and the equation for the topological state derivatives of the adjoint are obtained by differentiating the previous equation with respect to the shape Ω, that is:

$$\int_D \nabla \dot{p}_{x_0} \nabla \varphi + \alpha_\Omega \dot{p}_{x_0} \varphi \, dx = -\int_D 2\dot{u}_{x_0}\varphi \, dx - \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)p(x_0)\varphi(x_0), \tag{5.10}$$

then by testing (5.10) with $\phi = \dot{u}_{y_0}$, and equation (5.9) with $\phi = \ddot{u}_{x_0,y_0}$, equation (5.6) becomes:

$$\begin{aligned}D^2J(\Omega)(x_0)(y_0) &= \int_D 2\dot{u}_{x_0}\dot{u}_{y_0} + 2(u - u_d)\ddot{u}_{x_0,y_0}\, dx \\ &= -\int_D \nabla \dot{p}_{x_0} \nabla \dot{u}_{y_0} + \alpha_\Omega \dot{p}_{x_0}\dot{u}_{y_0}\, dx + \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)p(x_0)\dot{u}_{y_0}(x_0) \\ &\quad - \int_D \nabla p \cdot \nabla \dot{u}_{x_0,y_0} + \alpha_\Omega p \ddot{u}_{x_0,y_0}\, dx \\ &= \mathrm{sgn}_\Omega(y_0)(\alpha_1 - \alpha_2)u(y_0)\dot{p}_{x_0}(y_0) - \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)p(x_0)\dot{u}_{y_0}(x_0) \\ &\quad + \mathrm{sgn}_\Omega(y_0)(\alpha_1 - \alpha_2)\dot{u}_{x_0}(y_0)p(y_0) + \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)\dot{u}_{y_0}(x_0)p(x_0) \\ &= \mathrm{sgn}_\Omega(y_0)(\alpha_1 - \alpha_2)(u(y_0)\dot{p}_{x_0}(y_0) + \dot{u}_{x_0}(y_0)p(y_0)) \\ &= \int_D 2\dot{u}_{x_0}\dot{u}_{y_0}\, dx + \mathrm{sgn}_\Omega(y_0)(\alpha_1 - \alpha_2)\dot{u}_{x_0}(y_0)p(y_0) \\ &\quad - \mathrm{sgn}_\Omega(x_0)(\alpha_1 - \alpha_2)\dot{u}_{y_0}(x_0)p(x_0),\end{aligned}$$

or alternatively,

$$D^2J(\Omega)(x_0)(y_0) = \int_D 2\dot{u}_{x_0}\dot{u}_{y_0}\,dx + \operatorname{sgn}_\Omega(y_0)(\alpha_1-\alpha_2)\dot{u}_{x_0}(y_0)p(y_0) + \operatorname{sgn}_\Omega(x_0)(\alpha_1-\alpha_2)\dot{u}_{y_0}(x_0)p(x_0). \tag{5.11}$$

Next, the following adjoint equation is introduced:

$$\int_D \nabla q_{x_0}\cdot\nabla\varphi + \alpha_\Omega q_{x_0}\varphi\,dx = -\int_D 2\dot{u}_{x_0}\varphi\,dx, \tag{5.12}$$

then the second topological derivative becomes:

$$\begin{aligned} D^2J(\Omega)(x_0,y_0) = {} & \operatorname{sgn}_\Omega(y_0)(\alpha_1-\alpha_2)q_{x_0}(y_0)u_{x_0}(y_0) \\ & + \operatorname{sgn}_\Omega(y_0)(\alpha_1-\alpha_2)\dot{u}_{x_0}(y_0)p(y_0) \\ & + \operatorname{sgn}_\Omega(x_0)(\alpha_1-\alpha_2)\dot{u}_{y_0}(x_0)p(x_0). \end{aligned} \tag{5.13}$$

Recalling that a stationary point Ω^* of J satisfies:

$$DJ(\Omega^*)(x) \geq 0 \quad \text{for all } x \in D\setminus\partial\Omega^*.$$

The aim is to apply a Newton-type algorithm, to do so the previous inequality is formulated as an equation, that is:

$$\min\{DJ(\Omega^*)(x),0\} = 0 \quad \text{for all } x \in D\setminus\partial\Omega^*. \tag{5.14}$$

In terms of the generalized topological derivative, this condition can be written as:

$$\max\{\mathcal{D}J(\Omega^*)(x),0\} = 0 \quad x\in\Omega, \tag{5.15}$$

$$\min\{\mathcal{D}J(\Omega^*)(x),0\} = 0 \quad x\in D\setminus\Omega. \tag{5.16}$$

The target is to drive the generalized topological derivative to zero, to formulate the algorithm one needs to differentiate between points inside and outside the domain Ω and since H^1 finite elements of order 1 are used, the degrees of freedom of the system corresponds to the number of nodes in the mesh, so for a finite number $N\in\mathbb{N}_0$ of points the following index sets can be defined:

$$\mathscr{P}(\Omega) := \{x_1,\dots,x_N\}\subset D\setminus\partial\Omega,$$

$$\mathscr{P}_{in}(\Omega) := \{i\in\mathscr{P} : x_i\in\Omega\},\quad \mathscr{P}_{out}(\Omega) := \{i\in\mathscr{P} : x_i\in D\setminus\overline{\Omega}\},$$

subsequently, one can denote the generalized topological derivative discrete vector as:

$$\mathcal{D}J(\Omega)_{\mathscr{P}} = \begin{pmatrix} DJ(\Omega)_{\mathscr{P}_{in}} \\ -DJ(\Omega)_{\mathscr{P}_{out}} \end{pmatrix}, \tag{5.17}$$

the process then follows the following algorithm:

Algorithm 3 Direct Newton Level Set Algorithm.

Require: Initial level-set function ψ_0, stopping tolerance $\tau > 0$, maximum number of iterations $k_{\max} \in \mathbb{N}$

for $k = 0, 1, 2, \ldots, k_{\max}$ **do**
  Compute the destination state u_d
  Compute the state u from the system for the weak form (5.4)
  Compute the adjoint p from (5.10)
  Compute the first topological state derivative from (5.7)
  Compute $\theta_k = \arccos\left(\frac{(\psi_k, DJ(\Omega_k))_{L^2(D)}}{\|\psi_k\|_{L^2(D)}\|DJ(\Omega_k)\|_{L^2(D)}}\right)$
  if $\theta_k < \tau$ **OR** $N_{\text{opt}} \geq \text{Int}(0.995 \times N)$ **then**
    Stop with approximate minimizer Ω_k
  end if
  Compute the second topological derivative from (5.13)
  Define the direction vector $\alpha \in \mathbb{R}^N$ as the solution to:

$$\begin{pmatrix} (D^2J(\Omega_k))_{\mathscr{P}_{k,in},\mathscr{P}_{k,in}} & -(D^2J(\Omega_k))_{\mathscr{P}_{k,in},\mathscr{P}_{k,out}} \\ -(D^2J(\Omega_k))_{\mathscr{P}_{k,out},\mathscr{P}_{k,in}} & (D^2J(\Omega_k))_{\mathscr{P}_{k,out},\mathscr{P}_{k,out}} \end{pmatrix} (\alpha) = -\mathcal{D}J(\Omega_k)_{\mathscr{P}_k}$$

  Update the level-set function: $\psi_{k+1} = \psi_k + \alpha$
end for

5.2.1 Test case for the direct Newton level set algorithm

In this section, $\alpha_1, \alpha_2 \in \mathbb{R}$ constants are considered, it is known from section 3.3 that the clover shaped optimizing domain Ω^* has adjoint value zero for the cost function (2.27)and hence the topological derivative at said optimizer is also zero.

It is then observed that after just one step, the direction of the level-set function is very close to the destination clover shape represented by ψ_d. This indicates that the initial step in the algorithm effectively captures the correct direction towards the optimal shape.

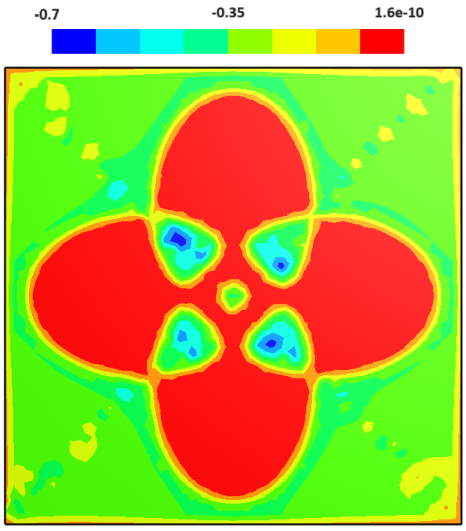


Figure 5.1: Direction after 1 Iteration.

Algorithm Performance

Despite the promising initial direction, the Newton-Level set Algorithm tends to be slow. This slowness can be attributed to several factors rooted in the underlying theory and computational requirements:

- **Complexity of Solving PDEs:** Each iteration of the algorithm involves solving a weak form of the PDE to find the state u. This process is computationally intensive, particularly for large domains D and fine discretization.
- **Adjoint Equations and Derivative Calculations:** The algorithm requires solving adjoint equations to compute the gradients necessary for the optimization process. This step is essential for calculating the topological derivatives but adds to the overall computational load, especially the adjoint equation (5.10). As for this equation, computing the topological state derivatives (Equations (5.7) and (5.12)) is necessary, which also requires solving linear systems that are computationally demanding.
- **Topological state Derivative Complexity:** The calculation of the first and second topological state derivatives involves solving large linear systems, which is inherently slow. As per the theoretical framework, these derivatives are crucial for the Newton step in the optimization but significantly increase the computation time due to the size and complexity of the systems involved.
- **Sparse Matrix Operations:** While the algorithm leverages the sparse structure of the matrices involved, the assembly and solution of these sparse systems still require considerable computational effort. Despite the computational efficiency of sparse solvers, the operations remain time-consuming for large-scale problems.

5.3 General Lagrange-Newton's Method for Constrained Optimization

To avoid some of the previously mentioned issues, another approach to Newton's method is considered, which utilizes the definition of the Lagrangian function and hence avoids the direct calculation of the cumbersome topological state derivative.

In general terms, the optimization problem is formulated as follows:

$$\min_{u \in \mathcal{U}} J(u),$$

subject to the PDE constraint (in weak form):

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega yv \quad \text{in } \Omega, \qquad \forall v \in H^1(\Omega). \tag{5.18}$$

Here, $\mathcal{U} \subset L^2(\Omega)$ represents an appropriate function space, typically a Sobolev space such as $H^1(\Omega)$, which ensures that u possesses the required regularity. The function $J : \mathcal{U} \to \mathbb{R}$ is the objective functional or the criterion to be minimized, and $y \in L^2(\Omega)$ is the given data in the constraint equation.

To solve this problem using Newton's method, we employ the method of Lagrange multipliers, which introduces auxiliary variables (Lagrange multipliers) to transform the constrained problem into an unconstrained one.

Definition 5.1 (Lagrangian Function)**.** *The Lagrangian function incorporates the constraint into the objective functional using the adjoint variable $p \in H^1(\Omega)$ a Lagrange multiplier:*

$$\mathcal{L}(y,u,p) = J(u) + \int_\Omega \nabla u \cdot \nabla p \, dx - \int_\Omega yp \, dx.$$

Definition 5.2 (KKT Conditions [8])**.** *The necessary conditions for optimality, known as the Karush-Kuhn-Tucker (KKT) conditions, are derived by setting the gradients of the Lagrangian with respect to y, u and p to zero, that is:*

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial y}(y,u,p)(v_1) &= -\int_\Omega p v_1 \, dx = 0, \\
\frac{\partial \mathcal{L}}{\partial u}(y,u,p)(v_2) &= \frac{\partial J}{\partial u}(v_2) + \int_\Omega \nabla p \cdot \nabla v_2 \, dx = 0, \\
\frac{\partial \mathcal{L}}{\partial p}(y,u,p)(v_3) &= \int_\Omega \nabla u \cdot \nabla v_3 \, dx - \int_\Omega y v_3 \, dx = 0, \\
&\text{with } v_1 \in L^2(\Omega),\, v_2\,, v_3 \in H^1(\Omega).
\end{aligned}$$

Newton's Method: To apply Newton's method,the classical Hessian of the Lagrangian is used. The first and second order derivative of the Lagrangian are calculated at the current guess (y_k, u_k, p_k), the target solving the resulting system of linear equations iteratively. The linearized system at iteration k is given by:

$$H(y,u,p) := \left(\begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial y^2} & \frac{\partial^2 \mathcal{L}}{\partial y \partial u} & \frac{\partial^2 \mathcal{L}}{\partial y \partial p} \\ \frac{\partial^2 \mathcal{L}}{\partial u \partial y} & \frac{\partial^2 \mathcal{L}}{\partial u^2} & \frac{\partial^2 \mathcal{L}}{\partial u \partial p} \\ \frac{\partial^2 \mathcal{L}}{\partial p \partial y} & \frac{\partial^2 \mathcal{L}}{\partial u \partial p} & \frac{\partial^2 \mathcal{L}}{\partial p^2} \end{pmatrix} (y,u,p) \right) \begin{pmatrix} \delta y_k \\ \delta u_k \\ \delta p_k \end{pmatrix} = - \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial y} \\ \frac{\partial \mathcal{L}}{\partial u} \\ \frac{\partial \mathcal{L}}{\partial p} \end{pmatrix} (y,u,p). \tag{5.19}$$

Algorithm

1. **Initialize:** Choose an initial guess (y_0, u_0, p_0).

2. **Iteration:** For each iteration k, solve the linear system:

$$(H_k(y_k,u_k,p_k)) \begin{pmatrix} \delta y_k \\ \delta u_k \\ \delta p_k \end{pmatrix} = - \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial y} \\ \frac{\partial \mathcal{L}}{\partial u} \\ \frac{\partial \mathcal{L}}{\partial p} \end{pmatrix} (y_k,u_k,p_k).$$

3. **Update:**

$$\begin{aligned}
y_{k+1} &= y_k + \delta y_k, \\
u_{k+1} &= u_k + \delta u_k, \\
p_{k+1} &= p_k + \delta p_k.
\end{aligned}$$

4. **Convergence Check:** Check if the updates $\delta y_k, \delta u_k$ and δp_k are sufficiently small. If they are, stop the iteration.

The function spaces involved in this formulation are crucial for ensuring the well-posedness and convergence of the algorithm. The choice of Sobolev spaces $H^1(\Omega)$ for u reflects the underlying PDE constraints and the nature of the optimization problem. The mappings between these spaces, particularly the differential operators and their adjoints, are essential for accurately representing the problem and applying Newton's method effectively.

5.4 Lagrange-Newton Method in Level Set Algorithm

The computation of the second derivative is costly as each evaluation $D^2J(\Omega)(x_i, x_j)$ requires the solution of a partial differential equation. Therefore, it is convenient to formulate the Lagrange-Newton algorithm in terms of adjoint variables, which leads to the solution of a larger system, namely a system of size $3N$ instead of N.

Let Ω be a shape that is not optimal, then the idea is to apply the Newton algorithm only to the points which don't satisfy the optimality Condition 2.68, that is x_i, $\forall i \in \mathscr{P}(\Omega)$, where $DJ(\Omega)(x_i) < 0$. For this purpose, we introduce the sets:

$$\mathscr{A}_{in}(\Omega) := \{i \in \mathscr{P}_{in}(\Omega) : DJ(\Omega)(x_i) < 0\}, \quad \mathscr{A}_{out}(\Omega) := \{i \in \mathscr{P}_{out}(\Omega) : DJ(\Omega)(x_i) < 0\},$$

and similarly:

$$\mathscr{I}_{in}(\Omega) := \{i \in \mathscr{P}_{in}(\Omega) : DJ(\Omega)(x_i) \geq 0\}, \quad \mathscr{I}_{out}(\Omega) := \{i \in \mathscr{P}_{out}(\Omega) : DJ(\Omega)(x_i) \geq 0\},$$

so that the set $\mathscr{I}(\Omega) = \mathscr{I}_{in}(\Omega) \cup \mathscr{I}_{out}(\Omega)$ represent the indices of optimal degrees of freedom, and the set $\mathscr{A}(\Omega) = \mathscr{A}_{in}(\Omega) \cup \mathscr{A}_{out}(\Omega)$ represents the indices of non optimal degrees of freedom.

Definition 5.3. *Let $L : A \times H^1(D) \times H^1(D) \to \mathbb{R}$ be a shape functional which is twice topologically differentiable at Ω, with $A \subset \Xi$. Then we introduce the notation of topological differentiation at $x_0, y_0 \in D \setminus \partial\Omega$:*

$$\partial_\Omega L(\Omega, u, p)(x_0) := \lim_{\varepsilon \to 0} \frac{L(\Omega_\varepsilon(x_0), u, p) - L(\Omega, u, p)}{|\omega_\varepsilon|}, \tag{5.20}$$

$$\partial^2_\Omega L(\Omega, u, P)(x_0)(y_0) := \lim_{\varepsilon \to 0} \frac{\partial_\Omega L(\Omega_\varepsilon(y_0), u, p)(x_0) - \partial_\Omega L(\Omega, u, p)(x_0)}{|\omega_\varepsilon|}. \tag{5.21}$$

For index sets $\mathscr{A}$ and $\mathscr{B}$ we introduce the notation:

$$\partial_\Omega J(\Omega)_{\mathscr{A}} := (DJ(\Omega)(x_i))_{i \in \mathscr{A}},$$

and

$$\partial^2_\Omega J(\Omega)_{\mathscr{A}\mathscr{B}} := (D^2J(\Omega)(x_i, x_j))_{i \in \mathscr{A}, j \in \mathscr{B}}.$$

Thus $\partial_\Omega J(\Omega)_{\mathscr{A}}$ is a column vector and $\partial^2_\Omega J(\Omega)_{\mathscr{A}\mathscr{B}}$ is a rectangular matrix.

Lemma 5.1. *For the Lagrangian:*

$$\mathcal{L}(\Omega, u, p) := \int_D (u - u_d)^2 \, dx + \int_D \nabla u \cdot \nabla p + \alpha_\Omega up - f_\Omega p \, dx. \tag{5.22}$$

With u_d as in (3.3), $u, p \in H^1(D)$ and $\Omega \subset D$ the partial derivatives with respect to each of the three variables Ω, u and p are calculated as follows:

$$\partial_\Omega \mathcal{L}(\Omega, u, p)(x_j) = sgn_\Omega(x_j)(\alpha_1 - \alpha_2)u(x_j)p(x_j) - sgn_\Omega(x_j)(f_1 - f_2)p(x_j), \tag{5.23}$$

$$\partial_u \mathcal{L}(\Omega, u, p)(\phi_i) = \int_D 2(u - u_d)\phi_i \, dx + \int_D \nabla p \cdot \nabla \phi_i \, dx + \int_D \alpha_\Omega p \phi_i \, dx, \tag{5.24}$$

$$\partial_p \mathcal{L}(\Omega, u, p)(\psi_i) = \int_D \nabla u \cdot \nabla \psi_i \, dx + \int_D \alpha_{\Omega_i} u \psi_i - \int_D f_\Omega \psi_i \, dx, \tag{5.25}$$

$$\forall \phi_i, \psi_i \in H^1(D), i \in \mathscr{P}(\Omega)$$

Lemma 5.2. *Under the setting of the previous Lemma, the second derivatives of the Lagrangian (5.22) are given as follows:*

$$\partial_{\Omega\Omega} \mathcal{L}(\Omega, u, p)(x_j)(x_i) = 0, \tag{5.26}$$

$$\partial_{uu} \mathcal{L}(\Omega, u, p)(\phi_i)(\phi_j) = \int_D 2\phi_i \phi_j \, dx, \tag{5.27}$$

$$\partial_{pp} \mathcal{L}(\Omega, u, p)(\psi_i)(\psi_j) = 0 \tag{5.28}$$

$$\partial_{u\Omega} \mathcal{L}(\Omega, u, p)(\phi_i)(x_j) = sgn(x_j)(\alpha_1 - \alpha_2)\phi_i(x_j)p(x_j) \tag{5.29}$$

$$\partial_{p\Omega} \mathcal{L}(\Omega, u, p)(\psi_i)(x_j) = sgn(x_j)(\alpha_1 - \alpha_2)\psi_i(x_j)u(x_j) - sgn(x_j)(f_1 - f_2)\psi_i(x_j), \tag{5.30}$$

$$\partial_{up} \mathcal{L}(\Omega, u, p)(\phi_i)(\psi_j) = \int_D \nabla \phi_i \cdot \nabla \psi_j \, dx + \int_D \alpha_\Omega \phi_i \psi_j \, dx. \tag{5.31}$$

$$\forall \phi_i, , \psi_j \in H^1(D), i, j \in \mathscr{P}(\Omega)$$

For the Lagrangian in the previous lemma, one can define the derivatives separately on each of the previously defined index sets, obtaining the notion for discrete vectors as follow:

$$F(\Omega, u, p) := \begin{pmatrix} \partial_\Omega \mathcal{L}(\Omega, u, p)_{\mathscr{I}_{\text{in}}} \\ -\partial_\Omega \mathcal{L}(\Omega, u, p)_{\mathscr{I}_{\text{out}}} \\ \partial_\Omega \mathcal{L}(\Omega, u, p)_{\mathscr{A}_{\text{in}}} \\ -\partial_\Omega \mathcal{L}(\Omega, u, p)_{\mathscr{A}_{\text{out}}} \\ \partial_u \mathcal{L}(\Omega, u, p)_{\mathscr{P}} \\ \partial_p \mathcal{L}(\Omega, u, p)_{\mathscr{P}} \end{pmatrix} \tag{5.32}$$

Where the vector $F(\Omega, u, p)$ consists of six vectors, each has the length of the corresponding index set, then the linear system to be solved with the generalized Jacobian of this function is then given by:

$$\begin{pmatrix} \mathbf{Id} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Id} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{in}} & (\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{in}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -(\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{out}} & -(\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{out}} \\ \mathbf{0} & \mathbf{0} & (\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{uu}\mathcal{L} & \partial_{up}\mathcal{L} \\ \mathbf{0} & \mathbf{0} & (\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{pu}\mathcal{L} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \alpha_{\mathscr{I}_{in}} \\ \alpha_{\mathscr{I}_{out}} \\ \alpha_{\mathscr{A}_{in}} \\ \alpha_{\mathscr{A}_{out}} \\ \delta u \\ \delta p \end{pmatrix} = -F(\Omega, u, p). \tag{5.33}$$

Where each of the elements of the matrix on the left-hand side is a matrix in itself, with the number of rows corresponding to the number of elements in the index set on which

the first derivative is performed. This means that if the set $\mathscr{P}(\Omega)$ has N elements, and the set $\mathscr{A}_{\text{in}}$ has N_1 elements, then the matrix $\partial_{uu}\mathcal{L}$ is a square $N \times N$ matrix, and the matrix $(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{\text{in}}}$ is a rectangular $N \times N_1$ matrix.

The target now is to solve the previous linear system in an iterative manner to drive the solution vector $\big(\alpha_{\mathscr{I}_{in}},\ \alpha_{\mathscr{I}_{out}},\ \alpha_{\mathscr{A}_{in}},\ \alpha_{\mathscr{A}_{out}},\ \delta u,\ \delta p\big)^T$ to zero, at which point the KKT conditions in Definition 5.2 and subsequently the optimality conditions (2.68) are satisfied. This is achieved by the following algorithm:

Algorithm 4 Lagrange-Newton-Level Set Method

Require: Initial domain $\Omega_0 \subset D$, number of points $N \in \mathbb{N}_0$, points $x_1, \dots, x_N \in D \setminus \Omega_0$, iteration counter $k = 0$, maximum number of iterations k_{max} level set function $\psi_0 : D \to \mathbb{R}$ with $\Omega_0 = \{x : \psi_0(x) < 0\}$

1: **for** $k = 0, 1, 2, \dots, k_{max}$ **do**

2: Define a vector $\alpha \in \mathbb{R}^{3N}$ as follows: The components $\alpha_{\mathscr{A}_{\text{in}_k}}$ and $\alpha_{\mathscr{A}_{\text{out}_k}}$ are given as part of the solution of

$$\begin{pmatrix} \mathbf{Id} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Id} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{in}} & (\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{in}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -(\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{out}} & -(\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{out}} \\ \mathbf{0} & \mathbf{0} & (\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{uu}\mathcal{L} & \partial_{up}\mathcal{L} \\ \mathbf{0} & \mathbf{0} & (\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{pu}\mathcal{L} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \alpha_{\mathscr{I}_{\text{in}},k} \\ \alpha_{\mathscr{I}_{\text{out}},k} \\ \alpha_{\mathscr{A}_{\text{in}},k} \\ \alpha_{\mathscr{A}_{\text{out}},k} \\ \delta u_k \\ \delta p_k \end{pmatrix} = -\big(F(\Omega, u, p)\big)$$

3: **if** $N_{\text{opt}} \geq \text{Int}(0.995 \times N)$ **then**

4: Stop with approximate minimizer Ω_k

5: **end if**

6: Make a level set update $(\psi_{k+1})_i := (\psi_k)_i + \alpha_i$ for $i = 1, \dots, N$ and set $\Omega_{k+1} := \{x : \psi_{k+1}(x) > 0\}$.

7: $k \leftarrow k + 1$

8: **end for**

The corresponding linear equations of the system (5.33) are:

$$\alpha_{\mathscr{I}_{in}} = -(\partial_\Omega \mathcal{L})_{\mathscr{I}_{in}} \tag{5.34}$$

$$\alpha_{\mathscr{I}_{out}} = (\partial_\Omega \mathcal{L})_{\mathscr{I}_{out}} \tag{5.35}$$

$$(\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{in}}\delta u + (\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{in}}\delta p = -(\partial_\Omega \mathcal{L})_{\mathscr{A}_{in}} \tag{5.36}$$

$$-(\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{out}}\delta u - (\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{out}}\delta p = (\partial_\Omega \mathcal{L})_{\mathscr{A}_{out}} \tag{5.37}$$

$$(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{in}}\alpha_{\mathscr{A}_{in}} - (\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{out}}\alpha_{\mathscr{A}_{out}} + \partial_{uu}\mathcal{L}\delta u + \partial_{up}\mathcal{L}\delta p = -(\partial_u \mathcal{L})_{\mathscr{P}} \tag{5.38}$$

$$(\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{in}}\alpha_{\mathscr{A}_{in}} - (\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{out}}\alpha_{\mathscr{A}_{out}} + \partial_{pu}\mathcal{L}\delta u = -(\partial_p \mathcal{L})_{\mathscr{P}}. \tag{5.39}$$

So that the values of $\alpha_{\mathscr{I}_{in}}$ and $\alpha_{\mathscr{I}_{out}}$ are set to the values of the right hand F, which is the topological derivative, as in those points the optimality condition is satisfied so it's not required to solve for the Newton direction there.

Moreover, the coupling between the direction terms $\alpha_{\mathscr{A}_{in}}$ and $\alpha_{\mathscr{A}_{out}}$ with the state and the adjoint increments δu and δp is due to equations (5.38) and (5.39), for if the term $(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}}$ is equal to zero, then the coupling is due to the latter only.

5.4.1 Lagrange-Newton Method Application on a Sparse System

We consider the cost functional:

$$J(\Omega) = \int_D (u - u_d + 10\,xy)^2\,dx, \tag{5.40}$$

with u_d as in (3.3), subject to, the weak form of the Poisson equation with $u \in H^1(D)$ solving:

$$\int_D \nabla u \cdot \nabla\varphi + u\varphi\,dx = \int_D f_\Omega\varphi\,dx \quad \forall\varphi \in H^1(D), \tag{5.41}$$

with u_d being prescribed the Lagrangian:

$$\mathcal{L}(\Omega, u, p) := \int_D (u - u_d)^2\,dx + \int_D \nabla u \cdot \nabla p + up - f_\Omega p\,dx. \tag{5.42}$$

Then the right hand side of the linear system $F(\Omega, u, p)$ in equation (5.32) is given by:

$$F(\Omega,u,p) = \begin{pmatrix} -(f_1 - f_2)p_\Omega(x_1) \\ \vdots \\ -(f_1 - f_2)p_\Omega(x_N) \\ \\ \int_D 2(u-u_d)(\varphi_1)\,dx + \int_D \nabla(\varphi_1)\nabla p + \alpha_\Omega(u)(\varphi_1)p\,dx \\ \vdots \\ \int_D 2(u-u_d)(\varphi_N)\,dx + \int_D \nabla(\varphi_N)\nabla p + \alpha_\Omega(u)(\varphi_N)p\,dx \\ \\ \int_D \nabla u \cdot \nabla(\psi_1) - f_\Omega(\psi_1)\,dx \\ \vdots \\ \int_D \nabla u \cdot \nabla(\psi_N) - f_\Omega(\psi_N)\,dx \end{pmatrix} \tag{5.43}$$

which leads to the Jacobian:

$$\partial F(\Omega, u, p) := \begin{pmatrix} \partial^2_{\Omega\Omega}\mathcal{L} & \partial^2_{u\Omega}\mathcal{L} & \partial^2_{p\Omega}\mathcal{L} \\ \partial^2_{\Omega u}\mathcal{L} & \partial^2_{uu}\mathcal{L} & \partial^2_{pu}\mathcal{L} \\ \partial^2_{\Omega p}\mathcal{L} & \partial^2_{up}\mathcal{L} & \partial^2_{pp}\mathcal{L} \end{pmatrix}. \tag{5.44}$$

This system is of the form:

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} & X \\ \mathbf{0} & M & D \\ X^T & D & \mathbf{0} \end{pmatrix}, \tag{5.45}$$

with:

$$X := -(f_1 - f_2)\psi_j(x_i), \quad i,j = 1,\dots,N, \tag{5.46}$$

$$M := \left(\int_D 2\varphi_i\varphi_j\,dx\right), \quad i,j = 1,\dots,N. \tag{5.47}$$

$$D := \left(\int_D \nabla\varphi_i \cdot \nabla\psi_j +\right), \quad i,j = 1,\dots,N, \tag{5.48}$$

where, φ_i, ψ_j are basis of $H^1(D)$ and x_i are points in D.
If linear finite elements are chosen for the discrtisation, So $\{x_1, \ldots, x_N\}$ are the nodes in the triangulation of D and we choose the Lagrange basis $\{\varphi_1, \ldots, \varphi_N\} = \{\psi_1, \ldots, \psi_N\}$, such that $\varphi_i(x_j) = \delta_{ij}$. With this choice, the matrix X becomes essentially the identity matrix

$$(X)_{ij} = (-(f_1 - f_2))\,\delta_{ij}. \tag{5.49}$$

To avoid the issue with the irregularity caused from the coupling of the direction,the state and the adjoints increments in the Newton system, we augment the solution by progressing only the level set function ψ_k with the solution from the Newton system, meanwhile the state and the adjoint are obtained by solving the PDE at each level set ψ_k, that is achieved by setting the last by setting the last $2N$ elements of the vector in 5.50 to zero, that is:

$$F(\Omega, u, p) = \begin{pmatrix} -(f_1 - f_2)p_\Omega(x_1) \\ \vdots \\ -(f_1 - f_2)p_\Omega(x_N) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \tag{5.50}$$

using this right hand side in Algorithm 4 leads to the convergence in 10 steps, in the sense that after 10 steps 99.5% of elements are satisfying the optimality conditions 2.68, the cost of the optimal shape is $J = 2240.37$.

Figure 5.2: Lagrange-Newton System Solution.

Despite converging to the same result as in Figure 3.1, the progression of solution is different as can be seen in the following figures:

5.4.2 Lagrange-Newton test case with a jump in linear term

Similarly, we can apply this to the system with the piece wise jump in the linear term, that is changing the constraint to:

$$\int_D \nabla u \cdot \nabla \varphi + \alpha_\Omega u \varphi \, dx = \int_D f_\Omega \varphi \, dx \quad \text{for all } \varphi \in H^1(D). \tag{5.51}$$

Figure 5.3: Shape Evolution for Algorithm 4.

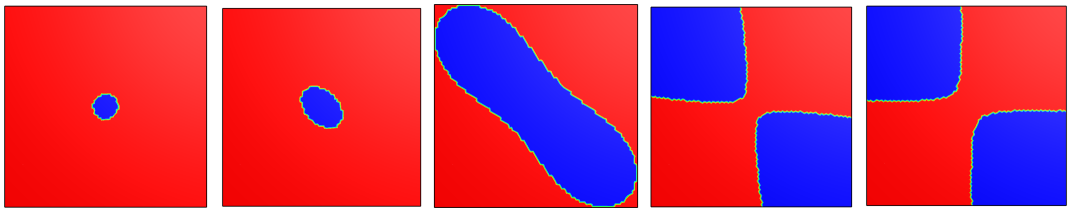


Figure 5.4: Shape Evolution for Algorithm 1.

with α_Ω as in the previous cases. the difference in this case will be in the formulation of the Jacobian (5.44), as it will take the form:

$$\begin{pmatrix} \mathbf{0} & Y & X \\ Y^T & M & D \\ X^T & D & \mathbf{0} \end{pmatrix}, \tag{5.52}$$

with:

$$X := (\alpha_1 - \alpha_2)\psi_j(x_i)u(x_j) - (f_1 - f_2)\psi_j(x_i), \quad i, j = 1, \dots, N, \tag{5.53}$$

$$Y := ((\alpha_1 - \alpha_2)\varphi_j(x_i))p(x_i), \tag{5.54}$$

$$M := \left(\int_D 2\varphi_i\varphi_j \, dx \right), \quad i, j = 1, \dots, N, \tag{5.55}$$

$$D := \left(\int_D \nabla\varphi_i \cdot \nabla\psi_j + \alpha_\Omega\varphi_i\psi_j \, dx \right), \quad i, j = 1, \dots, N. \tag{5.56}$$

When setting $\alpha_1 = 7$ and $\alpha_2 = 2$, Algorithm 4 converges in 27 iteration to the following shape of cost $J = 2735.47$:

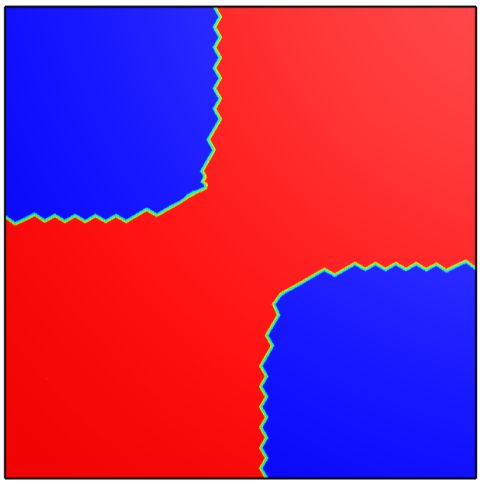


Figure 5.5: Minimizing Domain (Blue).

Similar to the previous case, the solution evolves differently than the same case solved by Algorithm 1 as shown in the following figures:

Figure 5.6: Shape Evolution for Algorithm 4.

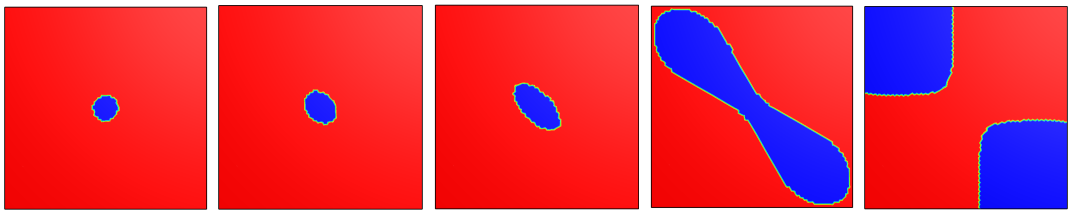


Figure 5.7: Shape Evolution for Algorithm 1.

Convergence

The number of elements in the set $\mathscr{A}(\Omega) = \mathscr{A}_{in}(\Omega) \cup \mathscr{A}_{out}(\Omega)$ indicates the number of finite elements which are still away from the optimality conditions. As the newton system progresses, $\mathscr{I}(\Omega)$ has less and less element, this number of elements can be used as an indication for convergence to compare with other algorithms, such comparison with the convex combination method from chapter 3 is provided in the following figure.

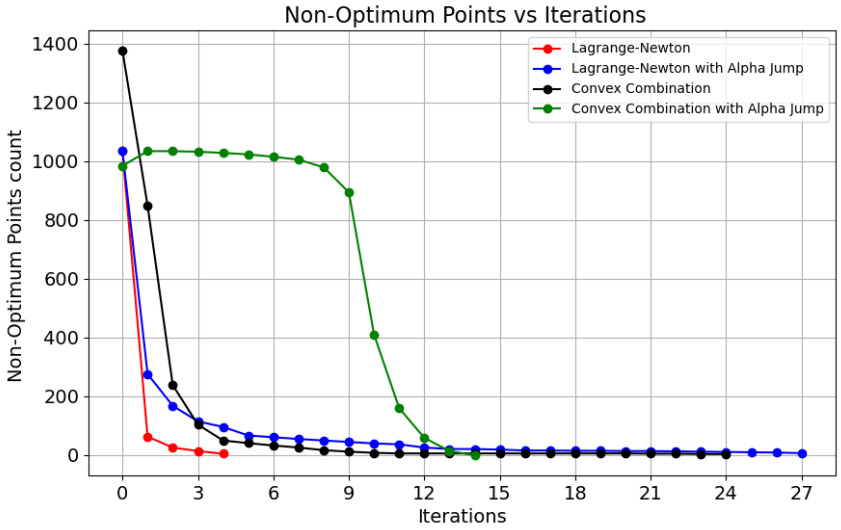


Figure 5.8: Evolution of The Number of Non-Optimal Points Comparison.

It's' clear from the figure that the drop in the number of non optimal point is significant in the first few steps of the Lagrange-Newton algorithm, which makes it's convergence faster than the level-set algorithm utilizing only the first topological derivative.

Alternative Formulation

In the previous example, the solution converged to the 99.5% optimum elements criterion in a relatively small number of iterations, however, each step requires solving $3N \times 3N$ linear system in order to solve for the Newton direction. Since the elements of the direction vector corresponding to indices in $\mathscr{I}_{in}(\Omega) \cup \mathscr{I}_{out}(\Omega)$ (elements already satisfying optimality condition 2.68) are set to $(\partial_\Omega \mathcal{L})_{\mathscr{I}_{in}(\Omega)}$ and $-(\partial_\Omega \mathcal{L})_{\mathscr{I}_{out}(\Omega)}$ respectively due to the formulation of the linear system, one can avoid solving for this points to make the system progressively smaller. Namely at each iteration, the following system is solved:

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} & (\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{in}} & (\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{in}} \\ \mathbf{0} & \mathbf{0} & -(\partial_{\Omega u}\mathcal{L})_{\mathscr{A}_{out}} & -(\partial_{\Omega p}\mathcal{L})_{\mathscr{A}_{out}} \\ (\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{u\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{uu}\mathcal{L} & \partial_{up}\mathcal{L} \\ (\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{in}} & -(\partial_{p\Omega}\mathcal{L})_{\mathscr{A}_{out}} & \partial_{pu}\mathcal{L} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \alpha_{\mathscr{A}_{in}} \\ \alpha_{\mathscr{A}_{out}} \\ \delta u \\ \delta p \end{pmatrix} = \begin{pmatrix} \partial_\Omega \mathcal{L}_{\mathscr{A}_{in}} \\ -\partial_\Omega \mathcal{L}_{\mathscr{A}_{out}} \\ \partial_u \mathcal{L} \\ \partial_p \mathcal{L}. \end{pmatrix} \tag{5.57}$$

It's important to note that the points represented by indices in $\mathscr{A}(\Omega)$ or $\mathscr{I}(\Omega)$ are not contingent in space, which means the vectors representing the derivatives at these sets (either on the right hand side or in the Jacobian) are also irregular.
Therefore,the direction vector obtained from this system can often be irregular or exhibit oscillations. These irregularities can negatively impact the convergence and stability of the optimization process. To mitigate this issue, a smoothing technique is employed, which is integrated into the iterative loop of the algorithm.

Smoothing The smoothing is achieved by solving an additional PDE that enforces a smoothness constraint on the direction vector. The smoothed direction vector is then used in place of the original direction vector in the optimization algorithm. The smoothing PDE is given by:

$$-\epsilon\Delta w + w = \alpha, \tag{5.58}$$

Where α is the direction vector and ϵ is chosen to be 10^{-5}.

With this formulation, the algorithm reaches the stopping criteria in the same number of iterations as the $3N \times 3N$ system, yet at each step the system size is $(2N + (N1 + N2))$ and $(N1 + N2)$ are the number of non optimal points, this can be seen in the following figure:

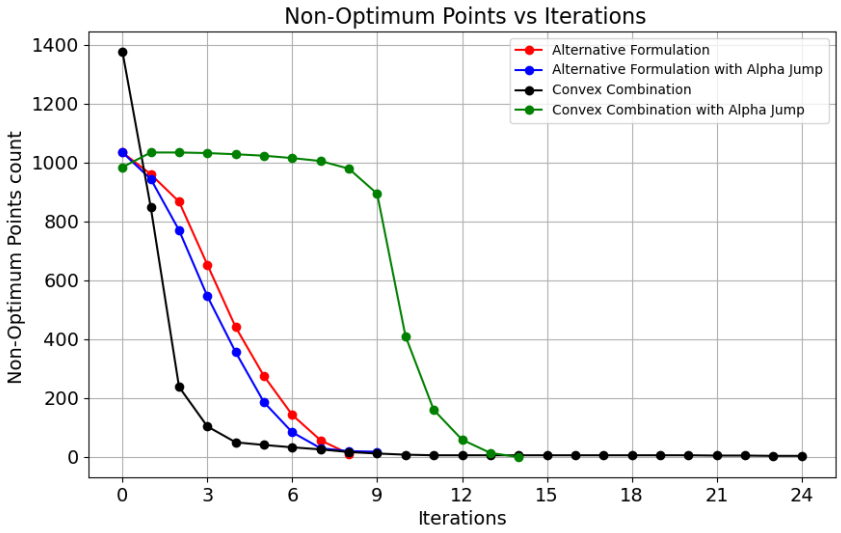


Figure 5.9: Evolution of The Number of Non-Optimal Points with Alternative Formulation.

5.5 Parallels to Control Problem

Due to the novelty of the approach, it's challenging to find reference cases in order to verify the obtained results, hence it's beneficial to compare the topological optimization problem to the well known control optimization problem.

5.5.1 Control Problem

In this section, we consider a minimization problem where the criterion is constrained by an elliptic linear PDE, that is:

$$\min_u F(u) = \min_u \int_D (u - u_d)^2$$

subject to:

$$-\Delta u + u = y \tag{5.59}$$

with homogeneous Neumann boundary conditions, or in weak formulation, u is the solution of:

$$\int_D \nabla u \cdot \nabla v \; + \int_D uv = \int_D vy \qquad \forall v \in H^1(D). \tag{5.60}$$

Here $u_d \in H^1(D)$ is a given target function.

For equation (5.60), the condition on the right hand side $y \in L^2$ is sufficient for the existence and uniqueness of the solution [1], so in this case the optimization variable $y \in L^2(D)$, unlike the previous case, where the variable of optimization was $\psi \in H^1(D)$. The Lagrange-Newton equation for this problem reads:

$$\mathcal{L}(y,u,p) := \int_D (u-u_d)^2\,dx + \int_D \nabla u \cdot \nabla p + up - yp\,dx, \tag{5.61}$$

with $u, p \in H^1(D)$ and $y \in L^2(D)$. Now the target is to solve the Linear system:

$$\partial^2\mathcal{L}(y,u,p)\,\alpha = -\partial\mathcal{L}(y,u,p), \tag{5.62}$$

where $\partial^2\mathcal{L}(y,u,p)$ is the Jacobian matrix of the Lagrangian, $\partial\mathcal{L}(y,u,p)$ is its gradient and α is the direction.
The gradient vector and the Jacobian matrix are then given as follows:

$$\partial\mathcal{L}(y,u,p) = \begin{pmatrix} \partial_y\mathcal{L}(y,u,p) \\ \partial_u\mathcal{L}(y,u,p) \\ \partial_p\mathcal{L}(y,u,p) \end{pmatrix}, \quad \partial^2\mathcal{L}(y,u,p) := \begin{pmatrix} \partial^2_{yy}\mathcal{L} & \partial^2_{yu}\mathcal{L} & \partial^2_{yp}\mathcal{L} \\ \partial^2_{uy}\mathcal{L} & \partial^2_{uu}\mathcal{L} & \partial^2_{up}\mathcal{L} \\ \partial^2_{py}\mathcal{L} & \partial^2_{pu}\mathcal{L} & \partial^2_{pp}\mathcal{L} \end{pmatrix}(y,u,p) \tag{5.63}$$

in the discrete form, the components of the gradient vector are:

$$\partial_y\mathcal{L}(y,u,p)(\gamma_i) = -\int_D \gamma_i p\,dx, \tag{5.64}$$

$$\partial_u\mathcal{L}(y,u,p)(\phi_j) = 2\int_D (u-u_d)\phi_j\,dx + \int_D \nabla\phi_j \cdot \nabla p + \phi_j p\,dx, \tag{5.65}$$

$$\partial_p\mathcal{L}(y,u,p)(\psi_k) = \int_D \nabla u \cdot \nabla\psi_k + u\psi_k - y\psi_k\,dx, \tag{5.66}$$

$$\psi_k \in H^1(D)\,,\ \phi_j \in H^1(D)\,,\ \forall\gamma_i \in L^2(D),$$

and the Jacobian is given by:

$$\partial^2\mathcal{L}(y,u,p) := \begin{pmatrix} \partial^2_{yy}\mathcal{L} & \partial^2_{yu}\mathcal{L} & \partial^2_{yp}\mathcal{L} \\ \partial^2_{uy}\mathcal{L} & \partial^2_{uu}\mathcal{L} & \partial^2_{up}\mathcal{L} \\ \partial^2_{py}\mathcal{L} & \partial^2_{pu}\mathcal{L} & \partial^2_{pp}\mathcal{L} \end{pmatrix}(y,u,p) = \begin{pmatrix} \mathbf{0} & \mathbf{0} & X \\ \mathbf{0} & M & D \\ X^T & D^T & \mathbf{0} \end{pmatrix}. \tag{5.67}$$

As the Lagrangian is linear in p and y, the matrices $\partial^2_{yy}\mathcal{L}$ and $\partial^2_{pp}\mathcal{L}$ are zeros and from (5.65) or (5.64), the matrix $\partial^2_{yu}\mathcal{L}$ is also zero. The matrices M, D, and X are given as follows:

$$\partial^2_{uu}\mathcal{L}(y,u,p) = M := \left(2\int_D \varphi_i\varphi_j\,dx\right), \qquad i,j = 1,\dots,N, \tag{5.68}$$

$$\partial^2_{up}\mathcal{L}(y,u,p) = D := \left(\int_D \nabla\varphi_i \cdot \nabla\varphi_j + \varphi_i\varphi_j\,dx\right), \qquad i,j = 1,\dots,N, \tag{5.69}$$

$$\partial^2_{yp}\mathcal{L}(y,u,p) = X := \left(2\int_D \gamma_i\varphi_j\,dx\right), \quad i = 1,\dots,\overline{N}, j = 1,\dots,N, \tag{5.70}$$

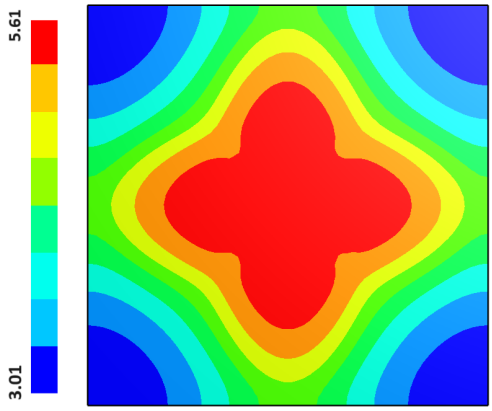


Figure 5.10: State Solution of The Control Problem.

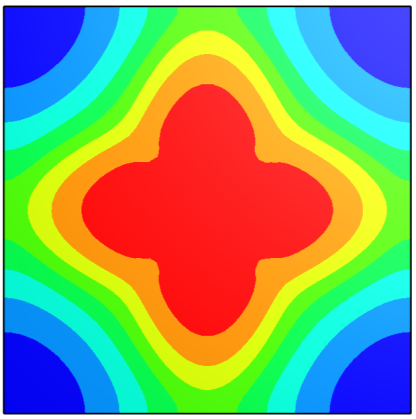


Figure 5.11: Reference State Solution.

where N is the number of degrees of freedom in the space $H^1(D)$ and $\overline{N}$ is the number of degrees of freedom in the space $L^2(D)$.

During the solution of the linear system, issues were encountered related to the singularity or near-singularity of the matrix involved, possibly due to the nature of the matrix X as it's assembled over two different spaces ($L^2(D)$ and $H^1(D)$). This necessitated the use of regularization to ensure the stability and solvability of the system.

Regularization was implemented by adding a small value ($\epsilon = 10^{-8}$) to the diagonal elements of the matrix. Specifically, we added $\epsilon \cdot I$, where ϵ is a small constant and I is the identity matrix of appropriate size. This helps in improving the conditioning of the matrix, thereby avoiding the singularity issues during factorization.

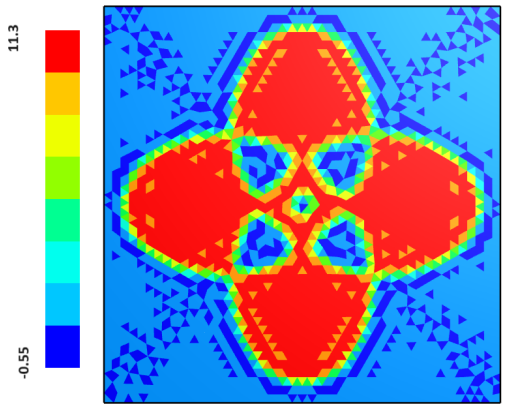


Figure 5.12: Control Variable Solution.

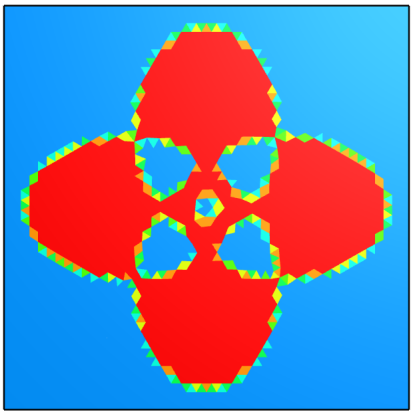


Figure 5.13: Reference Control Variable Solution.

Chapter 6

Interpolation of Level Set Functions in Finite Element Methods

In topology optimization and shape optimization problems, the representation and manipulation of geometries play a crucial role. The method used in this work for representing shapes is the level set method, where the shape of interest is described by the level set function. This implicit representation allows for a flexible and robust handling of complex geometries, including those undergoing significant topological changes. However, the numerical implementation of the level set method within the framework of the Finite Element Method (FEM) presents certain challenges. Specifically, the presentation of the boundary of the level set i.e. the curve $\psi = 0$, since the algorithms presented considered sufficient optimality conditions 2.68 and calculates values of the level set function at the degrees of freedom of the solution space, which are located at the nodes of the triangular elements of the mesh so the location of the points forming the curve $\psi = 0$ has to be approximated. This is where the interpolation functions presented in this chapter become indispensable.

Furthermore, the expansion of the level set function to higher-order elements, namely $P2$ elements, would enhance the accuracy of the interpolation. Higher-order elements allow for better approximation of curved geometries and more accurate computation of gradients, which are crucial for the convergence and efficiency of optimization algorithms.

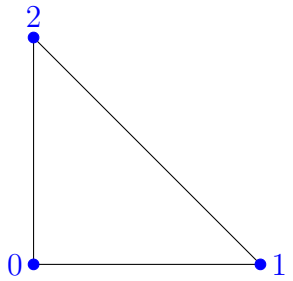
Definition 6.1. *(Finite Element): The triple (T, V_T, Ψ_T) is called a finite element, where T is a bounded set (a triangle in our 2D cases), V_T is Finite dimensional functional space of dimension $N_T \in \mathbb{N}_0$ where N_T is the number of local degrees of freedom (For V_T = the space of polynomials of degree 1 in 2D, $N_T = 3$) and the Basis function Ψ_T consisting of (N_T) linearly independent basis functions on (V_T) .*

Definition 6.2. *(Equivalent Elements): For convenience, finite element complexes are constructed such that all elements are affine equivalent to a reference element $(\hat{T}, V_{\hat{T}}, \Psi_{\hat{T}})$. where two finite elements (T, V_T, Ψ_T) and $(\hat{T}, V_{\hat{T}}, \Psi_{\hat{T}})$ are called equivalent if there exists an affine linear functional ($F(T)$) such that:*

- $T = F(\hat{T})$,
- $V_T = \{\hat{v} \circ F^{-1} : \hat{v} \in V_{\hat{T}}\}$,
- $\Psi_T = \{\psi_T^i : V_T \to \mathbb{R} : v \to \psi_{\hat{T}}^i(v \circ F)\}$.

6.1 First Order Polynomial Basis

We consider Lagrange finite elements of order one, the reference triangle $\hat{T}$ of the reference element $(\hat{T}, V_{\hat{T}} = H^1(\hat{T}), \Psi_{\hat{T}})$ in an (x, y) coordinate system has vertices $(0, 0)$, $(1, 0)$, and $(0, 1)$, corresponding to local vertex numbers 0, 1, and 2, respectively.



The P1 element has linear functions $\tilde{\varphi}_r(x, y)$ as basis functions, $r = 0, 1, 2$. Since a linear function $\tilde{\varphi}_(x, y)$ in 2D is on the form $a_r x + b_r y + c_r$, and hence has three constants a_r, b_r, and c_r.
Indeed, a function $v_h \in V_{\hat{T}}$ is uniquely defined by its values $v(x_j)$ on the nodes. Consequently, one can obtain a nodal basis of $V_{\hat{T}}$ by functions characterized by:

$$\Phi_i(x_j) = \delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}. \tag{6.1}$$

Requiring $\varphi_r = 1$ at node number r and $\varphi_r = 0$ at the two other nodes, gives three linear equations to determine a_r, b_r, and c_r. The result is:

$$\begin{aligned} \varphi_0(x, y) &= 1 - x - y, \\ \varphi_1(x, y) &= x, \\ \varphi_2(x, y) &= y. \end{aligned} \tag{6.2}$$

For a triangular element, the ratio $[s]$ is calculated, which is the ratio between the area cut by the level set interface and the whole triangle, where the position of the interface is determined according to the nodal values (ψ_0, ψ_1, ψ_2).

Accordingly, the following cases arise:

Case0: The case where all the 3 nodal values have the same sign is trivial, as $s = 1$ in case all values are negative and $s = 0$ if all values are positive.

Case1: for if $\psi_0 = 0$ and $sgn(\psi_1) = sgn(\psi_2)$,
in this case the entirety of the element lies beyond the interface, hence for $sgn(\psi_1) = sgn(\psi_2) = -1$ the element is inside the domain $s = 1$,
and $s = 0$ for $sgn(\psi_1) = sgn(\psi_2) = 1$.

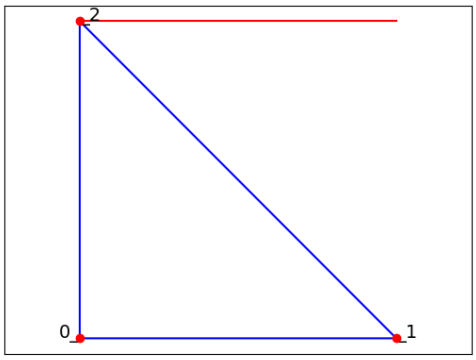


Figure 6.1: Case 1 - P1 Interpolation.

Case2: for if $\psi_0 = 0$ and $sgn(\psi_1) \neq sgn(\psi_2)$.
Let $sgn(\psi_2) = -1 \Rightarrow sgn(\psi_1) = 1$, since ψ is linear on the edge, $\exists$ a point O on the side $\overline{2-1}$ where the level set interface passes through.
For $\overline{2-1}$ of length L, and $\overline{2-O}$ of length $\overline{L}$, the ratio $\frac{\overline{L}}{L}$ is given by $\frac{-\psi_1}{\psi_2-\psi_1}$.
The area of the reference is given by $A = L \times L_1 \times Cos(\theta)$ where L_1 is the length of the side $\overline{2-0}$ and θ is the angle at node 2. Hence,

$$s = \frac{L_1 \times \overline{L} \times \cos(\theta)}{L \times L_1 \times \cos(\theta)} = \frac{-\psi_1}{\psi_2-\psi_1}.$$

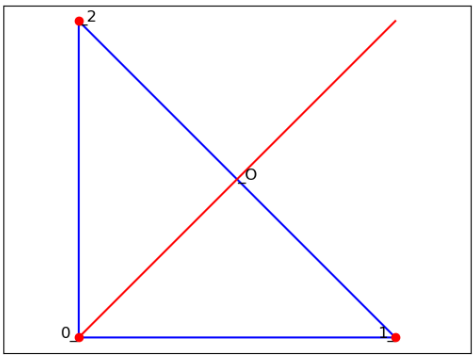


Figure 6.2: Case 2 - P1 Interpolation.

For if the signs are reversed, $s = 1 - \frac{-\psi_1}{\psi_2-\psi_1}$.

Case3: for if $\psi_0 = \psi_1 = 0$. The line $\overline{0-1}$ is cut by the interface, hence $s = 1$ for $\psi_2 > 0$ and $s = 0$ for $\psi_2 < 0$.

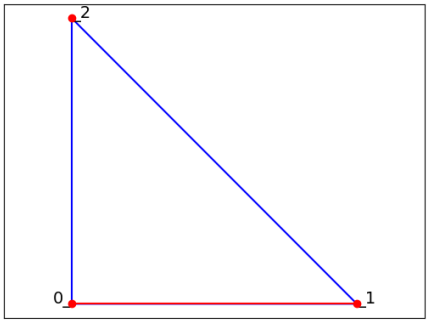


Figure 6.3: Case 3 - P1 Interpolation.

Case4: for if non of the nodal values is a zero. Let $\psi_1 > 0, \psi_2 > 0$ and $\psi_0 < 0$ Similar to case 2, since ψ is linear on the edges, $\exists$ two points O_1 and O_2 on the sides $\overline{0-1}$ and $\overline{0-2}$ respectively, where the level set interface passes through.

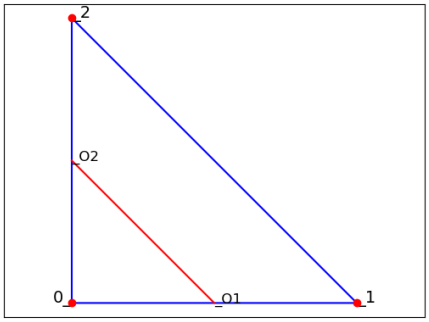


Figure 6.4: Case 4 - P1 Interpolation.

Let $\overline{0-1}$ and $\overline{0-2}$ have lengths L_1 and L_2 respectively, and the intersected parts have lengths $\overline{L}_1$ and $\overline{L}_2$ measured from point 0.
By linearity of ψ on the edges, the ratios $\frac{\overline{L}_1}{L_1}$ and $\frac{\overline{L}_2}{L_2}$ are given by, $\frac{-\psi_0}{\psi_1-\psi_0}$ and $\frac{-\psi_0}{\psi_2-\psi_0}$ respectively. And the area of the reference triangle is given by $A = L_1 \times L_2 \times Cos(\theta)$ where θ is the angle at node 0. and the area of the cut triangle is given by $A^- = \overline{L}_1 \times \overline{L}_2 \times Cos(\theta)$ Hence,

$$s = \frac{A^-}{A} = \frac{\overline{L}_1 \cdot \overline{L}_2}{L_1 \cdot L_2} = \frac{\psi_0^2}{(\psi_1 - \psi_0) \cdot (\psi_2 - \psi_0)},$$

for if the signs are reversed,

$$s = 1 - \frac{\psi_0^2}{(\psi_1 - \psi_0) \cdot (\psi_2 - \psi_0)}.$$

6.2 Second Order Polynomial Basis

To gain better accuracy, it's desired to considered basis of second degree polynomials, where the position of interface would be determined by 6 nodal values, and the reference element would be as follows:

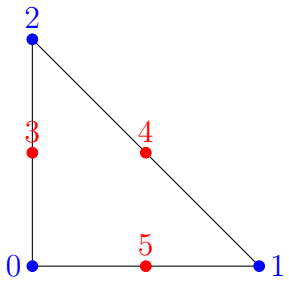


Figure 6.5: Reference Element for Second Order Polynomial Basis.

On each side of the triangle, the level set function follows the following second order polynomial:

$$a\psi^2 + b\psi + c = 0.$$

Given the values ψ_{start}, ψ_{end}, and ψ_{mid}, the constant of the polynomial are given by:

$$a = 2(\psi_{\text{start}} + \psi_{\text{end}} - 2\psi_{\text{mid}}),$$

$$b = 4\psi_{\text{mid}} - 3\psi_{\text{end}} - \psi_{\text{start}},$$

$$c = \psi_{\text{end}}.$$

The location of the level set interface is determined according to the signs and values of each of the six nodal values, which arises the following cases:

Case0:
Similar to the linear basis setting, the case where all the 6 nodal values have the same sign is trivial, as $s = 1$ in case all values are negative and $s = 0$ if all values are positive.

Case1:
For if all the nodal values are nonzero with one having the opposite sign from the others, the calculation of the volume fraction would be different if the different sign is on a vertex or a midpoint, which is clarified in the following two sub-cases:

Sub-case1.1:
Let $sgn(\psi_0) = -1$ and $sgn(\psi_i) = 1$ for $i = 1, 2, ...5$, the area included by the curve representing $\psi = 0$ could be approximated by an area of a triangle, this approximation is acceptable due to the nature of finite element meshes and the error inside a single element caused by the curvature must be minor.

Hence the ratio s between the included area and total area can be calculated as follows:

$$s = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2},$$

where $\overline{l_1}$ and $\overline{l_2}$ are the intersection length on the sides $\overline{01}$ and $\overline{02}$ of lengths l_1 and l_2 respectively.

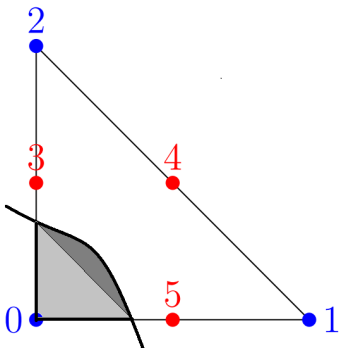


Figure 6.6: Second Order Interpolation Sub-Case 1.1.

Sub-case1.2:
Let $sgn(\psi_4) = -1$ and the rest of the nodal values have positive values, the volume fraction would be the ratio between the area under the parabola and the total triangle area:

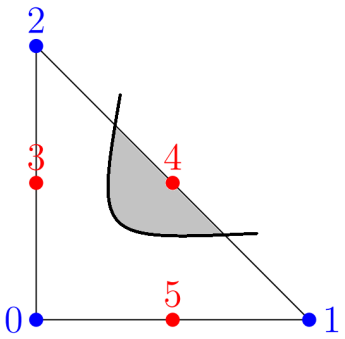


Figure 6.7: Second Order Interpolation Sub-Case 1.2.

Case2:
In case all the nodal values are nonzero with two of them having opposite sign from the other four, one should carefully consider the combination between the relative location of the two different signed nodes dictates the method of calculating the volume fraction, which leads to the following sub-cases:

Sub-case2.1:
For if the two nodes with different signs are a vertex and one of its consecutive midpoints, the case is similar to Sub-case 1.1, the volume fraction is approximated as ratio between two triangle.

$$s = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2}.$$

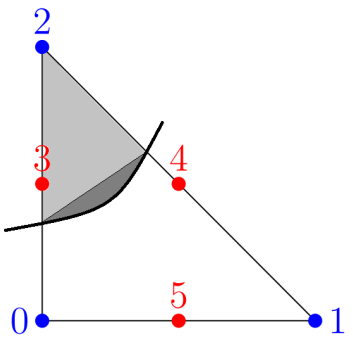


Figure 6.8: Second Order Interpolation Sub-Case 2.1.

Sub-case2.2:
For if the two nodes with different signs are a vertex and the mid point opposite to it, the volume fraction would be the sum of two ratios; the ratio between the triangle areas and the ratio between the area included by the parabolic curve around the mid point, that is:

$$s = s_1 + s_2,$$

$$s_1 = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2}, \qquad s_2 = \frac{\text{Parabolic Area}}{\text{Triangle Area}}.$$

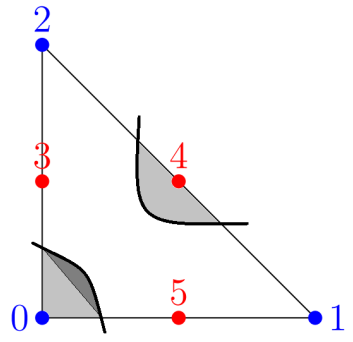


Figure 6.9: Second Order Interpolation Sub-Case 2.2.

Sub-case2.3:
For if the two nodes with different signs are two vertices, the volume fraction would be the sum of two ratios, each of them being approximated as a ratio of areas of triangle, that is:

$$s = s_1 + s_2,$$

$$s_1 = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2}, \qquad s_2 = \frac{\overline{l_3} \times \overline{l_4}}{l_3 \times l_4}.$$

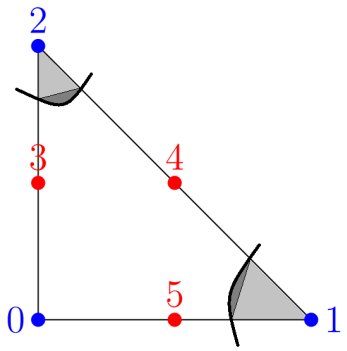


Figure 6.10: Second Order Interpolation Sub-Case 2.3.

Sub-case2.4:
Finally, for if the two nodes with different signs are two mid points, the volume fraction would be the sum of two ratios, each of them being the ratio between the area included by the parabolic curve and the total triangle area.

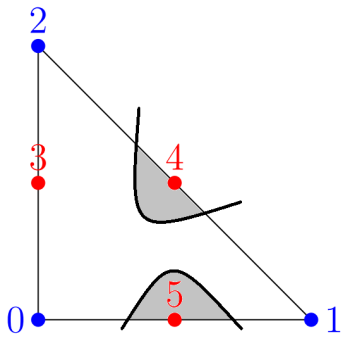


Figure 6.11: Second Order Interpolation Sub-Case 2.4.

Case 3:
In case all the nodal values are nonzero with three being positive valued and three being negative values, this case is also similar to Sub-Case 1.1 as the volume fraction is also approximated by the ratio between areas of triangles.

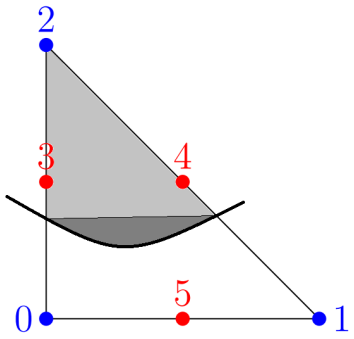


Figure 6.12: Second Order Interpolation Case 3.

Case 4:
In case all the nodal values are nonzeros of the same sign except for one which is exactly zero, the case is similar to Case 0 as the volume fraction would be $s = 1$ in case said sign is negative and $s = 0$ otherwise.

Case 5:
For if there is one nodal value that is exactly zero and one nodal value that has an opposite sign than the other four, then one needs to consider the relative location of zero and opposite sign, due to to the quadratic nature of the level set function this boils down to the following two Sub-Cases:

Sub-Case 5.1:
In case the zero is on a vertex, the opposite valued node should be one of the consecutive mid points of said vertex.

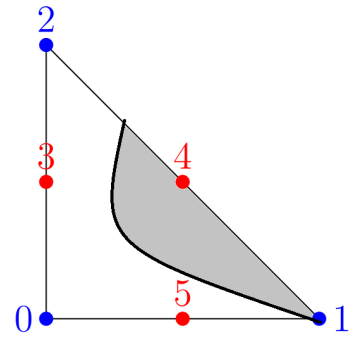


Figure 6.13: Second Order Interpolation Sub-Case 5.1.

Let $\psi_1 = 0$, $sgn(\psi_4) = -1$ and $sgn(\psi_i) = 1$ for $i \in \{0, 2, 3, 5\}$, then the volume fraction s will be the ratio between the area included by the parabola over the total triangle area.

Sub-Case 5.2:
In case the zero is on a midpoint, the opposite sign valued node should be on one of the vertices of the line bisected by the zero valued node.

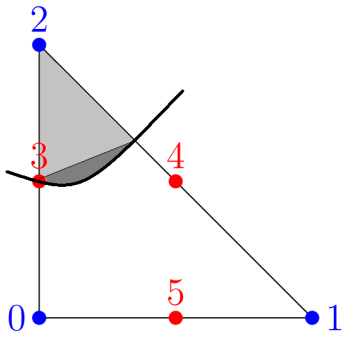


Figure 6.14: Second Order Interpolation Sub-Case 5.2.

Let $\psi_3 = 0$, $sgn(\psi_2) = -1$ and $sgn(\psi_i) = 1$ for $i \in \{0, 1, 4, 5\}$, then the volume fraction s will be approximated by the area of two triangles, that is:

$$s = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2}.$$

Case 6:
Finally, for if two of the nodal values are exactly zero, then due to the quadratic nature of the level set function it's impossible to have two consecutive nodes with zero values, it's also impossible to have two zero valued nodes and all the other nodes having the same sign,hence only the following two Sub-Cases arise:

Sub-Case 6.1: For if the two zeros are on two vertices, the nodal value between them must have an opposite sign to the other three nodal values of the triangle.

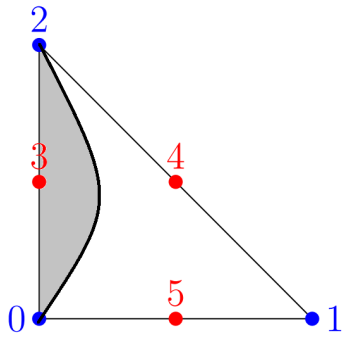


Figure 6.15: Second Order Interpolation Sub-Case 6.1.

Let $\psi_0 = \psi_2 = 0$ and $sgn(\psi_3) = -1$, then the volume fraction s will be the ratio between the area included by the parabola over the whole line $\overline{02}$ and the area of the entire triangle.

Sub-Case 6.2: For if the two zeros are on two midpoints, their common vertex's nodal value must have an opposite sign to the other three nodal values of the triangle.

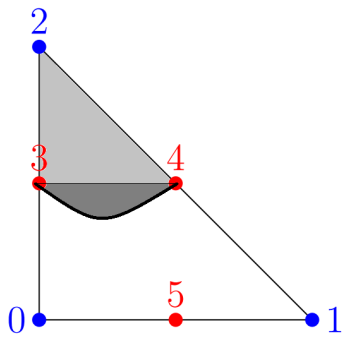


Figure 6.16: Second Order Interpolation Sub-Case 6.2.

Due to the approximation done when calculating this type of area ratio, this case will have a fixed volume fraction of $s = 1/4$, that is:

$$s = \frac{\overline{l_1} \times \overline{l_2}}{l_1 \times l_2}, \qquad \overline{l_1} = \frac{l_1}{2}, \overline{l_2} = \frac{l_2}{2}.$$

In all of the previous cases if the signs are reversed, $s = 1 - s$.

The computational methods and datasets used in this chapter and the previous chapters are available in the public GitHub repository at `https://github.com/amrmohd97/thesis-topo_optimization_codes`. This repository includes all scripts and data necessary to reproduce the results discussed in this thesis.

Chapter 7

Conclusion

The representation of spatial domain through the level set function enables a flexible and accurate optimization process, particularly in scenarios where the design space exhibits complexity. By allowing the values of the level-set function to change incrementally, the different algorithms discussed previously can effectively navigate the design space to reach local minima.

Integrating this representation with the concept of topological derivative proves useful in addressing different structural optimization problems. This combination is particularly effective in linear elasticity problems, as discussed in [3] and [5], where the optimization process must account for PDE constraints that has domain-dependent coefficients in the primary part. Although the algorithms 1 and 2 may require a relatively large number of iterations in some cases, they can still yield results which are close to practical applications [3] for example in the triple loaded bridge case discussed in Chapter 4. The incorporation of loose line search techniques within these algorithms aids in improving convergence, helping to strike a balance between computational efficiency and accuracy.

Higher order methods utilizing the topological state derivative introduced in [4] and the second order topological derivative have shown great potential in significantly enhancing the efficiency of the optimization process, yet they also come with a set of challenges. The algorithm 3 for example can reach the minimizing domain in one iteration for the case that the topological derivative at said minimizer is zero. Moreover, despite increasing the size of the linear system to be solved, algorithm 4 in its two formulations converges in a relatively small number of iterations. Regularity issues that arise can be circumvented by enforcing specific regularity using the equation (5.58) or by leveraging the PDE solver of NGSolve to augment the algorithm as discussed in examples 5.4.1 and 5.4.2.

Finally, it is crucial to calculate the location of the domain's boundary ($\psi = 0$). Within the framework of finite element methods (FEM), this can be efficiently achieved through simple linear interpolation. Additionally, the accuracy of this approximation can be further enhanced by employing second-order interpolation, provided that the nodal values of a second-order H^1 function are available. This approach would offer a more precise representation of the boundary, improving the overall effectiveness of the optimization process.

Bibliography

[1] Habib Ammari and Hyeonbae Kang. *Reconstruction of small inhomogeneities from boundary measurements*, volume 1846 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2004.

[2] Samuel Amstutz. Sensitivity analysis with respect to a local perturbation of the material property. *Asymptot. Anal.*, 49(1-2):87–108, 2006.

[3] Samuel Amstutz and Heiko Andrä. A new algorithm for topology optimization using a level-set method. *J. Comput. Phys.*, 216(2):573–588, 2006.

[4] Phillip Baumann, Idriss Mazari-Fouquer, and Kevin Sturm. The topological state derivative: an optimal control perspective on topology optimisation. *J. Geom. Anal.*, 33(8):Paper No. 243, 49, 2023.

[5] Sebastian Blauth and Kevin Sturm. Quasi-Newton methods for topology optimization using a level-set method. *Struct. Multidiscip. Optim.*, 66(9):Paper No. 203, 21, 2023.

[6] Peter Gangl and Kevin Sturm. A Newton levelset algorithm using first and second order topological derivatives. *(in preparation)*, 2024.

[7] Peter Gangl, Kevin Sturm, Michael Neunteufel, and Joachim Schöberl. Fully and semi-automated shape differentiation in `ngsolve`. *Struct. Multidiscip. Optim.*, 63(3):1579–1607, 2021.

[8] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[9] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[10] Walter Rudin. *Functional analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill, Inc., New York, second edition, 1991.

[11] Kevin Sturm. Topological sensitivities via a Lagrangian approach for semilinear problems. *Nonlinearity*, 33(9):4310–4337, jul 2020.