



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Superior d'Enginyeries Industrial,  
Aeroespacial i Audiovisual de Terrassa

# Study of optimal shapes for lightweight material design

**Document:**

Final report

**Author:**

Joel Carrasco Bañales

**Director - Codirector:**

Alex Ferrer Ferre - Juan Manuel Podestá

**Titulation:**

Degree in aerospace technology engineering

**Announcement:**

Spring 2022.

END OF STUDIES PROJECT

# Abstract

The present end of studies project tries to assimilate the connections between the geometrical and physical domain in order to design optimal microstructures with the desired properties in 2D. The link between both domains is established by means of the crystallographic point groups, which relate the topology of the minimum volume unit and its symmetries with the elastic tensor. Therefore, there are two pre-processing variables that play a determining role on the way to the optimal topology: the shape of the mesh and the symmetries of the material distribution inside it.

For this reason, in the present study a shape generator and unit cell meshing algorithm is implemented and a topological optimizer code is used to distribute geometrically the material inside the unit cells in order to obtain the desired elastic tensor (resolution of the inverse problem) while minimizing the amount of material used. In order to obtain the desired material properties, the capacity of the topological optimizer to generate the necessary geometric symmetries in the microstructure that guarantee the physical symmetries required by the design target tensor is evaluated.

Therefore, during the course of the study there will be a theoretical review of topological optimization, crystallography and geometric and tensor symmetries, the development of the structure and operation of the mesh generator code and a practical study of the optimizer's capacity to obtain the tensors designed with the selected lattice topologies. At the same time, the essential organizational concepts and main differences between the programming used in the meshing algorithm, that is object-oriented programming, and modular or functional programming, are also reviewed.

# Resumen

El presente proyecto de fin de estudios trata de asimilar las conexiones entre el dominio geométrico y el físico para diseñar microestructuras óptimas con las propiedades deseadas en 2D. El vínculo entre ambos dominios se establece mediante los grupos de puntos cristalográficos, que relacionan la topología de la unidad de volumen mínima y sus simetrías con el tensor elástico. Por lo tanto, aparecen dos variables de pre-proceso que juegan un papel determinante en el camino hacia la topología óptima: la forma de la malla y las simetrías de la distribución del material en su interior. Por este motivo, en el presente estudio se implementa un algoritmo generador de mallas con las formas de las celdas unitarias y se utiliza un código optimizador topológico para distribuir geoméricamente el material dentro de las celdas unitarias con el fin de obtener el tensor elástico deseado (resolución del problema inverso) minimizando la cantidad de material utilizado. Para obtener las propiedades deseadas del material, se evalúa la capacidad del optimizador topológico para generar las simetrías geométricas necesarias en la microestructura que garanticen las simetrías físicas requeridas por el tensor objetivo de diseño.

Por tanto, en el transcurso del estudio se realizará una revisión teórica de la optimización topológica, la cristalografía y las simetrías geométricas y tensoriales, el desarrollo de la estructura y funcionamiento del código generador de mallas y un estudio práctico de la capacidad del optimizador para obtener los tensores diseñados con las topologías de red seleccionadas. Al mismo tiempo, también se revisan los conceptos organizativos esenciales y las principales diferencias entre la programación utilizada en el algoritmo de mallado, es decir, la programación orientada a objetos, y la programación modular o funcional.

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Aim . . . . .	1
1.2	Scope . . . . .	1
1.3	Requirements . . . . .	2
1.4	Justification . . . . .	3
<b>2</b>	<b>State of the art</b>	<b>4</b>
<b>3</b>	<b>Theoretical development</b>	<b>6</b>
3.1	Introduction to topology optimization . . . . .	7
3.1.1	Density-based methods . . . . .	8
3.1.2	Boundary variations methods: Level Set . . . . .	8
3.2	The inverse problem . . . . .	9
3.3	Elastic sym. & micro. configuration . . . . .	10
3.3.1	Periodic micro-architecture materials . . . . .	11
3.3.2	Type of tensors according to its symmetries . . . . .	17
3.3.3	Connection between geometrical and physical symmetry . . . . .	19
3.3.4	Neumann's principle . . . . .	19
<b>4</b>	<b>Mesher performance</b>	<b>21</b>
4.1	Variables to be defined by the user . . . . .	21
4.2	Boundary nodes disposition mechanism . . . . .	24
4.3	Interior nodes disposition mechanisms . . . . .	26
4.4	Creation of elements: connections between nodes . . . . .	28
4.5	Implementation of boundary conditions . . . . .	29
4.6	Organization of the code . . . . .	30
4.7	Performance verification . . . . .	34
<b>5</b>	<b>Micro-structures design methodology</b>	<b>35</b>
<b>6</b>	<b>Practical assessments</b>	<b>37</b>
6.1	Convergence of results . . . . .	37
6.2	Demonstration of the Neumann's principle . . . . .	42
6.3	Top Opt in micro-structure design . . . . .	45
6.3.1	Isotropic target tensor . . . . .	46



---

6.3.2	Orthotropic target tensor . . . . .	50
6.4	Analysis of the non-Voronoi rhomboid cell . . . . .	54
<b>7</b>	<b>Future project expansions</b>	<b>57</b>
<b>8</b>	<b>Analysis and evaluation of the environmental, social and technological implications</b>	<b>59</b>
<b>9</b>	<b>Conclusions</b>	<b>61</b>

# List of figures

2.1	Topological optimization using Ansys interface . . . . .	4
3.1	Logical steps order comparison between traditional design and topology optimization. . . . .	7
3.2	2D design domain created by the level set model [9]. . . . .	9
3.3	Compatible and incompatible polygons when applying the crystallographic restriction theorem. . . . .	11
3.4	Demonstration of tessellation compliance in the hexagon and non-compliance in the octagon . . . . .	12
3.5	Determination of Bravais lattice, Voronoi cell and motif of the pegasus tessellation artistic wallpaper created by Escher. . . . .	13
3.6	Types of Bravais lattices in the plane [4]. . . . .	13
3.7	Bravais lattices with parallel lines indicating the mirror planes [4]. . . . .	14
3.8	Explicative figure of the notation used for the plane groups. . . . .	15
3.9	Voronoi cells for the seventeen point groups in 2D indicating the symmetry elements that compose each one. The colours inside the cell identify the phases of the composite [4]. . . . .	16
4.1	Representation of the required inputs to generate a 4-sided polygon. . . . .	22
4.2	Representation of the required inputs to generate a 6-sided polygon. . . . .	23
4.3	Boundary obtained by the mesh creator for the five inputs specified in 4.1 . . . . .	24
4.4	Representation of the data used for the calculation of the coordinates of the vertex nodes. Note that the numeration of the nodes is also shown. . . . .	25
4.5	Boundary interior master and slave nodes in the six-sided polygon of figure 4.4. Note that the numeration of the nodes is also shown. . . . .	26
4.6	Procedure followed by the intersection mesher. The figure shows the original polygon and the first iteration of the intersection mesher. . . . .	27
4.7	Procedure followed by the division of diagonals mesher. The figure shows the real polygon and the first scale with the nodes of the master sides. . . . .	28
4.8	Numbered square mesh with the master nodes in green and the slave nodes in red together with the master-slave matrix. . . . .	30
4.9	UML diagram of the Mesh Creator code. . . . .	33
4.10	Performance of the code using different inputs. . . . .	34
6.1	Convergence of the squared mesh tensor upon the increase of divUnit. . . . .	39
6.2	Convergence of the Rectangular Primitive mesh tensor upon the increase of divUnit. . . . .	40
6.3	Convergence of the Hexagon mesh tensor upon the increase of divUnit. . . . .	40

6.4	Convergence of the Rectangular Centered mesh tensor upon the increase of divUnit.	40
6.5	Convergence of the Oblique mesh tensor upon the increase of divUnit. . . . .	41
6.6	Convergence of the Rhomboid mesh tensor upon the increase of divUnit. . . . .	41
6.7	Comparative of the convergence of shaped mesh tensors upon the increase of DivUnit.	42
6.8	Undeformed state with the forced micro-architecture of each of the Voronoi cells and indication of the maximum plane group symmetry associated. . . . .	45
6.9	Non-isotropy coefficient and relative error for all the Voronoi cells and initial conditions considered . . . . .	49
6.10	Relative error for all the Voronoi cells and initial conditions considered when forcing an ortotropic tensor. . . . .	54
6.11	Identification of the design variables of the rhomboid shape. . . . .	55
6.12	Display of the rhomboid mesh ( $a = b$ , $\alpha = 60^\circ$ ), the deformed tessellated horizontal, vertical and shear tension states from left to right. . . . .	55
6.13	Demonstration of the relation between the non-Voronoi rhomboid cell and the regular hexagon. . . . .	56

# List of tables

3.1	Connection between the elastic symmetry classes with the plane groups and Voronoi cells . . . . .	19
4.1	Input data required to obtain the desired cell shape and divisions. . . . .	23
6.1	Relative error obtained for each number of divisions for the squared mesh. . . . .	39
6.2	Demonstration of the fulfillment of Neumann's theorem by means of the correspondence between the symmetry and shape of the micro-cell with the type of symmetry of the elastic tensor. In the first column we can find from left to right and from top to bottom, the mesh shape and divisions, the deformed for the horizontal, vertical and shear state of stress. . . . .	44
6.3	Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the circle inclusion. . . . .	47
6.4	Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the full of material cell. . . . .	48
6.5	Results obtained for every Voronoi cell when an isotropic tensor and using as initial state the inclusion of a void square. . . . .	48
6.6	Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the inclusion of void scattered holes. . . . .	49
6.7	Results obtained for every Voronoi cell when forcing an ortotropic tensor and using as initial state the inclusion of void circle. . . . .	51
6.8	Results obtained for every Voronoi cell when forcing an ortotropic tensor and using as initial state the full of material cell. . . . .	52
6.9	Results obtained for every Voronoi cell when forcing an ortotropic tensor and using as initial state the inclusion of a void square. . . . .	52
6.10	Results obtained for every Voronoi cell when forcing an ortotropic tensor and using as initial state the inclusion of void scattered holes. . . . .	53
6.11	Design parameters of the rhomboid mesh alongside with the tensor obtained and its symmetry. . . . .	56
7.1	Identification of future tasks to expand the study. . . . .	57

# Chapter 1

## Introduction

### 1.1 Aim

The aim of the project is to develop a new material design methodology to achieve the required material response by considering different shapes and symmetries in the periodic cell. It wills to characterize the relation between the size and symmetry of the periodic cell, the optimal micro-structure achieved and the overall elastic behaviour for the design of lightweight meta-materials.

### 1.2 Scope

The study to carry on in this project will include:

- Development of the code to implement different cell shapes. The objective is to code rectangular, squared, hexagonal and oblique shaped meshes.
- Development of the code that ensures that the repetition of the pattern obtained in the optimizer fits when trying to comply with the tessellation property (the pattern covers a 2D surface without overlaps or gaps).
- Development of the code to force different symmetries inside the mesh. Code able to force the appearance of different point and plane group symmetries.
- Study of the symmetries of the homogenized elasticity tensor when testing different shapes and symmetries to the mesh introduced in the topology optimizer. The objective is to obtain the set of characteristics of the mesh that ensure a particular homogenized elasticity tensor symmetry, ensuring at the same time the isotropy, orthotropy or anisotropy of the properties in the micro-structure obtained with the optimizer.
- Unification of the code in MATLAB classes to perform the object-oriented pattern of programming.

The study to carry on in this project will NOT include:

- Development of the topology optimization code. The optimization code used is the one provided by Alex Ferrer (Swan-master).

### 1.3 Requirements

The requirements of the project are divided in two groups. The first one is related with the restrictions for the final solution and the other with the procedure requirements for good understanding, storage and post-process of the code developed.

- Group 1 Restrictions for the final solution:
  - FR1: Determine the relationship between the shape and symmetry of the mesh with the properties of the 2D constitutive tensor obtained with the topology optimizer.
  - FR2: Ensure that all the results satisfy the tessellation principle (the pattern obtained must cover a surface guaranteeing that there are no overlaps or gaps).
- Group 2 Procedure requirements:
  - PR1: The code must be developed using the MATLAB software.
  - PR2: The changes in the code must be committed to an online repository called GitHub. This process enables the programmer or programmers to have an easy access to the code and its different versions.
  - PR3: The code must be developed using the object-oriented pattern of programming. This allows users to follow an ordered code, easier to understand.
  - PR4: The results of the optimizer must be post-processed with GiD.
  - PR5: Ensure the continuous proper performance of the code testing critical classes, functions and results.

## 1.4 Justification

As materials science has advanced, new methods have appeared that have allowed humans to create materials with the desired properties. One of these is topological optimization, which tries to minimize the cost function, being conditioned by certain imposed constraints [1][2]. This project tries to move in the same direction, focusing on one of the most elementary parts of a material, the micro-structure, playing with it to establish a connection between the shape of the micro-structure and its symmetry with the elastic behavior of the topologically optimized material and the symmetry of the elasticity tensor [3]. This would show that the restrictions imposed in the minimization of the cost can be combined with the shape of the cell for obtaining the desired properties when optimizing, thus giving the user great control over the material properties to be obtained [4] [5]. In addition, the project not only seeks to generate the code that allows the creation of the meshes, but it should also have the programming pattern dedicated to objects, thus making the code more orderly, understandable and easily editable for future programmers who work with it.

## Chapter 2

# State of the art

Topological optimization is a structural analysis process that, by solving multiple finite element problems, leads to structural solutions that generally seek to reduce the amount of material used without compromising any of its capabilities. Therefore, it is a tool of high importance for those industries where weight plays a determining role (orthopedic industry, automotive, aerospace...) and also allows the development of additive manufacturing technologies (SLA, SLS, SLM...), by the need of generating highly complex shapes. For this reason, some CAD software such as Altair, SolidWorks and Ansys incorporate this technology.

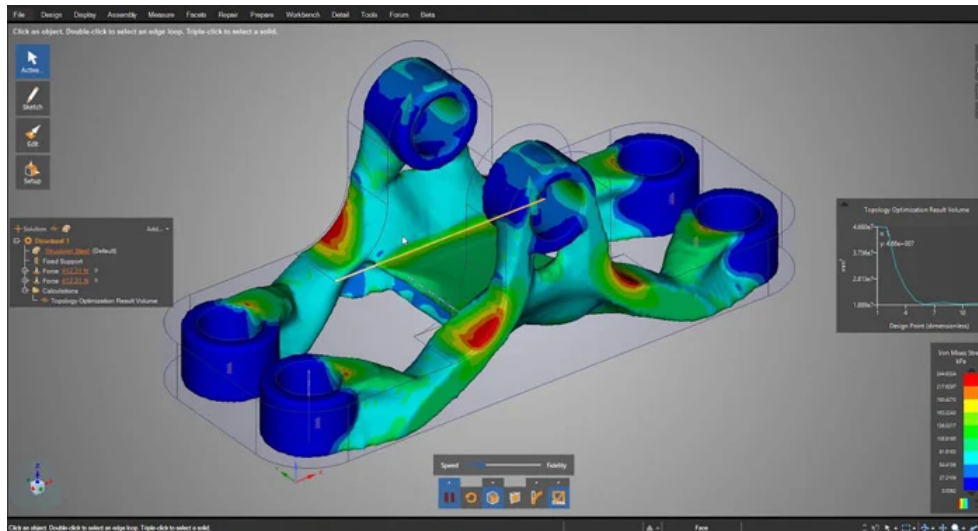


Figure 2.1: Topological optimization using Ansys interface

These programs focus on the optimization of macro-structures, while this project focuses on the optimization of structures at the microscopic level, hence, at the most elementary level of matter distribution in materials. There is no commercial software for this case specialized in the optimization of micro-structures and that is why, the state of the art is dominated by algorithms and investigations carried on by researchers. During the last few years, the research has been focused on studying those parameters of the elastic constitutive tensor that are controllable by the user of the topological optimization algorithm. Among these studies, the one performed by Amstutz [2010] [2] shows that



by knowing the parametric composition of the inverse of the orthotropic compliance tensor  $C^{-1}$ , they manage to optimize positions of the constitutive tensor to obtain a micro-architecture with maximum horizontal stiffness or optimum Poisson's ratio, among others.

$$C^{-1} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & 0 \\ -\frac{\nu_{21}}{E_2} & \frac{1}{E_2} & 0 \\ 0 & 0 & \frac{1}{G} \end{bmatrix} \quad (2.1)$$

In this matrix appear material design parameters, such as Young's modulus in the  $\hat{e}_1$  and  $\hat{e}_2$  directions of orthotropy  $E_1$  and  $E_2$ , shear modulus  $G$  and Poisson's ratios  $\nu_{12}$  and  $\nu_{21}$ .

But the development not only has advanced in the direction of optimizing some positions of the constitutive tensor. Some authors such as Ferrer et al. [2021] [1] try to obtain the optimal topology of a structure with an homogenized constitutive tensor  $C^h$  forcing it to be as similar as possible to a pre-established tensor  $C^*$ . This procedure is called the inverse problem, and in this case the parameters that make up the desired solution are known, and it is the micro-architecture that is unknown.

The work done by Podestá et al. [2019] [4] goes further, not only conditioning the optimizer to obtain a target tensor (inverse topological optimization problem) but also facilitating its convergence by choosing the appropriate shape and symmetries in the unit cell. In this way, it is shown that during the pre-processing phase, some design variables are chosen to make it easier or harder to reach certain symmetries in the 2D constitutive tensor (isotropy, orthotropy, tetragonal).

Other authors such as Barbarosie et al. [2017] [6] and Rossi et al. [2021][5] try to solve the inverse problem by focusing instead of optimizing the micro-structure of the whole unit cell, the minimum unit cell symmetry is optimized, thus exploiting the symmetry to simplify the number of variables to work with.

## Chapter 3

# Theoretical development

Over the years, nature has tended towards simplicity and effectiveness in its geometry and functionality. Therefore, it evolves and optimizes towards symmetrical geometries. The reflection of this, for example, is the honeycomb structure, which offers a great rigidity in comparison with its weight, obtaining an optimal geometry that requires a fair amount of material generating translations of hexagons in a predefined space. Through this theoretical analysis, we try to analyze the relationships between the symmetries of the constitutive tensor and the shapes and symmetries of the unit cells for 2D micro-structure problems, as well as the pre-processing factors that condition the optimizer (mesh shape, symmetries of the initial topology...). Once these relationships are understood, the goal is to emulate the evolution of nature in artificially developed micro-structures, optimizing materials for the desired functions.

The tool that will allow the optimization of the micro-structure will be the topological optimizer, which will distribute the material inside the unit cell in such a way that the shape functions involved in the problem are optimized and the constraints imposed on the resolution of the problem are respected. The problem to be solved is of the inverse problem type, therefore, the main objective of the optimizer is to try to adapt the distribution of the material to the maximum so that the tensor of the micro-structure obtained is equal to the tensor imposed by the user. But, it is not only possible to condition the optimizer with the shape functions and constraints, it is possible to condition its capacity with the choice of the shape and dimensions of the unit cell, that is, the optimization domain and the symmetries of the micro-structure generated inside. This means that some domains to be optimized will have the capacity to generate symmetries in the elasticity tensors that others will not be able to, thus making the pre-processing phase of great importance for the success of the design.

Since we are trying to optimize the periodic micro-structures of the materials, during the theoretical development, crystallographic arguments related to point groups and Bravais lattices are taken into account.

### 3.1 Introduction to topology optimization

As an optimization tool, topology optimization allows, by means of a part concept (mesh shape and boundary conditions), to obtain the optimal structure for a specific application, thus saving the need to constantly redesign and test the created models. The attached diagrams show the savings that a topological optimizer offers:

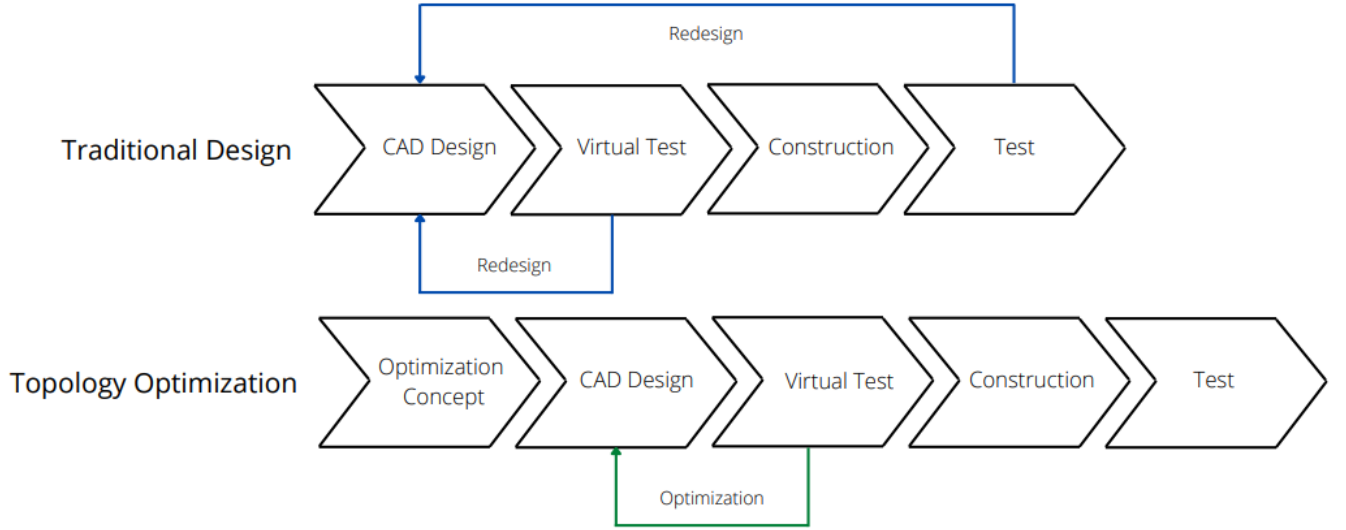


Figure 3.1: Logical steps order comparison between traditional design and topology optimization.

Therefore, it is a technique that provides a solution to one of the most common engineering problems: the design of the geometric configuration of a body that allows optimizing a function while ensuring that the constraints and boundary conditions are satisfied. The technique seeks for the optimal design.

Every optimization problem is based on the minimization of a function or set of shape functions (functions that provide a unique numerical value) called cost, while trying to satisfy certain equality, inequality or box constraints.

$$A = \begin{cases} \min_x & J(x) & (\text{cost}) \\ \text{s.t.} & a(x) = 0 & (\text{equality constraints}) \\ & b(x) < 1 & (\text{inequality constraints}) \\ & 0 \leq x \leq 1 & (\text{box constraints}) \end{cases} \quad (3.1)$$

We assume any microscopic domain  $D$ , whose inner material is a periodic composite consisting of two isotropic and elastic phases. The state of each of these phases is governed by the following characteristic function:

$$\chi(\bar{x}) = \begin{cases} 1 & \bar{x} \in \Omega \\ 0 & \bar{x} \notin \Omega \end{cases} \quad (3.2)$$

where  $\Omega$  is the set of all coordinates  $\bar{x}$  that belong to the domain  $D$  and in which the first of the phases appears. Trying to optimize the conditions specified in 3.1, the optimizer tests different distributions of the  $\Omega$  function, combining both phases of the material so that the smallest possible fraction of material is used, causing the  $\Omega$  domain to be the zone of presence of material and the  $\Omega_{void} = [\bar{x} \in D | \bar{x} \notin \Omega]$  the zone with absence of material.

### 3.1.1 Density-based methods

Normally, the optimization problem to be solved consists of a domain with a given number of finite elements, where the main objective is to determine the state of each of these elements (1 or 0, black or white, full or void) by minimizing the objective function. This usually results in a highly complex problem when dealing with a discrete state variable. Density-based methods solve this problem by replacing the discrete variable with a continuous variable and an iterative method that allows driving the solution towards a discrete scenario for each element (full/void). This is achieved with an interpolation function, where the continuous design variables are explicitly interpreted as the material density in each element,  $\rho_e$ . The density values from the interpolation are set to the range  $0 \leq \rho_e \leq 1$ , where 0 means absence of material and 1 means presence of material. As discussed above, it is necessary to combine the interpolation process with a method that allows penalizing the result to force the solution to a discrete result (as provided by SIMP [7] and RAMP [8] techniques). Density-based methods are usually governed by the following formulation for a static linear finite element analysis:

$$B = \begin{cases} \min & f(\rho, U) \\ \text{s.t.} & K(\rho)U = F(\rho) \\ & g_i(\rho, U) \leq 0 \\ & 0 \leq \rho \leq 1 \end{cases} \quad (3.3)$$

where  $f$  is the objective function,  $\rho$  is the vector of the density assigned to each design variable,  $U$  is the vector of displacements,  $K$  is the stiffness matrix,  $F$  is the force vector and  $g_i$  are the constraints.

### 3.1.2 Boundary variations methods: Level Set

A more recently developed method that, unlike density-based methods, relies on implicit functions that define structural contours instead of explicitly parameterizing the design domain. In the Level Set method, the contours are represented as the level zero curve of a scalar function  $\Phi$ . Boundary variations and merge, as well as the need for new hole appearances are performed on this scalar function while optimizing.

The shape and geometry of the contour are modified by controlling the movement of the level set according to the physical problem and the optimization conditions.

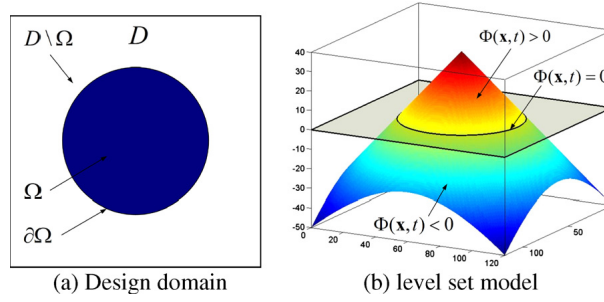


Figure 3.2: 2D design domain created by the level set model [9].

The figure above indicates the boundary contour between the material and non-material zone, defined as a level set isosurface of the 3D scalar function  $\Phi$  specified at the beginning of the section, for a specific step of time in the optimization. The contour is defined as:  $\delta\Omega = x(t) : \Phi(x(t), t) = k$  where  $k = 0$ , defining the 2D circle inside the  $D$  domain. The material-filled zone is that defined by the  $\Omega$  domain, and is characterized as the zone where  $\Phi(x, t) > 0$ . Conversely, the area that does not belong to  $\Omega$  (the area where  $\Phi < 0$ ) is in a state of absence of material.

Level Set method is governed by the following formulation for the minimization of the compliance with volume constraints:

$$C = \begin{cases} \min & c(U, \Phi) = U^T K(H(\Phi))U \\ \text{s.t.} & K(H(\Phi))U = F \\ & \int_{\Omega} \frac{H(\Phi)\delta\Omega}{V_0} \leq V^* \end{cases} \quad (3.4)$$

where  $V_0$  is the total geometric volume,  $V^*$  is the target volume fraction and  $H(\Phi)$  is the Heaviside function.

Note that the Heaviside function  $H(\Phi)$  represents the zones above and below the level set isosurface (presence and absence of material respectively), causing it to be equal to 1 for positive values of  $\Phi$  and 0 otherwise.

## 3.2 The inverse problem

Problem with high interest from the point of view of materials design, since it allows to obtain the configuration of the micro-architecture of a two-phase composite (phase of presence and absence of material) whose elasticity tensor is equal to one preset by the user. Therefore, by defining the working domain  $D$  and the desired tensor for the material  $C^*$ , the optimizer would generate the  $\chi$  distribution forming the  $\Omega$  domain. For this reason, the problem involves both the macroscopic scale, from which the homogenized tensor is calculated in each iteration ( $\sigma = C^h : \varepsilon$ ) and the microscopic scale, where the material design process is carried out.

The problem is formulated as follows:

$$D = \begin{cases} \min & \int_D \chi d\Omega \\ \text{s.t.} & \|C^h(\chi) - C^*\| = 0 \end{cases} \quad (3.5)$$

Note that the notation  $C^h(\chi)$  is used because of the dependence of the homogenized tensor on the material distribution in the working domain. It should also be noted that the resolution of the problem at the microscopic scale is solved considering periodic conditions of translation of cells, thus causing the fulfillment of the principle of tessellation at the macroscopic scale.

The presented approach allows to obtain the distribution of material with the desired properties, thus facilitating the design of the materials. However, it also causes implicit variables to appear that condition the optimizer's ability to converge to the desired tensor. Among these conditions are the size and shape of the domain  $D$ , the initial distribution of  $\Omega$  and the symmetries generated in the designed micro-architecture. This is because certain shapes and symmetries have a higher affinity with certain objective tensors, causing that the election should be made carefully according to the study carried out in the subsequent sections.

### 3.3 Relationship between elastic symmetries and micro-architecture configuration

In this section we will try to analyze the possible symmetric distributions of the two-phase composite material mentioned so far within the preset domain. This analysis will be based on providing the reason for the relationships established in the function  $C^h(\chi)$ , that is, how the elastic properties of the material are related by means of the constitutive tensor with the spatial distribution of the material in the micro-structure. This will be done based on the explanation on crystallographic principles and constraints for materials with periodic micro-architecture in the plane.

To understand the above relationship, all possible periodic micro-architectures will first be categorized by the point groups and the associated Bravais lattice in the plane. Then, the possible symmetries by means of the plane groups will be shown and the types of elastic constitutive tensors will be studied. Finally, the relations between the physical symmetries of the constitutive tensor with those of the material configuration will be established by means of the Neumann's principle.

#### Point group symmetries

Transformation or set of isometric transformations that keep the internal distribution of the minimum units of symmetry invariant. Each of these transformations is called a symmetry element, and the set of symmetry elements that form the geometric distribution of a material are the point groups. These transformations that form the groups are characterized by maintaining a fixed point of the geometry, therefore, two types of elements are distinguished:

- Rotations (denoted by the letter  $n$  in the notation used): transformation based on the rotation of the geometry of the minimum unit of symmetry a certain angle  $\alpha$  with respect to an axis orthogonal to the plane passing through the selected fixed point.
- Reflections (denoted by the letter  $m$ ): mirror-like symmetry on lines that belong to the plane and pass through the point that is held fixed. It should be noted that mirror lines are formed by the intersection between the plane of reflection and the working plane.

Once the elements that can compose a group have been presented, the two possible distinctions of point groups in the plane are made:

- Cyclic groups: groups composed only of rotationally symmetric elements. This group includes

$n$  rotations of  $\frac{360}{n}$  around the axis perpendicular to the plane.

- Dihedral groups: group that combines rotation elements and reflection elements, considering that all those lines generated by rotation are also mirror lines. Again, in this group there can be two cases:
  - Subgroup  $nm$  ( $n$  odd): the separation between the  $n$  reflection lines is  $\frac{180}{n}$ , therefore,  $2n$  symmetry elements appear ( $n$  rotations and  $n$  reflections).
  - Subgroup  $nmm$  ( $n$  even): in this case, the  $n$  rotations generate  $\frac{n}{2}$  mirror lines, whose separation will be the same as for the rotations. But, in this case, not only these mirror lines appear, but also those generated by the bisectors of the already existing mirror lines. This is because the combination of a reflection and a rotation of  $\frac{360}{n}$  is equivalent to a set of reflections forming an angle of  $\frac{180}{n}$  with respect to one of the original mirror lines, causing that in these subgroups appear two independent possibilities to consider the symmetries of the geometry.

Based on these considerations, from the geometrical point of view, there are infinite groups of point symmetries, since the only restriction is that  $n$  is an integer. On the other hand, from the crystallographic point of view, these point groups are considerably reduced due to the constraints of the crystallographic restriction theorem and the Bieberbach theorems. These theorems show that due to the peculiarities of lattices with translational symmetry (tessellation), if there is a displacement vector between two points of the lattice, then, this vector must be present throughout the domain. A few examples are given below to facilitate the understanding of the explanation:

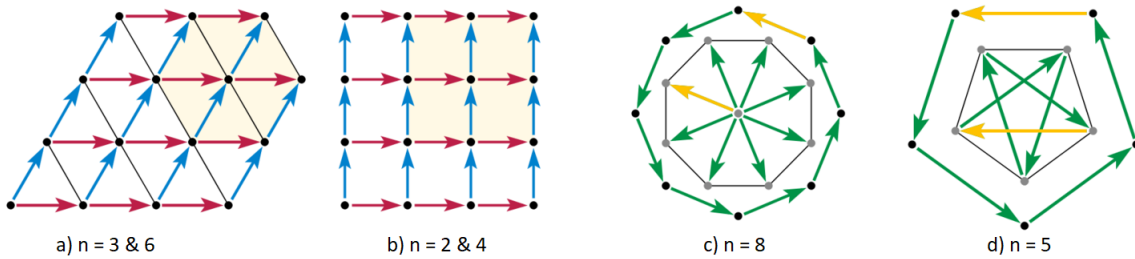


Figure 3.3: Compatible and incompatible polygons when applying the crystallographic restriction theorem.

Observing the previous figure and considering the theorem mentioned above, the origin of all the vectors corresponding to the sides of the polygon (with the rotational symmetry to be evaluated) can be placed at the same point, thus making them act as radial vectors that form the nodes of the original polygon. As shown in figure (3.3) this happens for hexagons and squares, but not for octagons and pentagons. It is observed, for example, that the vectors of the net of the octagon, only allow to obtain this same polygon with reduced area, making it non-compliant with crystallographic restrictions.

### 3.3.1 Periodic micro-architecture materials

Type of materials composed by the translational repetition of the micro-structure pattern around the 2D space selected. The micro-architecture remains invariant under discrete translations along

the two primitive vectors of the lattice. Thus, the material is generated respecting the following characteristic function:

$$f(\vec{x} + \vec{t}) = f(\vec{x}) \quad \text{where} \quad \vec{t} = \omega_1 \vec{a}_1 + \omega_2 \vec{a}_2 \quad (3.6)$$

where  $\vec{a}_1$  and  $\vec{a}_2$  are the primitive vectors that define the lattice and the directions to generate the translational symmetry and  $\omega_1$  and  $\omega_2$  are the scaling factors (define the distance of the translation with respect to the reference material or unit cell).

Any material with a periodic micro-architecture has a Bravais network composed of a minimum unit with translational periodicity whose shape ensures the tessellation principle. For this to be fulfilled, it must be ensured that the micro-architecture of the minimum unit of the network covers a predefined space in 2D, guaranteeing continuity in the topology of the network without overlaps or gaps. This is also a justification that shows why a Bravais lattice on a material with a periodic micro-architecture can be composed by a square or a regular hexagon, but in any case, by an octagon.

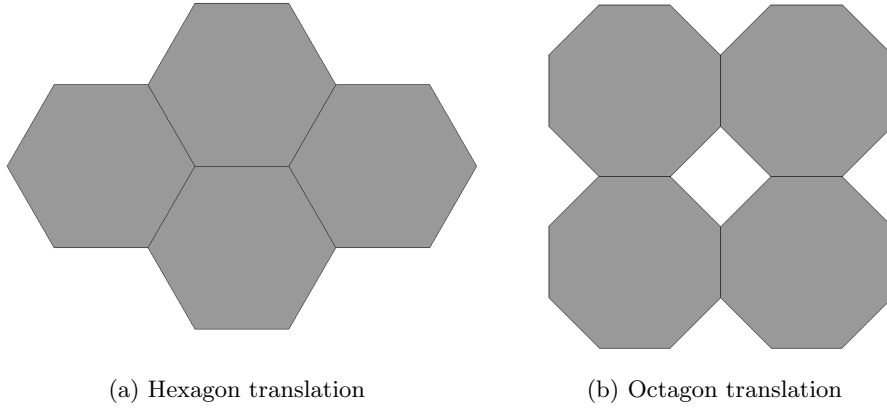


Figure 3.4: Demonstration of tessellation compliance in the hexagon and non-compliance in the octagon

### Identification and parametrization of Bravais lattices

Every configuration that complies with the principle of tessellation or translational symmetry contains an associated Bravais lattice and motif. The lattice refers to the shape of the unit cell obtained in the pattern, while the motif refers to the pattern that generates the distribution of material within that unit cell. The way to identify both parameters is based on the steps specified below and complemented with the figure 3.5:

1. Identify the points of the pattern that have the same state. In the case of the example figure, the tail of the pegasus is chosen and marked with green points.
2. Join these points by identifying the primitive vectors of the network.
3. Identify the shape of the mesh obtained by means of the Vornoi cells (cells obtained by generating the perpendicular in the center of union between two nodes of the pattern with the same state of the material and intersecting between them to obtain a closed shape) and the configuration of the material inside the cell.



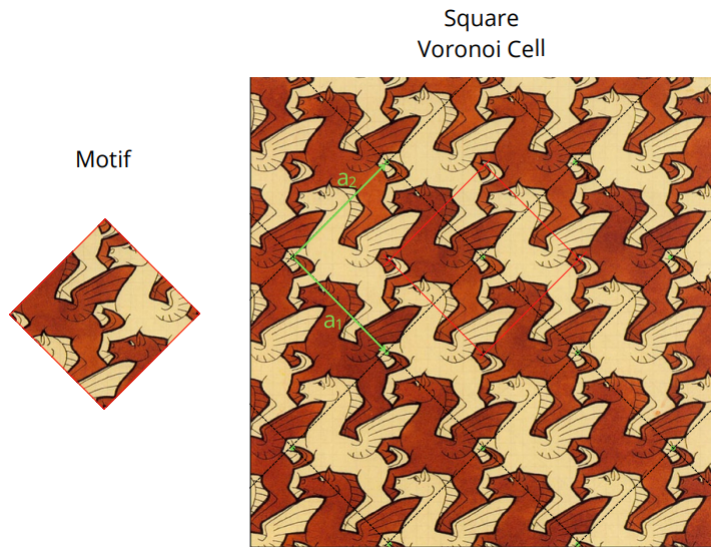


Figure 3.5: Determination of Bravais lattice, Voronoi cell and motif of the pegasus tessellation artistic wallpaper created by Escher.

As can be deduced from the figure above, the type of Bravais lattice depends on the relationship between the modulus of the primitive vectors and the angle formed between them:

$$\omega = \frac{\|\vec{a}_2\|}{\|\vec{a}_1\|} \tag{3.7}$$

$$\zeta = \arccos\left(\frac{\vec{a}_2 \cdot \vec{a}_1}{\|\vec{a}_2\| \cdot \|\vec{a}_1\|}\right) \tag{3.8}$$

Based on these relationships, the types of Bravais lattices and, consequently, of Voronoi cells can be classified in five groups: Hexagonal, Square, Rectangular primitive, Rectangular centered and Oblique. The following table depicts the characteristics and relation between parameters of each type of cell:

<p><b>Oblique</b></p> <p><math> a_1  \neq  a_2 </math> <math>\zeta &lt; \pi/2</math></p>	<p><b>Rectangular Centered</b></p> <p><math>2(a_2 \cdot a_1) =  a_1 ^2</math> <math>\zeta &lt; \pi/2</math></p>	<p><b>Square</b></p> <p><math> a_1  =  a_2 </math> <math>\zeta = \pi/2</math></p>	<p><b>Hexagonal</b></p> <p><math> a_1  =  a_2 </math> <math>\zeta = 3\pi/2</math></p>
	<p><b>Rectangular Primitive</b></p> <p><math> a_1  \neq  a_2 </math> <math>\zeta = \pi/2</math></p>		

Figure 3.6: Types of Bravais lattices in the plane [4].

### Bravais lattices' related point group

As specified in previous sections, translational symmetry characteristics combined with crystallographic constraint theorems limit the infinite geometric point groups to five. These are characterized by having  $n$  rotational symmetries, where  $n \in [2, 3, 4, 6]$  with one or two mirror line systems. Looking at the possibilities in the point groups and the available unit cell shapes, they are associated with each other according to the notation  $nmm$  (where  $n$  refers to rotation symmetries and  $mm$  to two mirror line systems) to form the lattice systems:

- Group 2: group associated with the oblique lattice, where the only possibility is double rotation symmetry.
- Group 2mm: group associated to the rectangular cells, both primitive and centered. In these there are two rotation symmetries and two systems of reflection lines separated by  $90^\circ$  (conventional lines and bisector).
- Group 4mm: group associated with square lattices. In this group there are four rotation symmetries and four systems of reflection lines (conventional and bisector).
- Group 6mm: point group associated with the hexagonal cell. In this group there are six rotation symmetries and six reflection line systems (conventional and bisector).

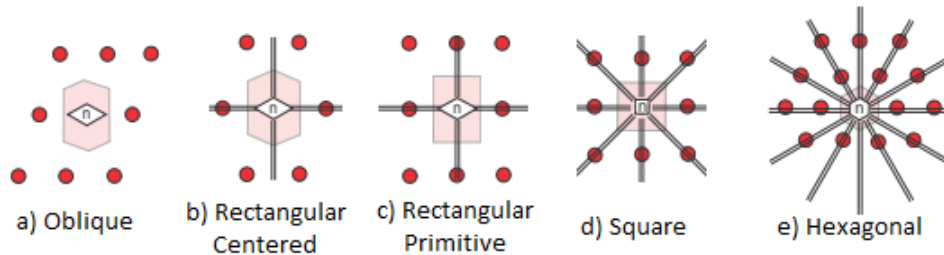


Figure 3.7: Bravais lattices with parallel lines indicating the mirror planes [4].

It should be noted that for the cases considered, the rotations of  $n$  around the axis coming out of the plane passing through the center of the cell coincide with the conventional mirror lines (the rest are created by bisector). Bravais grids are formed from the tessellation of a minimum unit of volume, called unit cell. The unit cells shown in the figures above correspond to a Voronoi cell, which preserves the same symmetries as a Bravais lattice and therefore maintains the point group.

### Plane groups

In this new section, instead of going into the symmetries of the lattices, as was previously done for the point groups, the present study focuses on analyzing the symmetries of the motif, therefore, the symmetries formed by the distribution of the two-phase material within the unit cell.

This analysis leads to the appearance of a new symmetry element in addition to rotation and reflection. This is the glide reflection, which is characterized by a reflection through a mirror line and then a translation parallel to the same line. It should be noted that by applying the inverse procedure (first translation and then reflection), the result remains invariant with respect to the regular procedure.

With this new symmetry element, all the components that make up the plane groups are now

available. The plane groups are defined as the set of symmetry elements that identify and characterize a mosaic, with a mosaic being understood as a symmetrical and periodic distribution of the two-phase composite mentioned since the beginning.

Therefore, as shown in the figure (3.5), a mosaic has an associated Bravais lattice and a motif that defines the plane group of the 2D domain considered. Again, as with the point groups, there is a restricted number of plane groups. Yegraf Fyodorov was the Russian crystallographer who in 1891 discovered that with only 17 plane groups it was possible to explain the symmetries of all possible mosaics. All these groups are specified in the figure shown below 3.9, where the Voronoi cells associated to the corresponding lattices with different distributions of material inside (motif) and the symmetry elements composing each group can be observed:

- Reflection lines indicated in black.
- Number of rotation symmetries indicated in the center of the polygon in red.
- Glide reflection lines and their translation direction indicated in blue.

Also note the notation used to name each of the groups. The letters p or c at the beginning of the notation of each group refer to whether the lattice of that plane group is primitive (only contains one internal node, summing the fraction of node of each of the corners) or centered (contains two internal nodes, one in the center and the other adding the node fraction of each corner). Next, the number of rotation symmetries appearing in the composite is indicated. Finally, in some cases, after the above, the terms referring to reflections appear. The first possibility is to have g or gg, indicating one or two glide reflections respectively. Secondly, there is the possibility of finding m or mm, indicating the presence of one or two mirror line systems.

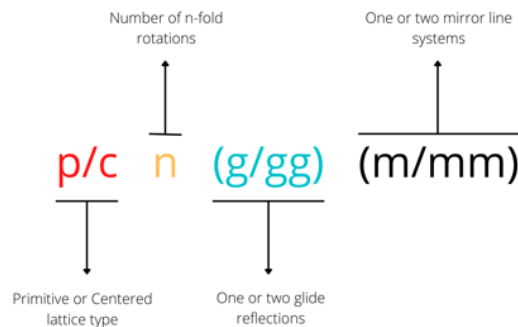


Figure 3.8: Explicative figure of the notation used for the plane groups.

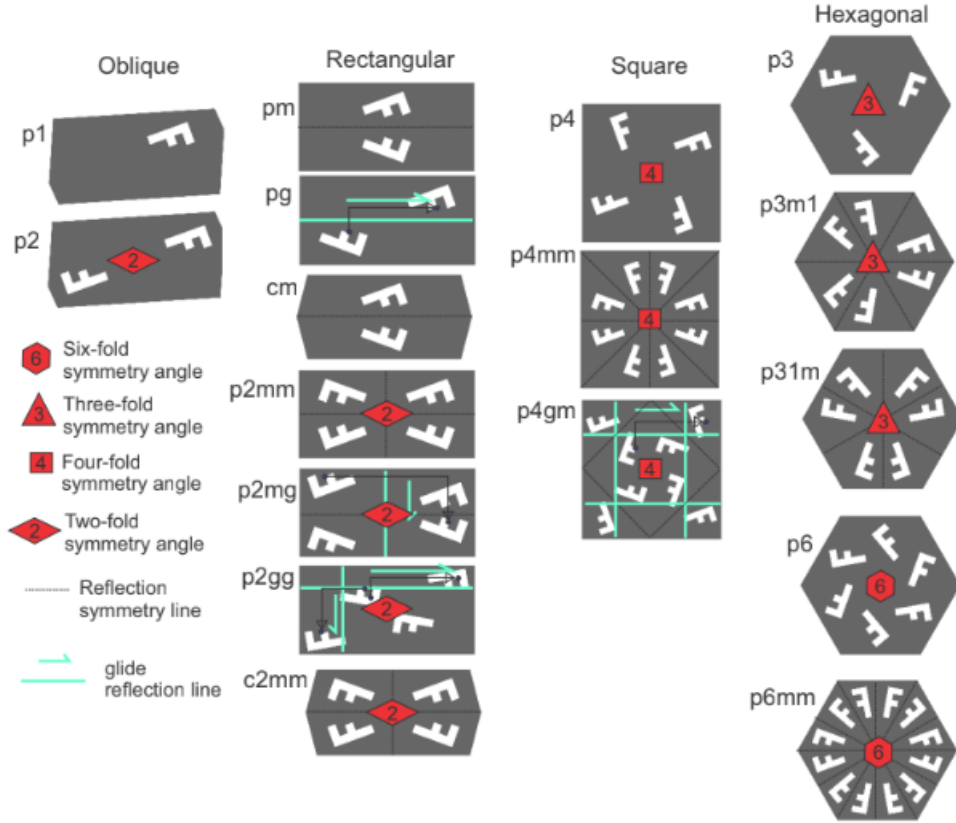


Figure 3.9: Voronoi cells for the seventeen point groups in 2D indicating the symmetry elements that compose each one. The colours inside the cell identify the phases of the composite [4].

It is likely that the point group symmetries of the Bravais lattice of the mosaic will be broken when the motif has a lower symmetry than that of its associated lattice. Therefore, the presented mosaic symmetry group of the mosaic is a subgroup of the one characterizing the underlying lattice. According to this property, the relations between the point groups and the 17 possible plane groups are therefore established. To characterize all the mosaics, ten point groups are necessary: 1, 2, m, 2mm, 4, 4mm, 3, 3mm, 6 and 6mm. These relationships can be seen in the table 3.1, where the plane groups are classified in the ten group points mentioned above according to the common symmetry elements between lattices and motifs. An example of this would be the point group 2mm, which is related to the plane groups p2mm and c2mm since they also contain two rotations and two mirror line systems. However, the relationships become more complicated when glide reflections are added, since the point groups do not include this element of symmetry. When analyzing the groups containing these symmetry elements, it is determined that the classification is done by substituting the g of the plane group to the m for point groups. For example, the plane group p2mg, is interpreted as p2mm, and is therefore classified in the point group 2mm. This is because performing a mirror symmetry of a crystal with glide symmetry yields the same crystal with a translation of half the unit cell size parallel to the line of displacement. Therefore, when considering an infinite crystal, both the original and the reflected and translated crystals are indistinguishable when considering the effective material properties.

## Crystal systems

Classification of the types of crystals by their point group symmetries. It should be noted that each crystal is formed by its associated lattice and its motif with the symmetry given by one of the plane groups with the associated point group, as presented in the previous section. That is why crystal systems are defined as the collection of crystals that share the same point groups with an identical set of compatible lattices. Using this criterion, four crystal systems are distinguished that allow classification of the ten point groups considered: oblique, rectangular, square and hexagonal. For example, crystals with point groups 4 and 4mm are both compatible with the same type of lattices, in this case, square lattices. Therefore, they belong to the square crystal system.

### 3.3.2 Type of tensors according to its symmetries

Considering the fact that the problem to be solved is of the linear elastic type, the objective of the tensor  $C$  is to relate the stress field with the deformation field, forming a matrix equation that generates the constitutive equations of the problem. Since the problem is linear elastic, these stress-strain relationships can be explained by Hooke's law. For one of the simplest cases, such as the uni-axial one, the constitutive equation would be the following:

$$\sigma = E \cdot \epsilon \quad (3.9)$$

where  $E$  is the elasticity modulus of the material.

The extension of a simple problem to a completely general one (three-dimensional solid) is established by means of a fourth order tensor  $C$ , whose components are the elastic coefficients which are constant given the material.

$$\sigma = C \cdot \epsilon \quad (3.10)$$

The assumption that the problem is linear elastic restricts the operating range of the material to its limit of proportionality, thus ensuring adequate results for small deformations.

Since the problem to be solved is of the 2D type and assuming that the elastic tensor is defined in a coordinate system aligned with the symmetry planes of the lattice point group (natural coordinates), the resolution of the problem is affected as follows:

- Simplification of the fourth-order tensor to a nine-component square symmetric matrix.
- Plane stresses and strains.
- Simplification of the tensor typology to four depending on the symmetry of the material distribution properties.
- Appearance of a new variable,  $\theta$ , which relates the canonical basis to the natural basis. It is also considered as a new elastic coefficient. The method to transfer an arbitrary tensor  $C$  from Cartesian coordinates to the natural coordinate basis is performed following the Auffray pattern [3], in which, by using the invariants of the elastic coefficients, the type of symmetry of the tensor, its appearance in the new basis and the  $\theta$  angle necessary to access it from its definition in the original basis are determined.

The symmetry in the elastic properties are properly classified according to the point group of the material [10]. Therefore, by applying isometric transformations compatible with each point group

on the 2D elastic tensor, the following types of tensors are derived, ordered from highest to lowest order of symmetry:

- **Isotropic O(2)**: type of tensor defined from elastic coefficients that is characterized by keeping the material properties identical in all directions of evaluation. This tensor is described below as a matrix in Voigt notation (use of engineering shear):

$$C = \begin{bmatrix} a_1 & a_2 & 0 \\ a_2 & a_1 & 0 \\ 0 & 0 & \frac{a_1 - a_2}{2} \end{bmatrix} \quad (3.11)$$

- **Tetragonal D4**: type of tensor with the need for four elastic coefficients to define it. The material properties are identical in the directions defining the natural base but vary when evaluating the shear. The tensor type is shown below:

$$C = \begin{bmatrix} a_1 & a_2 & 0 \\ a_2 & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \quad (3.12)$$

note that the fourth of the elastic coefficients corresponds to the rotation necessary to go from canonical axes to natural axes. This element is avoided in isotropic materials, since any basis is the natural basis because of tensor symmetries and properties.

- **Orthotropic D2**: tensor defined from five elastic elements. They contain two orthogonal axes of double rotational symmetry, which implies that their properties depend completely on the direction of evaluation. These tensors are of the following form:

$$C = \begin{bmatrix} a_1 & a_2 & 0 \\ a_2 & a_4 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \quad (3.13)$$

- **Fully anisotropic Z2**: tensor with the smallest number of symmetries that requires six elastic coefficients to be defined. This results in a material whose mechanical properties have no symmetry with respect to any plane or axis. The tensor has the following peculiarities:

$$C = \begin{bmatrix} a_1 & a_2 & a_5 \\ a_2 & a_4 & -a_5 \\ a_5 & -a_5 & a_3 \end{bmatrix} \quad (3.14)$$

### 3.3.3 Connection between geometrical and physical symmetry

Having presented all the characteristics of the physical and geometric domain of micro-architecture, the table below establishes the links between the two by means of point group symmetries:


















Elastic Symmetry class	Crystal system	Point group	Plane group	Compatible Voronoi cell					
				Oblique	Rectangular Primitive	Rectangular Centered	Square	Hexagonal	
Z2	Oblique	1	p1						
		2	p2						
D2	Rectangular	m	p1						
			pg						
			cm						
		2mm	p2mm						
			p2mg						
			p2gg						
			c2mm						
D4	Square	4	p4						
		4mm	p4mm						
			p4gm						
O(2)	Hexagonal	3	p3						
		3m	p3m1						
			p31m						
		6	p6						
		6mm	p6mm						

Table 3.1: Connection between the elastic symmetries (column 1), the crystal systems (column 2), the lattice point groups (column 3), the motif plane groups (column 4) and the compatible Voronoi cells (column 5) that can accomplish the characteristics displayed in previous columns.

It follows that to achieve the elastic tensor characteristics of the first column, one of the Voronoi cells shown in the last column, combined with one of the plane group symmetries indicated in the same row for the motif is required. For example, if a micro-architecture is desired to be obtained whose elastic tensor is of the D4 type, the squared Voronoi cell will be necessary jointly with a plane group symmetry of type p4, p4mm or p4gm for the internal distribution of the two phases of the composite.

### 3.3.4 Neumann's principle

Fundamental postulate of crystal physics that allows establishing the connection between the symmetry elements of the micro-architecture geometry with the symmetries appearing in the elastic constitutive tensor. Neumann's principle states the following: "the symmetry elements of any physical property (optical, magnetic, thermal and mechanical properties) of a crystal must include the symmetry elements of the crystal's point groups" [11]. This principle supports and explains the relationships obtained in previous sections, and is also demonstrated by the examples given in the practical section.

However, it should be noted that this principle does not state that the symmetry elements of a physical property must be the same as those corresponding to the point group of the crystal. In fact, sometimes physical properties may possess a symmetry greater than that indicated by the crystal

point group. Therefore, according to Neumann, the table 3.1 should be interpreted as follows: a unit cell with a given shape and a given internal symmetry ensures a symmetry equal to or higher than that of the tensor type specified in the corresponding row. For example, a square cell with a motif with plane group  $p2mm$ , corresponding to point group  $2mm$ , ensures at least a symmetry tensor  $D2$  (orthotropic). This means that it could also be the case to obtain a symmetry tensor  $O2$  (isotropy).



## Chapter 4

# Code performance and composition

Regarding the available possibilities of shapes in 2D unit cells reviewed in previous chapter, the objective is to develop a compact code which allows the user to create the desired mesh (between the Bravais lattices options) introducing a set of common parameters.

### 4.1 Variables to be defined by the user

The first step consists in analyzing which are the required variables to introduce by the user to perform each of the polygons. Taking into account the possible shapes of the Bravais lattices, it is observed that there is only the possibility of having polygons with 4 or 6 sides.

On the one hand, the analysis of 4-sided polygons includes:

- **Squares:** quadrilateral polygon with equal length sides ( $a_0$ ) and angles (as all the angles are equal, each one must have  $90^\circ$ ). That's the reason why the user only needs to define the length of the side.

$$c = a_0 \quad \theta = 90 \quad (4.1)$$

- **Rectangles:** quadrilateral polygon with two different lengths to define ( $a_0$  and  $a_1$ ) and the same  $90^\circ$  angle in the intersection between sides.

$$c = [a_0, a_1] \quad \theta = 90 \quad (4.2)$$

- **Rhomboids:** quadrilateral polygon with the parallel sides of equal length, which translates into two variables of length ( $a_0$  and  $a_1$ ) and an angle between the horizontal and the side following the base in the counterclockwise direction ( $\theta_1$ ).

$$c = [a_0, a_1] \quad \theta = \theta_1 \quad (4.3)$$

Taking into account the elemental definitions, it is clear to see that the rhomboid is the most general case, since by assigning the appropriate input parameters it would be possible to have a rectangle or a square. Hence, the inputs for the program to generate a four-sided polygon will be a vector of lengths ( $\vec{c}$ ) with two components and a vector of angles ( $\vec{\theta}$ ) composed of an initial angle to define the inclination of the lower side of the polygon and a second angle that will define the inclination

between the horizontal and the second side.

$$\vec{c} = [a_0, a_1] \quad \vec{\theta} = [\theta_0, \theta_1] \quad (4.4)$$

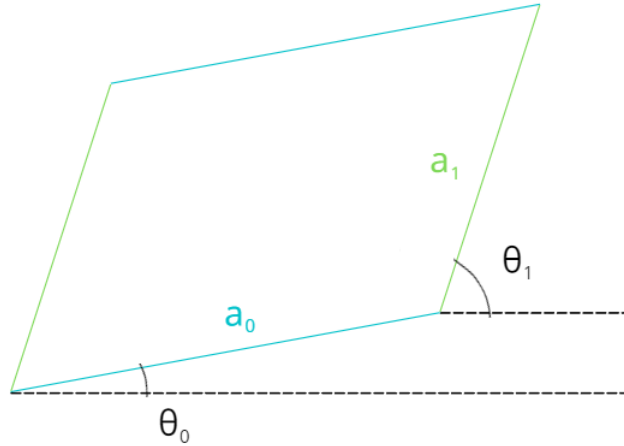


Figure 4.1: Representation of the required inputs to generate a 4-sided polygon.

On the other hand, the analysis of the 6-sided polygons includes:

- **Regular Hexagons:** 6-sided polygon with all the sides with the same length and an angle of  $60^\circ$  between sides. This causes the only possible input to be the length of the sides.

$$c = a_0 \quad \theta = 60 \quad (4.5)$$

- **Irregular Hexagons:** 6-sided polygon, in this particular case, with equal length in parallel sides, which translates in three side-lengths to define ( $a_0$ ,  $a_1$  and  $a_2$ ) and two angles  $\theta_1$  (angle between the horizontal and the side next to the base in the counterclockwise direction) and  $\theta_2$  (angle between the horizontal and the next side, corresponding with the length  $a_2$ ).

$$c = [a_0, a_1, a_2] \quad \theta = [\theta_1, \theta_2] \quad (4.6)$$

In this case, the most particular case is the irregular hexagon, as defining properly the input variables, the user can generate a regular hexagon. In the same way as in four-sided polygons, a vector of lengths ( $\vec{c}$ ) and another one of angles ( $\vec{\theta}$ ) will be defined, but in this case, each vector will have three components.

$$\vec{c} = [a_0, a_1, a_2] \quad \vec{\theta} = [\theta_0, \theta_1, \theta_2] \quad (4.7)$$

As said before,  $\theta_0$  is the angle between the lower side and the horizontal.

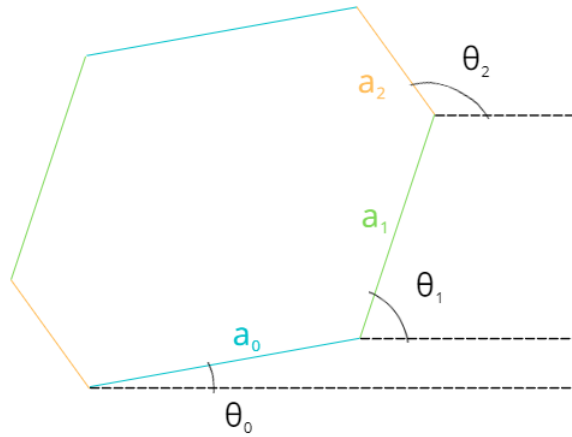


Figure 4.2: Representation of the required inputs to generate a 6-sided polygon.

With this, the definition of the contour of the polygon has been done. But it is also necessary for the user to define the density of the mesh ( $r = \text{divUnit}$ ). Hence, another input of type integer must be introduced. This value will define the number of equal divisions per unitary length of the side. Taking this into account, every side will contain two nodes, one in each vertex and the following nodes between them:

$$n^{int} = r \cdot c_i - 1 \quad (4.8)$$

where  $i$  refers to the desired position in the vector  $\vec{c}$ .

Once the analysis is done, the variables to be introduced by the user are:  $\vec{c}$ ,  $\vec{\theta}$  and  $\text{divUnit}$ .

As a guide of use of the code, the following table shows which are the input variables to introduce to obtain each one of the shapes considered and some limitations to take into account with the angles selected (setting  $\theta_0$  to 0 in all cases):

Cell shape	$\vec{c}$	$\vec{\theta}$	$\text{divUnit}$	Remarks
Square	[a,a]	[0,90]	2	-
Rectangle	[b,a]	[0,90]	2	-
Rhomboid	[b,a]	[0, $\alpha$ ]	2	$\alpha \neq 0, 180$
Regular Hexagon	[a,a,a]	[0,60,120]	2	-
Irregular Hexagon	[a,b,c]	[0, $\alpha,\beta$ ]	2	$\alpha \neq 0, 180$ $\alpha < \beta < \alpha + 180$

Table 4.1: Input data required to obtain the desired cell shape and divisions.

As an example:  $a = 1$ ,  $b = 2$ ,  $c = 1$ ,  $\alpha = 60^\circ$  and  $\beta = 100^\circ$ .

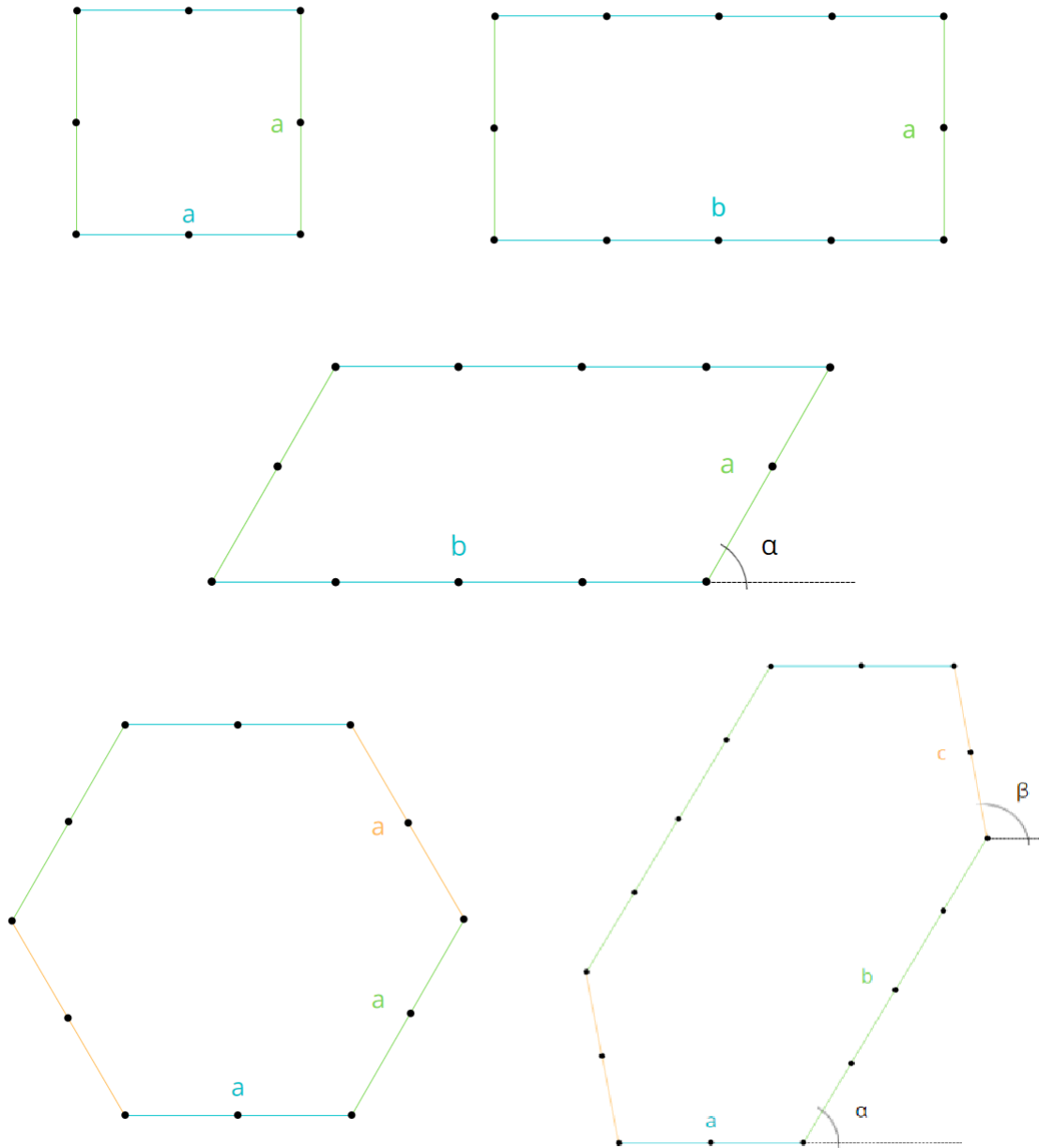


Figure 4.3: Boundary obtained by the mesh creator for the five inputs specified in 4.1

## 4.2 Boundary nodes disposition mechanism

Looking at the study done in the last section it is clear to see that the initial data introduced by the user, defines the location of the boundary nodes. In this section, the study is centered in the mechanism used by the MATLAB code to set the boundary nodes in the position specified by the user with the inputs. For this case, the mechanism is general for 4-sided and 6-sided polygons, that's why the explanation will use the most general situation, the irregular hexagon. Note that, all the shapes exposed in 4.3 comply with the tessellation principle, so that they have at least equal sides and angles two by two (causing the spatial translation of these, making them fit into a

pre-established space in 2D without leaving gaps and maintaining continuity). This makes that the inputs introduced by the user, define directly the master nodes (nodes of the first half of the polygon) and indirectly the slave nodes (nodes of the second half of the polygon that have the same succession of lengths as the first half, but supplementary angles with respect to the horizontal). To make it easier, the code differentiates between vertex nodes and interior boundary nodes in the computation of the coordinates:

- **Vertex Nodes:** first coordinates set by the code defining the shape of the selected polygon. On the one hand, the equation used to calculate the coordinates of the master nodes is as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = c_0 + c_i \cdot \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} \quad (4.9)$$

where  $c_0$  are the coordinates of the previous node (the first vertex is always set at the origin of coordinates),  $c_i$  is the vector of lengths for the succession of master sides and  $\theta_i$  is the vector of angles with respect to the horizontal axis for the master sides and  $i$  is a parameter that selects the successive positions in the input vectors from 1 to  $\frac{sides}{2}$ .

On the other hand, the equation used to compute the coordinates of the slave nodes is as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = c_0 + c_i \cdot \begin{pmatrix} \cos(\theta_i + 180) \\ \sin(\theta_i + 180) \end{pmatrix} \quad (4.10)$$

where, the only difference with respect to 4.9 is that to complete the polygon shape, rotate the angles defined for the master sides (user-defined) by 180 degrees.

The following figure is shown as an example (as in last examples, it has been considered that  $\theta_0 = 0$ ):

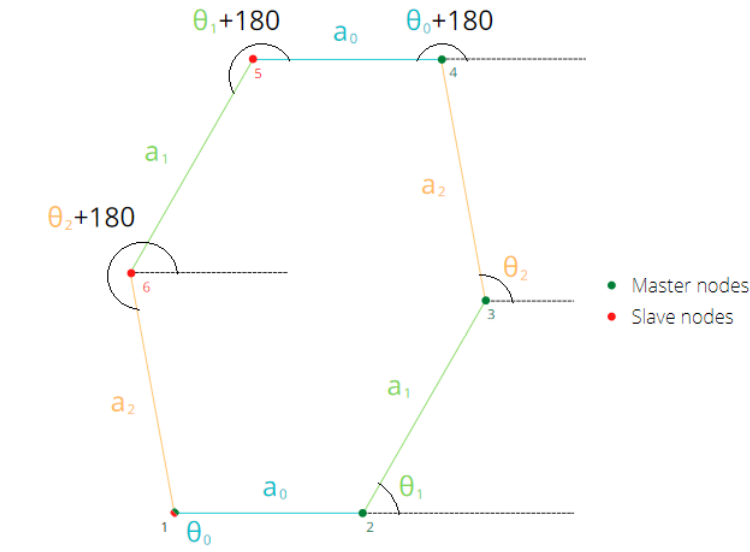


Figure 4.4: Representation of the data used for the calculation of the coordinates of the vertex nodes. Note that the numeration of the nodes is also shown.

The figure clearly shows the initial node (red/green), the pure master vertices (green) and the

pure slave vertices (red). The first node is considered a slave even though it is defined during the master calculation phase, since the slave calculation process retrieves it again to verify that all calculations performed so far are correct.

- **Interior Boundary Nodes:** nodes located between consecutive nodes. As said before, the number of divisions of every side is directly related with the length of the side and the unitary divisions considered by the user ( $divUnit$ ) as equation 4.8 shows. The equation used to obtain the coordinates of the internal master nodes is as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = c_0 + \frac{c_i}{div_i} \cdot \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} \quad (4.11)$$

where  $div_i = c_i \cdot divUnit$  is the number of divisions corresponding to the  $i$  side.

At a glance, it can be seen that the distance between two consecutive vertices ( $c_i$ ) is divided by the number of divisions of the side ( $div_i$ ) to generate equidistance between nodes. In this case,  $c_0$  will be the first vertex of the side (the smallest in terms of numbering) if the coordinates of the first internal node of each side are calculated and if not, its value will be that of the last node calculated.

The calculation for the internal slave nodes is identical to that of the equation 4.11 but in this case, using the supplementary angle to the one introduced in the vector  $\vec{\theta}$ .

As an example of the performance of the code to this point, the following indicative figure is provided:

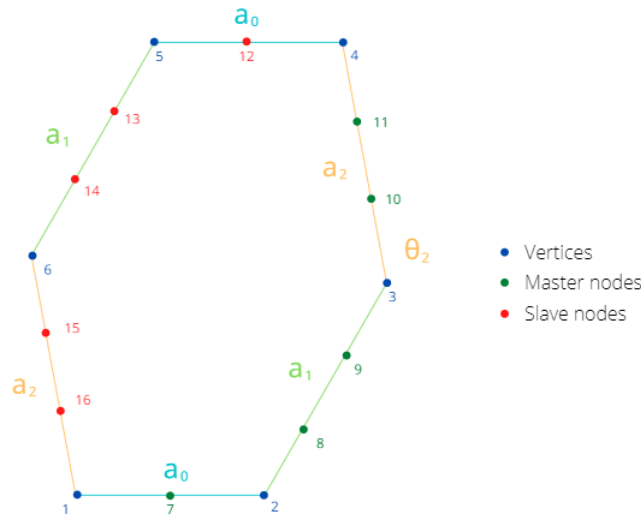


Figure 4.5: Boundary interior master and slave nodes in the six-sided polygon of figure 4.4. Note that the numeration of the nodes is also shown.

### 4.3 Interior nodes disposition mechanisms

Once the nodes that make up the boundary have been generated, it is time to calculate the internal nodes. Due to the peculiarities of each polygon, two methods are presented, where the first one is used for four-sided polygons and the second one for six-sided polygons.

- **Computation of coordinates by intersections:** A way to calculate the coordinates of the nodes by intersecting lines parallel to the sides for four-sided polygons. This method is based on the following steps:
  - 1. Obtaining the vectors of contiguous sides ( $v_1$  and  $v_2$ ). It should be noted that these vectors will allow to obtain all the interior nodes of the mesh.
  - 2. Obtaining the points to generate the first two lines to intersect ( $p_1$  and  $p_2$ ). We start with the points closest to the axis of coordinates of the contiguous sides (without including the vertices).
  - 3. Generation of the equation of both of the lines. Note that the first line is generated with the point  $p_1$  and the vector  $v_2$  of the adjoining side. For the second line in this case, the point  $p_2$  and the adjoining side vector  $v_1$  are used.
  - 4. Obtaining the coordinates of the node by the intersection between the two lines.
  - 5. Move to the next point on the first side. Note that the second line remains invariant until all the nodes of the contour of the first side are completed. When this point is reached, we move to the next node of the second side.

The following image clarifies the explanations made:

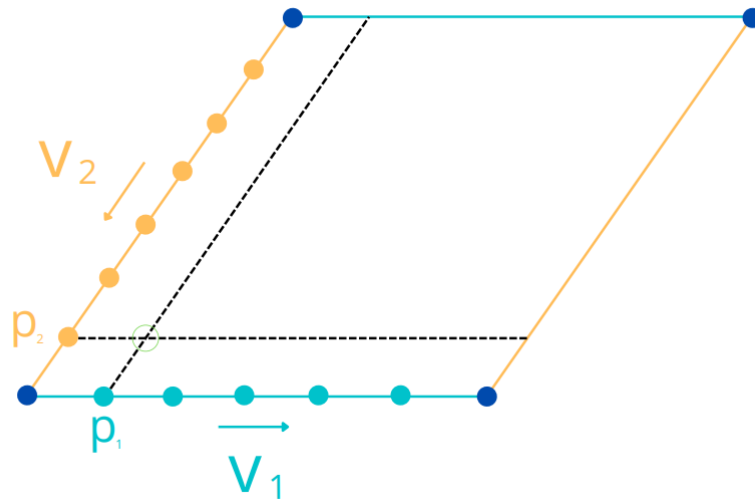


Figure 4.6: Procedure followed by the intersection mesher. The figure shows the original polygon and the first iteration of the intersection mesher.

- **Computation of coordinates by division of diagonals:** Second method to compute the coordinates of the interior nodes of 6-sided meshes by the division of the diagonals and reduction of the contour scaling factor. The method is based on the following steps:
  - 1. Computation of the coordinates of the node on the center of the 6-sided polygon by the intersection of two of the three diagonals.
  - 2. Generation of the the six vectors that connect every vertex with the central node (diagonals).

- 3. Division of the vectors of the diagonals according to maximum value of  $div$  and computation of the new vertices of the scaled polygon.
- 4. Connection of the consecutive nodes of the scaled polygon and meshing of each line taking into account to the following rule for the number of divisions when reducing the length of the sides when scaling: *the length distance between two consecutive nodes in the interior of the polygon should not be higher than the distance between two nodes in the boundary*. When this rule is not followed, the algorithm reiterates with a higher number of divisions until it is fulfilled. Highlight that after every scaling iteration, the number of divisions of every side is reduced by one.
- 5. Repetition from step 2, taking as reference the vertices of the new scaled polygon. This repetition is made until the last division of the diagonals.

The following figure makes it easier to understand the above steps:

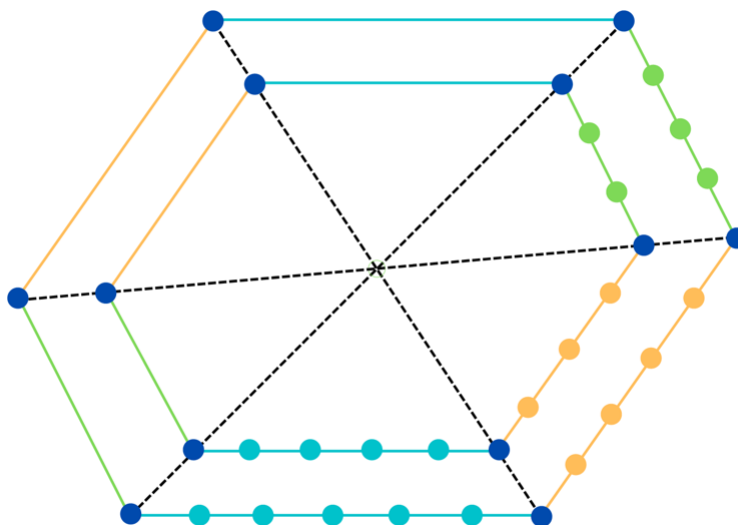


Figure 4.7: Procedure followed by the division of diagonals mesher. The figure shows the real polygon and the first scale with the nodes of the master sides.

Note that this internal node meshing strategy is fully functional for any even-sided polygon with opposite sides of equal length.

#### 4.4 Creation of elements: connections between nodes

Once all the necessary nodes for the mesh have been created, the algorithm is in charge of generating triangular unions formed by three nodes each, thus obtaining the elements. The connection between the nodes is made by proximity using the MATLAB precomputed function *delaunay*. This allows to make the triangular connections quickly and effectively, but, on the other hand, it requires certain filters to avoid the following problems:

- Generation of linear elements in the contour in those polygons with non-straight sides. The trigonometric functions used when placing the contour nodes cause the nodes to not lie exactly on the side line, causing the *delaunay* function to detect these differences and generate an



erroneous element. This is solved by requiring the algorithm to prevent the generation of an element if it contains three contour nodes.

- Disappearance of the corners of four-sided polygons. The above solution is perfect for six-sided polygons, but in four-sided polygons, where the corner elements consist of three nodes, it is insufficient. Therefore, the code implements a new filter around the previous solution in which it is required that elements are not removed when the node numbering sequence corresponds to a corner in a four-sided polygon.

## 4.5 Implementation of boundary conditions

Every finite element problem requires boundary conditions, and this is no exception. As discussed in the theoretical section, the microstructures obtained must be able to comply with the tessellation property. This is achieved by forcing the nodes on opposite sides to have the same topology during optimization and therefore, it is necessary to provide the optimizing algorithm with master-slave nodes pairs. On the other hand, it is also necessary to establish Dirichlet boundary conditions, which for this particular case, are based on fixing the vertices of the polygons.

The code implements these conditions as follows:

- **Obtention of the master-slave nodes:** type of boundary condition where the master node gets a condition and the slave copies it. The topology optimization code needs the list of master-slave nodes of every polygon created so as to ensure the tessellation property. The steps followed by the algorithm to obtain the master-slave nodes are as follows:
  - 1. The first step is to obtain the normal vectors of each of the sides of the polygon, trying to make them come out to the area created by them. To obtain this vector, what the code does is to obtain the director vector of each of the sides that make up the polygon, and then normalize and orthogonalize them.
  - 2. Obtaining all those vectors that have as origin one of the vertices and destination all the nodes that form the contour. The destination node of these vectors will be the possible candidates to be master-slave.
  - 3. The next step consists in revising which of the vector candidates are orthogonal to the normal vectors of the sides forming the polygon. In this step, the code saves in an ordered way (remember that the contour numbering is done counterclockwise for the interior nodes) the nodes of each of the sides that comply with the orthogonality in a matrix that has as much rows as master-slave nodes and as much columns as sides has the polygon. In this way, the master-slave nodes on each side are arranged in column vectors within the master-slave matrix.
  - 4. Now, it is necessary to know the parity between the sides, that is, to determine which sides should be related to each other, in order to copy the conditions of the other. These relations are established two by two following the rule that the orthogonal vectors of their sides are parallel. In this way, the code groups in pairs the lines that form the polygon by checking the parallelism between orthogonal vectors.
  - 5. The code groups the previously obtained nodes according to the parity of the sides.

The first half of the lines will act as masters (considering the first lines, those with a lower number of nodes than the rest) and the second half will act as slaves. The result of the imposition of these boundary conditions is a matrix where the numbering of the master node is related to that of the slave, taking into account that for the translation symmetry to be fulfilled, it is necessary to invert the order of numbering of the column vectors corresponding to slave sides.

A simple example is shown below to complete the explanations:

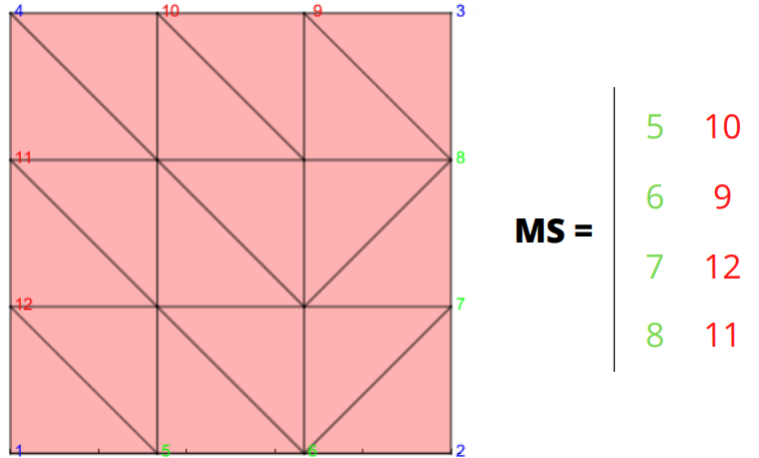


Figure 4.8: Numbered square mesh with the master nodes in green and the slave nodes in red together with the master-slave matrix.

It can be seen that in the master-slave matrix, the master nodes are ordered in the first column, while the slave nodes on each side are numbered inverted in order to guarantee translation symmetry.

Note that this methodology is completely general for any polygon with even sides where the opposite ones have equal length.

- **Application of the prescribed displacements (Dirichlet boundary conditions):** as discussed above, the code is responsible for constraining the movement of the two degrees of freedom of the corner nodes automatically. This is done by specifying the numbering of the nodes to be constrained, their degrees of freedom and the prescribed value (in this case null), during the creation of the .m file (file generated to subsequently submit the mesh to the topological optimization phase).

## 4.6 Organization of the code: object oriented programming & UML diagram

One of the characteristics that differentiate the code presented in the previous sections is that it is organized following an object oriented pattern of programming instead of patterns centered in functions or logic. This is implemented using classes instead of functions, where the object is defined as a data field that has certain exclusive attributes and methods. This way of programming, despite having greater programmatic and logical complexity, has several benefits, among which are

efficiency, reusability of methods and scalability (ease of increasing the algorithm's capabilities). To explain the organization of the classes that make up the Mesh Creator code, the Unified Modeling Language (UML) diagram is used to show the attributes of the objects, the methods of each of the classes and the relationships established between them. The diagram can be seen in figure 4.9, where only the attributes and public functions are reflected, that is, those that are accessible by the user. Each box in the diagram represents a class, whose name is reflected at the top of the box. Next, the attributes of the object of the class appear and in the lower subdivision of the box, the exclusive methods of the class appear. As for the relationships between classes, two can be found in the diagram:

- **Composition:** relationship of association in which the parts exist only in association with the compound. This relationship is represented by a black arrow.
- **Inheritance:** relationship between a general superclass and its more specific subclasses. These subclasses acquire all the properties of the superclass, but not all the properties of the subclasses are contained in the superclass. This relationship is represented by a white arrow.

Once the casuistry of the UML diagram is known, we proceed to explain the work developed by each of the classes from the top to the bottom, in order of class execution:

- *MeshCreator*: Brain class, from which most of the code functionalities are executed. From it you can obtain the coordinates and connectivities of the mesh, draw the mesh and create the .m file.
- *NodesCalculator*: Class in charge of calculating the number of nodes necessary to initialize the coordinate variables of the vertex nodes, of the contour and of the whole mesh.
- *NodesCalculatorFactory*: Class that selects the subclass required for the calculation of the number of nodes based on the number of sides of the polygon selected by the user.
- *QuadrilateralNodesCalculator*: Class in charge of calculating the totality of the nodes of the quadrilateral mesh.
- *HexagonalNodesCalculator*: Class in charge of calculating all the nodes of the six-sided meshes.
- *NodeCoordinatesComputer*: Class in charge of calculating the coordinates of all the nodes of the mesh. By composition, there are three classes in charge of calculating the coordinates of the vertices (*VertexCoordinatesCalculator*), of the contour (*BoundaryCoordinatesCalculator*) and of the whole mesh (*TotalCoordinatesCalculator*).
- *TotalCoordinatesCalculatorFactory*: Due to the peculiarities of four-sided and six-sided meshing, this class takes care of directing the superclass to the corresponding subclass.
- *IntersectionCoordComputer*: Meshing class of internal nodes of four-sided polygons by the method of intersections.
- *DiagonalCoordComputer*: Meshing class of the internal nodes of six-sided polygons by the method of diagonal division and scaling.
- *NodesConnector*: Class in charge of establishing the connections between the nodes to form the triangular elements.

- *MasterSlaveComputer*: Class in charge of calculating the master-slave nodes.
- *MatlabFileWriter*: Class in charge of setting the Dirichlet boundary conditions and writing to the .m document.

In addition, although they do not appear in the UML diagram, three additional functions that facilitate the user's work with the algorithm are presented below:

- *createShapedMeshes*: Matlab function that allows the user to create the mesh with the desired shape by entering the length of the first half of the sides of the polygon, the angles with respect to the horizontal of the first half of the vertices, the number of divisions for each unit of side length and the desired name for the .m file created by the algorithm.
- *testQuadrilateralSolutions*: function in charge of verifying that the classes through which a four-sided polygon passes, work correctly. By default, this test checks that the code works correctly when working with a square of side length 1 and 3 divisions per side. This allows the programmer to verify the correct functioning of the code when making small modifications, as well as to quickly find possible faults by detecting in which class they occur.
- *testHexagonalSolutions*: test function that does the same job as the previous function but for six-sided polygons. In this case, by default, it checks the performance of the code for a hexagon with side length 1 and 3 divisions per side.

Finally, the GitHub link to view the MATLAB code is provided [here](#).

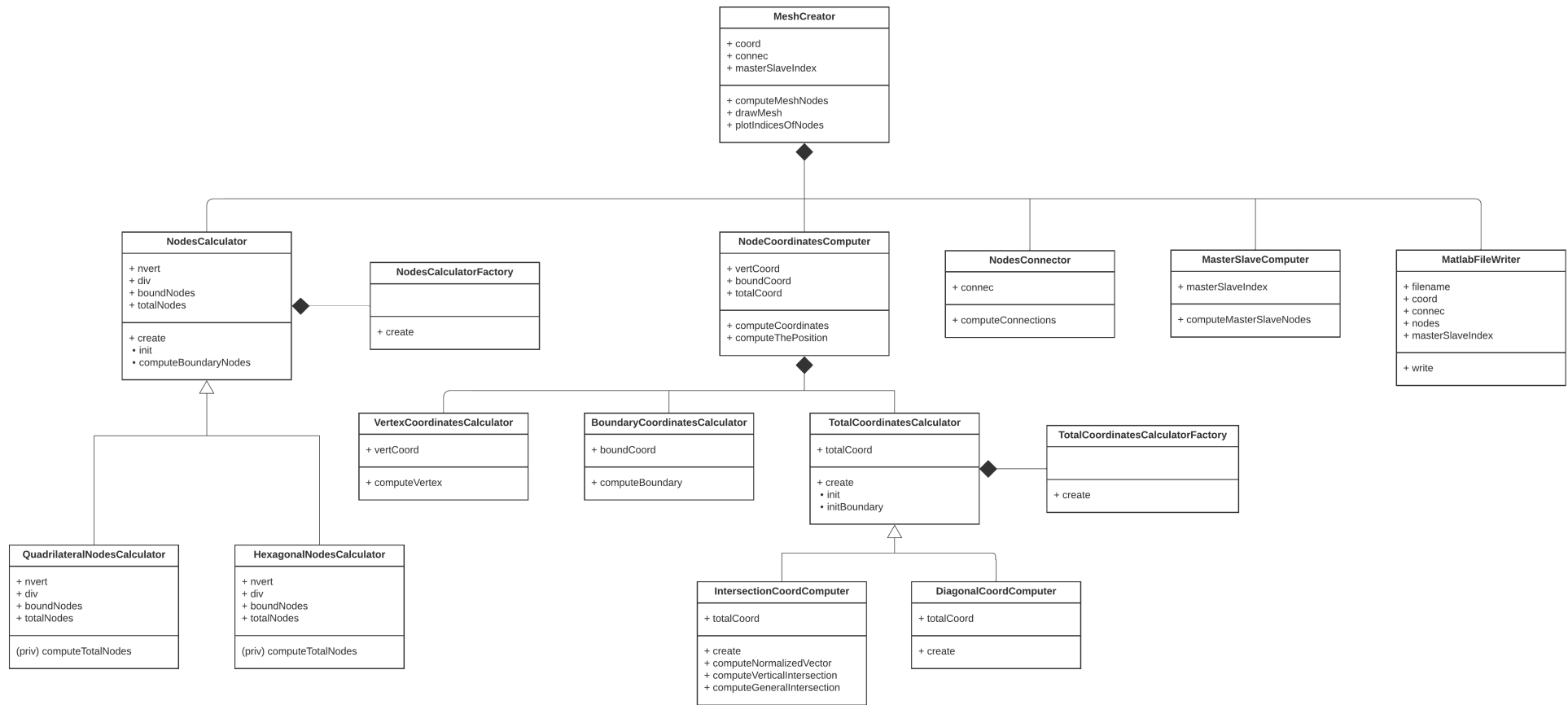


Figure 4.9: UML diagram of the Mesh Creator code.

### 4.7 Performance verification

All code needs a series of verifications before starting its use, in order to try to detect possible problems or cases not taken into account during the code development process. In this way, several checks are performed with different polygons of both four and six sides, observing the obtained parameters and those that can be expected taking into account the input parameters. The following are the examples verified using  $divUnit = 3$ , in which the numbering of the vertices is shown in blue and the numbering of the contour nodes in green or red depending on whether they are master or slave nodes:

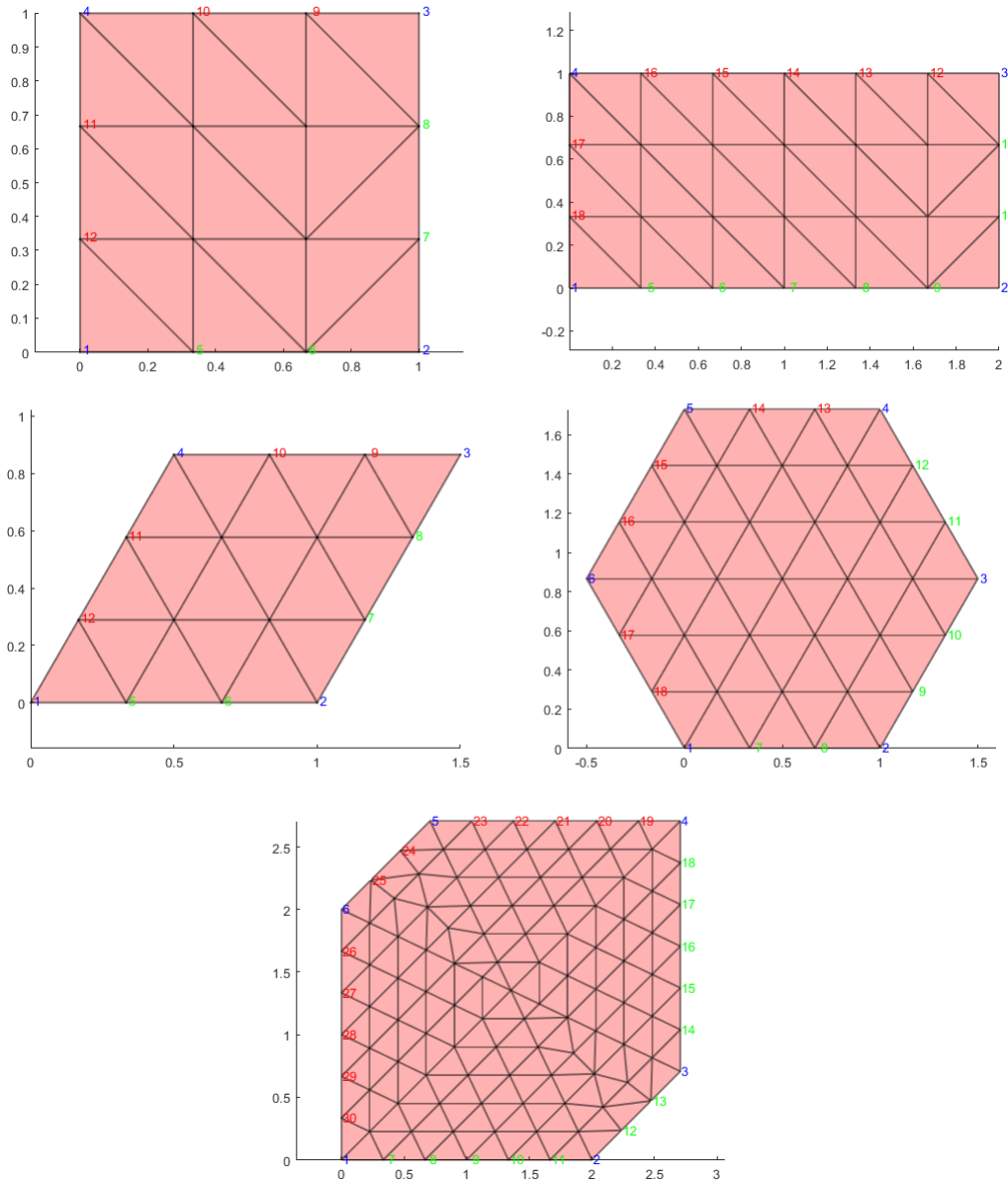


Figure 4.10: Performance of the code using different inputs.

## Chapter 5

# Micro-structures design methodology

Any design of material goes through a process in which a solution is proposed, its performance is interpreted and it is constantly iterated until the shape or distribution that fulfills the required function is achieved. As mentioned in previous sections, topological optimization eliminates this material testing time since the material is designed knowing the desired physical characteristics (inverse problem). The steps to obtain a topology with the desired properties by the designer are presented below as a guide:

1. Selection or elaboration of the elastic tensor according to the properties desired for the material. Of course, in this step it is advisable to transform the tensor to natural coordinates using the algorithm that implements the method of Auffray invariants, both to observe the tensor symmetries and to filter out those decimals that make it difficult to distinguish between the tensors of the section 3.3.2. The tensor obtained in the natural coordinates will be the target tensor  $C^*$  to be forced into the micro-structure by the optimizer.
2. Selection of the Voronoi cell shape with compatibility with the chosen tensor type, taking into account the relationships in the table 3.1.
3. Choice of the initial descriptive parameters of the cell: length of the sides, angles (if required) and initial topology. Of course, the selection of these parameters is made based on experience with numerous designs. That is why, in the practical section, a deeper analysis of the optimizer behavior with the variation of these preprocessing parameters is performed.
4. Use of the cell generator code to generate the desired shape with the conditions established in the previous point. At this point, a new parameter appears which is the number of divisions per side that make up the mesh of the cell. This selection can be made according to the convergence analysis and error evaluation carried out in the practical part.
5. Introduction of the conditions established with the mesh creator in the topological optimizer to proceed to solve the inverse problem. It should be noted that the creation and optimization of the code is performed in Cartesian coordinates.
6. Analysis of the symmetries obtained. Review of the distribution of material obtained in the cell and classification of the type of plane group. Ideally, the algorithm, as a post-processing, should be able to force one of the plane symmetries that ensure at least the desired tensor for the selected Voronoi cell. Without forcing these symmetries, the optimizer should be able

to achieve them by selecting a Voronoi cell related to the peculiarities of the tensor to be forced. Of course, as discussed in the theoretical section by means of Neumann's theorem, if an optimized Voronoi cell contains a plane symmetry that allows obtaining a tensor of type D4, this only ensures that the topology will have at least that symmetry, and may even reach isotropy if the appropriate symmetry conditions are met.

7. Transfer of the final tensor obtained from Cartesian coordinates to natural coordinates by means of the Auffray algorithm.
8. Culmination of the design of the desired micro-structure.



## Chapter 6

# Practical assessments

Once all the necessary knowledge has been compiled and all the essential concepts related to topological optimization and the meshing algorithm have been reviewed, it is time to test the theory reviewed and test the capability of both the optimizer and the mesher.

For this purpose, this practical part is organized in four essential points:

- Convergence study with mesh refinement. Observation of the ability of the meshing code to tend to reduce the error of the tensor obtained by increasing the number of divisions per side (divUnit). The study aims to provide the user with a tool to decide the number of elements of his mesh according to the rigor required by his results in contrast to the computational cost.
- Demonstration of Neumann's theorem and canonical deformations. Demonstration of the relationship between the geometrical and the physical domain by means of several examples. In addition, it is shown the fulfillment of the tessellation principle for each of the cells in their deformations under horizontal, vertical and shear stress states.
- Design of isotropic and orthotropic materials. Point destined to the testing of different Voronoi cells and initial conditions to evaluate the ability of the optimizer to generate the necessary plane symmetries to obtain the desired tensor.
- Study of the peculiarities of the non-Voronoi rhomboid cell. Review of the tessellation principle in principal deformations and the relation of its tessellated domain with that of a hexagon.

The objective of this practical study is to evaluate the behavior of the optimizer by varying the preprocessing conditions related to the cell shape, the initial optimization conditions and the number of mesh divisions to obtain the objective tensor. Therefore, it seeks to facilitate the designer's task by providing the necessary tools for the selection of initial conditions to ensure that its micro-architecture has the properties of the objective tensor.

### 6.1 Convergence of the results upon the increase of divisions

Basic point of any finite element study in which it is sought to assess the number of divisions required for each type of mesh to obtain rigor in the results without having an excessive computational cost. The first thing to note is the title of the section, where the increase in the number of divisions is mentioned, as if it were the application of the finite element method in 1D, instead of the number

of elements. This is because the user control variable is the number of divisions per unit side length ( $\text{divUnit}$ ) instead of the number of elements. In addition, assessing a priori the number of elements in six-sided meshes is very complex given the functionality of the algorithm in these meshes to densify the mesh if necessary.

To reflect the two phases of the composite, a hole has been introduced in the center of each of the meshes and the dimensions of the sides have been adjusted to ensure that the volume fraction of the composite is identical in all meshes and of value  $\frac{\pi}{4}$ .

The convergence with increasing number of divisions has been evaluated from the relative error between the determinant of the calculated tensor with respect to the real tensor:

$$\epsilon_{rel}(\%) = \frac{\|C_{calc} - C_{real}\|}{\|C_{real}\|} \cdot 100 \quad (6.1)$$

where  $C_{calc}$  is the calculated tensor for a given number of divisions and  $C_{real}$  is the "correct" or "zero-error" tensor. In this case, the tensors obtained for meshes of 300 divisions per unit side length and filtered by the symmetric trend tolerances of the Auffray algorithm have been assumed to be correct. This is the reason why correct and zero-error appear in quotation marks.

The procedure followed was based on evaluating the relative error of tensors generated from 25, 50, 100, 175 and 300 (without Auffray filter)  $\text{divUnit}$  meshes to fit an exponential curve passing through these points in a relative error vs number of divisions per unit side length plot. In this way, the exponential trend curve allows to represent the relative error of all points of interest for medium and large  $\text{divUnit}$ , since the exponential curve does not allow to adequately represent how the error is triggered for low divisions. But, since at these divisions, the error is high and of low interest, the exponential fit already adequately covers the expected range of work to be covered.

As an example, the procedure followed for the **square case** is explained below:

- 1. Calculation of the "real" tensor. For the case of the square, the result is as follows:

$$C_{real} = \begin{bmatrix} 0,6703 & 0,1921 & 0 \\ 0,1921 & 0,6703 & 0 \\ 0 & 0 & 0,1823 \end{bmatrix} \quad (6.2)$$

- 2. Obtention of the calculated tensors for each number of divisions and evaluation of the relative error by means of equation 6.1.

divUnit	$C_{calc}$	Relative Error (%)
25	$\begin{bmatrix} 0,6506 & 0,1811 & -0,0036 \\ 0,1811 & 0,6506 & -0,0036 \\ -0,0036 & -0,0036 & 0,1743 \end{bmatrix}$	$2,54 \cdot 10^{-3}$
50	$\begin{bmatrix} 0,6569 & 0,1864 & -0,0021 \\ 0,1864 & 0,6569 & -0,0021 \\ -0,0021 & -0,0021 & 0,1783 \end{bmatrix}$	$6,92 \cdot 10^{-4}$
100	$\begin{bmatrix} 0,6662 & 0,1897 & -0,0010 \\ 0,1897 & 0,6662 & -0,0010 \\ -0,0010 & -0,0010 & 0,1797 \end{bmatrix}$	$3,37 \cdot 10^{-5}$
175	$\begin{bmatrix} 0,6669 & 0,1908 & -0,0006 \\ 0,1908 & 0,6669 & -0,0006 \\ -0,0006 & -0,0006 & 0,1812 \end{bmatrix}$	$1,24 \cdot 10^{-5}$
300	$\begin{bmatrix} 0,6703 & 0,1921 & -0,0004 \\ 0,1921 & 0,6703 & -0,0004 \\ -0,0004 & -0,0004 & 0,1823 \end{bmatrix}$	$\approx 0$

Table 6.1: Relative error obtained for each number of divisions for the squared mesh.

- 3. Fit of an exponential curve to the points obtained in the above table:

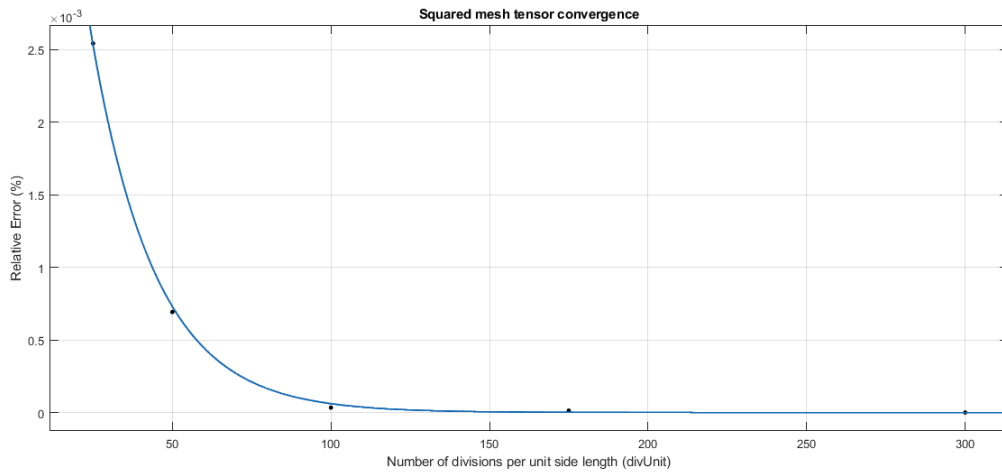


Figure 6.1: Convergence of the squared mesh tensor upon the increase of divUnit.

The figures below show the calculated points and the exponential fit for the rest of the cells:

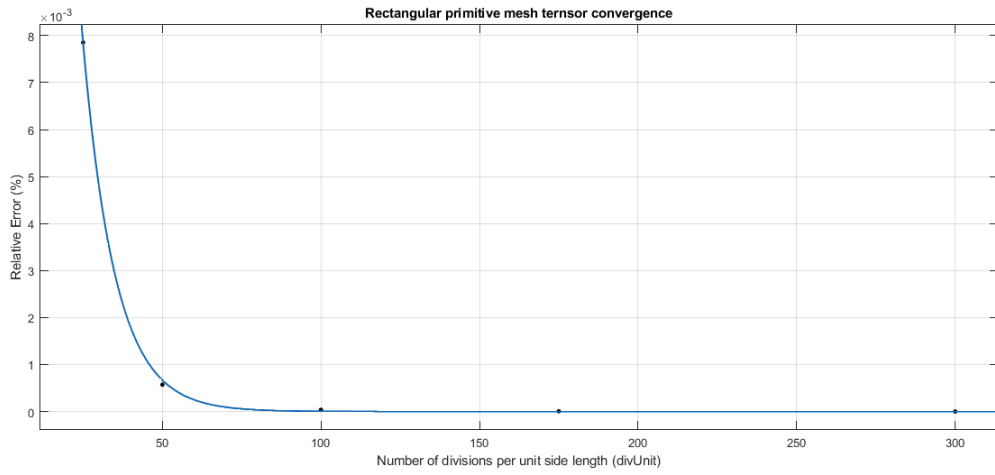


Figure 6.2: Convergence of the Rectangular Primitive mesh tensor upon the increase of divUnit.

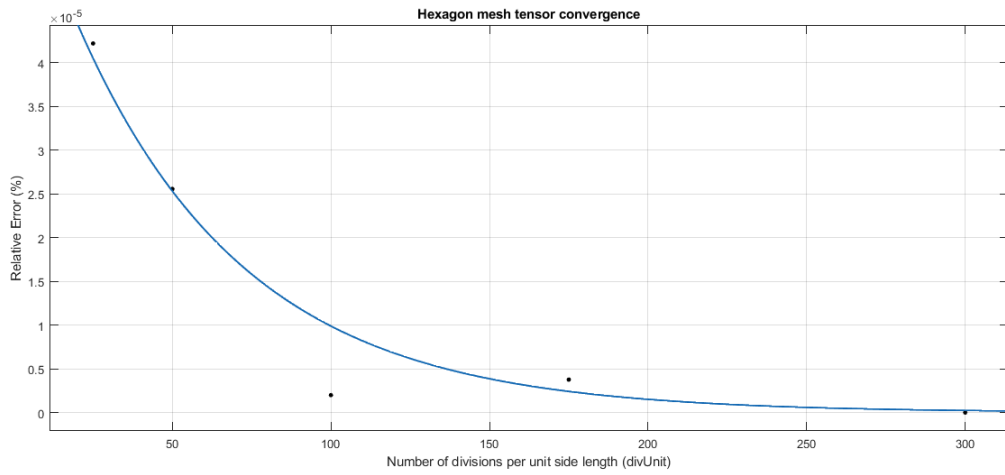


Figure 6.3: Convergence of the Hexagon mesh tensor upon the increase of divUnit.

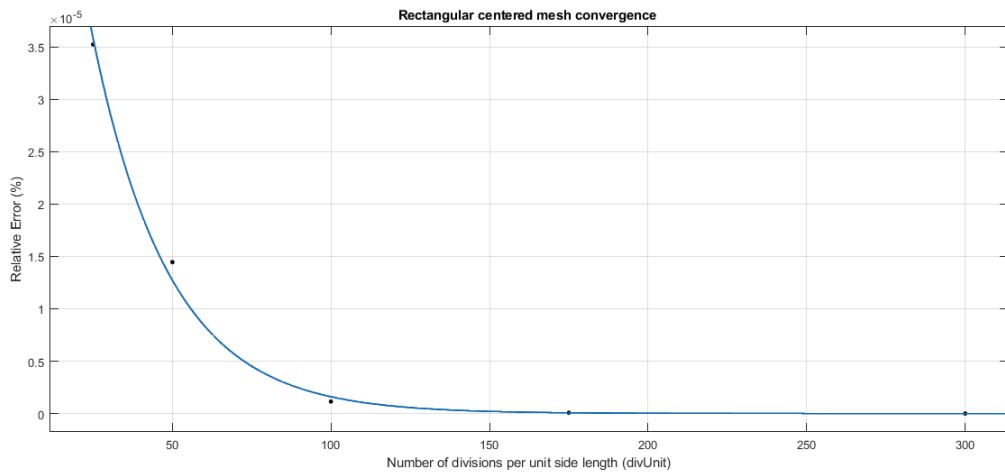


Figure 6.4: Convergence of the Rectangular Centered mesh tensor upon the increase of divUnit.

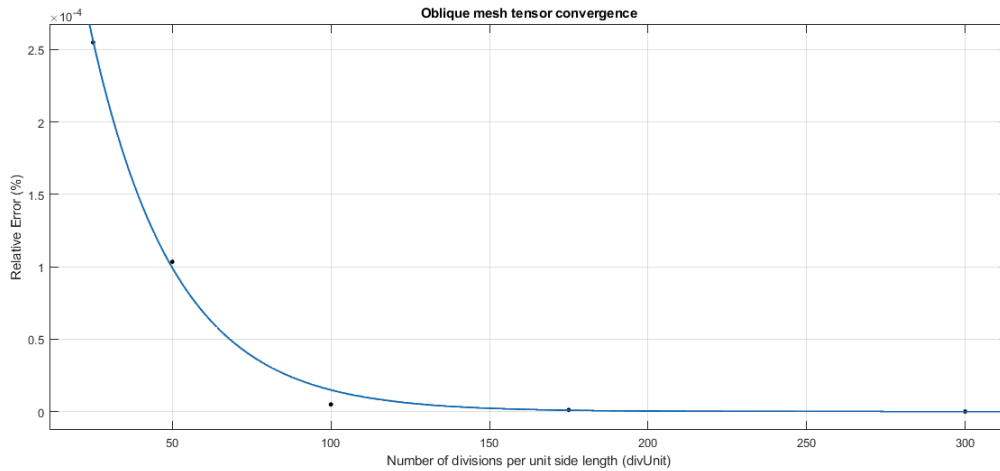


Figure 6.5: Convergence of the Oblique mesh tensor upon the increase of divUnit.

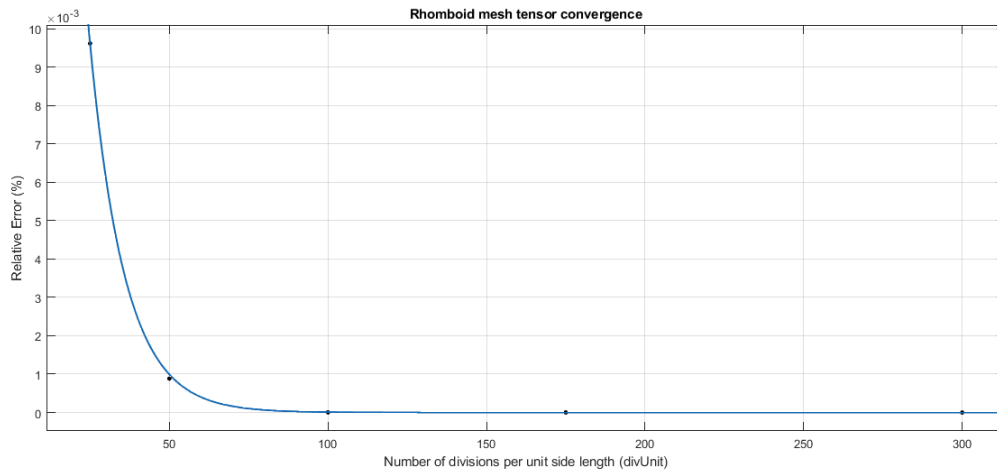


Figure 6.6: Convergence of the Rhomboid mesh tensor upon the increase of divUnit.

It is observed that the errors committed are of very low magnitude, achieving good results from 60 divisions for all the shapes considered with a relatively low computational cost. Again, it is important to note that the zone between 0 and 25 divisions does not correctly capture the behavior of the error with the exponential fit, so it is recommended to avoid it. A peculiarity observed in all cases is that from 25 elements onwards, the symmetry of the tensor is correctly shown when passing it through the Auffray filter. This is clearly observed in the case of the square, where from 25 elements up to 300, the D4 symmetry is maintained if the tensors are filtered by the Auffray algorithm (it evaluates the error made in positions 13 and 23 and their symmetrical positions of the tensor and eliminates them). This event is very useful for further studies, since with low computational cost, the symmetry characteristics of the tensor can be obtained.

For a better comparison of the error trend for all types of cells, the following figure is shown:

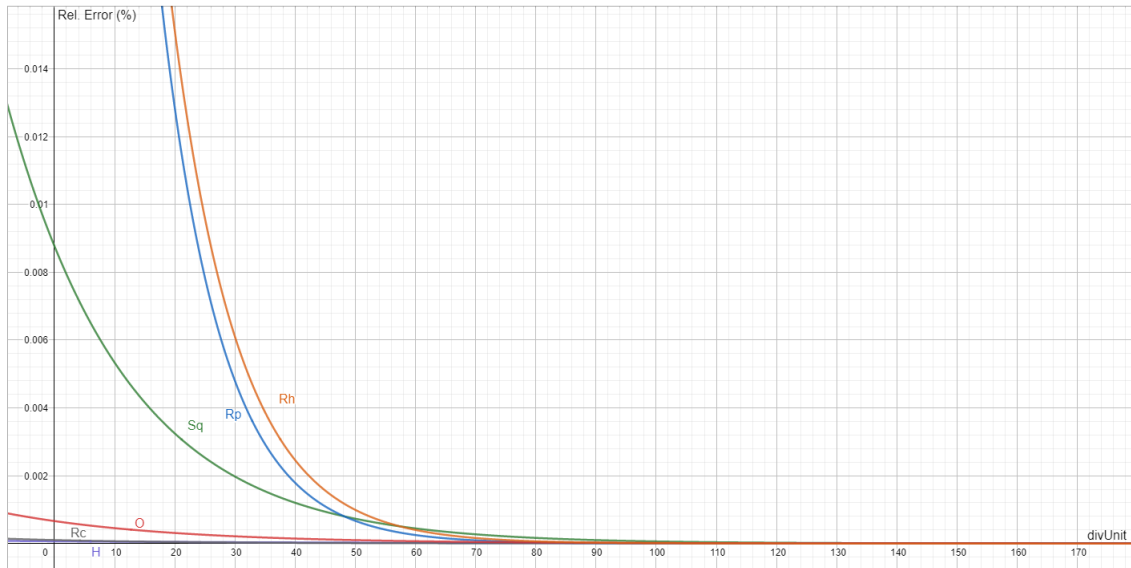


Figure 6.7: Comparative of the convergence of shaped mesh tensors upon the increase of DivUnit.

This comparison shows that four-sided polygons have a slightly higher error than six-sided polygons before convergence. This is due to the fact that the mesher densifies the hexagonal meshes to try to keep the distance between nodes invariant, generating, in general, a higher number of elements in the grid.

The differences between the curves of the four-sided polygons are due to the differences in the lengths of the sides of each of the polygons and their direct relationship with the number of divisions generated by means of the user-imposed variable `divUnit`. On the other hand, for the six-sided polygons, the error curves in the cells practically follow the same pattern despite varying side lengths, so it follows that the meshing tool is able to better separate the relationships between `divUnit` and side lengths by creating more elements.

In general, rigorous results are obtained from 60 elements and the error in the tensor converges to almost zero from 100 elements for all shapes.

## 6.2 Demonstration of the Neumann's principle

The Neumann principle postulates that there are connections between the symmetries in the micro-architecture and the symmetries in the elastic constitutive tensor by means of the point groups. The table 3.1 establishes these connections and through this section we seek to exemplify some of them to demonstrate the fulfillment of Neumann's principle. For these demonstrations the following methodology will be followed:

- Mesh generation by means of the shape creator algorithm. The number of divisions, as previously mentioned, must be greater than 25 divisions to guarantee that the symmetries in the tensor are shown. For the subsequent demonstrations, 60 `divUnit` have been chosen.
- Generation without topological optimization of the material distribution within the lattice to emulate the results of some of the crystallographic plane groups. This is done using the Level Set methodology and forcing a certain symmetry in each of the Voronoi cells.

- Obtention of the homogenized elastic tensor and identification of the symmetries of this one transferred to the natural basis by means of the Auffray algorithm.
- Tessellation of the deformations for the canonical stress states (horizontal, vertical and shear) using GiD to show the compliance of this essential principle and the behavior of the micro-structure in these situations.

The result of the present demonstration is a table showing the obtained tensor together with its symmetry and the generated topology together with its plane group. In this way, it will be possible to analyze in a simple way whether the relations specified in the table 3.1 are verified.

To proceed in a simple manner, a circular gap of material has been introduced in the center of each topology to achieve the maximum plane symmetry group of each cell type.

Before analyzing the results, it should be noted that the red zone of the deformations of the first column represents the zone of presence of material, while the blue zone is the zone of absence of material. The rest of the colors observed is a scale that allows to pass continuously from the zone without material to the zone with material. In this case, because the decision does not affect the evaluation of the tensor symmetries, it can be considered a zone without material.

By observing the table 6.2 in the first instance it is reported the fulfillment of the tessellation principle, an essential requirement to continue with the analysis. The results of each of the cells from the top to the bottom of the table are discussed below:

- The circle inside the oblique lattice forces a symmetry of type p2, since the rotation of half of the domain  $180^\circ$  generates the entire topology of the unit cell. A figure showing the topology is indicated below. It should be noted that the union between two opposite vertices is the one that generates the minimum unit symmetry to be rotated. According to table 3.1 the symmetry of the geometrical domain corresponds with a fully anisotropic elastic tensor coinciding with the results shown in the table of results 6.2
- The next cell is the rectangular centered one. As the table indicates it has a c2mm symmetry with two rotations and two systems of mirror lines. Therefore, the complete topology can be created by selecting a quarter of the unit cell, rotating it by  $180^\circ$  and then using one of the two mirror line systems (vertical or horizontal) to generate the entire micro-cell. As before, it complies with what was developed during the theory, by obtaining an orthotropic tensor.
- The following one is the rectangular primitive with the same symmetry as the rectangular centered case. The forced hole in the center of the topology causes two rotation operations and two mirror line systems to appear. This topology, having p2mm plane symmetry and an orthotropic elastic tensor, complies with the Neumann principle.
- The next case introduces four rotations and two sets of symmetry lines for the squared micro-cell. As expected, the p4mm plane group symmetry observed in the undeformed 6.8, allows the the tensor to have a tetragonal format.
- The fifth case is the diamond test, where the square is rotated  $45^\circ$  to check whether a tensor D2 or D4 is obtained. Since a material hole is introduced in the center, it could be assumed that the symmetry is of the c2mm type, but, there is a higher symmetry in this case associated with the square without rotating  $45^\circ$ , which is the p4mm. That is why, when passing the tensor in cartesian coordinates through the Auffray algorithm, it indicates that it is necessary

to rotate the tensor and the topology  $-45^\circ$  to obtain the tetragonal tensor.

- Finally, the maximum symmetry is studied by means of the hexagon. The plane group corresponding to this succession of symmetry operations is  $p6mm$ , where six rotations and two mirror line systems appear. Again, this case also fulfils the Neumann's theorem.

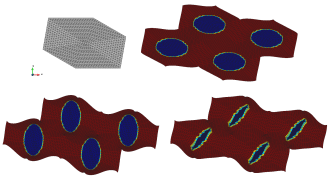
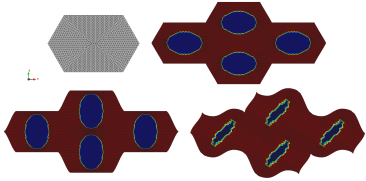
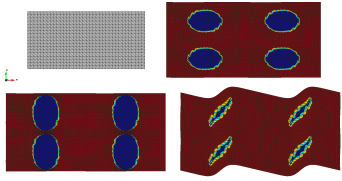
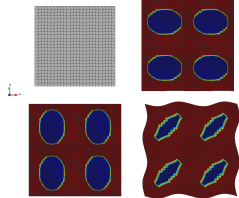
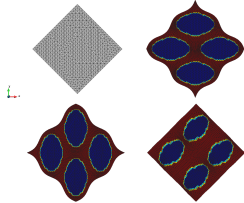
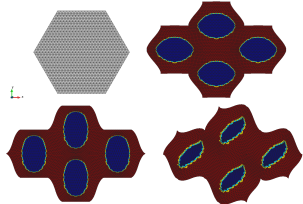
Cell mesh & deformed tessellation	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p2	$\begin{bmatrix} 0,5955 & 0,2038 & 0,0084 \\ 0,2038 & 0,6101 & -0,0087 \\ 0,0084 & -0,0087 & 0,2108 \end{bmatrix}$	Z2
	c2mm	$\begin{bmatrix} 0,7211 & 0,2227 & 0,0000 \\ 0,2227 & 0,7670 & 0,0000 \\ 0,0000 & 0,0000 & 0,2242 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,8157 & 0,2589 & 0,0000 \\ 0,2589 & 0,8769 & 0,0000 \\ 0,0000 & 0,0000 & 0,2525 \end{bmatrix}$	D2
	p4mm	$\begin{bmatrix} 0,6537 & 0,1864 & 0,0000 \\ 0,1864 & 0,6537 & 0,0000 \\ 0,0000 & 0,0000 & 0,1783 \end{bmatrix}$	D4
	p4mm	$\begin{bmatrix} 0,4045 & 0,0821 & 0,0000 \\ 0,0821 & 0,4045 & 0,0000 \\ 0,0000 & 0,0000 & 0,0651 \end{bmatrix}$	D4 ( $\theta = -45^\circ$ )
	p6mm	$\begin{bmatrix} 0,5931 & 0,1882 & 0,0000 \\ 0,1882 & 0,5931 & 0,0000 \\ 0,0000 & 0,0000 & 0,2025 \end{bmatrix}$	O(2)

Table 6.2: Demonstration of the fulfillment of Neumann's theorem by means of the correspondence between the symmetry and shape of the micro-cell with the type of symmetry of the elastic tensor. In the first column we can find from left to right and from top to bottom, the mesh shape and divisions, the deformed for the horizontal, vertical and shear state of stress.

The undeformed of each of these cases are presented below in the same order of appearance of the



table in order to identify more easily the symmetries they contain:

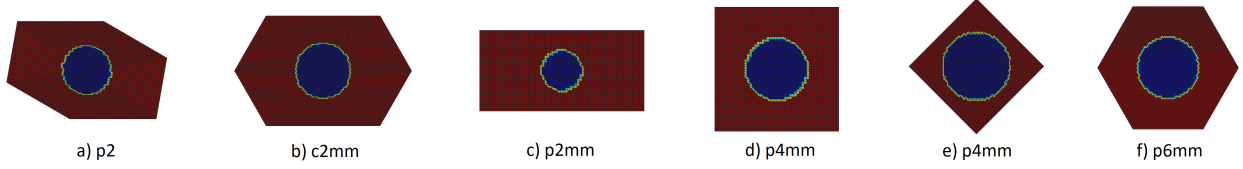


Figure 6.8: Undeformed state with the forced micro-architecture of each of the Voronoi cells and indication of the maximum plane group symmetry associated.

It is clear that the examples in the table comply with both essential principles of the present study: that of tessellation (presenting continuity in the deformed and undeformed by translating the unit cells in a 2D space without overlapping or leaving gaps) and that of Neumann 3.1 (the symmetries in the physical characteristics of the tensor are related to the geometric domain of the micro-architecture through the point groups and their associated plane groups).

### 6.3 Topology optimization in micro-structure design

Once the convergence capacity has been assessed and the ability to comply with the tessellation and Neumann principles has been demonstrated, it is time to put oneself in the shoes of a structure designer and try to obtain a topology that adapts to the target tensor. For this purpose, the inverse problem will be solved by means of the level set methodology, introducing perimeter penalization to avoid patterns with very narrow shapes and therefore, guaranteeing manufacturability, and volume inequality constraints to try to use the minimum amount of material in the designed micro-architecture. The mathematical problem to be solved is presented below:

$$P = \begin{cases} \min & \|C^h - C^*\| + \lambda_p \left( \frac{P}{P^*} - 1 \right) \\ \text{s.t.} & \frac{V}{V^*} - 1 \leq 0 \end{cases} \quad (6.3)$$

where  $C^*$  is the target elasticity tensor,  $\lambda_p$  the importance or weight assigned to the minimization of the perimeter ( $0 < \lambda_p \leq 1$ ),  $P^*$  is the perimeter target and  $V^*$  is the desired volume fraction. The present design seeks to observe the capacity of the optimizer to force the necessary symmetries to achieve the target tensor typology. For this purpose, the effect of different design variables selected during the preprocessing on the final result will be evaluated. These design variables are:

- Domain size and shape. Testing of the different Voronoi cells studied so far.
- Initial state of the distribution of material. Testing of different material configurations as a starting point for topological optimization.

The objective is to compare the solutions obtained with each situation to evaluate their effectiveness in designing the desired material.

Two inverse problems are posed in which the first one seeks in the convergence of the solution to an isotropic tensor and the second one seeks in an orthotropic tensor. The rest of the optimizer variables

are set as the following since they have provided the best quality results during the pre-design tests:

$$V^* = 0,5 ; \quad P^* = 1 ; \quad \lambda_p = 0,5$$

It should be noted that in order to maintain good quality results without compromising computational cost, 75 divisions per unit side have been selected for the four-sided cells and 60 for the six-sided cells for all the cases presented in the following subsections.

### 6.3.1 Isotropic target tensor

As discussed in the theoretical section, this is a type of material that has the same physical characteristics in all directions, which generates great importance in the search for optimal topologies that ensure this type of symmetries. According to the table 3.1 the unit cells with the highest affinity to isotropy are the hexagonal ones with p3, p3m1, p6 or p6mm plane group symmetries. This statement will be tested by testing all Vornoi cells combined with different initial conditions (circular inclusion, square inclusion, full, scattered holes...).

In order to numerically evaluate the capacity of each of the tests performed, two indicators are introduced:

- Coefficient of non-isotropy ( $\tau$ ): coefficient between 1 and 0 indicating the level of isotropy of an elastic tensor based on the following calculation [12]:

$$\tau = \frac{\|C^h - C^{iso}\|}{\|C^h\|} \quad (6.4)$$

where  $C^{iso}$  is the projection of  $C^h$  into a space of isotropic tensors and is obtained with the equation below specified:

$$C^{iso} = \begin{bmatrix} K^{iso} + G^{iso} & K^{iso} - G^{iso} & 0 \\ K^{iso} - G^{iso} & K^{iso} + G^{iso} & 0 \\ 0 & 0 & G^{iso} \end{bmatrix} \quad (6.5)$$

where,

$$K^{iso} = \frac{1}{8}(3C_{11}^h + 3C_{22}^h + 2C_{12}^h + 2C_{33}^h) - G^{iso} \quad G^{iso} = \frac{1}{8}(C_{11}^h + C_{22}^h - 2C_{12}^h + 2C_{33}^h)$$

It should be noted that the closer a tensor is to isotropy, the lower will be its  $\tau$  coefficient, being the null value, the maximum level of isotropy. This indicator will show the capacity of the optimizer to generate those symmetries that bring it closer to isotropy.

- Relative error ( $\epsilon_{rel}$ ): error made when trying to find the target tensor. This is calculated by the following expression:

$$\epsilon_{rel}(\%) = \frac{\|C^h - C^*\|}{\|C^*\|} \cdot 100 \quad (6.6)$$

This indicator will allow observing the capacity of the optimizer to approach to the desired tensor, regardless of whether it generates the symmetries conducive to isotropy.

Before starting the study it is necessary to fix the last and one of the most important design variables: the target tensor. The arbitrarily chosen isotropic target tensor is shown below:

$$C^* = \begin{bmatrix} 0,5931 & 0,1882 & 0,0000 \\ 0,1882 & 0,5931 & 0,0000 \\ 0,0000 & 0,0000 & 0,2025 \end{bmatrix} \quad (6.7)$$

A series of four tables are presented below, each showing the results for each of the four initial conditions considered for all Voronoi cell types.

For the case of circle inclusion as the initial condition, the following results are obtained:

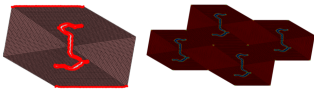
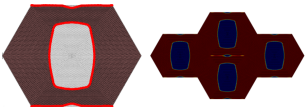
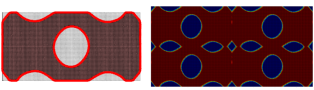
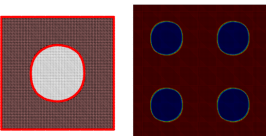
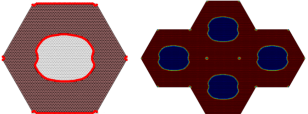
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p2	$\begin{bmatrix} 0,6055 & 0,1845 & 0,0129 \\ 0,1845 & 0,6867 & -0,0129 \\ 0,0129 & -0,0129 & 0,2215 \end{bmatrix}$	Z2
	c2mm	$\begin{bmatrix} 0,5976 & 0,1788 & 0,0000 \\ 0,1788 & 0,9260 & 0,0000 \\ 0,0000 & 0,0000 & 0,1680 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,5633 & 0,1865 & 0,0000 \\ 0,1865 & 0,4390 & 0,0000 \\ 0,0000 & 0,0000 & 0,1880 \end{bmatrix}$	D2
	p4mm	$\begin{bmatrix} 0,6649 & 0,1877 & 0,0000 \\ 0,1877 & 0,6649 & 0,0000 \\ 0,0000 & 0,0000 & 0,1773 \end{bmatrix}$	D4
	c2mm	$\begin{bmatrix} 0,6223 & 0,1894 & 0,0000 \\ 0,1894 & 0,5257 & 0,0000 \\ 0,0000 & 0,0000 & 0,1928 \end{bmatrix}$	D2

Table 6.3: Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the circle inclusion.

It should be noted that the first column indicates the optimized mesh (left) and the tessellation of four micro-cells (right). This comment applies for the rest of the initial conditions considered in the following tables.

For the case of full of material as the initial condition, the following results are obtained:

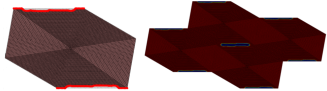
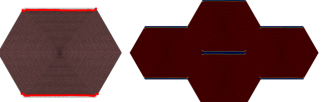
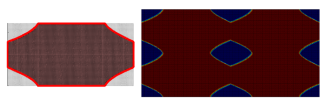
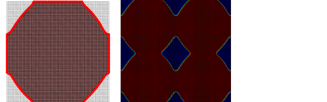
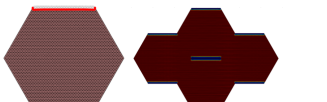
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p2	$\begin{bmatrix} 0,8693 & 0,1870 & -0,0010 \\ 0,1870 & 0,5395 & -0,0069 \\ -0,0010 & -0,0069 & 0,2446 \end{bmatrix}$	Z2
	c2mm	$\begin{bmatrix} 1,3656 & 0,1910 & 0,0000 \\ 0,1910 & 0,5601 & 0,0000 \\ 0,0000 & 0,0000 & 0,3264 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,7307 & 0,1893 & 0,0000 \\ 0,1893 & 0,6483 & 0,0000 \\ 0,0000 & 0,0000 & 0,1808 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,5866 & 0,1852 & 0,0000 \\ 0,1852 & 0,6853 & 0,0000 \\ 0,0000 & 0,0000 & 0,2033 \end{bmatrix}$	D2
	c2mm	$\begin{bmatrix} 1,0014 & 0,1848 & 0,0000 \\ 0,1848 & 0,5268 & 0,0000 \\ 0,0000 & 0,0000 & 0,2695 \end{bmatrix}$	D2

Table 6.4: Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the full of material cell.

For the case of square inclusion as the initial condition, the following results are obtained:

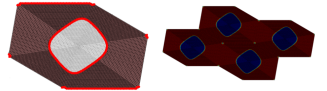
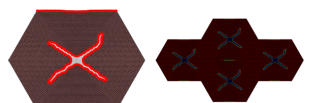
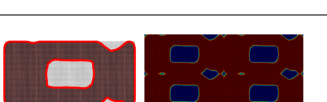
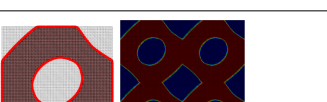

Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p2	$\begin{bmatrix} 0,4717 & 0,1877 & -0,0314 \\ 0,1877 & 0,4497 & -0,0179 \\ -0,0314 & -0,0179 & 0,1770 \end{bmatrix}$	Z2
	p2	$\begin{bmatrix} 0,8788 & 0,1773 & 0,0341 \\ 0,1773 & 0,8234 & 0,0074 \\ 0,0341 & 0,0074 & 0,2323 \end{bmatrix}$	Z2
	pm	$\begin{bmatrix} 0,6785 & 0,1840 & 0,0000 \\ 0,1840 & 0,5420 & 0,0000 \\ 0,0000 & 0,0000 & 0,1646 \end{bmatrix}$	D2
	pm	$\begin{bmatrix} 0,4408 & 0,0990 & 0,0000 \\ 0,0990 & 0,3818 & 0,0000 \\ 0,0000 & 0,0000 & 0,0848 \end{bmatrix}$	D2 ( $\theta = 33,5^\circ$ )
	cm	$\begin{bmatrix} 0,7445 & 0,1830 & 0,0000 \\ 0,1830 & 0,6843 & 0,0000 \\ 0,0000 & 0,0000 & 0,2264 \end{bmatrix}$	D2 ( $\theta = 8,1^\circ$ )

Table 6.5: Results obtained for every Voronoi cell when an isotropic tensor and using as initial state the inclusion of a void square.

For the last initial case, scattered holes, the results are below presented:

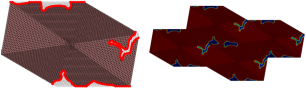
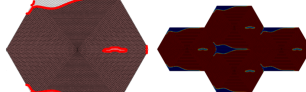
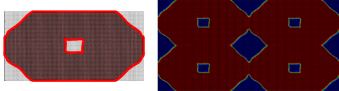
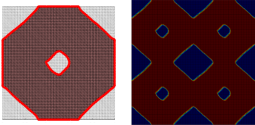
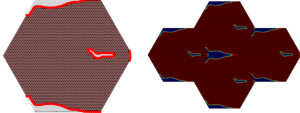
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p1	$\begin{bmatrix} 0,6668 & 0,1856 & 0,0392 \\ 0,1856 & 0,5372 & 0,0074 \\ 0,0392 & 0,0074 & 0,2190 \end{bmatrix}$	Z2
	cm	$\begin{bmatrix} 1,1833 & 0,1791 & 0,0000 \\ 0,1791 & 0,5388 & 0,0000 \\ 0,0000 & 0,0000 & 0,2797 \end{bmatrix}$	D2
	p2	$\begin{bmatrix} 0,6260 & 0,1807 & -0,0015 \\ 0,1807 & 0,6454 & -0,0011 \\ -0,0015 & -0,0011 & 0,1968 \end{bmatrix}$	Z2
	pm	$\begin{bmatrix} 0,5453 & 0,1863 & 0,0000 \\ 0,1863 & 0,4969 & 0,0000 \\ 0,0000 & 0,0000 & 0,1802 \end{bmatrix}$	D2
	p1	$\begin{bmatrix} 0,8773 & 0,1756 & -0,0064 \\ 0,1756 & 0,5263 & -0,0053 \\ -0,0064 & -0,0053 & 0,2439 \end{bmatrix}$	Z2

Table 6.6: Results obtained for every Voronoi cell when forcing an isotropic tensor and using as initial state the inclusion of void scattered holes.

With all the results presented, we proceed to their joint analysis. As mentioned above, two meters are available, one indicating the ability of the optimizer to force isotropy and the other showing the ability to adapt to the isotropic objective tensor. It should be noted that the optimizer does not force isotropy on the tensor, but tries to adapt to a predetermined isotropic tensor, which implies that all its positions try to approximate those of the objective without taking into account the relationships between positions. That is why, in the results it is not strange to find tensors with high  $\tau$  and very low  $\epsilon_r$ .

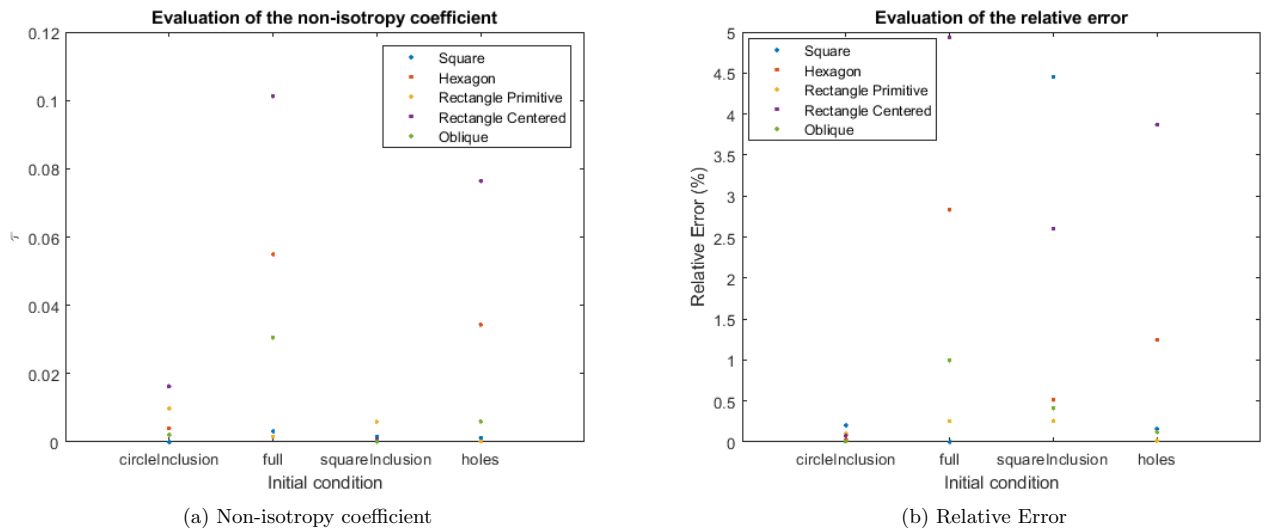


Figure 6.9: Non-isotropy coefficient and relative error for all the Voronoi cells and initial conditions considered

For the first of the initial conditions, the best isotropy behavior is that of the square, followed by the oblique and hexagonal lattice. In the case of the relative error, all are of low magnitude, with the square standing out above the average. In this case, this last mentioned form compromises in distance of its determinant to that of the target tensor to maintain the symmetries that bring it close to isotropy, unlike the hexagon, which approaches in few iterations to the target tensor, compromising in the level of isotropy.

For the initial full condition, in all the Voronoi cells it is observed a great difficulty to break the symmetry of the initial topology and therefore, to comply with the minimum desired volume for the preset iterations. That is why a great dispersion of the results is observed, highlighting the speed of convergence of the square and hexagon to the objective.

For the squareInclusion case, the results are similar to the circleInclusion. Very low results for the non-isotropy coefficient are observed, which indicates that the homogenized tensors are close to their associated tensor in the isotropy basis for each micro-cell. This could be due to the fact that both initial forms mentioned are compatible with the maximal plane group symmetries of each Voronoi cell, bringing the tensor and topology closer to isotropy. In reference to the relative error, it is observed in the topologies of the table 6.5 that they are not able to approach those suggesting that of the target tensor, causing again very scattered errors among the different shapes.

The last case is the most chaotic, in which the importance of the symmetry of the material distribution in the cell under initial conditions becomes evident. The six-sided cells were tested with an irregular distribution of three initial holes, causing the optimizer to have great difficulty in breaking this symmetry and approaching the target. The opposite happens with the four-sided shapes, in which five holes were arranged allowing to start the optimization with symmetries of the type 2mm. This is why better results are observed in the  $\tau$  and  $\epsilon_r$  plots.

In this way, it is shown that when forcing isotropy, not only are micro-cells more closely related to the target tensor, but also the symmetry of the initial topology must be properly selected. It will then be advisable to provide the maximum symmetry desired in the final topology, conditioning the optimizer so that it does not deviate excessively from this. For this reason, the square and hexagonal shape with circleInclusion and squareInclusion are the best options for this case.

### 6.3.2 Orthotropic target tensor

Orthotropy refers to the type of material that has different properties depending on the direction of the constituent axis of the selected cell for measurement. This is why it is expected to obtain a distribution with a high amount of material in the transverse distribution and less in the longitudinal (assuming that these are the constituent directions of the micro-cell) or vice versa. It is therefore expected that the best results will be observed for rectangular primitive cells, by initially conditioning the optimizer with an uneven distribution of material in one constituent direction of the micro-cell than in the other.

Again, to verify these theoretical propositions, all Voronoi cells will be tested with the same initial material distributions considered in the isotropy section.

For this case, the orthotropic target tensor is as follows:

$$C^* = \begin{bmatrix} 0,8157 & 0,2589 & 0,0000 \\ 0,2589 & 0,8769 & 0,0000 \\ 0,0000 & 0,0000 & 0,2525 \end{bmatrix} \quad (6.8)$$

The results obtained are presented below, classified in four tables according to the initial conditions of material distribution in the optimizer. For the case of circular inclusion, the following results are obtained:

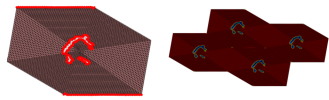
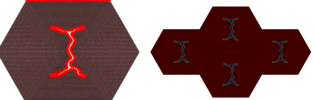
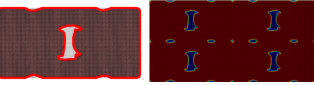
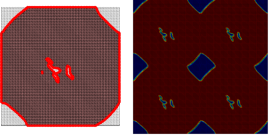
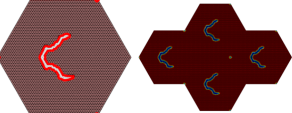
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p1	$\begin{bmatrix} 0,8208 & 0,2550 & 0,0150 \\ 0,2550 & 0,7823 & 0,0064 \\ 0,0150 & 0,0064 & 0,2709 \end{bmatrix}$	Z2
	cm	$\begin{bmatrix} 0,9412 & 0,2536 & 0,0000 \\ 0,2536 & 1,1479 & 0,0000 \\ 0,0000 & 0,0000 & 0,3042 \end{bmatrix}$	D2 ( $\theta = -88, 0^\circ$ )
	pm	$\begin{bmatrix} 0,7972 & 0,2642 & 0,0000 \\ 0,2642 & 0,8967 & 0,0000 \\ 0,0000 & 0,0000 & 0,2734 \end{bmatrix}$	D2 ( $\theta = -76, 1^\circ$ )
	p4	$\begin{bmatrix} 0,7885 & 0,2434 & 0,0000 \\ 0,2434 & 0,7885 & 0,0000 \\ 0,0000 & 0,0000 & 0,2582 \end{bmatrix}$	D4
	cm	$\begin{bmatrix} 0,7978 & 0,2581 & 0,0000 \\ 0,2581 & 0,8913 & 0,0000 \\ 0,0000 & 0,0000 & 0,2850 \end{bmatrix}$	D2 ( $\theta = -87, 5^\circ$ )

Table 6.7: Results obtained for every Voronoi cell when forcing an orthotropic tensor and using as initial state the inclusion of void circle.

The full of material initial case, generates the following results:



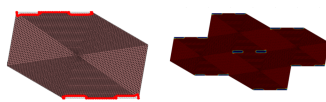
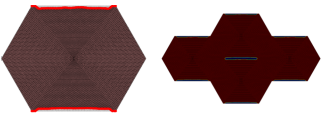
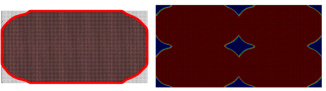
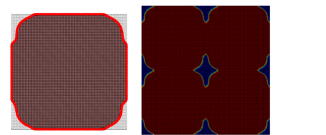
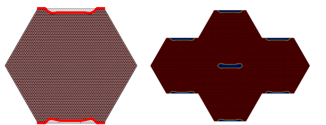
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p2	$\begin{bmatrix} 0,8900 & 0,2525 & 0,0021 \\ 0,2525 & 0,7470 & 0,0029 \\ 0,0021 & 0,0029 & 0,2755 \end{bmatrix}$	Z2
	c2mm	$\begin{bmatrix} 1,4028 & 0,2577 & 0,0000 \\ 0,2577 & 0,7538 & 0,0000 \\ 0,0000 & 0,0000 & 0,3628 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,8684 & 0,2525 & 0,0000 \\ 0,2525 & 0,7910 & 0,0000 \\ 0,0000 & 0,0000 & 0,2718 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,8708 & 0,2545 & 0,0000 \\ 0,2545 & 0,7359 & 0,0000 \\ 0,0000 & 0,0000 & 0,2796 \end{bmatrix}$	D2 ( $\theta = -88,3^\circ$ )
	c2mm	$\begin{bmatrix} 1,0379 & 0,2574 & 0,0000 \\ 0,2574 & 0,7538 & 0,0000 \\ 0,0000 & 0,0000 & 0,3039 \end{bmatrix}$	D2

Table 6.8: Results obtained for every Voronoi cell when forcing an orthotropic tensor and using as initial state the full of material cell.

For the square inclusion case, the results presented below are obtained:

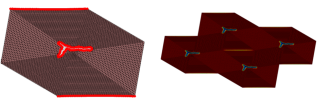
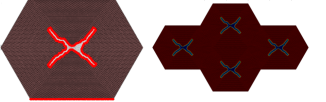
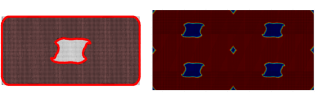
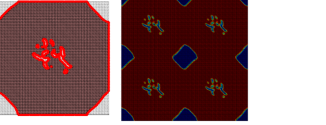
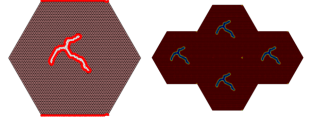
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p1	$\begin{bmatrix} 0,8728 & 0,2501 & 0,0014 \\ 0,2501 & 0,7545 & -0,0019 \\ 0,0014 & -0,0019 & 0,2715 \end{bmatrix}$	Z2
	c2mm	$\begin{bmatrix} 1,0496 & 0,2529 & 0,0000 \\ 0,2529 & 0,9848 & 0,0000 \\ 0,0000 & 0,0000 & 0,2990 \end{bmatrix}$	D2
	p2mm	$\begin{bmatrix} 0,8429 & 0,3004 & 0,0000 \\ 0,3004 & 0,7805 & 0,0000 \\ 0,0000 & 0,0000 & 0,2943 \end{bmatrix}$	D2 ( $\theta = 33,0^\circ$ )
	p4mm	$\begin{bmatrix} 0,7732 & 0,2308 & 0,0000 \\ 0,2308 & 0,7732 & 0,0000 \\ 0,0000 & 0,0000 & 0,2444 \end{bmatrix}$	D4
	cm	$\begin{bmatrix} 0,8923 & 0,2565 & 0,0000 \\ 0,2565 & 0,8523 & 0,0000 \\ 0,0000 & 0,0000 & 0,2896 \end{bmatrix}$	D2 ( $\theta = -88,0^\circ$ )

Table 6.9: Results obtained for every Voronoi cell when forcing an orthotropic tensor and using as initial state the inclusion of a void square.



Finally, the results for the scattered holes are presented in the following table:

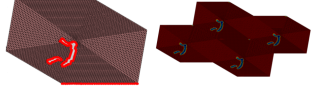
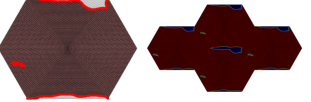
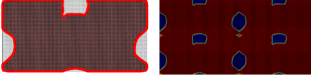
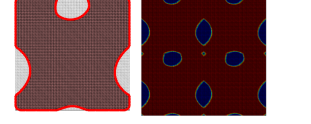
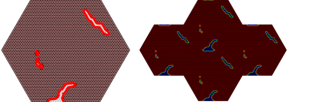
Optimized mesh & tessellated cell	Plane group symmetry	$C^h$ (in Natural basis)	Elastic symmetry class
	p1	$\begin{bmatrix} 0,7966 & 0,2545 & 0,0103 \\ 0,2545 & 0,8191 & -0,0118 \\ 0,0103 & -0,0118 & 0,2720 \end{bmatrix}$	Z2
	p1	$\begin{bmatrix} 1,3372 & 0,2563 & -0,0288 \\ 0,2563 & 0,7298 & -0,0106 \\ -0,0288 & -0,0106 & 0,3478 \end{bmatrix}$	Z2
	pm	$\begin{bmatrix} 0,7922 & 0,2515 & 0,0000 \\ 0,2515 & 0,8049 & 0,0000 \\ 0,0000 & 0,0000 & 0,2575 \end{bmatrix}$	D2
	pm	$\begin{bmatrix} 0,6693 & 0,2560 & 0,0000 \\ 0,2560 & 0,7443 & 0,0000 \\ 0,0000 & 0,0000 & 0,2571 \end{bmatrix}$	D2 ( $\theta = 87,0^\circ$ )
	p1	$\begin{bmatrix} 0,8490 & 0,2539 & -0,0120 \\ 0,2539 & 0,8157 & 0,0031 \\ -0,0120 & 0,0031 & 0,2961 \end{bmatrix}$	Z2

Table 6.10: Results obtained for every Voronoi cell when forcing an orthotropic tensor and using as initial state the inclusion of void scattered holes.

We observe that in many cases of the results presented the dominance of the distribution of material in one of the constituent directions of the cell in relation to the other is not notably observed, because the proposed objective tensor does not have a high level of orthotropy, presenting low differences between positions 11 and 22 of the tensor. Nevertheless, it is possible to evaluate the ability of each type of cell and initial conditions to condition the optimizer. The only difference with respect to the isotropy case is that there is no gauge of the capacity of the topology to generate the symmetries that bring the tensor closer to orthotropy. In its absence, the comparison will be made only according to the capacity of each of the tested conditions to approach the target tensor, by means of the relative error (see figure 6.10).

In the case of circle inclusion, it is observed that all the cells present a very low error in the final homogenized tensor with respect to the target. The only exception or the tensor that is the most out of tune is the rectangular centered one. The hexagonal, rectangular primitive and centered cells manage to reach the orthotropy, having even a low error except in the rectangular centered case, where it is evident that the optimizer has a great difficulty to converge at positions 11 and 22 of the tensor. It should also be noted that the square, reaches a higher symmetry level (tetragonal), since, in this case, it has not been able to break with the p4mm symmetry of the initial topology.

For the full material case, the optimizer is able to converge to orthotropy in all cases except the oblique one (high difficulties to force symmetries in this type of unit cell). Again, as in the isotropy case, this initial condition generates large scatter in the error, having the best results for the rectangular primitive case (low relative error and high convergence speed).

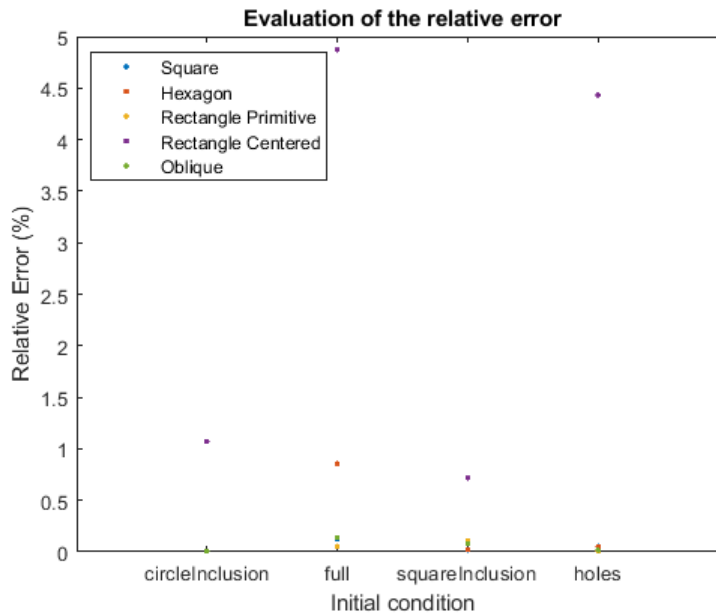


Figure 6.10: Relative error for all the Voronoi cells and initial conditions considered when forcing an orthotropic tensor.

Using as initial case the inclusion of an empty square in each micro-cell, the same symmetries in the tensors are observed as for the circle inclusion case. Again, there is low dispersion of the error results between cells, except in the rectangular centered case.

Finally, the scattered holes case provides the greatest disparity, having anisotropy in the six-sided micro-cells due to the low ability of the optimizer to break with the symmetric irregularity of the initial material scattering (three anti-symmetric holes). Therefore, the only tensors that achieve orthotropy are those associated with four-sided micro-cells, since the initial conditions provide symmetries close to those targeted (type 2mm).

With these results, the importance of selecting initial conditions with symmetry similar to the one required as a target to obtain the target tensor has been observed again. As predicted, the rectangle primitive is the cell with the highest convergence capability to the orthotropic tensors for the four initial conditions considered.

## 6.4 Analysis of the non-Voronoi rhomboid cell

So far, the behavior of the tensor of all the Voronoi cells has been analyzed against preprocessing variations in the mesh. Given the ability of the meshing code to represent any four-sided and six-sided shape, the study of the tensor symmetries of a rhomboidal topology with a central hole is proposed, varying slightly the basic shape parameters of the mesh. The figure below shows the three design parameters of the rhomboidal shape:

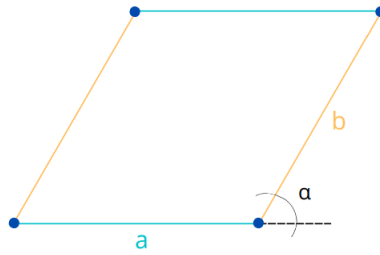


Figure 6.11: Identification of the design variables of the rhomboid shape.

The convergence analysis carried out in the first section of the practical part 6.1 shows that for a study of symmetries in the tensor, it would be sufficient to choose a mesh with `divUnit` greater than 25. For this particular case, 70 divisions have been chosen to obtain more rigorous results without considerably increasing the computational cost. We start this study with the rhomboid of  $a = b$  and  $\alpha = 60^\circ$ . The following figure shows the tessellated results of the deformations of the corresponding lattice in the canonical stress states, together with the initial mesh:

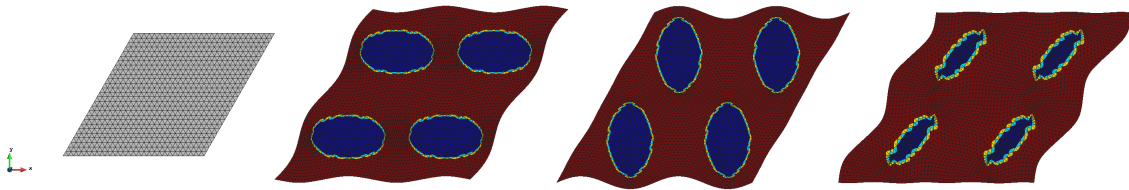


Figure 6.12: Display of the rhomboid mesh ( $a = b$ ,  $\alpha = 60^\circ$ ), the deformed tessellated horizontal, vertical and shear tension states from left to right.

Surprisingly, the tensor obtained in this case is isotropic, which opens the door to further study of the influence on the variations of the parameters that form the mesh. For this purpose, rhomboids with the characteristics specified below are tested and the symmetries obtained in the tensors are evaluated:

a,b relation	$\alpha$	C	Symmetry
a = b	$60^\circ$	$\begin{bmatrix} 0,5978 & 0,1995 & 0,0000 \\ 0,1995 & 0,5979 & 0,0000 \\ 0,0000 & 0,0000 & 0,1992 \end{bmatrix}$	O(2)
a $\neq$ b	$60^\circ$	$\begin{bmatrix} 0,6788 & 0,2337 & -0,0025 \\ 0,2337 & 0,6841 & 0,0025 \\ -0,0025 & 0,0025 & 0,2332 \end{bmatrix}$	Z2
a = b	$45^\circ$	$\begin{bmatrix} 0,5253 & 0,1783 & 0,0078 \\ 0,1783 & 0,5351 & -0,0078 \\ 0,0078 & -0,0078 & 0,1760 \end{bmatrix}$	Z2

Table 6.11: Design parameters of the rhomboid mesh alongside with the tensor obtained and its symmetry.

This shows that only the first of the tested configurations provides symmetries in the homogenized tensor. The explanation for this is based on the fact that this type of non-Voronoi cell is one third of a hexagonal Voronoi cell with p3 or p3m1 type plane group symmetries. Therefore, when tessellating the rhomboid, the Voronoi cell associated to the mosaic is a hexagon. The following figure clarifies the explanation:

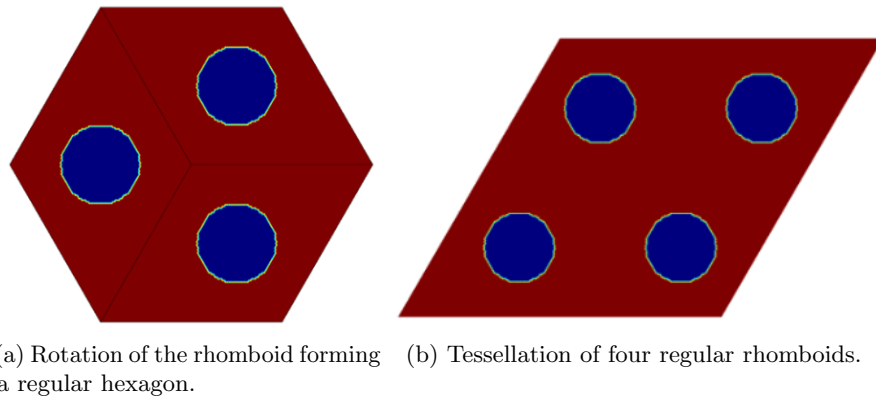


Figure 6.13: Demonstration of the relation between the non-Voronoi rhomboid cell and the regular hexagon.

It should be noted that the rotated hexagonal cell in the figure on the left generates the same pattern as the figure on the right, demonstrating that the hexagon is the Voronoi cell associated with the tessellation of the regular rhomboid.

By slightly varying the conditions of the regular rhomboidal cell ( $a = b, \alpha = 60^\circ$ ), one goes from the hexagonal to the oblique associated Voronoi cell, breaking the symmetries that allowed obtaining a tensor of the O(2) type, generating the Z2 tensor.

## Chapter 7

# Future project expansions

Every study has a logical order of accomplishment of its different sections or work packages. Of course, all of them include a roadmap to follow with all those points of relevance for the success of the research carried out. In most cases, the type available and the complexity of some of the tasks cause part of the roadmap work to be presented as a future extension of the study.

The following table lists the tasks that would follow the completion of this project and the estimated difficulty and time for its completion:

Task	Product	Difficulty	Time (weeks)
Extension of the meshing algorithm	Algorithm	B	3
Development of the symmetry forcing code	Algorithm	A	10
Change of the cost function	Algorithm	A	5
Investigation of performance of new preprocess conditions	Comparison	C	2
3D printing of a complex pattern	3D model	C	2

Table 7.1: Identification of future tasks to expand the study.

It shall be noted that the first column identifies the tasks with its name, the second column represents the final result of each task, the third one classifies the difficulty of each task from A to C where A is the higher grade of difficulty and the fourth column identifies the estimated time for the completion of each task considering 25 hours of work per week.

The work to be carried out in each of these tasks is explained below:

- The first task is based on editing the mesh generator code so that it creates it from the constituent parameters of the Bravais grids. Changing the input parameters of the meshing code to make the transition from the geometric to the crystallographic mindset. The idea would be to provide, instead of the parameters defining the shape of the cell, to provide the mesher with the parameters defining the Bravais lattice ( $\vec{a}_1$ ,  $\vec{a}_2$  and  $\zeta$ ) and have the mesher obtain the associated Voronoi cell, since each set of constituent Bravais parameters has a single associated Voronoi cell. This extension would only require a previous functionality to the current algorithm that would be in charge of translating the parameters of the Bravais lattice to the parameters that define the geometry of the Voronoi cell (parameters that currently define the mesh).

- The second task is one of those identified in the scope of work, which was not finally carried out due to the difficulty involved and the incompatibility with the methodologies used for meshing. It is the generation of a symmetry forcing algorithm in the topological optimizer. This would identify the minimum unit of symmetry of the plane group selected for the corresponding grid and would try to generate the symmetry operations required to obtain the plane group specified by the user. This process would not only entail the implementation of this algorithm, but would also require editing the mesh generator algorithm to avoid interference from improper meshing when selecting the minimum symmetry unit. Therefore, to avoid interference, it would be necessary to mesh in such a way that the limits of the minimum unit coincide with the edges of the elements, and in no case, the elements cross this edge. It is for these reasons that a high execution time and difficulty has been assigned.
- The third of the proposed tasks consists of modifying the cost function of the topological optimizer when solving inverse problems. The objective would be to minimize the distance between the homogenized and target tensors as before, but penalizing the fact that the topology breaks symmetries. In this way, the optimizer would be driven towards the desired target tensor, maintaining the symmetries of the initial topology. With this task, the effectiveness of the optimizer would be improved, providing the target symmetry in the initial tensor and using the cells related to the target tensor type. In this case, it is considered to be a task of high difficulty, since this task requires a high level of research to be implemented in the optimizing algorithm.
- The fourth of the proposed tasks is the testing of other initial conditions and mesh design parameters that help the user to have a greater number of possibilities or alternatives to classify the project. It is classified as a simple task, since the optimizing algorithm is already implemented and therefore, the completeness of this task is entirely dependent on the time required by the program to converge to the solution.
- Finally, the last of the proposed future tasks is the 3D printing of a complex optimized microstructure model. This task would ratify the potential of 3D printing to represent complex shapes in contrast to conventional manufacturing methods.

## Chapter 8

# Analysis and evaluation of the environmental, social and technological implications

Since the resources available in nature are limited, it is vitally important to use them as efficiently as possible. The lack of raw materials that is evident today, leads to the need to optimize structural designs to the maximum so that they resist given stresses using the minimum amount of material possible and thus reducing production costs. Therefore, the topological optimization method developed in this study allows:

- Minimize the excessive use of resources.
- Reduce the environmental impact for the procurement of raw materials.
- Reduce considerably the design time, avoiding the trial-and-error process, reducing again, the need to create prototypes that increase costs and the use of excessive raw materials.
- Improvement of the technological competitiveness of companies.
- Development of new additive manufacturing technologies. New 3D printing methods involve a technological development that makes it possible to represent highly complex topologies. But is it really a method that reduces environmental impact?
  - On the one hand, by reproducing products according to users' needs, mass production is reduced and with it, the excessive exploitation of raw materials and the generation of waste. Another advantage of this technology is that it makes it possible to break with the current model of offshored production in other countries, giving way to a local, customized and small-scale production system. This not only implies a saving in the amount of material used, but also eliminates the cost of imports and fuel pollution from transportation.
  - On the other hand, 3D printing has a number of main enemies: premature obsolescence, printing time, high energy consumption and its current dependence on materials from fossil fuels (plastics). Despite being great enemies of climate change and excessive consumption of raw materials, they have some alternatives. The ease of generating

customized spare parts without the need to wait for the hypothetical manufacturer to provide the spare part is something in favor in the field of premature obsolescence. The alternative use of recycled plastics from renewable sources (PLA) or the recent use of metals in micro-objects are presented as mitigation of environmental impact [13]. In relation to time and energy consumption, these two variables are directly related and depend on the type of 3D printing system used.



## Chapter 9

# Conclusions

Symmetries are natural tools that simplify problems and unite the links between geometry and physical properties of microstructures. The study takes advantage of the power and versatility of topological optimization to distribute the material inside the unit cell, in order to solve the inverse problem using the minimum amount of material possible, obtaining lighter structures, with high competitiveness and performance. Therefore, during the study, the existing relationships between Bravais lattices, Voronoi lattices, plane group symmetries and symmetries in the elastic tensor are reviewed, thus linking the geometrical domain with the physical domain by means of the point groups. The importance of the two basic principles of periodic microstructure generation: Neumann principle and tessellation should also be emphasized.

Based on the first of these, an algorithm was developed capable of reproducing the geometry of Voronoi cells to cover a pre-established 2D space. This algorithm has been a key tool in this study in addition to the basic methodology to obtain materials with the desired symmetry properties together with the elastic tensor and the material distribution has been elaborated. This is why the practical development demonstrates the fulfillment of the basic principles, as well as the importance of carefully selecting the pre-process variables (unit cell and initial material distribution conditions). Therefore, it is advisable to use the relationships established by means of the table 3.1 to select the type of cell and to use an initial material distribution compatible with the type of elastic tensor elaborated. More specifically, obtaining an isotropic elastic tensor is more favorable when using a hexagonal lattice with initial material distribution with plane group symmetries of type  $p6mm$  or equivalent.

Of course, despite the conclusion of this research, new lines of future development 7 open the door for the development of new methods that extend the convergence capability of the tensor to the objective by solving the inverse problem.

# References

1. FERRER, Àlex; GIUSTI, Sebastián Miguel. Inverse homogenization using the topological derivative. *Engineering Computations*. 2021.
2. AMSTUTZ, S; GIUSTI, SM; NOVOTNY, AA; SOUZA NETO, EA de. Topological derivative for multi-scale linear elasticity models applied to the synthesis of microstructures. *International Journal for Numerical Methods in Engineering*. 2010, vol. 84, no. 6, pp. 733–756.
3. AUFRAY, Nicolas; ROPARS, Pierre. Invariant-based reconstruction of bidimensional elasticity tensors. *International Journal of Solids and Structures*. 2016, vol. 87, pp. 183–193.
4. PODESTÁ, Juan Manuel; MÉNDEZ, CM; TORO, Sebastian; HUESPE, Alfredo Edmundo. Symmetry considerations for topology design in the elastic inverse homogenization problem. *Journal of the Mechanics and Physics of Solids*. 2019, vol. 128, pp. 54–78.
5. ROSSI, Nestor; PODESTÁ, Juan M; BRE, Facundo; MÉNDEZ, Carlos G; HUESPE, Alfredo E. A microarchitecture design methodology to achieve extreme isotropic elastic properties of composites based on crystal symmetries. *Structural and Multidisciplinary Optimization*. 2021, vol. 63, no. 5, pp. 2459–2472.
6. BARBAROSIE, Cristian; TORTORELLI, Daniel A; WATTS, Seth. On domain symmetry and its use in homogenization. *Computer Methods in Applied Mechanics and Engineering*. 2017, vol. 320, pp. 1–45.
7. BENDSØE, Martin P. Optimal shape design as a material distribution problem. *Structural optimization*. 1989, vol. 1, no. 4, pp. 193–202.
8. STOLPE, Mathias; SVANBERG, Krister. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*. 2001, vol. 22, no. 2, pp. 116–124.
9. LUO, Jun; LUO, Zhen; CHEN, Liping; TONG, Liyong; WANG, Michael Yu. A semi-implicit level set method for structural shape and topology optimization. *J. Comput. Phys.* 2008, vol. 227, pp. 5561–5581.
10. TING, Thomas Chi-tsai; TING, Thomas Chi-tsai. *Anisotropic elasticity: theory and applications*. Oxford University Press on Demand, 1996. No. 45.
11. NYE, John Frederick et al. *Physical properties of crystals: their representation by tensors and matrices*. Oxford university press, 1985.
12. MEILLE, Sylvain; GARBOCZI, Edward J. Linear elastic properties of 2D and 3D models of porous materials made from elongated objects. *Modelling and Simulation in Materials Science and Engineering*. 2001, vol. 9, no. 5, p. 371.
13. ALL3DP. *Tipos de impresoras 3D*. 2022. Available also from: <https://all3dp.com/es/1/tipos-de-impresoras-3d-tecnologia-de-impresion-3d/>.