Topological Optimisation Using Laguerre Diagram

Hussein Houdrouge

supervised by

Prof. Bruno Levy

July 2018

Contents

1 Introduction

Shape optimisation is a subdiscipline of applied mathematics and mechanical engineering. It is concerned with finding the shape that optimises one or multiple functions under certain constraints. This discipline has many applications in different areas such as civil engineering, aeronautics, and aerospace. For instance, engineers use it to design better bridges or more optimal wings for airplanes. For example, Figure 1 shows the variety of Topological Optimisation applications. In addition, the development in additive manufacturing (3D printing) will give a boost of such discipline since the new 3D printers will allow the manufacturing of more complex shapes resulted from Topological Optimisation algorithm.

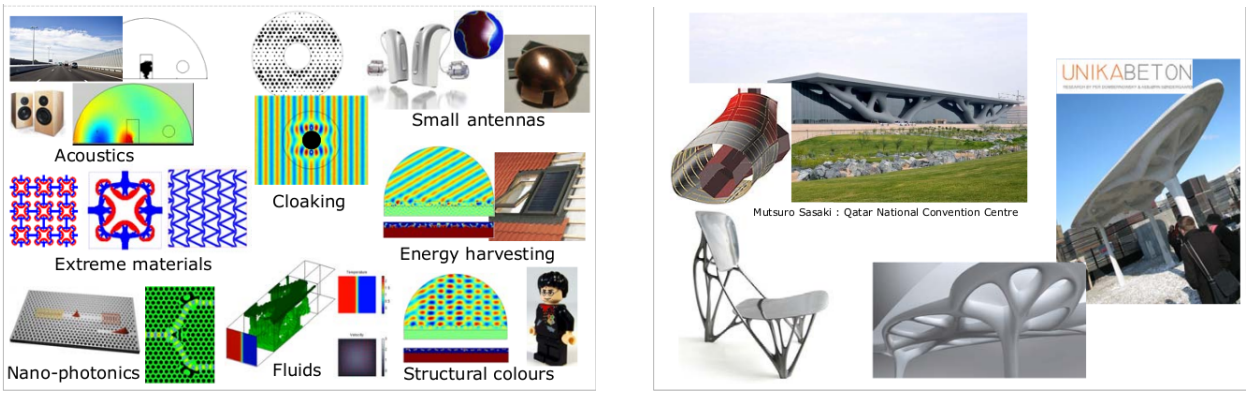

Figure 1: The application of Topoligical Optimisation. source [Sig].

In the rest of this document, we aim at two objectives. The first one is giving the minimal prerequisites to understand a shape optimization problem and how to solve it. This objective is treated in sections 2 and 3. Then, we want to introduce our method and what it requires to be understood. The second objective is treated in section 4. A reader who is more familiar with the subject can directly start from section 4.

2 Shape Optimisation

According to [dSA06], a shape optimisation problem is determined by these three components:

1. A **Model** describes the behaviors of a mechanical structure. Usually, this model is a system of partial differential equations such as the linear elasticity system which is described in section 4.1.1.

2. An **objective function** (or cost function) which is a function that we want to minimize or maximize by modifying our initial shape at each iteration.

3. An **admissible set** is the set of constraints over the variables of the problems. For instance, we can impose constraints over the volume and the surface of the shape.

Therefore, we can define the shape optimisation problem based on these three givens as follows:

Find the shape $\Omega \in \mathcal{U}_{ad}$ that minimise a given function $\mathbf{J}$,

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

where $\mathcal{U}_{ad}$ is the set of all admissible shapes, and u is a variable that depends on the model (usually solution of partial differential equation).

Shape optimization problems can be classified into three categories, respectively from the easiest to the hardest, Parametric, Geometric, and Topological.

1. **Parametric** Optimisation: The shape Ω is parameterized by a few number of variables such as thickness, diameter, or any other design variable. For concrete examples of this kind of optimization, we advise the reader to check chapter 2 of the following book [CK08].

2. **Geometric** Optimisation: Starting from the initial shape, we allow the spatial variables of the boundary to change without allowing any change in the topology i.e in two dimensions, holes are NOT allowed to appear/disappear inside the shape. A detailed discussion on this type of shape optimisation is presented in chapter 6 in [dSA06].

3. **Topological** Optimisation is the most general kind of shape optimization. This type of optimization requires no restriction on the shape or on the topology. For example, in two dimensions, holes can be created or removed in order to have an optimal shape.

Since in this work we are interested in Topological Optimisation, we give a standard example for such a problem. Let $D \in \mathbb{R}^2$ be a given domain, and let $\Omega \subset D$. We denote by $\partial\Omega$ the boundary of Ω. In our example, we decompose it into three parts. Γ_D represents the fixed part. Γ_N is also fixed, but there is a force applied on it. Γ represents the variable or movable part. We also say that Γ_D is the Dirichlet condition and Γ is the Neumann condition. A schematic representation is presented in Figure 2. The domain Ω is filled with a homogeneous, isotropic, elastic material. Moreover, the behaviours of Ω are captured by the following model known as Linear Elasticity System:

$$\begin{cases} \sigma = 2\mu e(u) + \lambda(tre(u))I & \text{in} \quad \Omega \\ \text{div}\sigma = f & \text{in} \quad \Omega, \\ u = 0 & \text{on} \quad \Gamma_D, \\ \sigma n = g & \text{on} \quad \Gamma_N, \\ \sigma n = 0 & \text{on} \quad \Gamma \end{cases}$$

In the presented equations, g and f are the applied forces on the boundary and on the structure, μ and λ are related to the used material, and they are known as Lamé Coefficients.

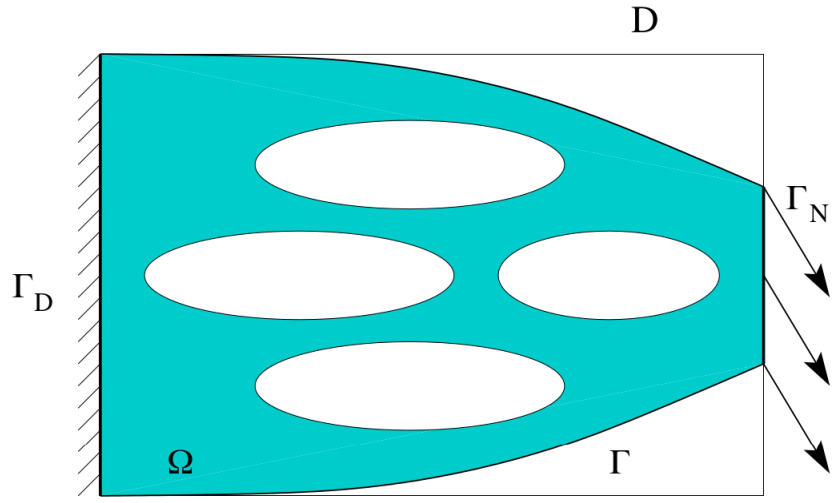

Figure 2: Topological Optimisation for a cantilever. source [All]

Furthermore, σ is the strain tensor and $e(u)$ is the stress tensor. They are related by Hook's Law. $u : \Omega \longrightarrow \mathbb{R}^2$ is the displacement vector field. We will discuss this model in details in section 4.1.1. We will restrict Ω to the following admissible set:

$$\mathcal{U}_{ad} = \{\Omega \subset \mathbb{R}^2 \text{ such that } \Gamma_D \bigcup \Gamma_N \subset \partial\Omega \text{ and } \int_\Omega dx = V_0\}$$

And we will chose to minimise the following function.

$$J(\Omega) = \int_\Omega f.udx + \int_{\partial_\Omega} g.udx$$

2.1 General Topological Algorithm

We will describe briefly and generally how to solve a Topological Optimisation problem. We illustrate this process in Figure 3. The first step is to initialize the shape and the mechanical model of the structure to be able to solve it and deduce the behaviour of the shape. The second step is to solve the model and deduce its behaviour. For this step, the standard technique is to use the Finite Element Method which we dedicate section 4.1.2 for it. Later, we conduct a sensitivity analysis to be able to change the shape towards an optimal one. This step, sensitivity analysis, is usually linked with the shape derivative that we will discuss in the next section. After conversion (or attaining the halting criterion), we can extract the shape and halt.

2.2 Shape Derivative

Shape derivative is one of the most important concepts in shape optimisation, especially for a gradient-based algorithm. In order to find the shape that optimises a given function,

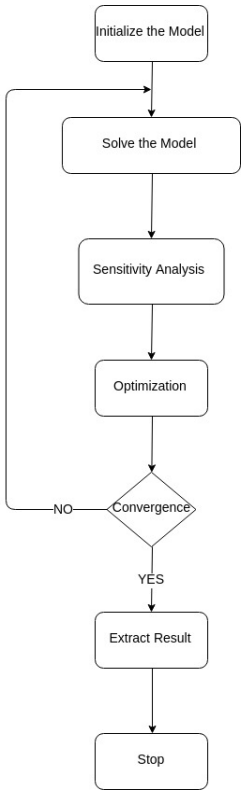

Figure 3: General Topological Optimisation Process

we want to compute the derivative of this function with respect to the shape itself (or the parametrization of the shape). We will begin this section by introducing the classical definition of the derivative of a functional with respect of a domain, then we will introduce two important notions of derivatives (Eulerian and Lagrangian derivative). In the end, we will show how the shape derivative is computed. However, in practice and in our work, we are not using these techniques directly but we are using a discrete approach.

First of all, the shape derivative must be understood in the sense of Fréchet derivative. For detailed definition of Fréchet derivative consider [Pug15].
Definition: Suppose that θ is a given vector field that deforms the initial shape Ω_0, and J is a functional of the domain of the the admissible set. We say that J is differentiable at Ω if the underlying function

$$\theta \longrightarrow J((I+\theta)\Omega)$$

is Frechet differentiable at $\theta = 0$. Then, there exists a linear transformation (map), $L = J'(\Omega)$ called the shape derivative at Ω such that,

$$J((Id+\theta)\Omega_0) = j(\Omega_0) + L(\theta) + o(\theta)$$

where The shape $\Omega = (Id+\theta)(\Omega_0)$ is defined by

$$\Omega = \{x + \theta(x) | x \in \Omega_0\}$$

Notice that this derivative tells us how to change the geometry to optimise the function not how to change the topology. To change the topology, one can use techniques such as the Bubble method [EKS94].

To compute this derivative, we need to recall how to change variables in integrals. We define the space of diffeomorphismes over R^2 by

$$\mathcal{T} = \{T \quad \text{such that} \quad (T - id) \in W^{1,\infty}, (T^{-1} - id) \in W^{1,\infty}(R^2, R^2)\}$$

We will mention the following lemma about the change of variables in the integrals.
Lemma 1: let Ω_0 be an open set in R^2. $T \in \mathcal{T}$, a diffeomorphime of R^2. for $1 \leq p \leq +\infty$. then $f \in L^p(T(\Omega_0))$ if and only if $f \circ T \in L^p(\Omega_0)$ and we have,

$$\int_{T(\Omega_0)} f dx = \int_{\Omega_0} f \circ T |det \nabla T| dx$$

and

$$\int_{T(\Omega_0)} f |det(\nabla T)^{-1}| dx = \int_{\Omega_0} f \circ T dx,$$

furthermore, $f \in W^{1,p}(T(\Omega_0))$ if and only if $f \circ T \in W^{1,p}(\Omega_0)$ and we have

$$(\nabla f) \circ T = ((\nabla T)^{-1})^t \nabla (f \circ T).$$

where W is a Banach space (We present its definition in the appendix).

Using the previous Lemma, the derivative of an integral of a function is given by the following proposition. For more details about the derivation and the proof of this proposition, we direct the reader to the following resources [dSA06] and [Dap13].
Preposition: let Ω_0 be an open, regular, and bounded set of R^2. let $f \in W^{1,1}(R^N)$, and J is a function of $\mathcal{C}(\Omega_0)$ in R defined by

$$J(\Omega) = \int_{\Omega} f(x) dx$$

then J is differentiable in Ω_0 and we have

$$J'(\Omega_0)(\theta) = \int_{\Omega_0} div(\theta(x) f(x)) dx = \int_{\partial\Omega_0} \theta(x).n(x) f(x) ds$$

for all $\theta \in W^{1,\infty}(R^2; R^2)$.

2.2.1 Derivative of a Function that Depends on the Domain

We have the function $u(\Omega, x)$ that is defined for all $x \in \Omega$, and that depends on the domain Ω. Usually this function can be the solution of partial differential equation on Ω. Our goal is to compute the derivative of this function with respect to the domain Ω.
There are two types of derivatives, **Eulerian derivative** (or shape derivative), and **Lagrangian derivative** (or material derivative). We use U and Y to denote the first and the second derivative respectively. This distinction is identical to the mechanical distinction where Eulerian is related to a fixed reference frame and Lagrangian to a non-fixed reference frame.

2.2.1.1 The Eulerian Derivative

For a fixed point x that belongs to the initial domain and to the deformed domain we can compute the new value of the function at a point x in the following manner,

$$u((Id+\theta)(\Omega_0), x) = u(\Omega_0, x) + U(\theta, x) + o(\theta).$$

where U is a continuous linear form in θ that we call the Eulerian derivative.

2.2.1.2 The Lagrangian Derivative

In this version of derivation, the point x is no longer fixed in one place but we are tracking the point by constructing the *transpoted* $\bar{u}(\theta)$ in the initial domain Ω_0 with simple change of variables for all $x \in \Omega_0$.

$$\bar{u}(\theta, x) = u((Id+\theta)(\Omega_0)) \circ (Id+\theta) = u((Id+\theta)(\Omega_0), x+\theta(x))$$

We obtain the Lagrangian derivative Y by deriving the function $\bar{u}(\theta, x)$ with respect to θ.

$$\bar{u}(\theta, x) = \bar{u}(0, x) + Y(\theta, x) + o(\theta).$$

where $\bar{u}(0, x) = \bar{u}(\Omega_0, x)$ and $Y(\theta, x)$ is a linear continuous form in θ.

Now we want to define the derivative of integral using the Lagrangian and the Eulerian derivative. We will simply state the derivatives and we will ignore some of the details that are presented in [dSA06]. we have the function

$$J(\Omega) = \int_{\Omega} u(\Omega) dx$$

In term of the Lagrangian derivative, the derivative is the following.

$$J'(\Omega_0)(\theta) = \int_{\Omega_0} (u(\Omega_0) div\theta + Y(\theta)) dx$$

In term of the Eulerian derivative,

$$j'(\Omega_0)(\theta) = \int_{\Omega_0} (U(\theta) + div(u(\Omega_0)\theta)) dx.$$

3 Related Works

Many techniques have been developed to solve the problem of Topological Optimisation. These methods are mainly different from each other by the way they represent the shape, and how to compute the sensitivity of the objective function [Dap13]. In the following subsections, we will discuss briefly some of these techniques and we will discuss the related points to our method if they exist.

3.1 SIMP Method

The SIMP is an abbreviation for Solid Isotropic Material with Penalisation. According to [GM12], the main idea of this method is to discretize the domain into small fixed elements such as small rectangles. Each element will be associated with a variable that denotes the density of the material inside it. Then, computing the gradient of the objective function with respect to this parameter, indicates how to change the density of each element in order to achieve an optimal shape. Therefore, the shape is allowed to change topology by assigning a low value for some elements which can be considered a void space. However, this method has a little in common with our method. The differences are in the representation of the shape and in the way the topology is changed. Our method requires moving elements (Laguerre cells), and it changes the topology by changing the position of the cells.

3.2 Level Set Method

The Level Set method uses implicit techniques to represent the shape. In other words, the shape is implicitly represented using a scalar function $\phi : R^d \longrightarrow R$. For instance, this function can be defined as follows.

$$\begin{cases} \phi(x) < 0 & \text{if} \quad x \in \Omega \\ \phi(x) = 0 & \text{if} \quad x \in \partial\Omega \\ \phi(x) > 0 & \text{if} \quad x \in \bar{\Omega}^c \end{cases}$$

Moreover, this representation allows us to translate the motion of evolving domain $\Omega(t)$ over a period $[0, T]$ to a partial differential equation for an associated time-dependent level set function $\phi(t, .)$ known as the Hamilton-Jacobi equation. This method is also based on gradient optimization techniques. Thus, a computation of the derivative of the objective function is done in order to reach the optimal shape representation. More details on this method are presented in [Dap13]. Obviously, this method is far from the one that we are trying to create especially in the way we represent our shape.

3.3 Topological Optimisation Using Simplicial Complex

This method is introduced in the following paper [CBN$^+$15]. This method represents the shape using a mesh of Simplicial Complex. It also uses a gradient based technique to optimise the shape. In addition, according to the sensitivity analysis, it deforms the shape and the position of the vertices using tools that edit the mesh discussed in [CBN$^+$15]. This method shares some similarity with ours concerning the moving elements. However, the way we constrain the volume and change the topology is different.

3.4 Topological Optimisation Using Evolutionary Techniques

The main idea behind this method is to represent the domain with an array of bits. Then starting from an initial shape, apply some genetic algorithms that are described in [dSA06], in order to reach the optimal shape. For further information on this kind of optimization,

consider the following book [ES15]. Some of these techniques introduce Voronoi Diagram (a special case of Laguerre Diagram) which makes it appear close to our method but the use of evolutionary techniques creates a big difference.

4 Topological Optimisation Using Laguerre Diagram

Before moving to our Topological Optimisation algorithm that is based on Laguerre Diagram we would like to introduce some preliminary knowledge in the following section.

4.1 Preliminary Knowledge

4.1.1 Linear Elasticity

Linear Elasticity System models the small deformation and displacement of a solid at rest in a given domain. In the following paragraphs, we will denote this domain by Ω which is an open bounded set of $\mathbb{R}^n$. Further, we will denote the force by f which is a function from Ω into $\mathbb{R}^N$. In order to understand the Linear Elasticity System, we will define two important quantities known as stress and strain (deformation or displacement), and we will state the connection between them. The following definitions and Figure 4 are inspired by [McG].

4.1.1.1 Stress

Stress is simply

$$\frac{\textbf{Force}}{\textbf{Area}}$$

There are two types of stress, **normal** stress (denoted by σ) and **shear** stress (denoted by τ). The stress is called **Normal** if the force is normal to the surface. It is called shear if the force is parallel to the surface. That is,

$$\sigma = \frac{F_{normal}}{Area} \quad \text{and} \quad \tau = \frac{F_{parallel}}{Area}$$

Stress components: In order to capture the complete stress state, we have to define many components. Take the example in Figure 1 taken from [McG]. The force F has two

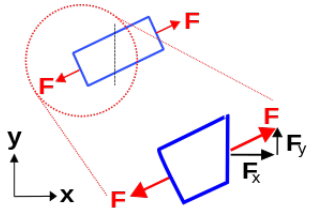

Figure 4: Simple stresses

components F_x and F_y. F_x is normal to A_x and F_y is parallel to A_x. Then two possible

stress can be defined the normal stress σ_{xx} and the shear stress σ_{xy}. That is,

$$\sigma_{xx} = \frac{F_x}{A_x} \quad \text{and} \quad \tau_{xy} = \frac{F_y}{A_x}.$$

Moreover, one can compute the stress with respect to A_y by cutting the figure horizontally. This will yield to,

$$\sigma_{yy} = \frac{F_y}{A_y} \quad \text{and} \quad \tau_{yx} = \frac{F_x}{A_y}.$$

We can notice in the notation that the first subscript refers to the area component and the the second one to the force component. According to [McG], τ_{xy} and τ_{yx} are equal. In addition, the stress is a tensor and it can be represented in the following way for the two dimensional case.

$$\sigma = \begin{pmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{yx} & \sigma_{yy} \end{pmatrix}$$

4.1.1.2 Strain

The strain is related to the deformation that is caused by the stress. It is the ratio of the deformation to the initial length. Two kind of strain are defined **Normal** strain and **Shear** strain. The **Normal** strain is the change in the displacement caused by the normal stress and the shear is the change in the displacement caused by the shear stress. Let $u(X, Y, Z)$ be the displacement field which is a function from Ω into $\mathbb{R}^N$ where X, Y, and Z are the reference position of the particles with respect to the body.
The normal strain is defined as follows for $N = 3$,

$$\epsilon_x = \frac{\partial u_x}{\partial X} \quad \epsilon_y = \frac{\partial u_y}{\partial Y} \quad \epsilon_z = \frac{\partial u_z}{\partial Z}$$

where $u_x = x - X = (\frac{X}{L_0})(L_f - L_0)$.
The equation for the shear strain,

$$\gamma_{xy} = \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X}$$

The strain is also a tensor, and it can be expressed as follows for $N = 2$,

$$e(u) = \begin{pmatrix} \epsilon_{xx} & \frac{\gamma_{xy}}{2} \\ \frac{\gamma_{yx}}{2} & \epsilon_{yy} \end{pmatrix}$$

Notice that the shear strain in the tensor representation is one-half of the mechanical shear strain [McG].
For a more general formula of the tensor of the strain (deformation), we refer to [ACC07],

$$e(u) = \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big)_{1 \le i,j \le N}$$

4.1.1.3 Hooke's Law

The two quantities defined above are related to each other by Hooke's law. It is described by [McG]. But we will just state the final result from [ACC07].

$$\sigma = 2\mu e(u) + \lambda tr(e(u))I$$

where λ and μ are the Lamé coefficients for the homogeneous isotropic material which occupiers Ω. These coefficients according to [ACC07] must satisfy

$$\mu > 0 \quad \text{and} \quad 2\mu + N\lambda > 0$$

4.1.1.4 Force Balance Equation

The balance of the force in the solid satisfies the following equation,

$$-\text{div}\sigma = f \quad \text{in} \quad \Omega$$

where the divergence is defined as follow,

$$\text{div}\sigma = \Big(\sum_{j=1}^{N} \frac{\partial \sigma_{ij}}{\partial x_j}\Big)_{1\leq i\leq N}$$

and more precisely,

$$-\sum_{j=1}^{N} \frac{\partial}{\partial x_j}\Big(\mu\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) + \lambda(\text{div}u)\delta_{ij}\Big) = f_i \quad \text{in} \quad \Omega$$

where $\text{div}u = tr(e(u))$ and $1 \leq i \leq N$. In addition to this equation we want to add boundaries conditions. The first condition is Dirichlet condition,

$$u = 0 \quad \text{on} \quad \Gamma_D.$$

The second condition is Neumann Boundary condition,

$$\sigma n = g \quad \text{on} \quad \Gamma_N,$$

where $\partial\Omega = \Gamma_N \cup \Gamma_D$ and g is a vector valued function.
Combine all the above, we get,

$$\begin{cases} \text{div}\sigma = f & \text{in} \quad \Omega, \\ u = 0 & \text{on} \quad \Gamma_D, \\ \sigma n = g & \text{on} \quad \Gamma_N, \end{cases}$$

4.1.2 Finite Element Method

The Finite Element Method is a numerical method to approximate the solution of Elliptic Boundary Value problems [ACC07]. The idea of this method is based on the Variational Approach (Weak Formulation). According to the Variational Approach, the Elliptic Value problems (or the PDE problems) are reduced to the following problem.

$$\text{find} \quad u \in V \quad s.t \quad a(u,v) = L(v) \quad \forall v \in V$$

where V is a Hilbert space, a is a bilinear form, and L is a linear form. The main idea is to replace the space V with a subspace V_h of finite dimension. Thus, the problem becomes

$$\text{find} \quad u_h \in V_h \quad s.t \quad a(u_h,v_h) = L(v_h) \quad \forall v_h \in V_h.$$

Solving the second problem is easier, and it only requires solving a linear system. The complete proof of this fact is presented in [ACC07]. The first step of the solution is to introduce a basis $(\phi_j)_{1\leq j\leq N_h}$ for V_h. So, u_h can be written as a linear combination of this basis.

$$u_h = \sum_{j=1}^{N_h} u_j \phi_j.$$

Consequently, the problem becomes,

$$\text{find} \quad U_h \in R^{N_h} \quad s.t \quad a\Big(\sum_{j=1}^{N_h} u_j\phi_j, \phi_i\Big) = L(\phi_i) \quad \forall 1 \leq i \leq N_h,$$

where $U_h = (u_1, ..., u_{N_h})$ is a vector of R^{N_h}. The last problem is equivalent to a linear system

$$\mathcal{K}_h U_h = b_h$$

where the entries of $\mathcal{K}_h$ and b_h are

$$(\mathcal{K}_h)_{ij} = a(\phi_j, \phi_i), \quad (b_h)_i = L(\phi_i), \text{ for } 1 \leq i,j \leq N_h.$$

To illustrate the process, we will give a simple example for this method in one dimension. This example is presented in full details in [ACC07]. For the domain, we will choose $\Omega =]0,1[$. The mesh of this domain simply will be the collection of points $(x_j)_{0\leq j\leq n+1}$ such that,

$$x_0 = 0 < x_1 < ... < x_n < x_{n+1} = 1.$$

where the point $x_j s$ (nodes or vertices) are equidistant, i.e,

$$x_j = jh, \quad \text{with} \quad h = \frac{1}{n+1}, 0 \leq j \leq n+1.$$

And our object is to find the solution u of the following problem.

$$\begin{cases} -u'' = f & in \quad]0,1[\\ u(0) = u(1) = 0 \end{cases}$$

Next, we will replace the space $H_0^1(\Omega)$ (Check the appendix for the definition of this space) with the space $\mathbb{P}_1$, the set of polynomials of real coefficient with one real variable with degree less than or equal to 1. This equation has a unique solution in the new space if $f \in L^2\Omega$. For the basis function we will choose,

$$\phi_j(x) = \phi\left(\frac{x - x_j}{h}\right).$$

where ϕ is the hat function defined as follow,

$$\phi(x) = \begin{cases} 1 - |x| & if|x| \leq 1, \\ 0 & if|x| > 1. \end{cases}$$

The Variations formulation for this problem is

$$\text{find } u_h \in V_{0h} \quad \text{such that} \quad \int_0^1 u_h'(x)v_h'(x)dx = \int_0^1 f(x)v_h(x)dx \quad \forall v_h \in V_{0h}.$$

If we replace u_h by its expression with respect to ϕ_j and we choose $v_h = \phi$, we obtain the following

$$\sum_{j=1}^{n} u_h(x_j) \int_0^1 \phi_j(x)'\phi_i(x)'dx = \int_0^1 f(x)\phi_i(x)dx.$$

Then, denote $U_h = (u_h(x_1), ..., u_h(x_n))$, $b_h = \left(\int_0^1 f(x)\phi_i(x)dx \right)_{1 \leq i \leq n}$, the stiffness matrix

$$\mathcal{K}_h = \left(\int_0^1 \phi_j'(x)\phi_i(x)'dx \right)_{1 \leq i,j \leq n}$$

and what is remaining is to solve

$$\mathcal{K}_h U_h = b_h$$

4.1.3 Laguerre Diagram - Power Diagram

Suppose we have a finite set of points $M \in E^d$ where E^d is an Euclidean metric space of dimension d. Voronoi Diagram associates with each point p in M a cell $V(p)$ such that,

$$V(p) = \left\{ \forall x \in E^d, d(x,p) < d(x,q), \forall q \in M - \{p\} \right\}$$

In this context, d denotes the Euclidean distance function. However, for some applications a modified notion of Voronoi Diagram is needed. For instance, we can associate with each point p in M a real number $w(p)$ that we call the weight of the cell. Then, the distance of a point in M to a point x can be measured as function of $d(x,p)$ and $w(p)$. The formula $d(x,p) - w(p)$ leads to a diagram whose planar instance can be interpreted as a Voronoi diagram for circles of centre p and radius $w(p)$ [Aur87]. If we measure the distance by the function $d(x,p)^2 - w(p)$, we obtains Laguerre Diagram (or Power Diagram) such as the one in Figure 6. According to [Aur87], the geometric interpretation is viewing p with weight

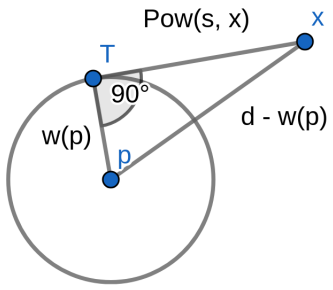

Figure 5: Geometric Representation of pow(x, s)

$w(p) \geq 0$ as the centre of a sphere $s = \{x \in E^d | d(x,p) = \sqrt{w(p)}\}$, and the distance function is known as the power of a point $x \in E^d$ with respect to s denoted by $pow(x,s)$. For instance in the planar case, $pow(x,s)$ is the length of segment from the tangent point between the line that passes by x and circle to x as it is shown in Figure 5. Laguerre Diagram has many properties. However, in our work, we are interested only in the following one. Given two points $s, t \in M$ with weights $w(s)$ and $w(t)$ respectively, then the points that satisfy $pow(x,s) = pow(x,t)$ describe a hyperplane

$$h : 2x(s-t) = w(t)^2 - w(s)^2 - t^2 + s^2$$

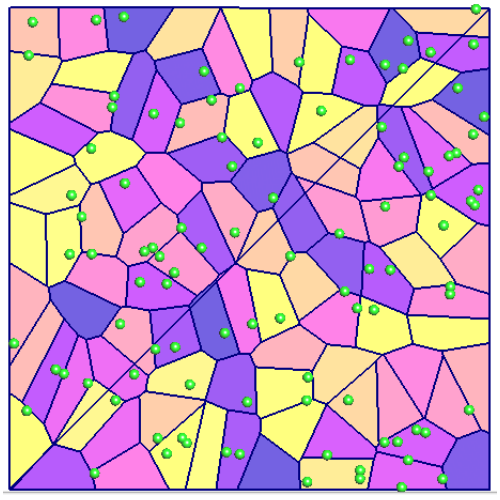

Figure 6: Instance of Laguerre Diagram (Notice that the seed points can be outside their cell).

4.1.4 Optimal Transport Theory

Optimal Transport is a mathematical theory that is concerned with the optimal allocations of resources. Furthermore, this theory provides us with the means to measure distances between functions, to compare function, and to interpolate between functions. This theory has important applications in computer graphics and computational physics such as fluid simulation. However, we are interested with this theory because it gives us the ability to preserve the volumes of Laguerre cells and by consequence we can satisfy the matter constraints of our topological optimisation problem. In this document and in our work, we are not concerned with the details of this theory, but for more details we direct the reader to these resources [LS18] and [Vil08].

4.2 Our Algorithm

4.2.1 General Description

We suppose that we have a finite regular space (or domain) denoted by $\mathcal{D}$. We suppose also that we have a set of n points $\mathcal{P}$ in $\mathcal{D}$ where these n points are partitioned into two disjoint sets of points, that is $\mathcal{P} = \mathcal{F} \cup \mathcal{E}$.

$$\mathcal{P} = \begin{cases} x_i \in \mathcal{F} & i = 1, ..., n_f, \\ y_i \in \mathcal{E} & i = 1, ..., n_e, \end{cases}$$

where $n = n_f + n_e$ and the set $\mathcal{F}$ is the set of the centres (seed) of a region that contains matters, and $\mathcal{E}$ is the set of centres (seed) of the region that is void. In addition, each points in $\mathcal{P}$ has a weight $\psi_i \in \mathcal{R}$. Then we will define the shape Ω.

Definition 4.1. $\Omega \subset D$ is the set of Laguerre cells defined as follows,

$$V_i^{\psi_i} = \left\{ ||x - x_i||^2 - \psi_i \leq ||x - x_j||^2 - \psi_j \quad \forall j \in \mathcal{F} \right\}$$

in other words,

$$\Omega = \bigcup_{i \in \mathcal{F}} V_i.$$

Furthermore, we want to define some constraints over Ω. The first constraint that we will impose is the volume constraint. We want to preserve the volume of Ω. i.e

$$|\Omega| = V_0$$

at any time. Another precise definition of the volume constraint is the following.

$$\begin{cases} |V_x| = \frac{V_0}{n_f} & \forall x \in \mathcal{F}, \\ \\ |V_y| = \frac{|D| - V_0}{n_e} & \forall y \in \mathcal{E}, \end{cases}$$

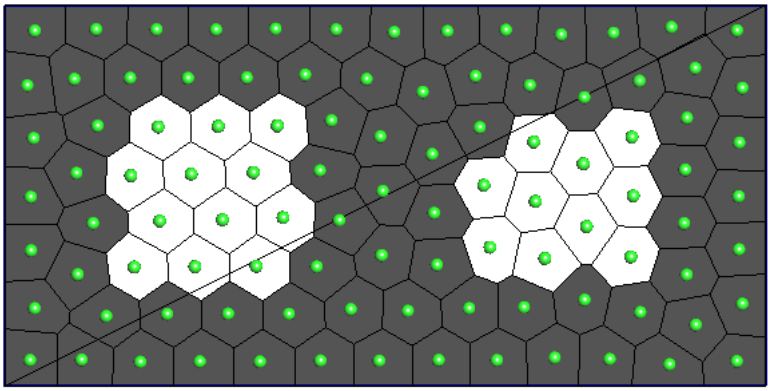

Figure 7: An instance of initial state where the dark cells represent the matter and the white cells represent the void.

To satisfy these constraints, we will make use of theorems and techniques from optimal transport theory. For instance, we know the following theorem.

Theorem 4.1. *There exists a unique ψ such that the above constraints are satisfied.*

The algorithm starts by computing the effect of a pulling down force on the initial structure (such as the one in Figure 7) using Finite Elements Method. After , we evaluate the gradient of the objective function which is in our case the compliance.

$$J(\Omega) = \int_{\Omega} f.udx$$

where f is the force applied on the structure and u is the displacement resulted from such a force. In our work, we want to compute the gradient with respect to the positioning of the seed points. For this reason, we will compute first the gradient of the objective function with respect to the boundary points of the mesh (Laguerre diagram in our cases) using the formula stated in [CBN+15]. Then, we compute the derivative of the boundary with respect to the seed points using a formula similar to the one presented in [LL10]. After, we combine these two computations together using the chain rule. That is, let x be the vector of the seed point and let q be a boundary vertex then,

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial q}\frac{\partial q}{\partial x} + \frac{\partial J}{\partial q}\frac{\partial q}{\partial \phi}\frac{\partial \phi}{\partial x}$$

The proofs of these computations are presented in the next two sections. However, for this moment we are ignoring the second term in our work. Later upon computing the gradient we moves our seed points in the direction that minimises the objective function. At this point, we compute the weights that preserve the volume of our Laguerre Diagram, then we compute the new diagram and we repeat the same steps until reaching a stopping criteria. In summary, our algorithm is the following.

1. Initialise Laguerre Diagram.

2. Apply a pull down force over the structure.
3. Compute the effect of this pull down force using Finite Element Method.
4. Compute the gradient of the objective function with respect to the boundary points of the Laguerre cells.
5. Compute the gradient of the boundary points of the Laguerre cells with respect to their seeds.
6. Use the chain rule to compute the derivative with respect to the seed of Laguerre cells.
7. Take a descent step according to the gradient.
8. Compute the new weights of Laguerre cells.
9. Go to 1 then 2 and 3, in case of convergence stop otherwise continue.

4.2.2 The Derivative of the Objective Function with Respect to the Boundaries Of Laguerre Cells

Our objective in this section is to prove the derivative of the objective function using the adjoint method presented in [Bra10] and [All15]. The use of this method allows a faster computation of the derivative. To begin our proof, we have the objective function J (the compliance).

$$J = u^T f$$

where u is the solution of

$$K(m, x)u = f(m, x) \tag{1}$$

and x represents the position of the nodes in the mesh (the boundaries of Laguerre cells). Furthermore, K is a symmetric matrix. The formula that we want to prove is the following.

$$\frac{\partial J}{\partial x} = -u^T \frac{\partial K}{\partial x} u + 2u^t \frac{\partial f}{\partial x}$$

In order to accomplish our task, we will make use of the following identity. let $u = u(x), \quad v = v(x)$, then

$$\frac{\partial (u.v)}{\partial x} = \frac{\partial u^t v}{\partial x} = u^T \frac{\partial v}{\partial x} + v^T \frac{\partial u}{\partial x}$$

Proof: we start by deriving J with respect to x.

$$\frac{\partial J}{\partial x} = \frac{\partial u^T f}{\partial x}$$

using the identity mentioned above, we get

$$\frac{\partial J}{\partial x} = \frac{\partial u^T f}{\partial x} = u^T \frac{\partial f}{\partial x} + f^T \frac{\partial u}{\partial x}$$

Now define the following Lagrangian function,

$$\mathcal{L}(u, \lambda) = J + \lambda^T (Ku - f)$$

we know that for u_Ω solution of the equation (1), we have

$$\mathcal{L}(u_\Omega, .) = J$$

Thus,

$$\frac{\partial \mathcal{L}}{\partial x}(u_\Omega, .) = \frac{\partial J}{\partial x}$$

We will derive this function with respect to x, so we get,

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial J}{\partial x} + \lambda^T \frac{\partial (Ku - f)}{\partial x} = u^T \frac{\partial f}{\partial x} + f^T \frac{\partial u}{\partial x} + \lambda^T \Big(\frac{\partial K}{\partial x} u + K \frac{\partial u}{\partial x} - \frac{\partial f}{\partial x} \Big)$$

$$= \big(f^T + \lambda^T K \big) \frac{\partial u}{\partial x} + u^T \frac{\partial f}{\partial x} + \lambda^T \frac{\partial K}{\partial x} u - \lambda^T \frac{\partial f}{\partial x}$$

now if we set u to u_Ω we know that the derivative of this function will be equal to the derivative of the objective function for any λ. Thus, we will choose λ that cancel the derivative of u with respect to x (which is hard to compute and the reason of using such method). i.e

$$f^T + \lambda^T K = 0$$

$$f^T = -\lambda^T k$$

$$(K u_\Omega)^T = -\lambda^T K$$

$$u_\Omega^T K^T = -\lambda^T K$$

since $K = K^T$

$$u_\Omega^T K = -\lambda^T K$$

$$\lambda = -u_\Omega$$

replacing λ by its value in the derivative of $\mathcal{L}$ give us what we want,

$$\frac{\partial \mathcal{L}}{\partial x}(u_\Omega, .) = \frac{\partial J}{\partial x} = u_\Omega^T \frac{\partial f}{\partial x} - u_\Omega^T \frac{\partial K}{\partial x} u_\Omega + u_\Omega^T \frac{\partial f}{\partial x} = 2 u_\Omega^T \frac{\partial f}{\partial x} - u_\Omega^T \frac{\partial K}{\partial x} u_\Omega$$

After computing this derivative, what remains is to compute is the derivative of the matrix K with respect to x. We choose to perform this competition using finite differences but later, we will propose a more faster method.

Figure 8: Boundary point constructed by the intersection of three lines

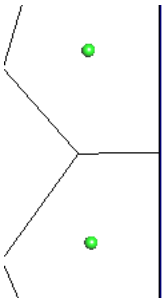

Figure 9: Boundary point constructed by the intersection of two lines (one belongs to Laguerre Diagram and the other to the boundary domain)

4.2.3 The Derivatives of the Boundaries with Respect to the Seed of Laguerre Cells

In this section, we are presenting the derivation of the derivative of the boundary points of the cell with respect to its seed. Similar computations are presented in [LL10]. Furthermore, we will restrict our computation for the two dimensional case. Since boundary points in setting are constructed either from the intersection of three lines (Figure 8) or from one line and the boundary of the domain (Figure 9), we will split our computation to two cases.

The first case (the intersection of three lines), let x_1, x_2 and x_3 be the seeds points and ψ_1, ψ_2, and ψ_3 be their weights respectively, and let q be the boundary point. From Laguerre diagram property presented in section 4.1.5 we know the following.

$$\begin{cases} (1) & d^2(q, x_1) - \psi_1 = d^2(q, x_2) - \psi_2 \\ (2) & d^2(q, x_1) - \psi_1 = d^2(q, x_3) - \psi_3 \end{cases}$$

where the function d is the Euclidean distance. Expanding both (1) and (2) yields to the following equations.

$$\begin{cases} (1) & 2q.(x_2 - x_1) = -||x_1||^2 + ||x_2||^2 + \psi_1 - \psi_2. \\ (2) & 2q.(x_3 - x_1) = -||x_1||^2 + ||x_3||^2 + \psi_1 - \psi_3. \end{cases}$$

Thus we can determine q by the following.

$$q = \frac{1}{2} \begin{pmatrix} x_2 - x_1 \\ x_3 - x_1 \end{pmatrix}^{-1} \begin{pmatrix} -||x_1||^2 + ||x_2||^2 + \psi_1 - \psi_2 \\ -||x_1||^2 + ||x_3||^2 + \psi_1 - \psi_3 \end{pmatrix}$$

we will call the first matrix A, and the vector B, so

$$q = \frac{1}{2}A^{-1}B$$

Using the matrix derivatives mentioned in [PM99]. we get the following expression.

$$\partial q = \frac{1}{2}\partial(A^{-1}B) = \frac{1}{2}(\partial A^{-1}B + A^{-1}\partial B)$$

$$= \frac{1}{2}(-A^{-1}\partial A A^{-1}B + A^{-1}B)$$

$$= \frac{1}{2}A^{-1}(-\partial A A^{-1}B + A^{-1}\partial B)$$

notice that $q = \frac{1}{2}A^{-1}B$, so we can replace $\frac{1}{2}A^{-1}B$ by q.

$$= A^{-1}(\frac{1}{2}\partial B - \partial A q)$$

Thus, we only need to compute ∂Aq and ∂B. The derivation of ∂Aq and ∂B with respect to $[x_1, x_2, x_3, \psi_1, \psi_2, \psi_3]$ are:

$$\partial(AQ) = \begin{pmatrix} -q & -q & 0 & 0 & 0 & 0 \\ -q & 0 & -q & 0 & 0 & 0 \end{pmatrix}$$

and

$$\partial B = \begin{pmatrix} -x_1 & x_2 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 \\ -x_1 & 0 & x_3 & \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

For the second case (the intersection of one line with the boundary), let x_1 and x_2 be the seed points and let ψ_1 and ψ_2 be their weights respectively, and let q be the boundary point and $l : (x, y).v = c$ is the given line. We know that,

$$\begin{cases} (1) & d^2(q, x_1) - \psi_1 = d^2(q, x_2) - \psi_2 \\ (2) & l : q.v = c \end{cases}$$

then expanding the first equality gives us the same result as before.

$$\begin{cases} (1) & 2q.(x_2 - x_1) = -||x_1||^2 + ||x_2||^2 + \psi_1 - \psi_2 \\ (2) & q.v = c \end{cases}$$

Thus q is determined by the following.

$$q = \begin{pmatrix} x_2 - x_1 \\ v \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{2}(-||x_1||^2 + ||x_2||^2 + \psi_1 - \psi_2) \\ c \end{pmatrix}$$

Similarly to the first case, using the derivation rules of [PM99], yields to

$$\partial q = A^{-1}(\partial B - \partial A q)$$

where A is the first matrix and B is the vector, and their derivatives eith respect to $\partial x = [x_1, x_2, \psi_1, \psi_2]$ are

$$\partial(AQ) = \begin{pmatrix} -q & -q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\partial B = \begin{pmatrix} -x_1 & x_2 & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

This ends the proof of the two derivatives.

4.2.4 Implementation Details

To experiment our work, we are using two main components Graphite which is a software for graphics modelling built on Geogram library for geometric algorithms. Furthermore, we are using MFEM library to solve Linear Elasticity system and analyse the behaviour of our structure. In addition, we are using Armadillo Library to do linear algebra operations.

4.3 Experimental Results

Currently, there are no significant results to show. However, we still working on debugging and improving our codes. For now, our experimentation is focused on the two dimensional case that we think might have some problems that we can overcome on the three dimensional case. From these problems, we think that in two dimensions a void cell might not be able to penetrate the matters cells because that will weak the structure thus the optimisation function must go over some local maximum to attain another minimum. But in three dimensions this problem might not exist. Another solution might be changing the initial configuration. However, we were able to get the following result in Figure 10 that shows a change in the topology of the structure in Figure 7. But the problem in the result presented in Figure 10

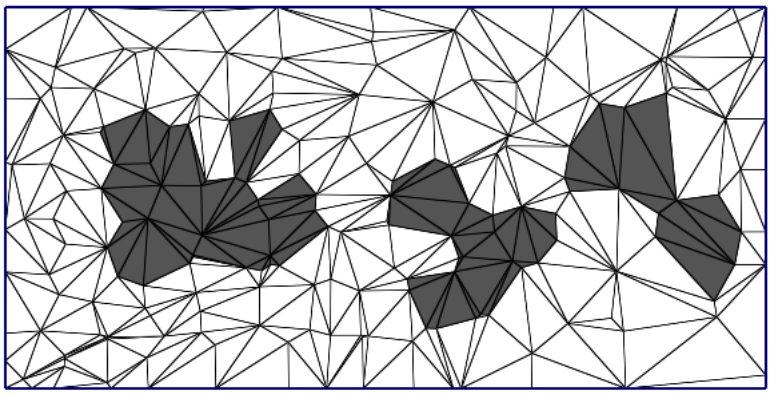

Figure 10: Change of the Topology of Figure 7 (but in this figure the white cells are the matter, and the black ones are the void).

is the stability of the objective function. The objective function does not decrease steadily

toward a minimum. Figure 11 shows how the objective function change in the experiment that gave the result of Figure 10. But after decreasing the descent step, we were able to get

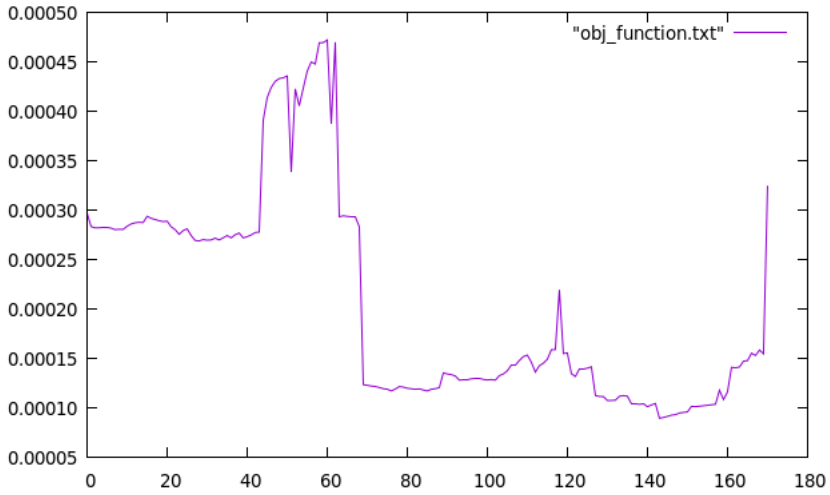

Figure 11: The objective function of Figure 10, the horizontal axis is the number of iteration, and the vertical one is the value of the objective function.

a steadier objective function such as the one in Figure 12.

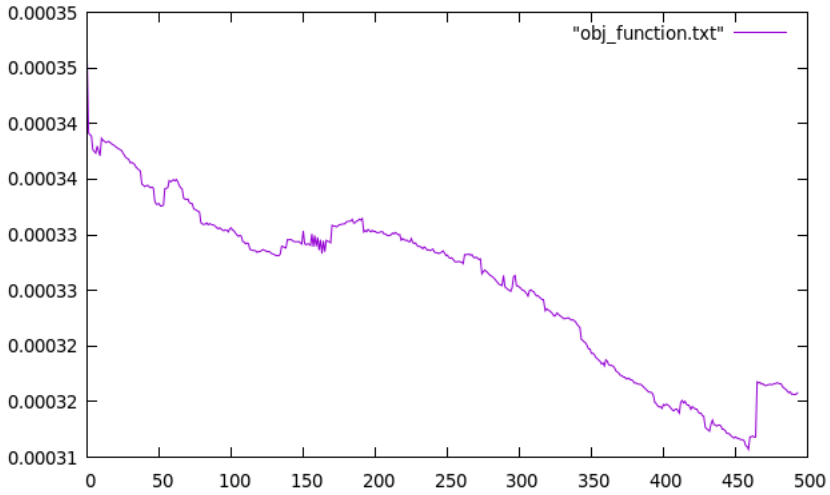

Figure 12: A steadier decrease of the objective function but with no significant change in the topology. (The horizontal axis is the number of iteration, and the vertical one is the value of the objective function.)

In summary, we were able to optimise the objective function. But this optimisation was not simultaneous with the optimisation of the topology of the shape.

4.4 Future Works

Our future plan is to work on the two dimensional model to be able to get a better result. For this reason, we will improve the quality of our code, and we will try better techniques to compute the derivatives. In case of success, we will work on the implementation of a three dimensional version. Then, we want to test with different objective functions.

5 Conclusion

In this work, we aimed at two things. The first is to present a brief introduction on shape optimisation. The second is to introduce a new algorithm to solve a shape optimisation algorithm. In this method, we have a new and unique representation of the shape using Lagurre Diagram. Furthermore, we are taking a new approach by allowing the cells of this diagram to change their positions and to preserve its volume using techniques from optimal transport theory. Unfortunately, our work is not yet complete, but further enhancements and improvements will be available in the future.

Appendices

Appendix A: In this appendix, we will just state the definition of the mathematical spaces used in this report. These definitions are presented and detailed in [ACC07].

Definition .1. Let v be a function of $L^2(\Omega)$, and Ω be an open set of $\mathbb{R}^N$. We say that v is differentiable in the weak sense in $L^2(\Omega)$ if there exist functions $w_i \in L^2(\Omega)$, for $i \in \{1, ..., N\}$, such that, for every function $\phi \in C_c^\infty(\Omega)$, we have

$$\int_\Omega v(x)\frac{\phi}{\partial x_i}(x)dx = -\int_\Omega w_i(x)\phi(x)dx.$$

Each w_i is called the ith weak partial derivative of v.

Definition .2. Let Ω be an open set of $\mathbb{R}^N$. The Sobolov space $H^1(\Omega)$ is defined by

$$H^1(\Omega) = \big\{v \in L^2(\Omega) \text{ such that } \forall i \in \{1, ...N\}\frac{v}{x_i} \in L^2(\Omega)\big\},$$

where $\frac{\partial v}{\partial x_i}$ is the weak partial derivative of v in the sense of definition 1.

Definition .3. For every integer $m \geq 0$, the Sobolev space $W^{m,p}(\Omega)$ is defined by

$$W^{m,p}(\Omega) = \{v \in L^p(\Omega) \text{ such that } , \forall \alpha with |\alpha| \leq |m|, \partial^\alpha v \in L^p(\Omega)\},$$

where the partial derivative $\partial^\alpha v$ is taken in the weak sense.

References

[ACC07] G. Allaire, A. Craig, and D.M.A. Craig. *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*. Numerical Mathematics and Scientific Computation. OUP Oxford, 2007.

[All] Gregoire Allaire. Optimal design of structures map 562.

[All15] Grégoire Allaire. A review of adjoint methods for sensitivity analysis, uncertainty quantification and optimization in numerical codes. *Ingénieurs de l'Automobile*, 836:33–36, July 2015.

[Aur87] F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[Bra10] Andrew M. Bradley. Pde-constrained optimization and the adjoint method. June 4, 2013 (original November 16, 2010).

[CBN+15] Asger Nyman Christiansen, Jakob Andreas Bærentzen, Morten Nobel-Jørgensen, Niels Aage, and Ole Sigmund. Combined shape and topology optimization of 3d structures. *Computers & Graphics*, 46:25–35, 2015.

[CK08] P.W. Christensen and A. Klarbring. *An Introduction to Structural Optimization*. Solid Mechanics and Its Applications. Springer Netherlands, 2008.

[Cé86] J. Céa. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. page 371–420, 1986.

[Dap13] Charles Dapogny. *Shape optimization, level set methods on unstructured meshes and mesh evolution*. PhD thesis, 2013. Thèse de doctorat dirigée par Allaire, Grégoire et Frey, Pascal Mathématiques Appliquées Paris 6 2013.

[dSA06] M. de Schoenauer and G. Allaire. *Conception optimale de structures*. Mathématiques et Applications. Springer Berlin Heidelberg, 2006.

[EKS94] H. A. Eschenauer, V. V. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1):42–51, Aug 1994.

[ES15] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer Berlin Heidelberg, 2015.

[GM12] Dheeraj Gunwant and Anadi Misra. Topology optimization of continuum structures using optimality criterion approach in ansys. 5:470–485, 11 2012.

[LL10] Bruno Lévy and Yang Liu. L_p centroidal voronoi tessellation and its applications. *ACM Trans. Graph.*, 29(4):119:1–119:11, 2010.

[LS12] S. Lyle and J. Salencon. *Handbook of Continuum Mechanics: General Concepts Thermoelasticity*. Springer Berlin Heidelberg, 2012.

[LS18] Bruno Lévy and Erica L. Schwindt. Notions of optimal transport theory and how to implement them on a computer. *Computers & Graphics*, 72:135–148, 2018.

[McG] Bob McGinty. Continuum mechanics. Lectures.

[Min02] Remo Minero. Numerical integration in more dimensions. Seminar, 2002.

[PM99] Thomas P. Minka. Old and new matrix algebra useful for statistics. 04 1999.

[Pug15] C.C. Pugh. *Real Mathematical Analysis*. Undergraduate Texts in Mathematics. Springer International Publishing, 2015.

[Sig] Ole Sigmund. Topology optimization state-of-the-art and future perspectives.

[Str07] W.A. Strauss. *Partial Differential Equations: An Introduction, 2nd Edition*. Wiley Global Education, 2007.

[TWH10] G.B. Thomas, M.D. Weir, and J. Hass. *Thomas' Calculus*. Always Learning. Pearson, 2010.

[Vil08] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.