
NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF CIVIL ENGINEERING

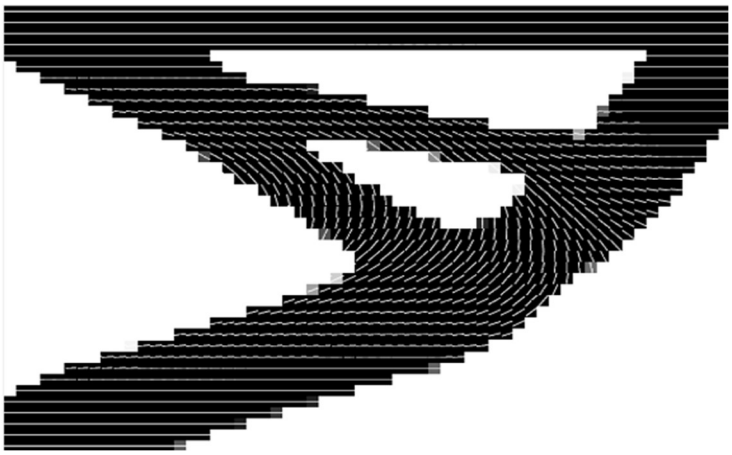
INSTITUTE OF STRUCTURAL ANALYSIS AND ANTISEISMIC REASEARCH

---

# MULTISCALE TOPOLOGY OPTIMIZATION

**DIPLOMA THESIS**

**CHRISTOS KOSTOPOULOS**



**Advisor**:

Vissarion Papadopoulos, NTUA Professor

Athens, July 2022

# Abstract

An efficient multiscale material and topology optimization scheme for fiber-reinforced structures is proposed. The goal was to find the structural topology and set of fiber orientations that minimized a fiber-reinforced structure's compliance using a multiscale framework.

Effective properties of the material were calculated using homogenization. The design domain was discretized in 4-node plane stress finite elements. The topology optimization problem was described by assigning a density variable and an orientation variable to each element using the solid isotropic material with penalization method. The fiber orientation was simulated by rotating the homogenized elasticity tensor appropriately. The homogenization took place only once, and it was not included in the optimization iterations.

A convolution filter was applied to both densities and orientations, aiming to avoid checkboard patterns, orientation discontinuities and convergence to local minima. The optimization problem was solved for densities and orientations concurrently using the gradient-based Method of Moving Asymptotes (MMA). The filter was applied adaptively to both densities and orientations in order to avoid dependency from initial conditions. Before the termination of optimization iterations elimination of intermediate densities was performed. The algorithm was applied to a carbon nanotube-reinforced structure and it proved to be stable and efficient, yielding checkerboard-free structures with smooth distribution of orientations. To conclude, we compared the results obtained by optimizing the topology and orientations to simple topology optimization. The proposed algorithm yielded superior and much more efficient results.

# Acknowledgements

I would like to acknowledge and give my warmest thanks to:

- My professor and thesis supervisor Vissarion Papadopoulos for his precious guidance and mentorship in the majority of my undergraduate years and my thesis. He ensured that I was conducting my thesis project under the most favorable circumstances. Additionally, he brought me in touch with plenty of very interesting research topics and guided me in enhancing my theoretical background and programming skills.

- PhD candidate Serafeim Bakalakos, who was tirelessly helping me with my C# code issues, answering theoretical background questions and improving my code's computational efficiency. He was my C# mentor making me a much more efficient coder and bringing me in touch with extremely useful programming tools. He also made me familiar with the values and procedures of research through inspiring conversations.

- My professor Savvas Triantafyllou for his precious guidance and mentorship. He gave me some of the most inspiring lectures in my undergraduate years imparting his passion for computational mechanics to me. He was always available to address all my thoughts and questions about plenty of computational mechanics topics.

- My family for their moral and physical support through my whole life and the values they imported to me. They gave me the opportunity to focus on my studies and thrive in every educational level.

# Contents

# Chapter 1

## 1 Introduction

Nowadays, the rapid advance of technology and the fact that knowledge is easily available through world wide web has increased the capabilities and competition in nearly every field of human activity. In this framework, efficiency is a key element to success for a researcher, an entrepreneur, or an employee. Each one of them endeavor to solve a problem with the minimum effort and cost (economic or computational). The most efficient solution will be the most commonly used. For instance, by formulating the most computationally efficient methodology to solve a problem, a researcher contributes vastly to his field since other research may integrate this efficient solution in their research activities. An entrepreneur who provides market with the most efficient service, greatly increases his profit margins as the cost for the provision of the service will be lower and the service price will be more enticing for the consumers.

Considering the significance of efficiency in our days, it is only rational for the Mathematical Optimization field to go through massive growth. Mathematical Optimization (or Mathematical Programming) is the branch of mathematics which investigates the optimal (the most efficient) way a problem can be solved. The resulting solution from optimization methodologies is mathematically proven to be the optimal for the formulated problem. Thus, the capability to properly express a specific problem with an optimization formulation is tantamount to the acquisition of the problem's optimal solution. Pursuing efficiency, researchers attempt to formulate Optimization problem expressions and algorithms for the problems they try to solve. Large companies are establishing simulation departments, in order to mathematically express the problems they endeavor to deal with and seek for the most efficient solution. Consequently, Mathematical Optimization field is radically growing with plenty of research being conducted on it.

The construction industry was included in these advances. Plenty of work and research projects has been dedicated in creating optimization formulations for the industry's problems. These efforts resulted in the formulation of three categories of optimization problems: Size Optimization, Shape Optimization and Topology Optimization. All those methodologies share

the same goal: The cheaper structure design which complies with all the safety, endurance and durability demands. Each methodology has its own way of approaching this goal. Size optimization algorithms optimize the section areas of a geometrically defined structure. Shape optimization algorithms optimize the shape of a structure with predefined topology (holes and boundaries). Topology optimization algorithms optimize the topology of a structure with the only information required being the design domain.

Each of those methodologies has its benefits and drawbacks and is formulated to address different needs. However, topology optimization has characteristics and capabilities which has gathered special interest from a great number of researchers. The fact that the only required geometrical information for the algorithm to operate is the design domain, creates two major benefits. Firstly, those algorithms are not restrained from a specific shape (size optimization) or topology (shape optimization) leading to a greater domain of available solutions. Secondly, structure's designer is not obliged to predefine the topology or the shape of the structure making topology optimization simpler to use in respect of selecting the design parameters. However, occasionally structure's designer may be sure about structure's geometry or topology. In that case, the use of topology optimization is an exaggeration and size or shape optimization could be implied. Because Topology Optimization solves a much more general problem than the other two types of optimizations, it is more computationally demanding. Thus, whenever the use of shape or size optimization is possible (when the shape or topology are strictly defined), we prefer those methodologies.

In my thesis, I decided to broaden the formulation of topology optimization with SIMP[3] in order to optimize the topology in multiple scales in composite fiber reinforced structures. Composite materials are used more and more in the construction industry. The opportunity researchers have, to modify the mechanical properties of a material according to the materials combined and the way they were merged, is making composite materials very enticing to them. The idea behind the proposed formulation is that we could not only optimize the topology of the structure in the macroscopic level, but also the fiber reinforcement orientation in the microscopic level. Different fiber orientations result to different constitutive laws to describe the behavior for a specific domain of the structure. If the structure will be discretized into finite elements, each finite element could have its own fiber orientation and as a result its own constitutive matrix. Considering this, I formulated the proposed optimization scheme in which the 'activity' (this determines the macroscopic topology) and the fiber orientations of each finite element is optimized. In order to model the composite behavior of the material, I used

homogenization and the optimization problem was solved using the ‘Method of Moving Asymptotes’ (MMA) algorithm.

The use of this multiscale topology optimization scheme in a variety of numerical examples had very encouraging results. The combination of optimal topology and fiber orientations resulted in excellent and very efficient use of the composite material. I hope that in the aforementioned current circumstances, this formulation may be a great contribution in the field of topology optimization.

# Chapter 2

# 2 Optimization

## 2.1 Introduction to Mathematical Optimization

### 2.1.1 The Optimization Problem

An optimization problem is the problem of finding the best solution from all the feasible solutions. This is achieved by maximizing or minimizing some function relative to some set, often representing a range of choices available in a certain situation. The function allows comparison of the different choices for determining which might be "best". A more formal expression of the problem is to find a solution in the feasible region which has the minimum value of the objective function.

This problem can be represented in the following way:

$$F(\mathbf{x}) \rightarrow \min,$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$$

$$x_i \in X_i\ ,\ i = 1, 2, \ldots, n$$

$$g_j(\mathbf{x}) \leq 0,\ j = 1, 2, \ldots, p$$

$$h_k(\mathbf{x}) = 0,\ k = 1, 2, \ldots, q$$

Where:

- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ denotes an array with length n which contains the design variables.
- $F(\mathbf{x})$: $\mathbb{R}^n \rightarrow \mathbb{R}$ denotes the objective function which is being minimized during the optimization.
- $X_i$ denotes the design variable domain.
- $g_j(\mathbf{x})$: $\mathbb{R}^n \rightarrow \mathbb{R}$ denotes the p in number inequality constraints.
- $h_j(x)$: $\mathbb{R}^n \rightarrow \mathbb{R}$ denotes the q in number equality constraints.

This is the expression of a minimization problem, which can be used slightly modified to describe a maximization problem as well, where the goal is to maximize the objective function. The maximization expression is exactly the same with the only difference being that the -F(**x**) function is minimized.

The methods used to solve optimization problems can either use objective function's derivative information or different methodologies in order to approach to the optimum solution. The first category of methods is called 'gradient based methods', while the second category is called 'gradient-free methods'. Examples of gradient methods are the gradient descent and the conjugate gradient. Some very useful gradient free tools are genetic algorithms, surrogate modeling and simulated annealing.

### 2.1.2 Design Variables

Design variables are the parameters that need to be specified in order to obtain the optimum solution in a mathematical optimization problem. For those specified values the objective function takes its minimum value. The $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ vector that consists of the specified design variables is called an optimum solution in respect of the problem's constraints. All the vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ whose design variables satisfy the constraints are considered a solution of the optimization problem, but we only seek for the optimum solution. If a solution satisfies all the problem's constraints, it is called a feasible solution. Otherwise, it is called an infeasible solution.

The Optimization problems can be differentiated according to their design variables type as follows:

- Problems with continuous design variables. The set of the possible values $X_i$ a continuous variable xi can take is an interval which is subset of $\mathbb{R}$ and has the form of $l_i \leq x_i \leq u_i$ , where li and ui denote the lower and the upper bound of the interval respectively. This is the most common type of design variables because in the vast majority of the optimization problems, we are able to choose the variable values for our parameters from continuous intervals.

- Problems with discrete design variables. This type of variables is commonly used in integer programming problems. Especially, when the design variables take only the binary values 0 and 1. There is a wide range of optimization problems where the integer

quantity of a parameter must be optimized. These parameters are expressed through discrete design variables.

- Problems whose discrete design variables can take values from finite sets $X_i$. This type of problems is extremely usual in the construction industry, where the values of construction material characteristics are given via industrial and regulative matrices. Even though physically the values of mechanical properties can be continuous, engineers tend to use discrete characteristic values.

- Problems with mixed types of design variables. These problems are commonly converted into problems with continuous design variables. For instance, the optimal design of a concrete beam is such a problem. The section dimensions are continuous design variables, whereas the rebar bar sizes and concrete stress capacity are discrete design variables.

The selection of the appropriate design variables to describe a problem is an extremely important part of the optimization process. An accurate selection leads to much more efficient optimization schemes. Attention must be paid in order to avoid dependence between design variables, otherwise the system's complexity will be unreasonably increased. Furthermore, the use of a variable whose variance has little effect in the objective function increases the computational cost in vain. Those variables are called secondary variables and they are linked with the primary variables through linear expressions (design variable linking). Sensitivity analysis is a useful tool for the design variable selection process, which addresses the aforementioned issue. Another useful methodology for improving optimization algorithms' efficiency is the 'design variable grouping'. By performing design variable grouping we decrease the computational cost and deal with variables with little effect in the objective function (instead of considering them as secondary).

### 2.1.3 Objective Function

Objective function is the criteria with which we can evaluate a solution and compare it to others. In an optimization problem our goal is to find the values of design variables which lead to the minimum value of objective function and satisfy the constraints. The $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ vector containing those values is the 'optimum feasible solution'. Objective function is very commonly a cost function.

Problem with one objective function or with multiple objective functions which are minimized simultaneously are called 'Single Objective Optimization'. However, it is very common to have multiple objective functions such as the minimization of one increasing the value of the others. For instance, assuming that we want to maximize the performance of a formula 1 car. We can simplistically formulate two objective functions: Straight-line speed and Corner speed and consider downforce as a design variable. Higher downforce leads to an increase of Corner speed but deteriorates the speed in the straights, while a lower downforce setup could result in higher speed in the straights. These problems are called 'Multi Objective Optimization' and they lack a univocal solution. The goal is to detect a set (Pareto Optimal Solutions) of optimum solutions which are considered equally efficient. None of the solutions outside of this set has a better effect on all the objective functions together.

Optimization problems are classified regarding to the type of their objective functions. Those can be either linear or nonlinear separating the mathematical programming in linear and nonlinear programming. Linear programming problems can be solved easier than the nonlinear ones.

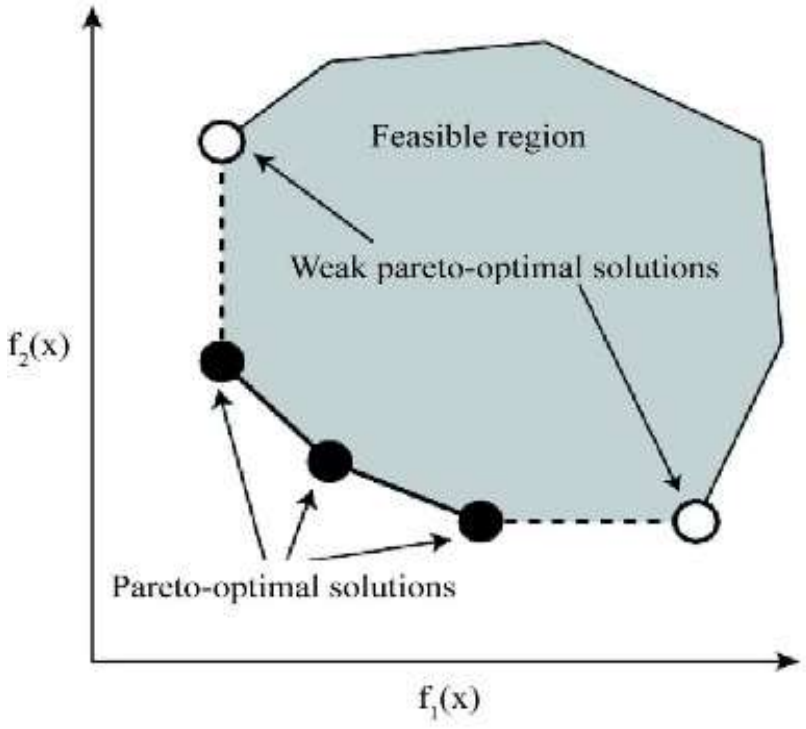



**Figure 2.1**
Visualization of Pareto-optimal solutions.

Another important objective function's characteristic is its convexity. If the function is convex, the detection of the optimum solution, if there is one, is quicker and certain. If the function is non-convex, it is possible for the optimization algorithm to converge in a non-optimum solution. To address this problem the problems complexity is increased and the detection of the optimum solution is uncertain.

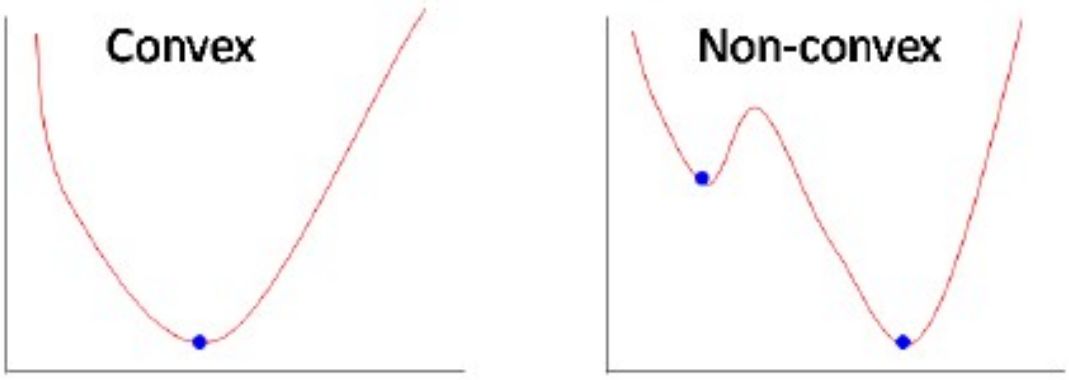



**Figure 2.2** Convex and Non-convex Objective functions

### 2.1.4 Global and Local Minimums

An objective function can have two types of minimums, global and local ones. Global minimum is the minimum value of the function on its whole domain and local minimum is the minimum value in a specific subdomain. Local minimums are always greater than the global minimum of the objective function.

In order to formally define the two minimums, we assume the objective function $F(\mathbf{x})$: $\mathbb{R}^n \rightarrow \mathbb{R}$. Then:

- $\mathbf{x}^* \in \mathbb{R}^n$ is called a **global minimum point** of F over $\mathbb{R}^n$ if $F(\mathbf{x}) \geq F(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}^n$. Then, $F(\mathbf{x}^*)$ is called the **global minimum value** of the objective function.

- $\mathbf{x}^* \in \mathbb{R}^n$ is called a **strict global minimum point** of F over $\mathbb{R}^n$ if $F(\mathbf{x}) > F(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}^n$. Then, $F(\mathbf{x}^*)$ is called the **strict global minimum value** of the objective function.

- $\mathbf{x}^* \in \mathbb{R}^n$ is called a **local minimum point** of F over $\mathbb{R}^n$ if there exists r > 0 for which $F(\mathbf{x}) \geq F(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}^n \cap B(\mathbf{x}^*, r)$. Then, $F(\mathbf{x}^*)$ is called a **local minimum value** of the objective function.

- $\mathbf{x}^* \in \mathbb{R}^n$ is called a **strict local minimum point** of F over $\mathbb{R}^n$ if there exists r > 0 for which $F(\mathbf{x}) > F(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}^n \cap B(\mathbf{x}^*, r)$. Then, $F(\mathbf{x}^*)$ is called a **strict local minimum value** of the objective function.

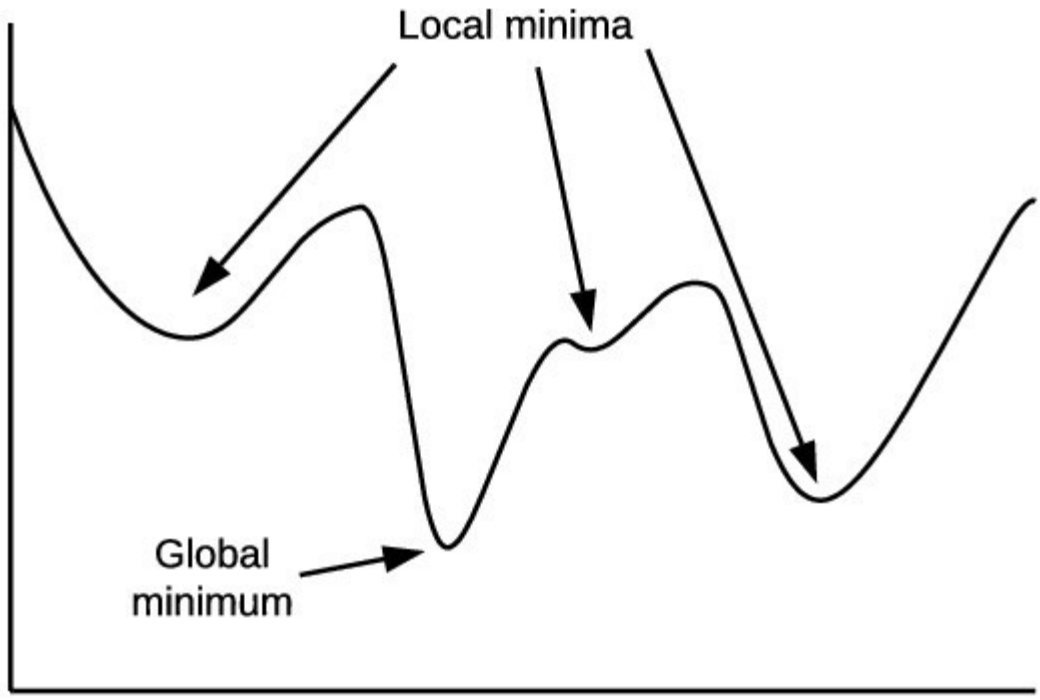



**Figure 2.3** Objective function with four local minimums from which one is a global minimum.

A global minimum is always considered as a local minimum, but the opposite does not apply. The aforementioned definitions apply only in the case of unconstraint optimization. In other case, they apply in the subdomain of the feasible solutions instead of the whole design variables domain.

In general, it is unknown whether an objective has a local or a global minimum. In the specific occasion where the objective function is continuous, the feasible design domain non-empty, closed and bounded the existence of a global minimum can be mathematically proven. In any

other case, it is impossible to predict whether a global minimum exists or not before the optimization. The feasible design domain is considered to be empty if there are numerous constraints which contradict with each other. In such an occasion, some constraints must be eliminated. Furthermore, feasible design domain is considered to be closed and bounded if the constraint functions are continuous and the inequalities are not strict.

The goal of an optimization algorithm is to detect the global minimum of the objective function. If the objective function is convex, there is only one global minimum which can be easily detected through the optimization method. Otherwise, the non-convex objective functions may have multiple local minimums and the optimization algorithm could converge to one of them resulting in a solution which is not the optimal. Therefore, the convexity of the objective function has a significant impact in the optimization's algorithm efficiency.

### 2.1.5 Constraint functions

From the optimization problem's mathematical definition, it is obvious that design variables have a specific domain from which they can receive their values. However, a combination of design variable values may result in irrational results. In order to prevent that, constraint functions are used to define the feasible domain of the problem. Constraint expressions can be either equalities or inequalities. A design variable selection which satisfies all of the constraint is called "feasible design" and a selection outside the feasible domain is called "non-feasible design". The goal in optimization problems is to acquire the optimum solution in the feasible design.

A very common equality constraint in optimization of structures problems is the equilibrium. The most important criterion for the selection of design variable values must be the capacity of the structure to carry the assigned loads. This criterion is inserted in the problem formulation as an equality constraint expression.

Inequality constraint functions can be active, inactive or unsatisfied. If $g(\mathbf{x}) < 0$ (where $g(\mathbf{x})$ is an inequality constraint function, according to the formulation) then the constraint is inactive. On the contrary, if $g(\mathbf{x}) = 0$ the constraint is considered to be an active one. In the case of $g(\mathbf{x}) > 0$, the constraint is unsatisfied. Similarly, the equality constraints can be either active or unsatisfied. If $h(\mathbf{x}) = 0$ (where $h(\mathbf{x})$ is an equality constraint function, according to the formulation) the equality constraint is active and if $h(\mathbf{x}) \neq 0$ the equality constraint is unsatisfied.

Special mention should be made to the common problem of over-constrained formulations. This is the most common reason for an optimization algorithm not to reach in a solution. If many constraints are applied simultaneously the feasible domain is an empty set. This means that there is not such a solution satisfying all the formulated criteria and that some constraints must be eliminated so that we obtain a feasible domain of solutions. Generally, equality constraints decrease the feasible domain much more than the inequality constraints. Let say that an optimization algorithm seeks a solution in a n-dimensional initial feasible domain. Inequality constraints decrease the n-dimensional feasible domain, while an equality constraint dictates a (n-1)-dimensional curve on which the solution must be found. This reduces the feasible domain's dimension leading to much fewer options for the design variable values selection. If we considered an optimization problem formulated in a 3-dimensional space, we can visualize the aforementioned argument. In the following visualization, the initial feasible domain is the whole $\mathbb{R}^3$ space. An inequality constraint reduces the feasible domain in such of a cylinder volume. An equality constraint, represented by a $\mathbb{R}^2$ surface, reduces the feasible domain in the cylinder's and volume's intersection, which belongs to the $\mathbb{R}^2$ space.

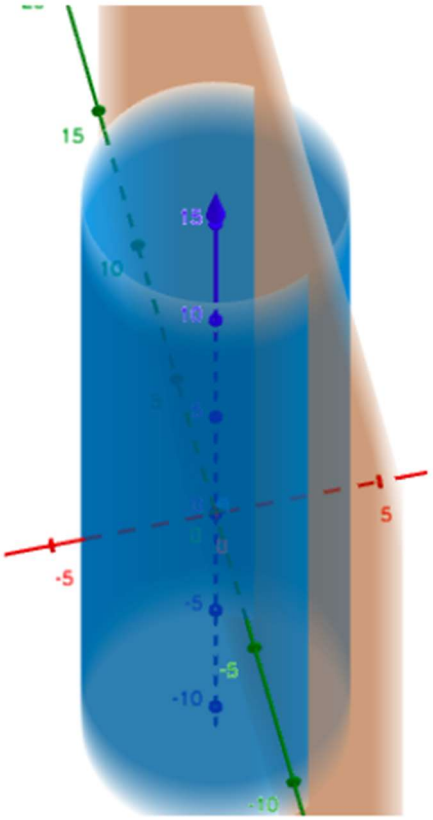



**Figure 2.4** Visualization of equality and inequality constraint functions in the $\mathbb{R}^3$ space.

### 2.1.6 Duality in mathematical optimization

In mathematical optimization theory, duality is the principle that optimization problems may be viewed from either of two perspectives, the primal problem or the dual problem. If the primal is a minimization problem, then the dual is a maximization problem (and vice-versa). If the primary problem has n design variables and m constraints, the dual problem has n constraints and m design variables. Solving any of those problems leads to the solution of the other, offering the option to solve the easiest of them.

Lagrangian duality is a very useful tool when formulating a dual problem to our primary problem. The basic idea in Lagrangian duality is to take the constraints of an optimization problem into account by augmenting the objective function with a weighted sum of the constraint functions.

## 2.2 Structural Design Optimization

In structural design optimization problems, a common goal is to minimize the cost of the construction. Because of this the objective function is an expression of structure's weight. However, the design variable and the constraint function selection may vary leading to different forms of structural optimization problems. The most common structural design optimization problems are:

- Size Optimization
- Shape Optimization
- Topology Optimization

### 2.2.1 Size Optimization

In a size optimization problem, the geometry of the structure (joints, member's lengths) is defined. Moreover, the load and restraint conditions are selected and cannot be changed through the optimization process. The goal is to specify the minimum cross-sectional dimensions while complying with specific structural constraints and mainly the equilibrium condition. Thus, the design variables are the structure members' cross-sectional dimensions. The objective function is the total weight of the structure which is easily calculated given the members' cross-sectional dimensions. A common example of a constraint function is to control

members, so that their stress is less than their material yield stress. The optimal design of a truss construction is a common size optimization problem.

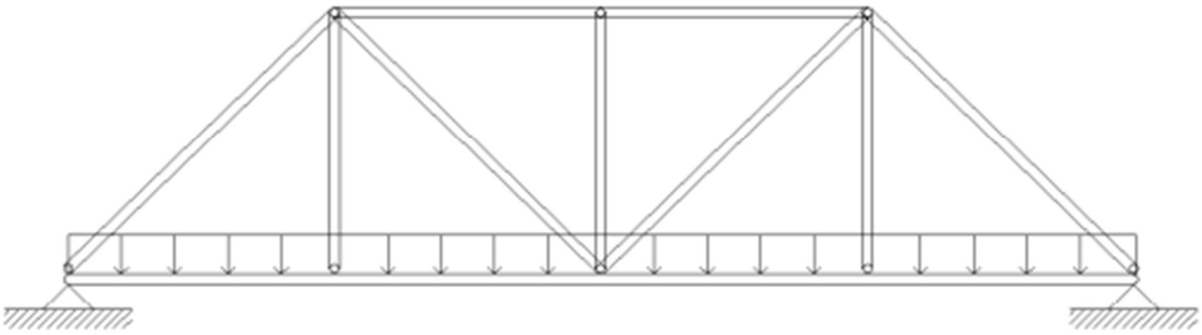


**Figure 2.5** Size Optimization problem
Given Geometry design, Load assignments, restraint assignments and initial selection of cross-sectional dimensions.

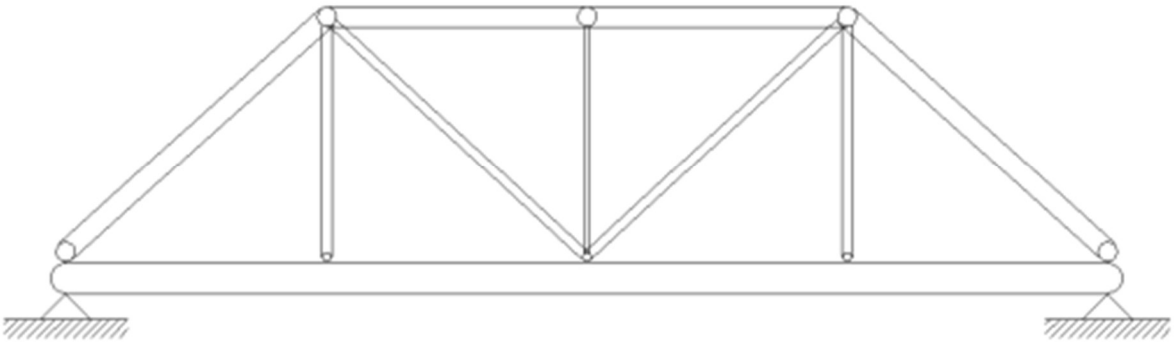


**Figure 2.6** Size Optimization's results
Optimal cross-sectional dimensions.

### 2.2.2 Shape Optimization

Shape optimization problems are size optimization problems with the structure's geometric characteristics (joints, member lengths) being design variables. The goal is to find a geometry of the structure which minimizes a given function (such as the weight of the structure) and yet simultaneously satisfies specific constraints. The topology of the structure is predefined, the algorithm optimizes only the geometry.

To solve any shape optimization problem would ideally mean to find the minimum of a specific cost function over a set of admissible domains. However, it turns out that very few adequate existence results are available. In general, the existence results for such problems are obtained, provided that some unrealistic constraints are imposed on the family of admissible domains.

Shape optimization is more complex than the pure sizing optimization. Since the shapes are continuously changing in the design process, careful consideration has to be paid to describe the changing boundary shape, to maintain an adequate finite element mesh, to enhance the accuracy of the sensitivity analysis, to impose proper constraints and to utilize existing optimization methods to solve the shape optimization problems.

An important characteristic of shape optimization is that is not immediately applicable. As it happens in the case of size optimization a topology and an initial geometry must be predefined in order to initialize the shape optimization process. The optimal shape of the boundaries is defined by the optimization, but the algorithm must be provided with an initial form of the boundaries. Especially for the internal boundaries, their number and initial position must be predefined.

A simple shape optimization problem is the pursuit of a beam's holes optimum design. Restraint and load conditions are given as well as an initial description of the internal boundaries (holes).

**Figure 2.7** Shape Optimization problem
Given Load assignments, restraint assignments, topology and initial external and internal boundaries (Geometry)

**Figure 2.8** Shape Optimization's results
Optimal Geometry

### 2.2.3 Topology Optimization

Topology optimization aims to determine the optimal form and layout of a structure with nothing else given but its external boundaries, the load and restraint conditions. A strict description of the domain in which the structure will be designed, and the load-restraint conditions are sufficient for this method to create itself the structure's optimal shape. This gives topology optimization a huge advantage in comparison to shape optimization where a description of structure's shape must be provided. Apart from the fact that topology optimization requires less data to return the optimal solution, it also has a larger domain of feasible designs as it is not bounded to the optimization of a specific shape. As a result, this method investigates many different layouts.

Contrary to size and shape optimization where design variables are parameters defining the shape and topology (joints, holes, member lengths), in topology optimization the only design variable becomes the material distribution which can be described in the $\mathbb{R}^2$ space by a function of the following form:

$$d(x,y) = \begin{cases} 1 & ,if\ material\ exsists\ at\ (x,y) \in \mathbb{R}^2 \\ 0, & if\ material\ does\ not\ exist\ at\ (x,y) \in \mathbb{R}^2 \end{cases}$$

However, usually in topology optimization methods $d(x,y)$ function's set of values is not discrete, taking values in the whole [0, 1] domain. The issue with this formulation is that values in (0, 1) lack of physical meaning as there is no such state as 'partially existing'. In real-life problems, in a specific coordinate material can either exist or not exist. In this type of methods this problem is handled with appropriate design variable filtering. However, some researchers accept values between 0 and 1 which can be interpreted as a material mesostructure with holes. Obviously, values higher than 1 and less than 0 lack of physical meaning as well.

Another common characteristic of the topology optimization methods is that the space in which the structure in formed is not the continuous $\mathbb{R}^n$ but a discretized domain (usually Finite Element Methods are used). Consequently, assuming that design space is a discretized with finite elements domain, $d(x,y)$ takes the following form:

$$d(i) = \begin{cases} 1, & if\ material\ exsists\ in\ the\ i-indexed\ finite\ element \\ 0, & if\ material\ does\ not\ exsists\ in\ the\ i-indexed\ finite\ element \end{cases}$$

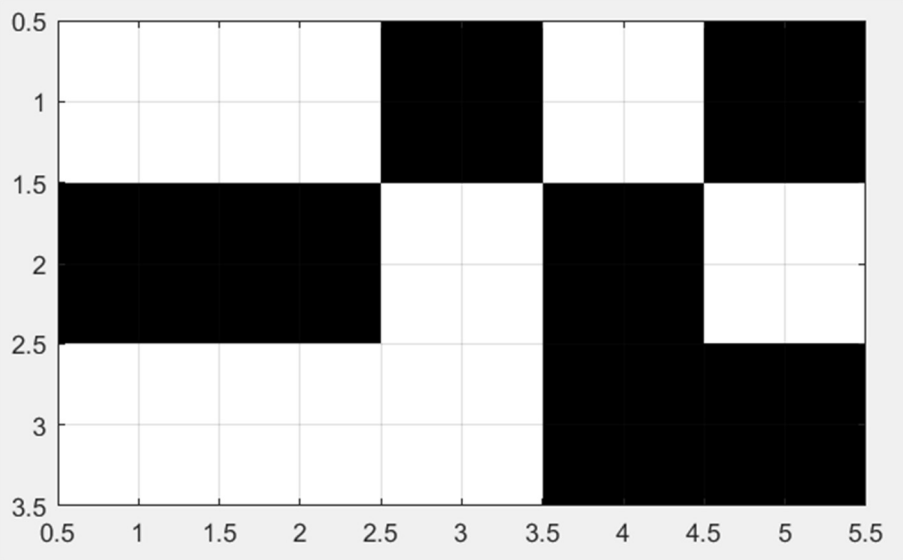



**Figure 2.9** $d(i)$ function with discrete set of values

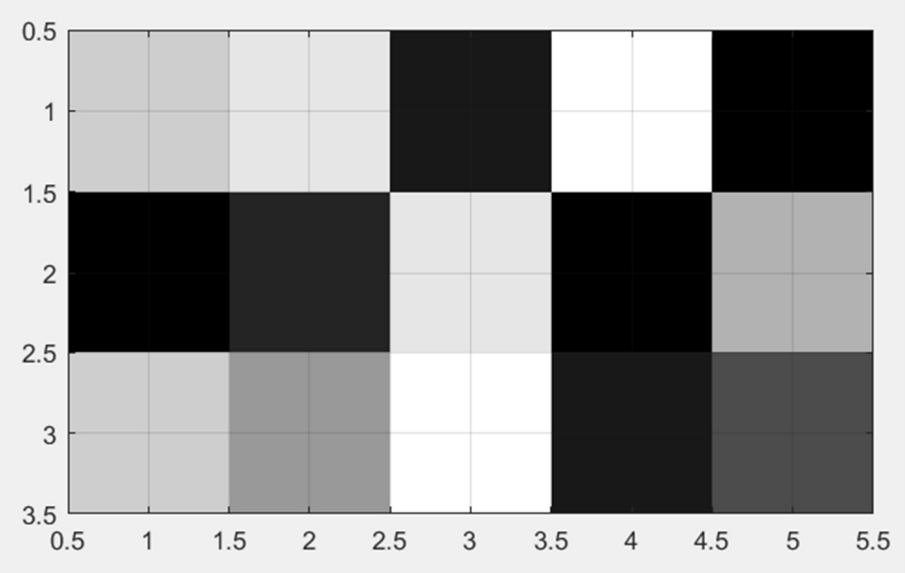



**Figure 2.10** $d(i)$ function with continuous set of values

The goal in the topology optimization methods is to determine a $d(i)$ function (this function is eventually the design variable) which minimizes the objective function. Initially, material is distributed evenly in the design domain and as a result $d(i)$ is constant in the whole domain. Then, material distribution is altered according to which finite element is more effective for the structure to carry the assigned loads. This way method finds the optimum material distribution $d(i)$. Typically, it is asked for objective function to be subject to a volume fraction constant.

**Figure 2.11** In the beginning of topology optimization methods, material is evenly distributed in the design

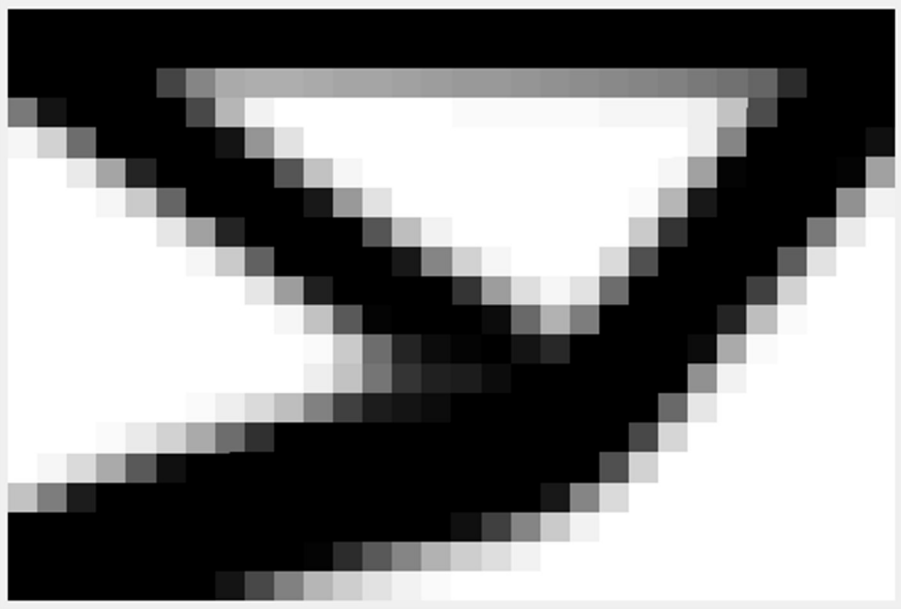


**Figure 2.12** Optimal material distribution after performing topology optimization in a 60x40 finite element discretized domain

An important drawback of topology optimization is that the output structure may be difficult or very expensive to be physically implemented. This happens because too complex and detailed topologies are usually the optimal solution to a problem. However, the growing technic of 3-D printing helps in topology optimization output implementation. As this technology improves, topology optimization application domain will be broadened.

**Figure 2.13** 3-D printed Topology Optimized chair

The most famous topology optimization methodologies are:

- Homogenization method (Bendsoe and Kikuchi, 1988)
- Solid Isotropic Material with Penalization (SIMP) (Bendsoe 1989, Buhl et al. 2000, Rozvany 2001)
- Simultaneous Analysis and Design (SAND) (Haftka 1985, Orozco 1997)
- Evolutionary Structural Optimization (ESO) (Xie and Stephen 1993)

## 2.3 Solid Isotropic Material with Penalization (SIMP)

### 2.3.1 Introduction and formulation

The distribution of the material by parameterization of design domain was first proposed by Bendsoe at the end of 1980's. The Solid Isotropic Material with Penalization (SIMP) method receive much attention in the early 1990's. The SIMP method sometimes also referred as

density method, works by keeping a fixed finite element discretization. Each Finite Element is then associated with a density function ρ(x) whose values lie between 0 and 1. 0 denotes a void and 1 denotes a solid.

Suppose we consider the material properties of a particular element as a function of its density, such that element densities can be interpreted as design variables which adjust the performance of the design. Let say we have a solid (ρ(x) =1) with the material property Young's modulus denoted by $E_s$. Then, the Young's modulus of elements with intermediate densities is given by $E_i = E_s\, \rho_i(x)$, where the subscript i, denotes a particular element. The intermediate densities can be considered as 'artificial material' which cannot be realized in practice. Therefore, it is needed to introduce a technique to penalize the intermediate densities such that it is represented as 0 or 1 for each element. This means the final design domain will contain only solids and voids. Thus, the fundamental equation that characterizes a SIMP approach with the penalty factor incorporated is given as:

$$E_i = \rho_i(x)^p \cdot E_s \; ; Vol = \int_{R^3} \rho^x dx$$

Where

$E_i$ : Young's modulus of the i$^{th}$ element in the design domain,

$\rho_i$ : Density parameter for the i$^{th}$ element,

$p$ : Penalization parameter for intermediate densities (usually, p ≥ 3),

$E_s$ : Solid material young's modulus,

$Vol$ : Total volume of distributed material in the design domain

$R^3$ : Three-dimensional design region

$x$ : The coordinate points in space for the centroid of a material element.

### 2.3.2 Topology Optimization with SIMP formulation

To mathematically express topology optimization with SIMP, we need to define the design variables, objective function and constraints of the problem. The design variable of the problem

is the density function ρ(x) ∈ [0, 1] for the discretized design domain. In this type of problems, we try to minimize structure's compliance and as a result, objective function is a compliance function with the penalization term for the intermediate densities integrated. In terms of constraints, there are 2 of them in this formulation.

The first one is the equilibrium condition, which is crucial as it secures that the structure will not collapse under the assigned loads. Considering that the structure's compliance is minimized instead of weight or cost, as it happens very commonly in structural optimization methodologies, there is nothing in the formulation which ensures us that the optimized result will be cheap.

This is why the second constraint sets a volume fraction equal to a predefined value. This means that the output structure's volume (and as a result the weight) has to be less than this predefined value multiplied with the initial volume (weight). Consequently, we can predefine how lighter the output structure will be by setting volume fraction equal to the wanted value.

Considering all the aforementioned ideas, topology optimization using SIMP is mathematically formulated as follows:

$$\min_{x}: \qquad c(\boldsymbol{x}) = \boldsymbol{U^T} \cdot \boldsymbol{K} \cdot \boldsymbol{U} = \sum_{i=1}^{N} E_i(x_i) \cdot u_i^T \cdot k_0 \cdot u_i$$

$$subject\ to: \qquad \frac{V(\boldsymbol{x})}{V_0} = f$$

$$\boldsymbol{K} \cdot \boldsymbol{U} = \boldsymbol{F}$$

$$\mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}$$

Where

$c$ : compliance

$\boldsymbol{U}$ : global displacement vector

$\boldsymbol{F}$ : force vector

$\boldsymbol{K}$ : global stiffness matrix

$u_i$ : element displacement vector

$k_0$ : element stiffness matrix

$\boldsymbol{x}$ : vector of design variables (element densities ρ)

$N$ : number of elements used to discretize domain

$V$ : material volume

$V_0$ : design domain volume

$f$ : prescribed volume fraction

## 2.4 Method of Moving Asymptotes (MMA)

### 2.4.1 Introduction to MMA

The Method of Moving Asymptotes is a method of structural optimization introduced by Krister Svanberg in 1987. In each step of an iterative process, a strictly convex approximating subproblem is generated and solved. The generation of these subproblems is controlled by so called 'moving asymptotes', which may both stabilize and speed up the convergence of the general process.

In order to describe this method, we consider an optimization problem of the general form:

P: minimize

$$F_o(\mathbf{x}) \qquad (\mathbf{x} \in \mathbb{R}^n)$$

Subject to,

$$F_i(\mathbf{x}) \leq \hat{F}_i\,, \qquad \text{for } i = 1, 2, \ldots, m$$

and

$$\underline{x}_j \leq x_j \leq \overline{x}_j, \qquad \text{for } j = 1,\ldots, n$$

where $\mathbf{x} = [x_i,\ldots, x,]^T$ is the vector of design variables, $F_o(\mathbf{x})$ is the objective function, typically the structural weight, $F_i(\mathbf{x}) < \hat{F}_i$ are behavior constraints, typically limitations on stresses and displacements, $\underline{x}_j$ and $\overline{x}_j$ are given lower and upper bounds ('technological constraints') on the design variables.

## 2.4.2 General description of the method

A well-established general approach for attacking such problems is to generate and solve a sequence of explicit subproblems according to the following iterative scheme:

**Step 0.** Choose a starting point $x^{(0)}$ and let the iteration index k = 0.

**Step I.** Given an iteration point $\mathbf{x}^{(k)}$, calculate $F_i(\mathbf{x}^{(k)})$ and the gradients $\nabla F_i(\mathbf{x}^{(k)})$ for i = 0,1, ..., m.

**Step II.** Generate a subproblem $P^{(k)}$ by replacing, in P, the (usually implicit) functions $F_i$ by approximating explicit functions $F_i^{(k)}$, based on the calculations from **Step I**.

**Step III.** Solve $P^{(k)}$ and let the optimal solution of this subproblem be the next iteration point $\mathbf{x}^{(k+1)}$. Let k = k + 1 and go to **Step I**.

The process is interrupted when some convergence criteria are fulfilled, or simply when the user is satisfied with the current solution $\mathbf{x}^{(k)}$.

MMA follows the general approach (**Steps 0-III**) described in the previous section. Thus, to define the method, it must be described:

a) how the approximation functions $F_i^{(k)}$ should be defined.
b) how the subproblem $P^{(k)}$ should be solved, given that the $F_i^{(k)}$ have been chosen.

### 2.4.3 Definition of the approximation functions

In order to answer to the first question, we assume the iteration point $\mathbf{x}^{(k)}$ (at iteration k), and the values of the parameters $L_j^{(k)}$ and $U_j^{(k)}$ are chosen, for j = 1,. . . , n, such that

$$L_j^{(k)} < x_j^{(k)} < U_j^{(k)}$$

Then, for each i = 0, 1,. . . m, $F_i^{(k)}$ is defined by

$$F_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^{n} \left( \frac{p_{ij}^{(k)}}{\mathrm{U}_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - \mathrm{L}_j^{(k)}} \right)$$

Where

$$\mathrm{p}_{ij}^{(k)} = \begin{cases} \left(U_j^{(k)} - x_j^{(k)}\right)^2 \cdot \partial F_i / \partial x_j, & \text{if } \partial F_i / \partial x_j > 0 \\ 0, & if \ \partial F_i / \partial x_j \leq 0 \end{cases}$$

$$\mathrm{q}_{ij}^{(k)} = \begin{cases} 0, & \text{if } \partial F_i / \partial x_j \geq 0 \\ -\left(x_j^{(k)} - L_j^{(k)}\right)^2 \cdot \partial F_i / \partial x_j, & if \ \partial F_i / \partial x_j < 0 \end{cases}$$

$$r_i^{(k)} = F_i\left(\boldsymbol{x}^{(k)}\right) - \sum_{j=1}^{n} \left( \frac{p_{ij}^{(k)}}{\mathrm{U}_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - \mathrm{L}_j^{(k)}} \right)$$

All derivatives $\partial F_i / \partial x_j$ are evaluated at $\mathbf{x} = \mathbf{x}^{(k)}$. Since $\mathrm{p}_{ij}^{(k)} and \ \mathrm{q}_{ij}^{(k)} \geq 0, F_i^{(k)}$ is a convex function. In particular, at $\mathbf{x} = \mathbf{x}^{(k)}$ the second derivatives are given by

$$\frac{\partial^2 F_i^{(k)}}{\partial x_j^2} = \begin{cases} \dfrac{2\,\partial F_i/\,\partial x_j}{\mathrm{U}_j^{(k)} - x_j^{(k)}}, & if\ \ \partial F_i/\,\partial x_j > 0 \\ -\dfrac{2\,\partial F_i/\,\partial x_j}{x_j^{(k)} - \mathrm{L}_j^{(k)}}, & if\ \ \partial F_i/\,\partial x_j < 0 \end{cases}$$

Thus, the closer $\mathrm{L}_j^{(k)}$ and $U_j^{(k)}$ are chosen to $x_j^{(k)}$, the larger become the second derivatives, the more curvature is given to the approximating functions $F_i^{(k)}$, and the more conservative becomes the approximation of the original problem. If, instead, $L_j^{(k)} = 0$ and $\mathrm{U}_j^{(k)} = +\infty$ for all j, then the $F_i^{(k)}$ becomes linear in the variable $x_j\ if\ \partial F_i/\partial x_j \geq 0$ and strictly convex in $x_j\ if\ \partial F_i/\partial x_j < 0$. Now, with the approximating functions $F_i^{(k)}$defined, the following subproblem, called $\mathrm{P}^{(k)}$ is obtained:

$\mathrm{P}^{(k)}$: minimize

$$\sum_{j=1}^{n} \left( \frac{p_{oj}^{(k)}}{\mathrm{U}_j^{(k)} - x_j^{(k)}} + \frac{q_{oj}^{(k)}}{x_j^{(k)} - \mathrm{L}_j^{(k)}} \right) + r_o^{(k)}$$

Subject to

$$\sum_{j=1}^{n} \left( \frac{p_{ij}^{(k)}}{\mathrm{U}_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - \mathrm{L}_j^{(k)}} \right) + r_i^{(k)} \leq \hat{F}_i, \quad for\ i = 1, \ldots, m$$

And

$$max\left\{\underline{x}_j, \mathrm{a}_j^{(k)}\right\} \leq x_j \leq min\left\{\overline{x}_j, \mathrm{b}_j^{(k)}\right\}, \qquad for\ j = 1, \ldots, n$$

Here, the parameters $\mathrm{a}_j^{(k)}$ and $\mathrm{b}_j^{(k)}$ are 'move limits' which are probably not very crucial. However, to avoid the possibility of any unexpected 'division by zero' while solving the subproblem, $\mathrm{a}_j^{(k)}$ and $\mathrm{b}_j^{(k)}$ should at least be chosen such that

$$L_j^{(k)} < a_j^{(k)} < x_j^{(k)} < b_j^{(k)} < U_j^{(k)}$$

### 2.4.4 Subproblem solution

In order to complete the definition of MMA, the subproblems must be solves. To simplify notation, we will from now on suppress the iteration index k on the coefficients in the subproblem. Further, we will write $a_j$ instead of $max\left\{\underline{x}_j, a_j^{(k)}\right\}$, $b_j$ instead of $min\left\{\overline{x}_j, b_j^{(k)}\right\}$, and $b_i$ instead of $\hat{F}_i - r_i$. Then, the problem $P^{(k)}$ may be written as follows:

$P^{(k)}$: minimize

$$\sum_{j=1}^{n}\left(\frac{p_0 j}{U_j - x_j} + \frac{q_0 j}{x_j - L_j}\right) + r_o$$

Subject to

$$\sum_{j=1}^{n}\left(\frac{p_{ij}}{U_j - x_j} + \frac{q_{ij}}{x_j - L_j}\right) \leq b_i, \qquad for\ i = 1, \dots, m$$

And

$$a_j \leq x_j \leq b_j, \qquad for\ j = 1, \dots, n$$

Where $p_{ij}, q_{ij} \geq 0\ and\ L_j < a_j \leq b_j < U_j$.

$P^{(k)}$ is a convex, separable problem. Therefore, a dual method analogous to the ones described in reference[1] could be used for its solution. The Lagrangian function corresponding to $P^{(k)}$ is given by

[1] *C. Fleury, 'Structural weight optimization by dual methods of convex programming', Int. j. numer. Methods eng. 14, (1979).*

$$l(x,y) = F_o(\boldsymbol{x}) + \sum_{i=1}^{m} y_i \cdot F_i^{(k)}(x)$$

Which, after trivial calculations, equals

$$r_o - \boldsymbol{y^T} \cdot \boldsymbol{b} + \sum_{j=1}^{n} l_j(x_j, \boldsymbol{y})$$

Where

$$\boldsymbol{b} = (b_1, \dots, b_m)^T, \qquad \boldsymbol{p}_j = (p_{1j}, \dots, p_{mj})^T, \qquad \boldsymbol{q}_j = (q_{1j}, \dots, q_{mj})^T,$$
$$\boldsymbol{y} = (y_1, \dots, y_m)^T$$

and

$$l_j(x_j, \boldsymbol{y}) = \frac{(p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})}{(U_j - x_j)} + \frac{(q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})}{(x_j - L_j)}$$

**y** is the vector of Lagrange multipliers or 'dual variables'.

Next, the 'dual objective function' W is defined, for **y** ≥ 0 as follows:

$$W(\boldsymbol{y}) = min_x \{ l(\boldsymbol{x}, \boldsymbol{y}); a_j \le x_j \le bj \quad for\ all\ j \}$$

$$= r_o - \boldsymbol{y^T} \cdot \boldsymbol{b} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{W_j}(\boldsymbol{y})$$

Where

$$\boldsymbol{W_j}(\boldsymbol{y}) \ = \ min_{x_j} \left\{ l_{j(x_j, \boldsymbol{y})}; a_j \le x_j \le b_j \right\}$$

We may assume that at least one of $p_{oj} + y^T \cdot p_j$ or $q_{oj} + y^T \cdot q_j$ is strictly positive. Then, the derivative of $l_j(x_j, \boldsymbol{y})$ with respect to $x_j$ is given by

$$l_j'(x_j, \boldsymbol{y}) = \frac{(p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})}{(U_j - x_j)^2} - \frac{(q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})}{(x_j - L_j)^2}$$

And the second derivative of $l_j(x_j, \boldsymbol{y})$ with respect to $x_j$ is given by

$$l_j''(x_j, \boldsymbol{y}) = \frac{2 \cdot (p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})}{(U_j - x_j)^3} + \frac{2 \cdot (q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})}{(x_j - L_j)^3}$$

Since $l_j''(x_j, \boldsymbol{y})$ is strictly positive, the derivative $l_j'(x_j, \boldsymbol{y})$ is strictly increasing in $x_j$, and we may reach to the following conclusions:

1. If $l_j'(a_j, \boldsymbol{y}) \geq 0$ then $x_j(\boldsymbol{y}) = a_j$
2. If $l_j'(b_j, \boldsymbol{y}) \leq 0$ then $x_j(\boldsymbol{y}) = b_j$
3. If $l_j'(a_j, \boldsymbol{y}) < 0$ and $l_j'(b_j, \boldsymbol{y}) > 0$ then $x_j(\boldsymbol{y})$ is the unique solution of the equation $l_j'(x_j, \boldsymbol{y}) = 0$.

We can verify that his unique solution is given by

$$x_j(\boldsymbol{y}) = \frac{(p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})^{(1/2)} \cdot L_j + (q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})^{(1/2)} \cdot U_j}{(p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})^{(1/2)} + (q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})^{(1/2)}}$$

since there is an explicit expression for $x_j(\boldsymbol{y})$, the dual objective function W(**y**) can also be expressed explicitly as follows:

$$W(\boldsymbol{y}) = r_o - \boldsymbol{y^T} \cdot \boldsymbol{b} + \sum_{j=1}^{n} \frac{(p_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{p_j})}{(U_j - x_j(\boldsymbol{y}))} + \frac{(q_{oj} + \boldsymbol{y^T} \cdot \boldsymbol{q_j})}{(x_j(\boldsymbol{y}) - L_j)}$$

Furthermore, the derivatives of W(y), with respect to the dual variables $y_i$, are given by:

$$\frac{\partial W}{\partial y_i} = -b_i + \sum_{j=1}^{n} \left( \frac{p_{ij}}{U_j - x_j(\boldsymbol{y})} + \frac{q_{ij}}{x_j(\boldsymbol{y}) - L_j} \right)$$

The dual problem corresponding to $P^{(k)}$ is the problem of maximizing $W(\boldsymbol{y})$ over the set of all **y** such that **y** ≥ 0.

D: maximize

$$W(\boldsymbol{y}) \qquad subject\ to \qquad \boldsymbol{y} \geq \mathbf{0}$$

D is therefore a rather 'nice' problem, which may be solved by a gradient method. Once the dual problem has been solved, the optimal solution of the (primal) subproblem $P^{(k)}$ is directly obtained by just plugging in the optimal dual solution **y** in the expressions for $x_j(\boldsymbol{y})$.

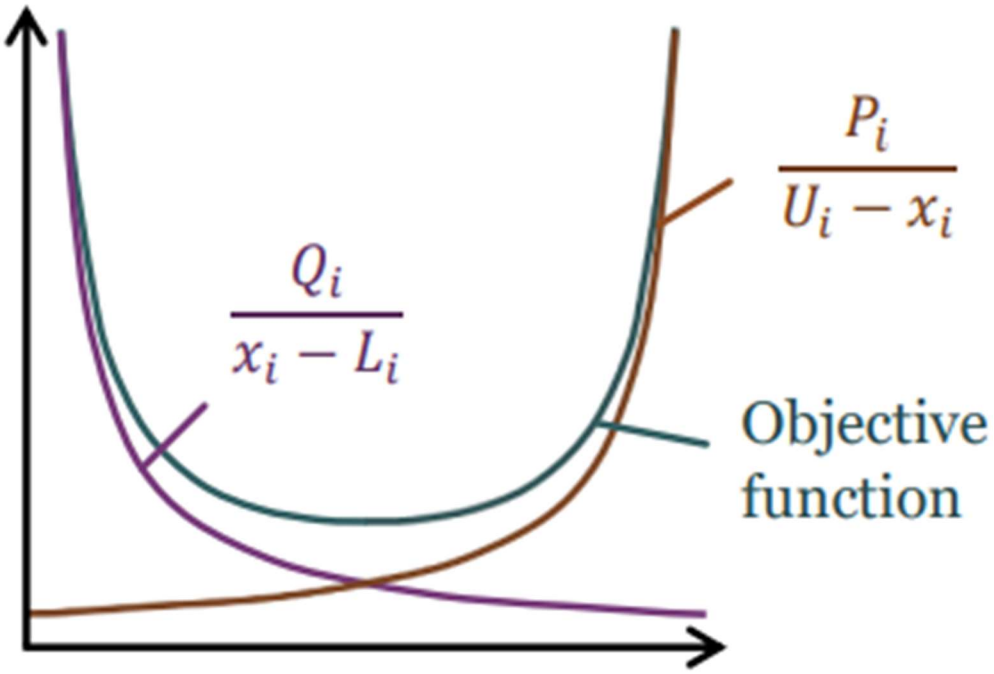



**Figure 2.14** Simple visualization of Method of Moving Asymptotes

# Chapter 3

# 3 Homogenization

## 3.1 Micro To Macro Transmission (MTMT)

The definition of a macroscopic overall response of a heterogeneous material with complex microstructure in an averaged or homogenized sense, often denoted as micro-to-macro transition (MTMT). Analytical approaches are restricted in many circumstances, especially with respect to the geometry of the representative microstructure and its constitutive response, which is often assumed to be linearly elastic. Several numerical methods have been developed in recent years which discretize fine-scale fields on the representative microstructure. These methods allow the analysis of general geometries and constitutive nonlinearities of composite cells and provide detailed information on fine-scale mechanisms.

C. Miehe and A. Koch developed a distinct family of algorithms and matrix representations of overall stresses and tangent moduli for discretized microstructures of heterogeneous materials undergoing small strains. In order to achieve that, they considered a microstructure $\mathbb{R} \subset \mathbb{R}^3$ with overall properties related to a homogenized macro-continuum $\mathbb{R} \subset \mathbb{R}^3$. Typically, the microstructure defines a representative volume (RV) of a nonlinear heterogeneous material such as inelastic composite, polycrystalline aggregate or particle assembly. The also consider a priori discretized models of the microstructures. The basic underlying approach is a partitioning of the displacement degrees of a discretized microstructure into those of the interior domain and those associated with the boundary of the microstructure.

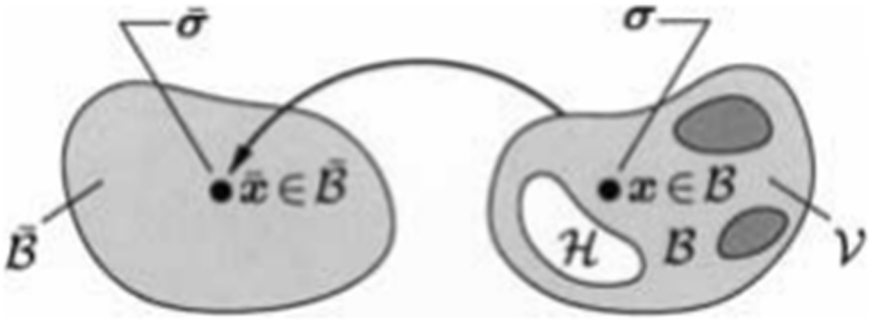



**Figure 3.1** Associated with a typical point $\bar{x} \in \bar{\mathbb{R}}$ of a homogenized macro-continuum $\mathbb{R} \subset \mathbb{R}^3$ is a microstructure $\mathbb{R} \subset \mathbb{R}^3$. The representative volume $V \subset \mathbb{R}^3$ that characterizes a representative part of the heterogeneous material.

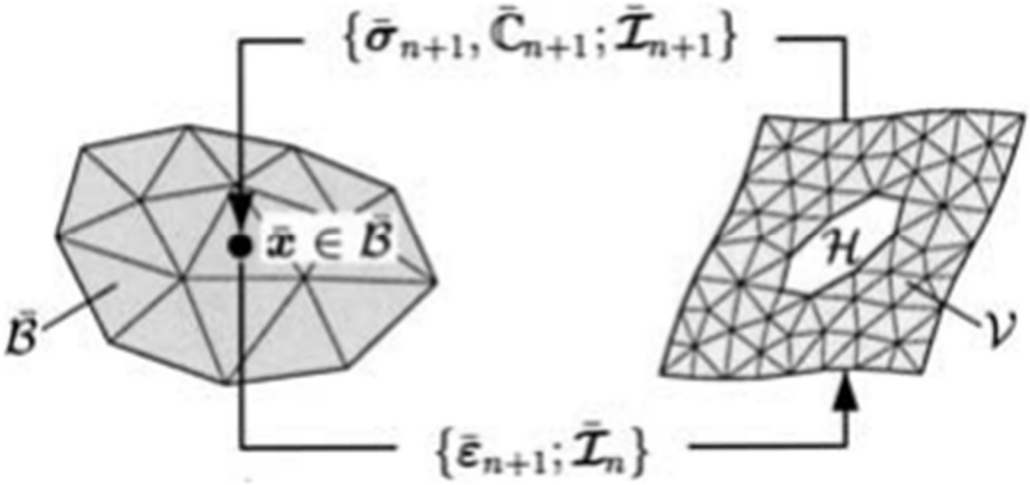



**Figure 3.2** Strain-driven computational treatment of a discretized microstructure.

## 3.2 Homogenization scheme

With the aforementioned ideas in mind, Miehe and Koch formulated a homogenization scheme. According to it, for a given macroscopic strain $\bar{\boldsymbol{\varepsilon}} = [\bar{\varepsilon}_{11},\ \bar{\varepsilon}_{22},\ 2 \cdot \bar{\varepsilon}_{33}]^T$ a displacement field u is applied to the boundary of the RVE according to the relation:

$$\boldsymbol{u}(\boldsymbol{x}) = \bar{\boldsymbol{\varepsilon}} \cdot \boldsymbol{x}\ at\ \boldsymbol{x} \in \partial V$$

where $\boldsymbol{x}$ denotes the position vector of a point on the boundary of the RVE and $\partial V$ denotes the boundary of the RVE. This relation constitutes the localization rule, that is, the rule that provides the boundary conditions that are applied on the RVE. In other words, the localization rule is used to transition from the macroscopic to the microscopic level.

After the solution of the boundary value problem resulting from the application of the localization rule, the macroscopic stress $\bar{\boldsymbol{\sigma}} = [\bar{\sigma}_{11}, \bar{\sigma}_{22},\ \bar{\tau}_{33}]^T$ is calculated as the volume average of the microscopic stress **σ** according to the relation:

$$\bar{\boldsymbol{\sigma}} = \frac{1}{|V|} \cdot sym(\int_V \boldsymbol{\sigma} dV)$$

where V denotes the volume of the RVE. This equation is the homogenization rule, that is, the rule that yields the macroscopic state variables as a function of the microscopic stress state. This means that homogenization rule is used to transit from the microscopic level back to the macroscopic one.

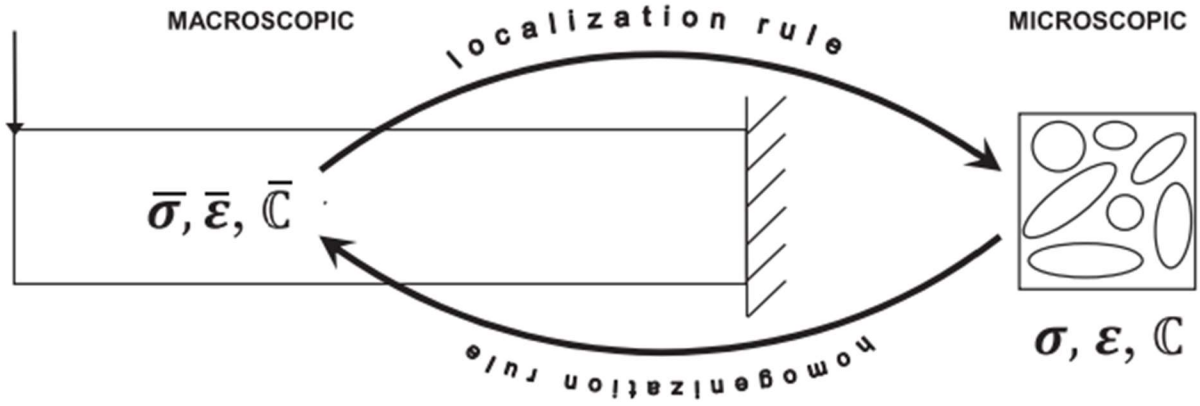



**Figure 3.3** Transition from the microscopic to the macroscopic level and vice versa, implementing the homogenization and localization rule respectively

The macroscopic tangent modulus $\bar{C}$ is calculated as the sensitivity of the macroscopic stress $\bar{\boldsymbol{\sigma}}$ with respect to the macroscopic strain $\bar{\boldsymbol{\varepsilon}}$ as follows:

$$\bar{C} := \partial_{\bar{\varepsilon}} \cdot \bar{\boldsymbol{\sigma}}$$

If the finite element method is used, the RVE is appropriately discretized and the formulation differs slightly. Further, quantities having **a** as an index refer to internal nodes, whereas quantities having **b** as an index refer to nodes on the boundary. If we denote the nodal displacement vector with **d**, the localization rule for every node **q** of the boundary can be written in matrix form as:

$$\boldsymbol{d_q} = D_{\boldsymbol{q}}^T \cdot \bar{\boldsymbol{\varepsilon}}$$

Where $D_{\boldsymbol{q}}$ is a matrix that is defined using the nodal coordinates $x_1$ and $x_2$ of the node as:

$$D_{\boldsymbol{q}} := \frac{1}{2}\begin{bmatrix} 2x_1 & 0 \\ 0 & 2x_2 \\ x_2 & x_1 \end{bmatrix}_q$$

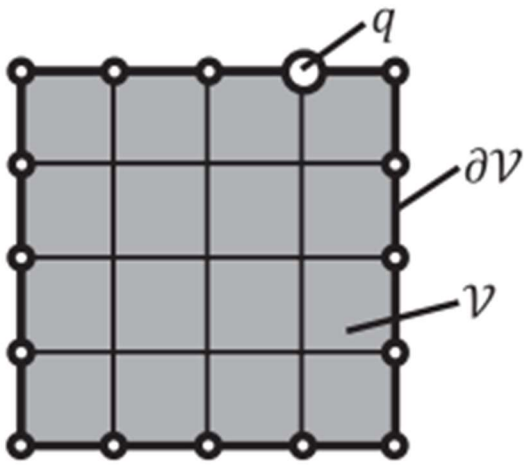



**Figure 3.4** Schematic representation of a discretized RVE.

The equilibrium of the RVE using Lagrange multipliers dictates that:

$$\boldsymbol{Q_a(x) = 0}$$

$$\boldsymbol{Q_b(x) - \lambda = 0}$$

$$\boldsymbol{d_b - D^T \cdot \bar{\varepsilon} = 0}$$

Where $[D \coloneqq D_1, D_2, \dots, D_M]$ is the global coordinate matrix for all M nodes lying on the boundary. $\boldsymbol{Q}$ denotes the internal nodal force vector and the Lagrange multiplier vector necessary for the enforcement of the boundary conditions with $\boldsymbol{\lambda}$. The above set of equations takes the following incremental form for an RVE:

$$\boldsymbol{Q_a(x) + K_{aa} \cdot \Delta d_a + K_{ab} \cdot \Delta d_b = 0}$$

$$\boldsymbol{Q_b(x) - \lambda + K_{ba} \cdot \Delta d_a + K_{bb} \cdot \Delta d_b - \Delta \lambda = 0}$$

$$\boldsymbol{d_b - D^T \cdot \bar{\varepsilon} + \Delta d_b - D^T \cdot \Delta \bar{\varepsilon} = 0}$$

where K denotes the tangential stiffness matrix of the RVE.

The solution of the problem is obtained using an iterative Newton–Raphson procedure, as follows: For a given macroscopic stress tensor $\bar{\boldsymbol{\varepsilon}}$ (that is $\Delta \boldsymbol{d_b} = \mathbf{0}$ ), the nodal displacements $\boldsymbol{d_a}$ are computed iteratively from the first equation according to the relations:

$$\boldsymbol{d_a} \leftarrow \boldsymbol{d_a} + \Delta \boldsymbol{d_a}, \qquad \Delta \boldsymbol{d_a} = \boldsymbol{K_{aa}^{-1}} \cdot \boldsymbol{Q_a}$$

The iterations continue until convergence is achieved in the sense that $||\boldsymbol{Q_a}||$ < tolerance. The macroscopic stress vector is then calculated as:

$$\bar{\boldsymbol{\sigma}} = \frac{1}{|V|} \cdot D \cdot \boldsymbol{Q_b}$$

And the macroscopic tangent modulus as:

$$\bar{C} = \frac{1}{|V|} \cdot D \cdot \bar{\boldsymbol{K}}_{\boldsymbol{bb}} \cdot D^T$$

where the condensed stiffness matrix $\bar{\boldsymbol{K}}_{\boldsymbol{bb}}$ is the result of static condensation:

$$\bar{\boldsymbol{K}}_{\boldsymbol{bb}} := \boldsymbol{K_{bb}} - \boldsymbol{K_{ba}} \cdot \boldsymbol{K_{aa}^{-1}} \cdot \boldsymbol{K_{ab}}$$

## 3.3 Embedded fiber element model

A very common case of making use of homogenization is when a constitutive relation is required to describe the behavior of a microstructure constituted by a matrix element with an embedded fiber.

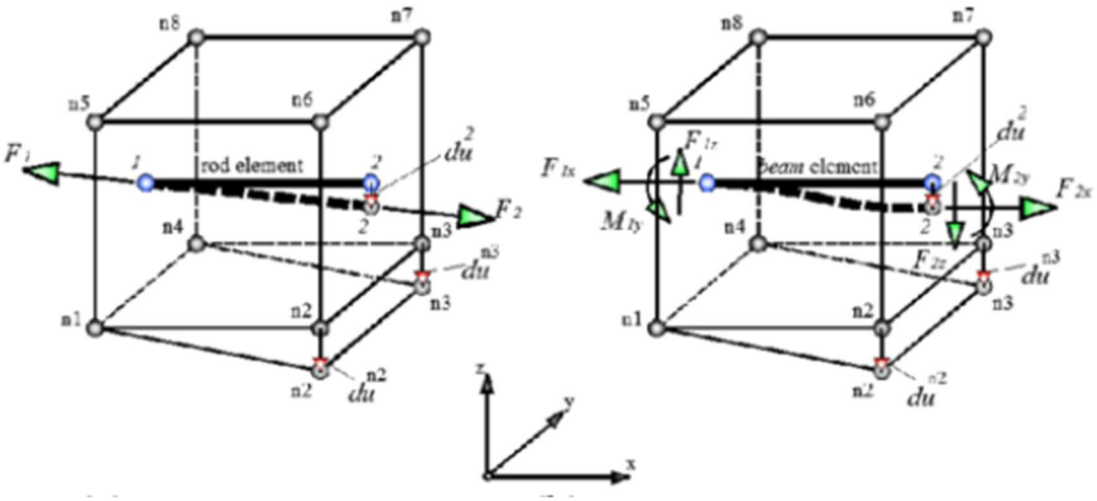



**Figure 3.5** 3-D representation of a matrix element with an embedded fiber element. In the right figure the embedded element is a truss member while in left is a beam element.

In 2-Dimensional space, we may assume that the matrix element is a Quad plain stress element with four nodes. The fiber element is considered to be a truss member with two nodes, as we want to simulate a fiber reinforcement behavior. Reinforcement elements in structural mechanics very commonly receive only axial forces and truss members are eligible to model this behavior. The displacements of the fiber element are the result of an interpolation of quad's displacements. In order to describe this procedure, we number the nodes and degrees of freedom and consider the dimensions as follows:

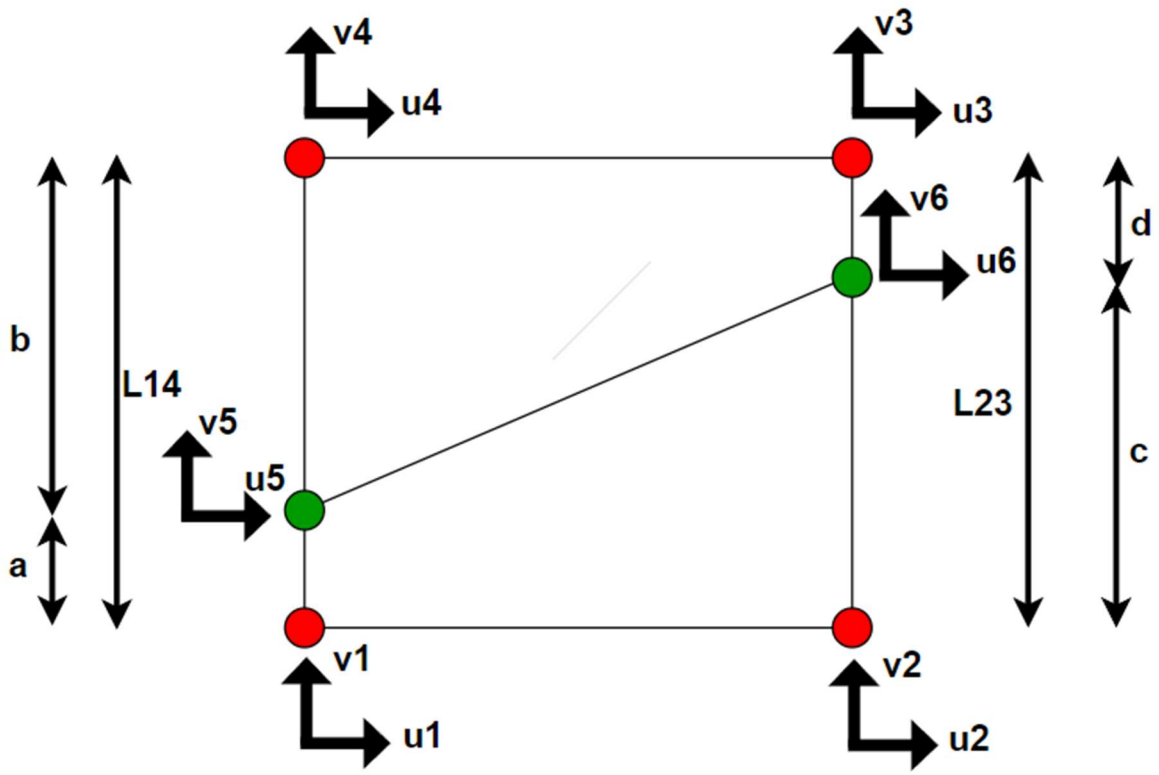



**Figure 3.6** Quad element with an embedded truss element. Naming of nodes, degrees of freedom and dimensions

In this example, displacements of 5 and 6 nodes will be the result of an interpolation of the nodes' (1, 4) and (2, 3) displacements respectively. According to that, we obtain the relations:

$$u_5 = \frac{b}{L_{14}} \cdot u_1 + \frac{a}{L_{14}} \cdot u_4$$

$$v_5 = \frac{b}{L_{14}} \cdot v_1 + \frac{a}{L_{14}} \cdot v_4$$

$$u_6 = \frac{d}{L_{23}} \cdot u_2 + \frac{c}{L_{23}} \cdot u_3$$

$$u_6 = \frac{d}{L_{23}} \cdot v_2 + \frac{c}{L_{23}} \cdot v_3$$

Then, we can express the dependence between fiber and quad element's displacements in the following matrix form:

$$\begin{Bmatrix} u_5 \\ v_5 \\ u_6 \\ v_6 \end{Bmatrix} = \begin{bmatrix} b/L_{14} & 0 & 0 & 0 & 0 & 0 & a/L_{14} & 0 \\ 0 & b/L_{14} & 0 & 0 & 0 & 0 & 0 & a/L_{14} \\ 0 & 0 & c/L_{23} & 0 & c/L_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & c/L_{23} & 0 & c/L_{23} & 0 & 0 \end{bmatrix} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$

Which can be recast in the form:

$$\boldsymbol{d}_{Truss} = [\boldsymbol{T}] \cdot \boldsymbol{d}_{Quad}$$

Similarly, we can write

$$\boldsymbol{P}_{Quad} = [\boldsymbol{T}]^T \cdot \boldsymbol{P}_{Truss}$$

This way, we transform truss element's equilibrium and after trivial calculations, we can express embedded element's stiffness matrix in respect to those of Quad and Truss elements as follows

$$[\boldsymbol{K}]_{Quad}^{emb} = [\boldsymbol{K}]_{Quad} + [\boldsymbol{T}]^T \cdot [\boldsymbol{K}]_{Truss} \cdot [\boldsymbol{T}]$$

# Chapter 4

# 4. Multiscale Topology Optimization of fiber-reinforced structures

## 4.1 Broadening Topology Optimization formulation

After a thorough investigation in topology optimization and musing on the possibilities this extremely useful methodology provides to us, the natural next step is the endeavor to alter the problem formulation in order to achieve different goals. Researchers' experimentations with objective functions, design variables and constraints has as an outcome the implementation of exceptional methods increasing the structure's efficiency according to various demands. For example, an objective function could be added altering the optimal layout according to a second criterion. This way, we can ensure that our solution behaves optimally according to two different criteria or even transform the problem into a 'multi-objective' one having to choose from a 'pareto family' of solutions.

As it is previously stated, topology optimization may converge in a solution the implementation of which could be difficult. More constraints could be formulated aiming to adjust the optimization's result in a form which may be easier to implement in a particular situation. Another goal we can achieve through adding rational constraints to the formulation is a more accurate description of the physics behind a problem.

For example, prompted by ideas such as those mentioned above, Jonathan B. Russ and Haim Waisman[12] proposed a new formulation incorporating local ductile failure constraints and buckling resistance into elastoplastic structural design in the context of extreme loading. To achieve that, they added the appropriate for the problem's physics constraints and used the following complex expression for the objective function:

$$\overbrace{-\omega_1 \cdot \frac{W(\theta,\{\bar{u}_i\},\{c_i\})}{W^{scale}}}^{Total\ work\ term} + \overbrace{\omega_2 \cdot \frac{B_{KS}(\theta,\bar{u}_L)}{B_{KS}^{scale}}}^{Buckling\ term} + \overbrace{\omega_3 \cdot AL(\theta,\{\bar{u}_i\},\{c_i\})}^{Ductility\ term}$$

Weight factors ($\omega_1$, $\omega_2$, $\omega_3$) were added to define the significance of each term in the contextual problem, increasing the efficiency of this formulation on many occasions.

Another way to alter the formulation of an optimization problem, is by inserting design variables. Doing that, we can optimize multiple parameters in a design problem. For example, in topology optimization apart from structure's density, which is the common design variable, other characteristics (if they could be appropriately modeled) can be optimized as well.

Investigating this type of generalization, we propose a topology optimization scheme for fiber-reinforced structures with two design variables: element density and fiber orientation. The goal was to find the structural layout and set of fiber orientations that minimized a fiber-reinforced structure's compliance.

## 4.2 Multiscale Analysis

Composite materials are becoming more and more popular in structural design. The rapid advance of the material technology broadens their application range and as a result the design options. In this framework, the capability to create optimized composite structures is increased. The advent of 3D printing technologies such as fused-deposition modeling has enabled the easy manufacturing of superior fiber-reinforced composites A topology optimization scheme, optimizing the composite material design, could take advantage of this technological growth providing us with demanding (in terms of implementation solutions) which can nowadays be implemented.

However, analyzing and designing structures made by nanomaterials is a computationally intensive task. Thus, multiscale analysis provides an attractive option, in that it breaks down the complex structural model into several layers of simpler ones. Multiscale optimization is a natural next step, using multiscale analysis to facilitate the computationally intensive task of optimization.

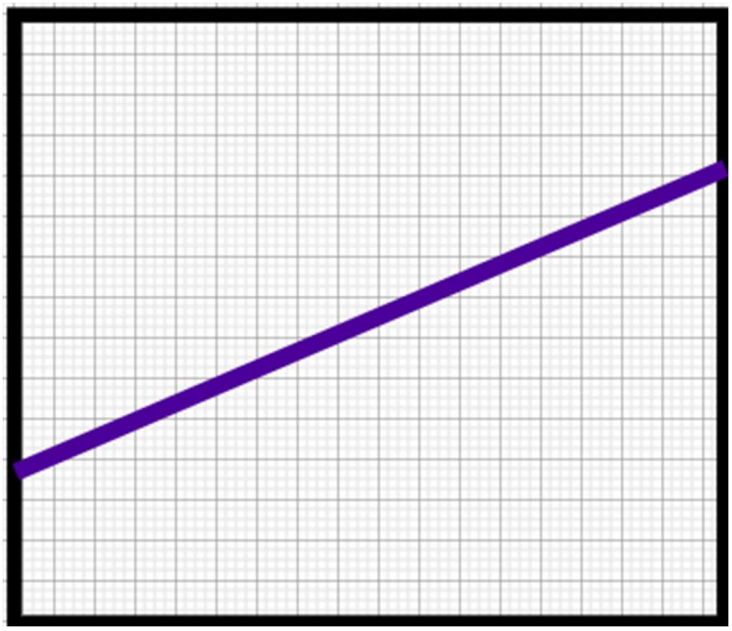


**Figure 4.1** Microscopic Level
A RVE consisting of a matrix material reinforced with a single horizontal fiber material is analyzed using the finite element method

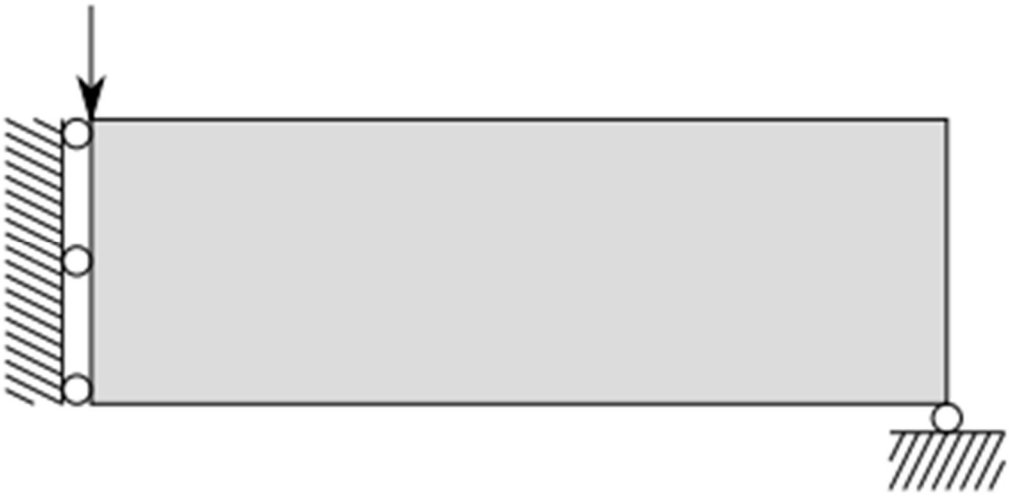


**Figure 4.2** Macroscopic Level
A cantilever beam consisting of the material described by the RVE of the previous level is analyzed using the finite element method.

# Chapter 4 Multiscale topology optimization of fiber-reinforced structures

Multiscale topology optimization refers to the task of optimizing the topology of structures that are complex enough to warrant the use of multiscale analysis. In the proposed scheme, the topology of the structure is optimized on multiple levels, assuming that the structure on the lower level is constant across the higher one. A common pattern in optimizing the topology in two scales is optimizing the microstructure for a discrete, representative set of stress loads, and use the results when optimizing the topology on the macroscopic level.

The most frequent formulation is optimizing the topology of a structure and the orientation of the material, using an orthotropic constitutive law to describe the material properties[6,10]. Gao and Ma[7] extended this formulation to also optimize the topology on a lower scale using a multiscale technique, whereas Groen and Sigmund[11] attempted to describe the same problem in a more computationally efficient manner, using a microstructure that consisted of a rectangular matrix with a single hole. This topology optimization problem was approximated by optimizing the dimensions of the hole. Regarding to optimizing material orientation, the material axes were commonly aligned to the direction of the principal stresses. In contrast, Gao and Ma[7] made no assumptions about the optimal orientation, instead treating the element orientations as additional variables in the optimization problem, using a gradient method to obtain a solution. In Garland and Fadel[10], the orientations were treated in a similar manner, but instead of using a gradient method to get the optimal orientations, the non-gradient golden section algorithm was employed. According to it, the entirety of the orientation domain was searched for the minimum in every iteration. Finally, in Esposito[9] an analytical relation for the optimal orientation of an orthotropic material was obtained as a function of the stress to which the material is subjected.

In the proposed optimization scheme, the computationally expensive task of multiscale analysis is executed only once, before the initialization of the optimization loop. We simulate the structure in the macroscopic level with a 4-node quad finite element mesh. Every finite element complies with its own constitutive law, which is obtained from the microstructure analysis. In order to model the microstructure, we create an RVE constituted by a 4-node quad element with an embedded horizontal truss element. Then, a homogenization is executed and we obtain the homogenized tensor that corresponds to the microscopic RVE with a horizontal fiber. The orientation of the fibers was optimized by rotating the homogenized tensor in the optimization loop.

We used first-order homogenization pertaining to linear boundary displacements. To calculate the properties of the homogenized material, displacements representing a strain were applied to the boundary of the RVE according to the following relation:

$$\boldsymbol{d_q} = D_q^T \cdot \bar{\boldsymbol{\varepsilon}}$$

Where $\boldsymbol{d_q}$ denotes the displacement applied on a boundary node $q$ of the RVE, $\bar{\boldsymbol{\varepsilon}}$ denotes the strain for which we want to calculate the homogenized properties of the RVE, and $D_{\boldsymbol{q}}$ is a matrix that is defined using the nodal coordinates $x_1$ and $x_2$ of the node as:

$$D_{\boldsymbol{q}} := \frac{1}{2}\begin{bmatrix} 2x_1 & 0 \\ 0 & 2x_2 \\ x_2 & x_1 \end{bmatrix}_q$$

The homogenized elasticity tensor $\bar{C}$ is then calculated by averaging the properties of the deformed RVE according to relation:

$$\bar{C} = \frac{1}{|V|} \cdot D \cdot \bar{\boldsymbol{K}}_{\boldsymbol{bb}} \cdot D^T$$

Where $|V|$ denotes the volume of the RVE and $\bar{\boldsymbol{K}}_{\boldsymbol{bb}}$ denotes the stiffness matrix of the RVE after statically condensing the internal nodes, according to relation:

$$\bar{\boldsymbol{K}}_{\boldsymbol{bb}} := \boldsymbol{K}_{\boldsymbol{bb}} - \boldsymbol{K}_{\boldsymbol{ba}} \cdot \boldsymbol{K}_{\boldsymbol{aa}}^{-\boldsymbol{1}} \cdot \boldsymbol{K}_{\boldsymbol{ab}}$$

In the above relation, $b$ denotes nodes on the boundary and $a$ denotes internal nodes.

## 4.3 Optimization problem

Regarding optimization, our goal was to minimize the macroscopic structure's compliance, in other words, minimize the work produced by the external loads. The design variables were twofold: a) the topology of the macroscopic structure, and b) the orientation of the reinforcement fiber for every macroscopic element.

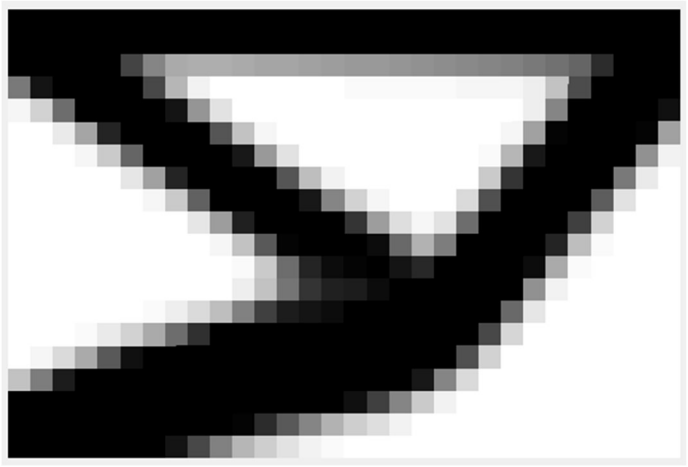


**Figure 4.4** Optimal Topology of macrostructure

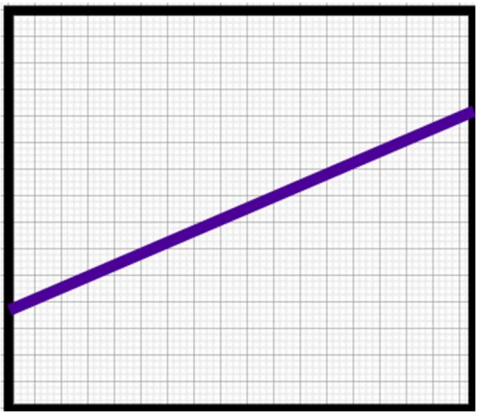


**Figure 4.5** Optimal reinforcement fiber orientation for every macroscopic element.

The problem can be stated as follows:

- Find a layout of the macroscopic structure that occupies volume $V_{goal}$ within a domain that is given by the relation:

$$V_{goal} = f * V \leq V$$

where $f$ is the target volume fraction and $V$ is the volume of the domain.

- And a spatial distribution of reinforcement fiber orientations $\theta_i$:

$$\theta_i \in [0, \pi] \ (rad)$$

- So that the work produced by the external loads, equal to the beam's potential energy, is minimized:

$$c = \boldsymbol{P}^{\boldsymbol{T}} * \boldsymbol{u} = \boldsymbol{u}^{\boldsymbol{T}} * \boldsymbol{K} * \boldsymbol{u} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{N}} \boldsymbol{u}_{\boldsymbol{i}}^{\boldsymbol{T}} * \boldsymbol{k}_{\boldsymbol{i}} * \boldsymbol{u}_{\boldsymbol{i}}$$

  In the above relation, $\boldsymbol{P}$ denotes the external nodal force vector, $\boldsymbol{u}$ denotes the nodal displacement vector, $\boldsymbol{K}$ denotes the global stiffness matrix of the structure, and $\boldsymbol{u_i}$ and $\boldsymbol{k_i}$ denote the nodal displacements and stiffness matrix of each element respectively.

The stiffness matrix of each element $\boldsymbol{k_i}$ is calculated using the following relation:

$$\boldsymbol{k_i} = \int_{\boldsymbol{V_i}} \boldsymbol{B}^{\boldsymbol{T}} * \mathbb{C}_{\boldsymbol{i}} * \boldsymbol{B}\, dV$$

Where $\mathbb{C}_{\boldsymbol{i}}$ denotes the material elasticity tensor for element $i$.

The orientation of the fibers was simulated by rotating the homogenized tensor that corresponds to the microscopic RVE with the horizontal fiber. Doing so allowed us to obtain cheap analytical relations for the derivative of the objective function with respect to orientations, as opposed to approximating them arithmetically. This also allowed us to use homogenization on a detailed model of the microstructure to accurately calculate the equivalent elasticity tensor without increasing the computational cost, since the computationally demanding task of homogenization only needs to be performed once.

The elasticity tensor $\mathbb{C}_{\mathbf{rot,i}}$ corresponding to fiber orientation $\theta i$was calculated as follows:

$$\mathbb{C}_{\boldsymbol{rot,i}} = \boldsymbol{R}(\theta_i)^{-\mathbf{1}} * \mathbb{C}_{\boldsymbol{unrot}} * \boldsymbol{T} * \boldsymbol{R}(\theta_i) = \boldsymbol{R}(-\theta_i) * \mathbb{C}_{\boldsymbol{unrot}} * \boldsymbol{T} * \boldsymbol{R}(\theta_i)$$

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta)^2 & \sin(\theta)^2 & 2 * \sin(\theta) * \cos(\theta) \\ \sin(\theta)^2 & \cos(\theta)^2 & -2 * \sin(\theta) * \cos(\theta) \\ -\sin(\theta) * \cos(\theta) & \sin(\theta) * \cos(\theta) & \cos(\theta)^2 - \sin(\theta)^2 \end{bmatrix}$$

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

The topology optimization problem was formulated using SIMP. Accordingly, a density variable $\rho_i$ was assigned to every macroscopic element. The density ranged continuously between 0 and 1, with 1 denoting presence and 0 denoting absence of material. Densities between 0 and 1 are interpreted as material mesostructures with holes. The material elasticity tensor for each element was expressed as a function of the density variable according to

$$\mathbb{C}_{\boldsymbol{penal,i}}(\rho_i) = \rho_i^p * \mathbb{C}_{\boldsymbol{full}}$$

Where $\mathbb{C}_{\boldsymbol{full}}$ denotes the elasticity coefficient of the material and $\rho_i^p$ , is the penalty function, that guides the density of every element towards 0 or 1. Higher values of penalty coefficient $p$ accelerate convergence, $p$ is increased adaptively during the optimization iterations avoiding convergence to local minima and ensure independence from initial conditions.

Finally, the elasticity tensor $\mathbb{C}_{\boldsymbol{i}}$ for element $i$ is obtained by first calculating the homogenized elasticity tensor $\mathbb{C}_{\boldsymbol{hom}}$ of the microscopic RVE, which corresponds to horizontal fibers and is the same for all macroscopic elements. The homogenized tensor is then rotated by the fiber orientation $\theta_i$, followed by applying the penalty function as follows:

$$\mathbb{C}_{\boldsymbol{i}}(\rho_i, \theta_i) = \rho_i^p * \mathbb{C}_{\boldsymbol{rot,i}}$$

Where:

$$\mathbb{C}_{\boldsymbol{rot,i}} = \boldsymbol{R}(\theta_i)^{-1} * \mathbb{C}_{\boldsymbol{hom}} * \boldsymbol{T} * \boldsymbol{R}(\theta_i)$$

Considering this relation, the element stiffness matrix can be expressed as follows:

$$\boldsymbol{k_i} = \rho_i^p * \boldsymbol{k_{rot,i}}$$

Where:

$$\boldsymbol{k_{rot,i}} = \int_{\boldsymbol{V_i}} \boldsymbol{B^T} * \mathbb{C}_{\boldsymbol{rot,i}} * \boldsymbol{B}\, dV$$

Therefore, the objective function takes the form:

$$c = \sum_{i=0}^{N} \rho_i^p * \boldsymbol{u_i^T} * \boldsymbol{k_{rot,i}} * \boldsymbol{u_i}$$

The aforementioned optimization problem was solved using a gradient-based algorithm. To apply the algorithm, the derivatives of the objective function with respect to all optimization variables were calculated as follows:

$$\frac{\partial c}{\partial \rho_i} = -p * \rho_i^{p-1} * (\boldsymbol{u_i^T} * \boldsymbol{k_{rot,i}} * \boldsymbol{u_i})$$

$$\frac{\partial c}{\partial \theta_i} = -\rho_i^p * (\boldsymbol{u_i^T} * \partial \boldsymbol{k_{rot,i}}/\partial \theta_i * \boldsymbol{u_i})$$

Where:

$$\frac{\partial \boldsymbol{k_{rot,i}}}{\partial \theta_i} = \int_{\boldsymbol{V_i}} \boldsymbol{B^T} * \partial \mathbb{C}_{\boldsymbol{rot,i}}/\partial \theta_i * \boldsymbol{B}\, dV$$

And:

$$\frac{\partial \mathbb{C}_{\boldsymbol{rot,i}}}{\partial \theta_i} = \frac{\partial \boldsymbol{R}(-\theta_i)}{\partial \theta} * \mathbb{C}_{\boldsymbol{hom}} * \boldsymbol{T} * \boldsymbol{R}(\theta_i) + \boldsymbol{R}(-\theta_i) * \mathbb{C}_{\boldsymbol{hom}} * \boldsymbol{T} * \frac{\partial \boldsymbol{R}(\theta_i)}{\partial \theta}$$

Where:

$$\frac{\partial \boldsymbol{R}(\theta)}{\partial \theta}$$

$$= \begin{bmatrix} -2 * \sin(\theta) * \cos(\theta) & 2 * \sin(\theta) * \cos(\theta) & 2 * (\cos(\theta)^2 - \sin(\theta)^2) \\ 2 * \sin(\theta) * \cos(\theta) & -2 * \sin(\theta) * \cos(\theta) & -2 * (\cos(\theta)^2 - \sin(\theta)^2) \\ -(\cos(\theta)^2 - \sin(\theta)^2) & (\cos(\theta)^2 - \sin(\theta)^2) & -4 * \sin(\theta) * \cos(\theta) \end{bmatrix}$$

The optimization problem was solved using MMA[13], which was analyzed in chapter 2.4. According to it, the objective function at point $\boldsymbol{x_0}$ is approximated by a convex hyperbolic one.

## 4.4. Orientations and Densities Filtering

Solution of the topology optimization problem described in the previous section with the aforementioned gradient-based algorithm is very sensitive to initial conditions, as it is very probable that the algorithm will converge to some local minimum with respect to orientations. The reason for this is that there is no guarantee that the set of the chosen initial orientations will not be close to a local minimum of the objective function for the problem under consideration. If the initial choice for the orientation for an element coincides or is close to a local minimum of the objective function, it is possible that the optimization will converge to that local minimum.

As a result, solutions that present localized discontinuities of the orientations are possible. This phenomenon is similar to checkerboard patterns, a well-known problem of topology optimization algorithms, wherein the topology optimization algorithm converges to a non-constructible black-and-white pattern. This kind of pattern has artificially high stiffness, as a result of using the finite element method to describe the problem.

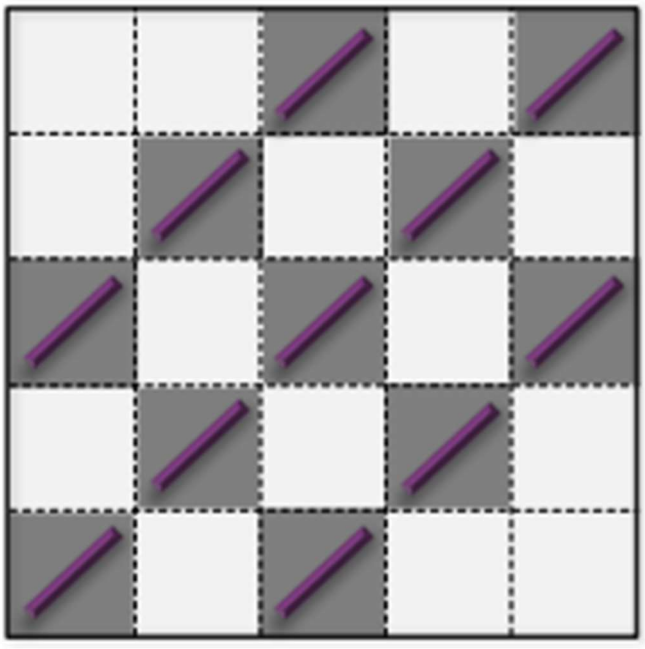


**Figure 4.6** Checkboard Pattern

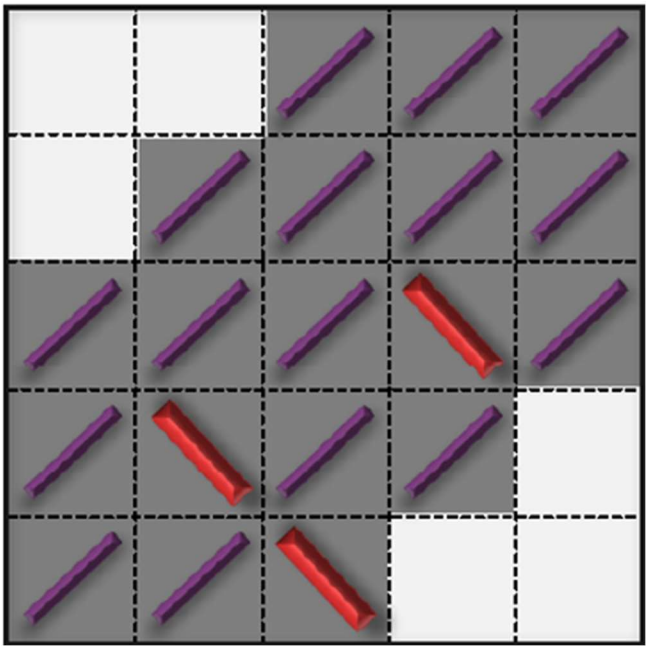


**Figure 4.7** Localized orientation discontinuities

To solve the aforementioned issues and improve the convergence and speed of the optimization algorithm, we used a convolution kernel to filter both densities and orientations. The convolution kernel works in a similar manner to blurring an image by calculating the value of a variable (be it density or orientation) for element $i$ as a weighted average of the value of the variable for the element as well as the element's neighbors according to the following relations:

$$\theta_i = \frac{1}{\sum_{j=1}^{N} w_{ij}} * \sum_{j=1}^{N} w_{ij} * \bar{\theta_j}$$

$$\rho_i = \frac{1}{\sum_{j=1}^{N} w_{ij}} * \sum_{j=1}^{N} w_{ij} * \bar{\rho_j}$$

Where $w_{ij}$ is a weight factor decreasing linearly with distance $\|r_{ij}\| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ of element j from element I according to:

$$w_{ij} = \begin{cases} 1 - \dfrac{\|r_{ij}\|}{\|r_{filter}\|} \; if \; \|r_{ij}\| \leq \|r_{filter}\| \\ 0 \; otherwise \end{cases}$$

Where $\|r_{filter}\|$ is the cutoff radius of the filter.

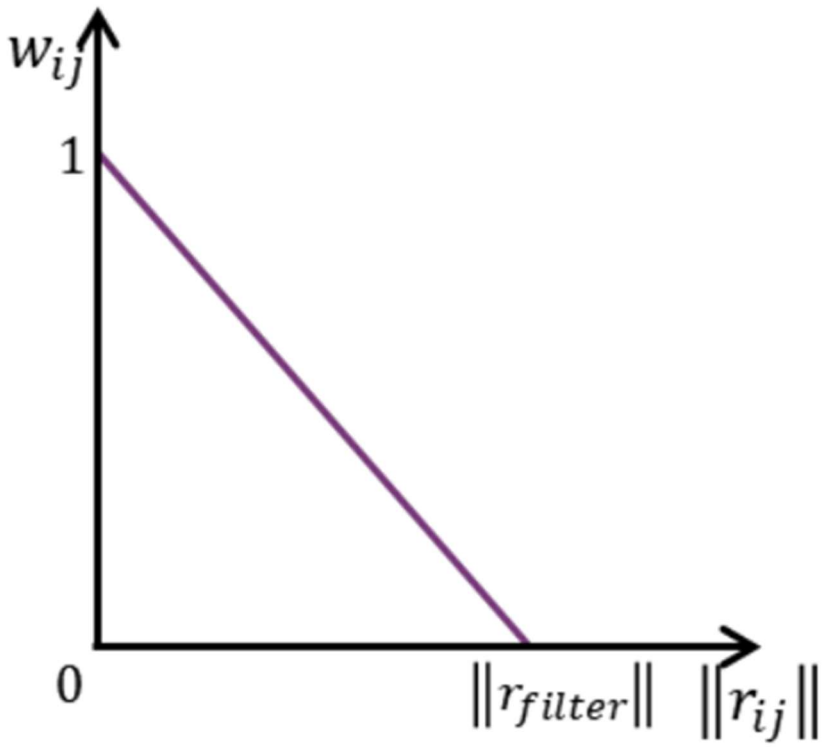



**Figure 4.8** Schematic representation of the weight factor $w_{ij}$ as a function of the distance between elements I and j.

Relations of $\theta_i$ and $\rho_i$ can be recast in matrix form as follows:

$$\boldsymbol{\theta} = \mathbb{G} * \overline{\boldsymbol{\theta}}$$

$$\boldsymbol{\rho} = \mathbb{G} * \overline{\boldsymbol{\rho}}$$

Where $\boldsymbol{\theta}$ and $\overline{\boldsymbol{\theta}}$ are vectors containing orientations $\theta_i$ and $\overline{\theta}_i$ respectively, $\boldsymbol{\rho}$ and $\overline{\boldsymbol{\rho}}$ are vectors containing densities $\rho_i$ and $\overline{\rho}_i$ respectively, and $\mathbb{G}$ is a transformation matrix whose elements are calculated as:

$$\mathbb{G}_{\mathrm{ij}} = \frac{w_{ij}}{\sum_{j=1}^{N} w_{ij}}$$

Those relations of filtered densities and orientations are subsequently substituted into objective function's relations, in order to express the objective function as a function of $\overline{\theta_j}$ and $\overline{\rho_j}$. Therefore, the optimization is performed with respect to "virtual" densities $\overline{\rho_j}$ and orientations $\overline{\theta_j}$. The quantities that describe the physical problem, called the "physical" variables $\rho i$ and $\theta i$, are derived by mapping the spatial distribution of quantities $\overline{\rho_j}$ and $\overline{\theta_j}$ to a smoother one, according to filtering relations.

In order to perform the optimization, we need to calculate the derivatives of the objective function with respect to $\overline{\rho_j}$ and $\overline{\theta_j}$. This is achieved by applying the chain rule as follows:

$$\frac{\partial f}{\partial \bar{\theta}_i} = \sum_{j=1}^{N} \frac{\partial f}{\partial \theta_j} * \frac{\partial \theta_j}{\partial \bar{\theta}_i}$$

$$\frac{\partial f}{\partial \bar{\rho}_i} = \sum_{j=1}^{N} \frac{\partial f}{\partial \rho_j} * \frac{\partial \rho_j}{\partial \bar{\rho}_i}$$

Those relations can be recast in matrix form as:

$$\frac{\partial f}{\partial \overline{\boldsymbol{\theta}}} = \mathbb{G}^{\mathrm{T}} * \frac{\partial f}{\partial \boldsymbol{\theta}}$$

$$\frac{\partial f}{\partial \overline{\boldsymbol{\rho}}} = \mathbb{G}^{\mathrm{T}} * \frac{\partial f}{\partial \boldsymbol{\rho}}$$

Where $\frac{\partial f}{\partial \overline{\boldsymbol{\theta}}}, \frac{\partial f}{\partial \boldsymbol{\theta}}$ are vectors containing the derivatives of the objective function with respect to virtual and physical orientations $\frac{\partial f}{\partial \overline{\theta}}, \frac{\partial f}{\partial \theta}$ and $\frac{\partial f}{\partial \overline{\boldsymbol{\rho}}}, \frac{\partial f}{\partial \boldsymbol{\rho}}$ are vectors containing the derivatives of the objective function with respect to virtual and physical densities $\frac{\partial f}{\partial \overline{\rho}}, \frac{\partial f}{\partial \rho}$.

Applying the aforementioned filters differentiates the flow of the optimization as follows: For every step of the optimization, the approximation for the virtual orientations $\overline{\boldsymbol{\theta}}$ and densities $\overline{\boldsymbol{\rho}}$ is calculated. These are transformed to their physical counterparts $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ by applying the filtering transformation tensor $\mathbb{G}$ το $\overline{\boldsymbol{\theta}}$ and $\overline{\boldsymbol{\rho}}$. Subsequently, the value of the objective function at ($\boldsymbol{\theta}$, $\boldsymbol{\rho}$) is calculated, as well as the derivatives of the objective function with respect to physical orientations $\frac{\partial f}{\partial \boldsymbol{\theta}}$ and densities $\frac{\partial f}{\partial \boldsymbol{\rho}}$. The derivatives of the objective function with respect to physical variables $\frac{\partial f}{\partial \boldsymbol{\theta}}, \frac{\partial f}{\partial \boldsymbol{\rho}}$ are then transformed to derivatives with respect to virtual variables $\frac{\partial f}{\partial \overline{\boldsymbol{\theta}}}$ and $\frac{\partial f}{\partial \overline{\boldsymbol{\rho}}}$ by applying the filtering transformation tensor $\mathbb{G}^{\mathrm{T}}$ to $\frac{\partial f}{\partial \boldsymbol{\theta}}, \frac{\partial f}{\partial \boldsymbol{\rho}}$. Finally, the filtered derivatives are returned to the MMA optimizer in order to proceed to the next optimization step. It is worth mentioning that filtering the derivatives can be omitted. However, calculating the derivatives with respect to the actual variables for which the optimization is performed instead of the physical ones, increases the accuracy of the optimizer approximations, which in turn accelerates convergence.

The radius of the density filter roughly corresponds to the minimum width of the members in the resulting optimal truss structure. Moreover, the radius of the orientation filter is proportional to the minimum radius of curvature of the fiber reinforcements. The minimum radius of curvature controls the waviness of the fiber reinforcements within a truss member, the relative angles between truss members, as well as the thickness of the truss junctions. As is the case for topology optimization, the choice of the density and orientation filter radius can greatly impact the end result. Both the problem of gray material, as well as the sensitivity of the result with respect to the choice of filter radius can be addressed using adaptive filtering.

## 4.5. Adaptive filtering and elimination of intermediate densities

The way filters were applied influenced the obtained result. For higher values of the orientation filter radius the algorithm converged faster, but the result could differ significantly from the global minimum and could depend on the initial orientations. In order to avoid local minima and ensure independence of the solution from initial conditions, an adaptive successive filtering of the orientations was implemented as follows:

- First, the initial virtual orientation for every element was chosen randomly between 0 rad and $\pi$ rad and a filter radius of 1.00m was selected. Every time the algorithm converged to a local minimum, the filter radius was increased by $1.00m$ (the element dimension of the discretized structure), until the desired filter radius $\|r_{oreint}^{conc}\|$ was reached. The relative change of the objective function with respect to the previous iteration was used to assess convergence. A local minimum was assumed to have been reached if the relative change of the objective function was less than 5e − 3. After the orientation filter had reached its target radius, the process was repeated for the density filter.

- To further control the rate of convergence, the density scope was gradually expanded throughout the optimization. While the orientation and density filter radius were increased, the density scope was kept limited. In our case, densities were allowed to vary between 0.4 and 0.7. This enabled the structure's topology to change as the radius of the orientation and density filter were increased. After the radius had reached their target values, the density bounds were gradually relaxed. The lower bound of the density was decreased by 0.05 and the upper bound increased by 0.05 every time the number of elements with intermediate densities dropped below a limit or the algorithm had reached a local minimum. In our case, the density bounds were relaxed every time the number of elements with virtual density between $\bar{\rho}min$ + 0.10 and $\bar{\rho}max$ − 0.10 dropped below 0.30 of the total number of elements, where $\bar{\rho}min$ and $\rho max\bar{}$ denote the lower bound and upper bound of the virtual density for the current iteration. A local minimum was assumed to be reached if the relative change of the objective function was less than 1e − 4. Moreover, in order to prevent the structure's stiffness matrix from becoming singular, densities were not allowed to drop to 0. Therefore, the density bounds were expanded until the lower bound was equal to 0.1 and the upper bound was equal to 1.

- The effect of the penalty coefficient on convergence was similar to that of the filter. For lower values, the algorithm was slow but stable. Conversely, for higher values of the penalty coefficient the algorithm converged faster, but it tended to converge more often to local minima. Moreover, higher values of the penalty coefficient yielded greater variation of the results with respect to initial orientations. To remedy this, the value of the penalty coefficient was increased gradually as follows: For the initial stage, while the filters were applied incrementally and the density bounds were expanded, the penalty coefficient was kept constant and equal to 1. After the density bounds had reached their final values and the algorithm had converged to a local minimum, the penalty coefficient was increased by one. Subsequently the penalty coefficient was increased by 1 every time the algorithm reached a local minimum, until it reached its prescribed value. The optimization was assumed to have converged to a local minimum if the relative change of the objective function was less than 1e − 2 .

- Using a density filter might result in a significant amount of grey material for the optimized topology. Apart from being non-constructible, topologies with gray material result in high values of the objective function, due to the penalization of intermediate densities. To avoid this, we gradually reduced the density filter radius at the end of the optimization to 0, which guided the algorithm to black-and-white topologies. It was important that the algorithm had converged before every reduction of the density filter radius, otherwise the reduction could lead the algorithm to a local minimum. We assumed that convergence was achieved when the following two criteria were met:

    - All virtual densities were either 0 or 1, with no gray material. In reality, this goal was unattainable. Instead, it was enough to check whether the elements with density between 0.20 and 0.90 exceeded in number $5e-2$ of the total number of elements of the structure.

    - The relative change of the objective function with respect to the previous optimization iteration was below $1e-3$.

- Finally, every time the algorithm converged to a local minimum, the density filter radius was decreased by 10%. The optimization was terminated when all the following criteria were met:

    - The density filter radius was equal to 0.

    - The number of elements with density between 0.20 and 0.90 dropped below

$2e-3$ of the total number of elements.

- The relative change of the objective function with respect to the previous optimization iteration was below $5e-4$.

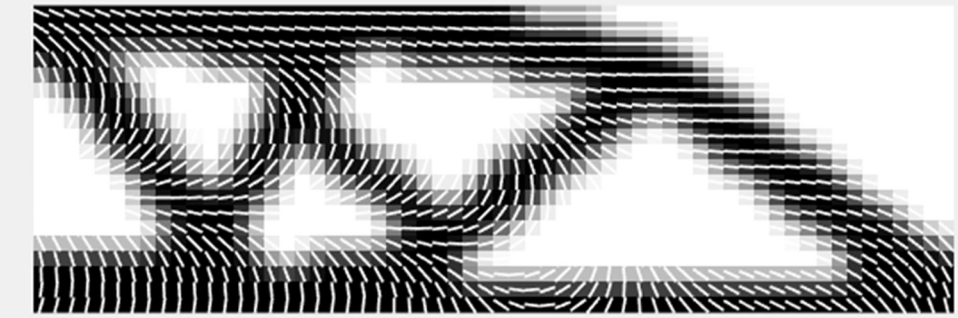


**Figure 4.9** Result without elimination of intermediate densities.

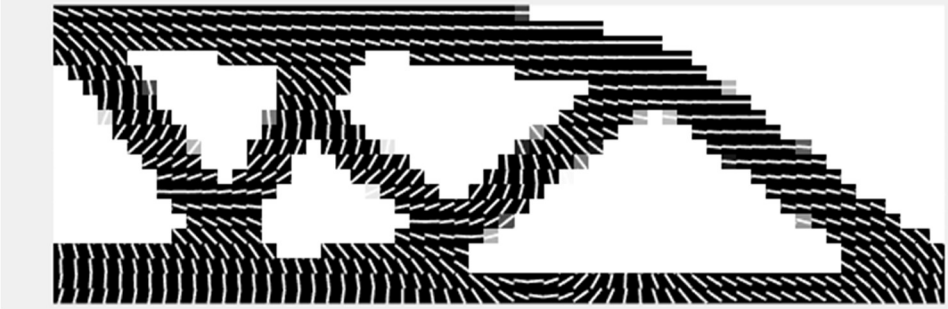


**Figure 4.10** Result with elimination of intermediate densities.

## 4.6. Flow Chart and Overview of the Solution Strategy

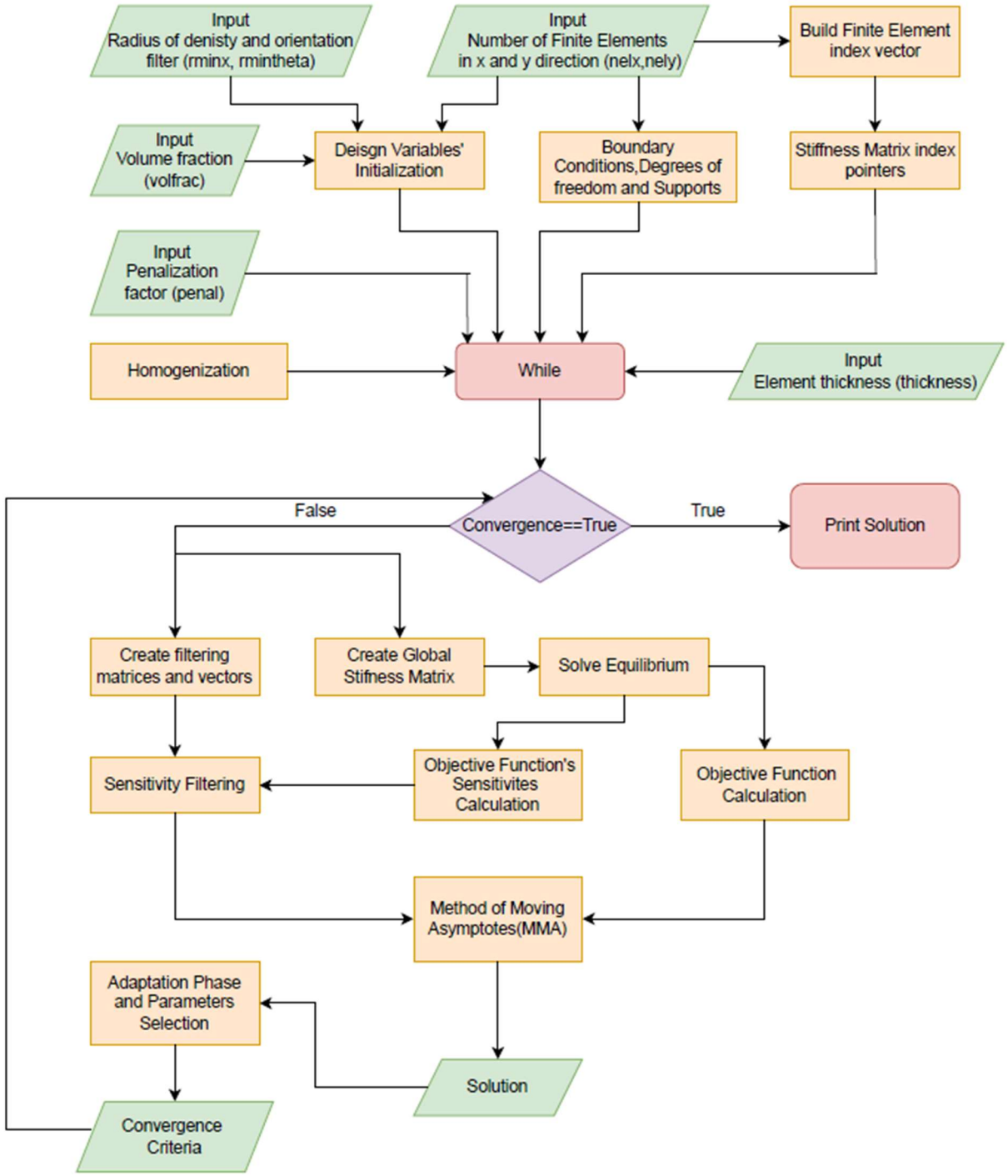



**Figure 4.11** Proposed Scheme's diagram flow.

The flow of the optimization algorithm can be summed up as follows:

***Initialization***

1. Calculate the homogenized elasticity tensor of the RVE corresponding to horizontal fibers $\mathbb{C}_{\boldsymbol{hom}}$.

2. Assign a random virtual orientation $\bar{\theta}_i$ between [0, $\pi$] and virtual density $\bar{\rho}i$ equal to the target volume fraction $f$ to every macroscopic element.

3. Set the radius of the orientation filter equal to 1. Set the radius of the density filter equal to 1. Set the density scope to $[0.4, 0.7]$. Set the value of penalty coefficient $p$ equal to 1. Set adaptation phase to "Orientation filter increase".

***Adaptation of optimization parameters***

4. Orientation filter increase: If adaptation phase is not set to "Orientation filter increase", skip this step. Check if a local minimum has been reached. If yes, increase the radius of the orientation filter $\|r_{orient}\|$ until a target value $\|r_{oreint}^{conc}\|$. If the target value is reached, set adaptation phase to "Density filter increase".

5. Density filter increase: If adaptation phase is not set to "Density filter increase", skip this step. Check if a local minimum has been reached. If yes, increase the radius of the density filter $\|r_{dens}\|$ until a target value $\|r_{dens}^{conc}\|$. If the target value is reached, set adaptation phase to "Density boundary expansion".

6. Density boundary expansion: If adaptation phase is not set to "Density boundary expansion", skip this step. Check if intermediate densities have dropped below a cutoff or a local minimum has been reached. If yes, expand the density scope until target [0.1, 1]. If the target boundaries are reached, set adaptation phase to "Penalty coefficient increase".

7. Penalty coefficient increase. If adaptation phase is not set to "Penalty coefficient increase", skip this step. Check if a local minimum has been reached. If yes, increase the penalty coefficient until a target value. If the target value is reached, set adaptation phase to "Elimination of intermediate densities".

8. Elimination of intermediate densities. If adaptation phase is not set to "Elimination of intermediate densities", skip this step. Check if a local minimum has been reached. If yes, decrease the radius of the density filter until 0.

***Adaptation of optimization parameters***

9. Apply the orientation and density filter to transform virtual variables $\overline{\boldsymbol{\theta}}$, $\overline{\boldsymbol{\rho}}$ to their physical counterparts $\boldsymbol{\theta}$, $\boldsymbol{\rho}$.

10. Calculate the elasticity tensor for every element according to its physical orientation $\theta_i$ and physical density $\rho_i$.

11. Calculate the value of the objective function $f$ as well as its derivatives $\frac{\partial f}{\partial \boldsymbol{\theta}}$, $\frac{\partial f}{\partial \boldsymbol{\rho}}$ with respect to physical variables $\boldsymbol{\theta}$, $\boldsymbol{\rho}$.

12. Calculate the derivatives of the objective function $\frac{\partial f}{\partial \overline{\boldsymbol{\theta}}}$, $\frac{\partial f}{\partial \overline{\boldsymbol{\rho}}}$ with respect to virtual variables $\overline{\boldsymbol{\theta}}$, $\overline{\boldsymbol{\rho}}$.

13. Use MMA to obtain the values of the virtual variables $\overline{\boldsymbol{\theta}}$, $\overline{\boldsymbol{\rho}}$ for the next iteration as a function of $f$, $\frac{\partial f}{\partial \overline{\boldsymbol{\theta}}}$, $\frac{\partial f}{\partial \overline{\boldsymbol{\rho}}}$.

***Termination and post-processing***

14. Check if the global minimum has been reached. If not, go to adaptation phase (step 4).

# Chapter 5

# 5. Numerical Examples

## 5.1. Example Description

In order to evaluate the results of the proposed scheme, two algorithms were created in C#. The first algorithm is an implementation of *'Efficient topology optimization in MATLAB using 88 lines of code'* [2] using MMA as an optimizer. This is a very computationally efficient algorithm of topology optimization with SIMP, taking advantage of MATLAB sparse matrices, matrix and vector operators. Furthermore, this algorithm performs as many procedures as possible outside of optimization loop, gaining significant computational advantage in comparison to other algorithms. Matrix and vector operators, matrix builders and sparse matrices were implemented with C#.

Computational efficiency is a matter that concerned us greatly during the design and the implementation of the proposed multiscale topology optimization scheme. Multiscale analysis, solving the equilibrium, Method of Moving Asymptotes, expressing objective function and its derivatives with respect to 2 design variables (plenty of matrix operations) could be very computationally demanding tasks if complexities were not kept to minimum levels. Therefore, a great effort was put into writing the code in the most computationally efficient way possible. Considering this, our simple Topology Optimization code had to be as optimized as possible.

In order to showcase the proposed methodology, a cantilever beam consisting of carbon nanotube-reinforced polymer was optimized using a multiscale scheme on two levels. Regarding the spatial distribution of the nanotube orientations at the microscopic level, and the topology at the macroscopic level. A nodal load of 1 kN on the top right node and pinned boundary conditions in the left side of the structure where applied. The structure was discretized with 2400 finite elements (60x40). The Young's modulus was taken equal to E=1GPa, and Poisson's ratio equal to v=0.3. The microscopic RVE consisted of a rectangular polymer matrix in which a single straight nanotube was embedded. The material configuration corresponded to approximately 0.5% nanotube volume fraction. The nanotube was modeled

using equivalent truss elements and the matrix using plane stress elements. The Young's modulus for the truss' material was set equal to $E$ = 100GPa.

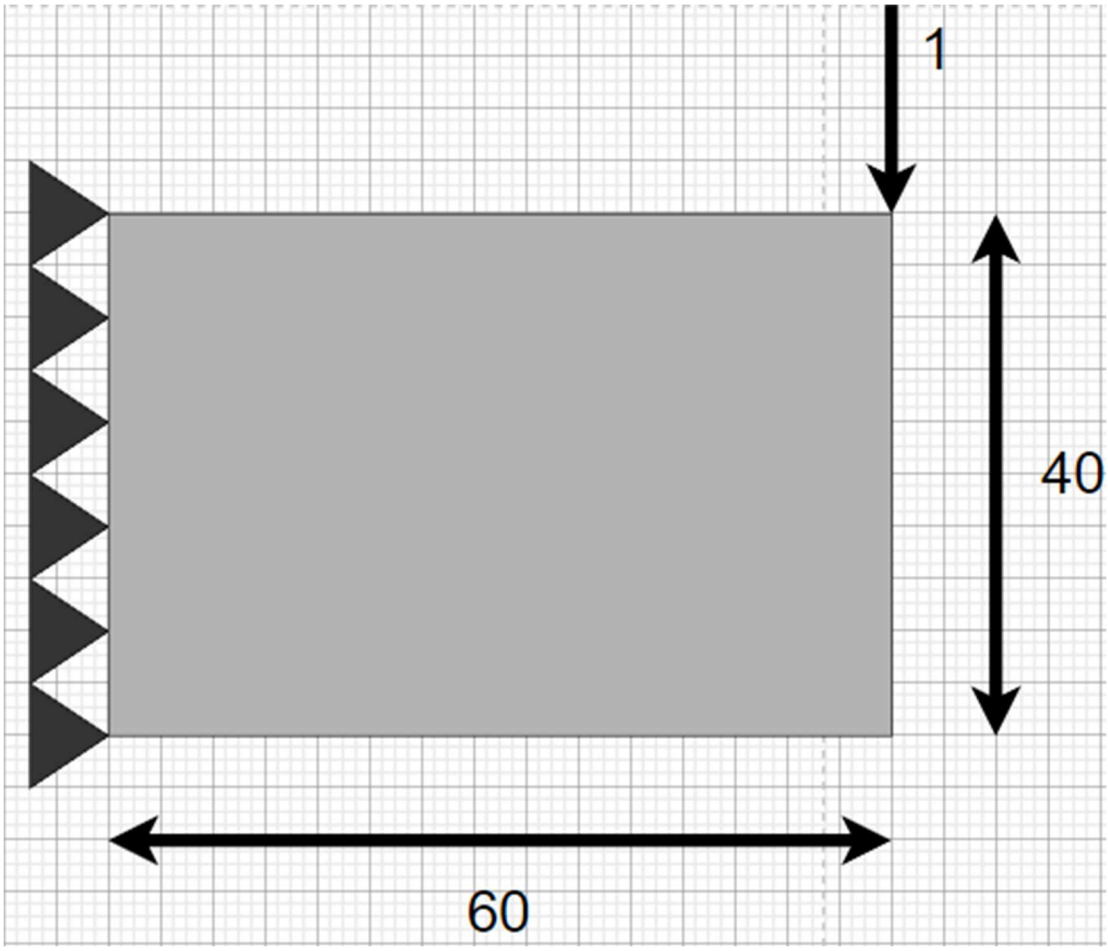



**Figure 5.1** Macroscopic structure

- Cantilever beam
- 60x40 finite elements
- Pinned boundary conditions in the left side
- Nodal load in the top right corner

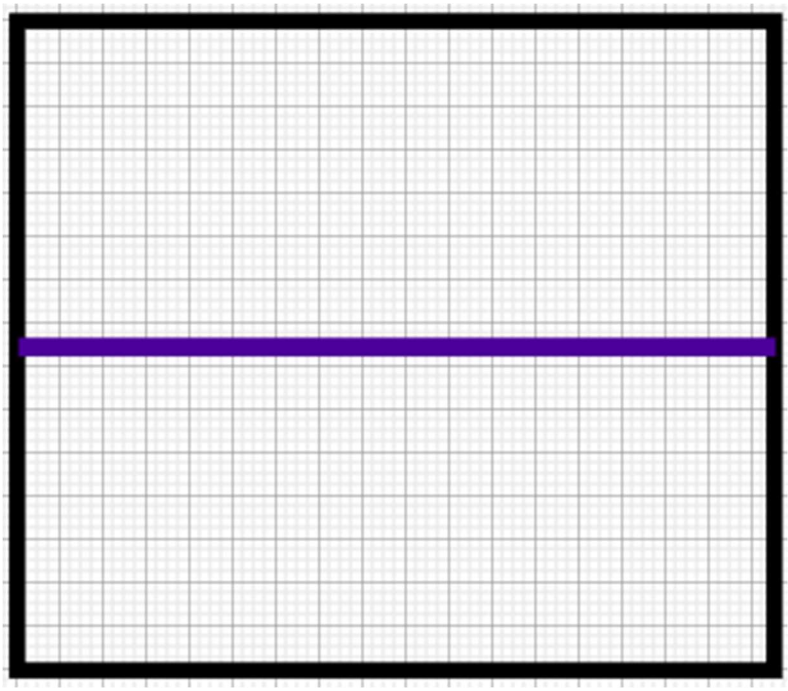


**Figure 5.2** Microscopic RVE

- Nanotube: equivalent truss elements
- Polymer matrix: plane stress elements

## 5.2. Topology Optimization results

**Vectorized SIMP using MMA in C# vs Sigmund's Algorithm (OC) in MATLAB**

Firstly, we compare the results of our Topology Optimization with SIMP code (using Method of Moving Asymptotes as optimizer) with the one Sigmund[3] proposed (using Optimality Criteria as optimizer). We needed to check the accuracy of our results, as well as our algorithm's computational efficiency. The parameters were set equal to both algorithms as follows: a prescribed volume fraction of 0.5, a penalization coefficient of 6 and a density filter radius of 2.

The resulted Topology from the two algorithms is similar and ours was much faster as we expected.

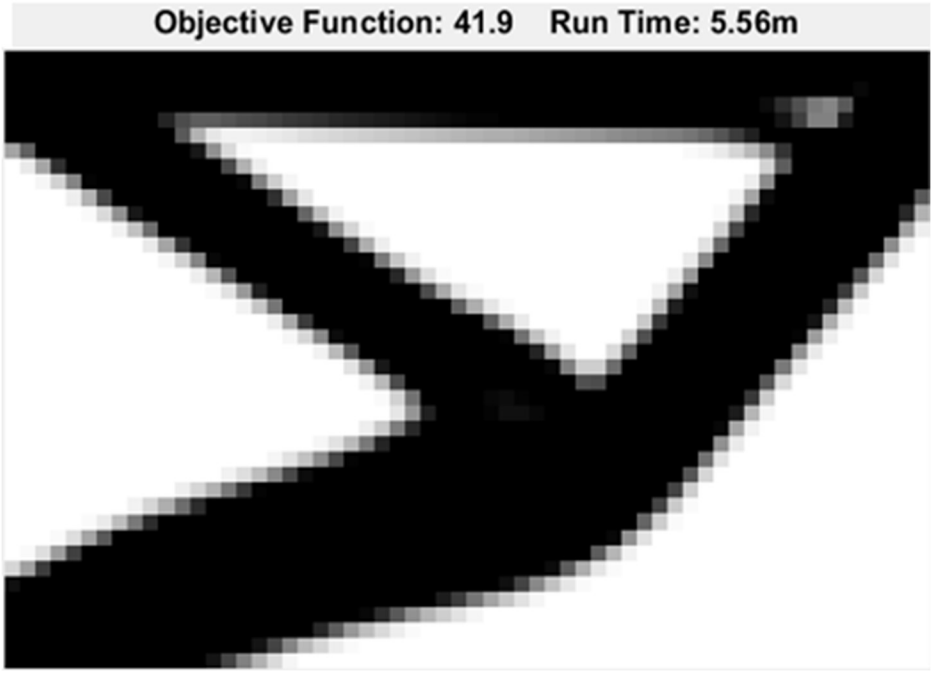



**Figure 5.3** Sigmund's Algorithm's resulted topology, objective function and run time values.

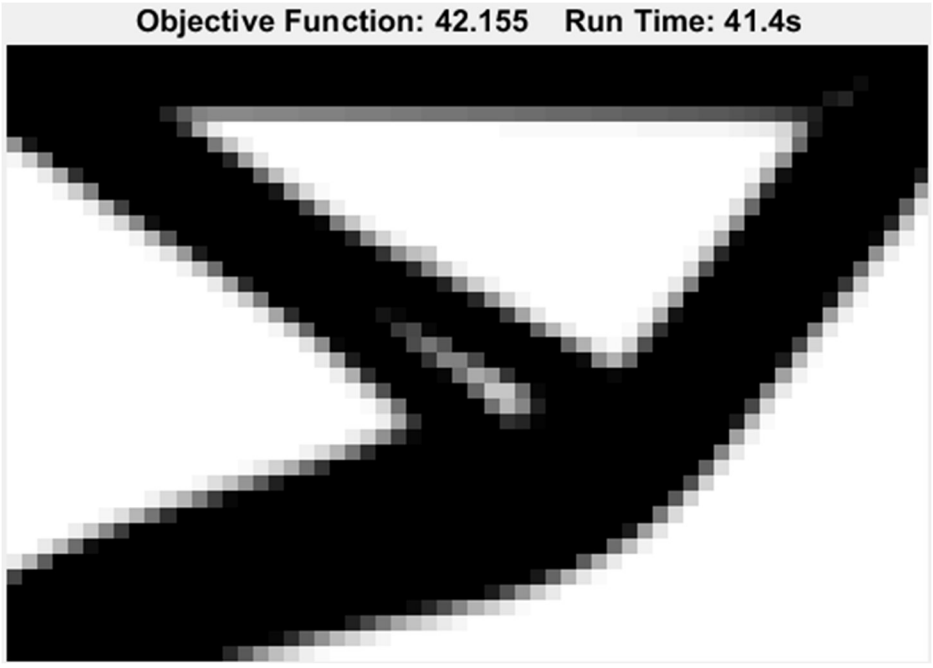
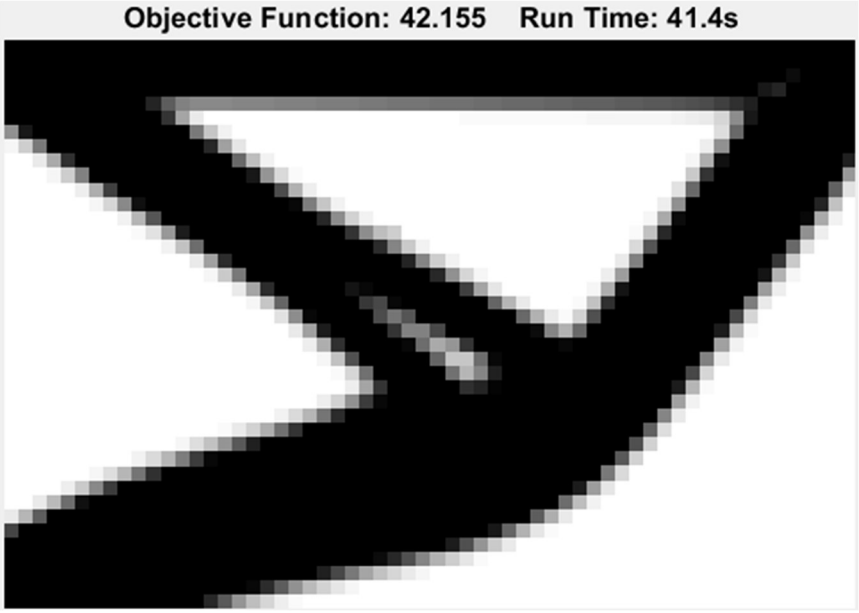


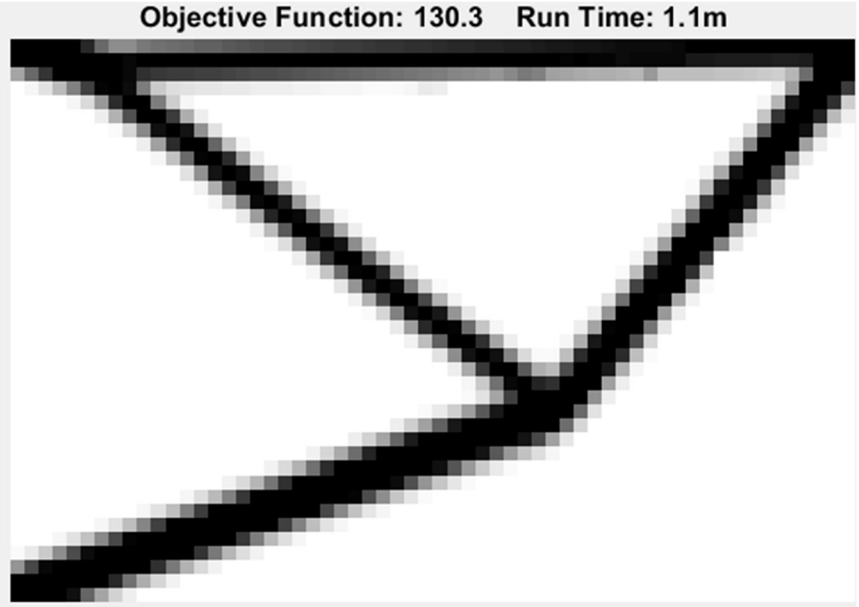

**Figure 5.4** Vectorized SIMP's (using MMA in C#) resulted topology, objective function and run time values.

**Volume fraction investigation**

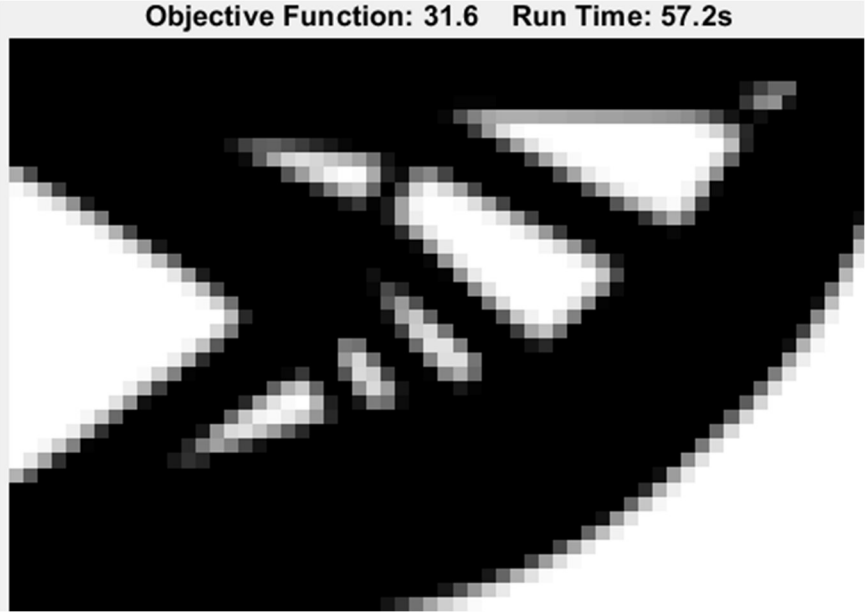
It is crucial to understand our simple Topology optimization code's sensitivity to problem's parameters. Firstly, we are going to investigate algorithm's behavior for different values of the prescribed volume fraction. We performed 3 different optimizations with volume fractions of 0.5, 0.2 and 0.7. For greater values of volume fraction, we obtained denser topologies and smaller values of the objective function as it was expected. Volume fraction's physical meaning is the percentage of material existing in the design domain and objective function is an expression of work performed from the external loads. This means that the greater objective function is, the less stiff our structure is. Thus, it is rational for smaller values of volume fraction to obtain greater values of the objective function, as the structure is constructed by less material and is more compliant. Run times was similar for all optimizations without a specific pattern for their insignificant variations.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2




- Volume fraction: 0.2
- Penalization coef.: 6
- Density filter radius: 2

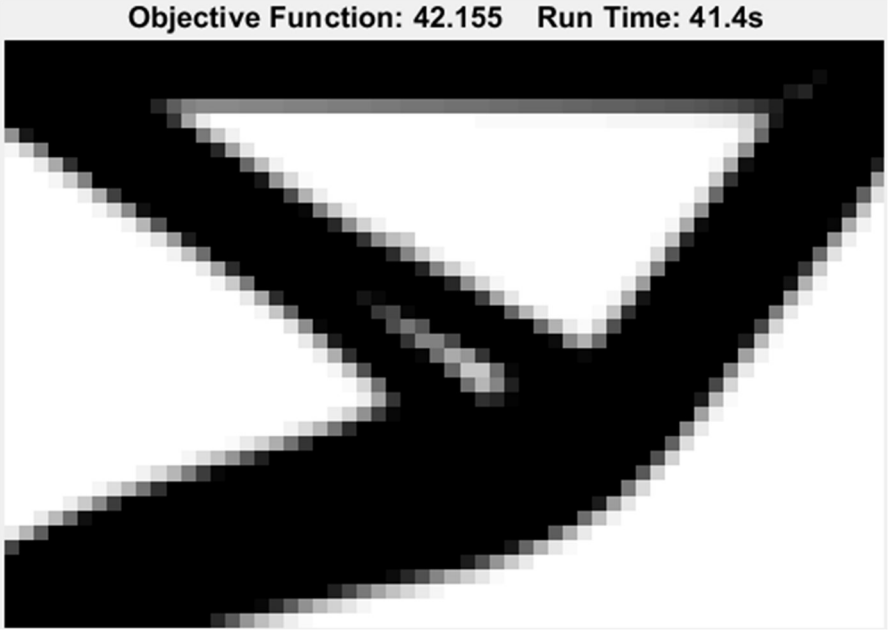



- Volume fraction: 0.7
- Penalization coef.: 6
- Density filter radius: 2

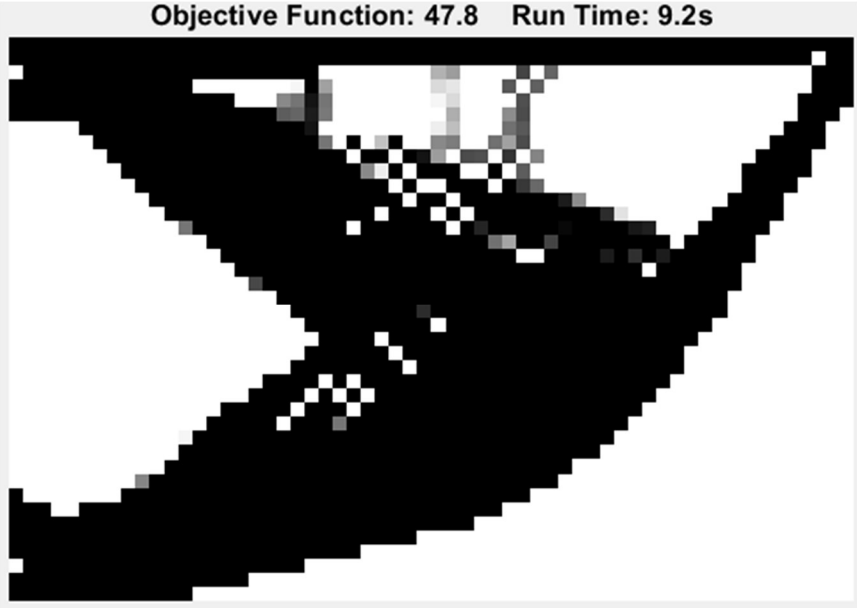
**Figures 5.5** Resulted topologies, objective function and run time values for different values of the volume fraction

**Density filter radius investigation**

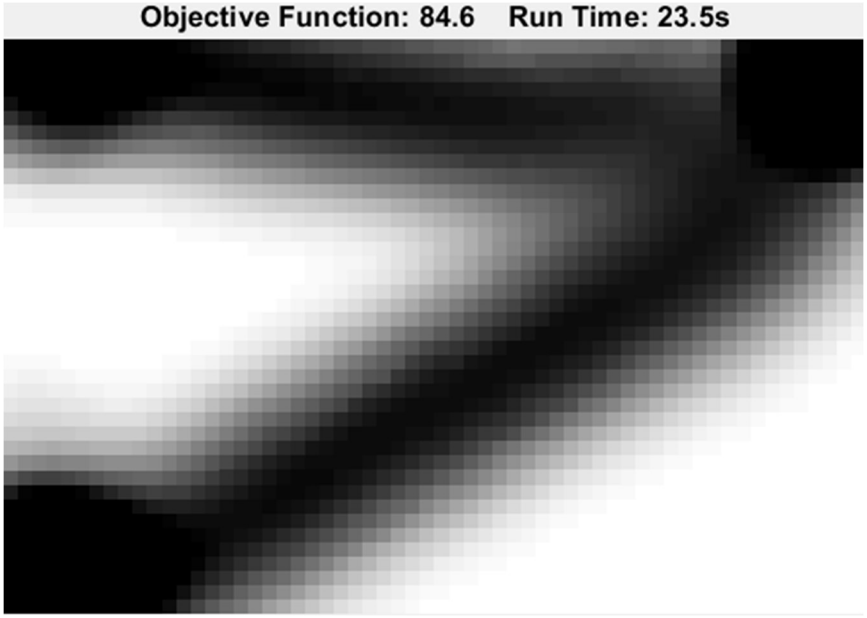
The second and very significant parameter we investigated is the density filter radius. We performed 3 different optimizations with density filter radiuses of 2,1 and 10. In the first occasion we obtained a clear, constructible topology. In the second example, where we practically have no filtering, we obtain a topology full of checkboard patterns as we anticipated (see chapter 4.4). In the third example, we obtain a very blur topology as a result of the big density filtering radius. This value means that each element's density is the weighted average of a large number of values. Thus, densities near 0 or 1 are very difficult to be reached and gray areas (intermediate densities) are extremely common.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 1

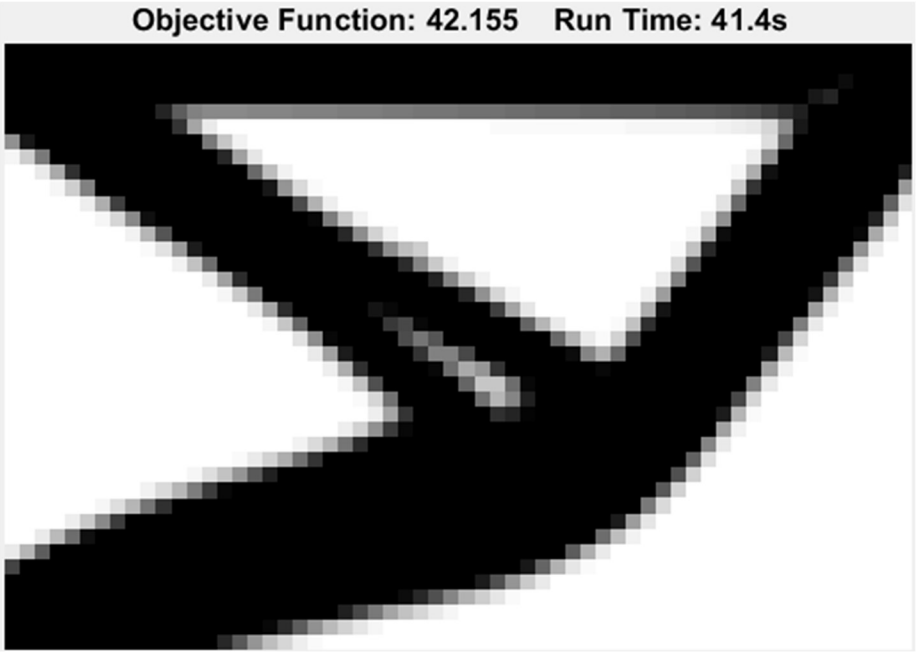



- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 10

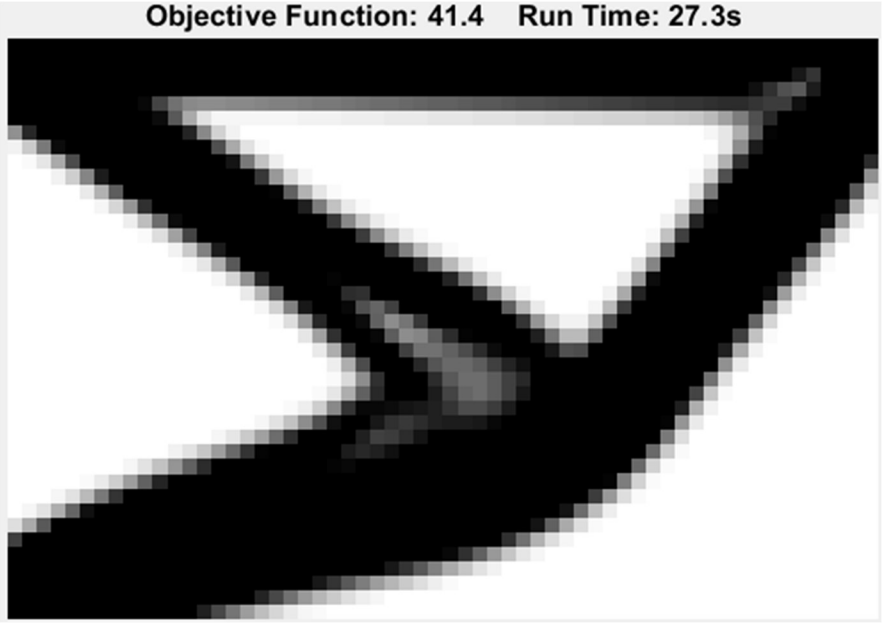
**Figures 5.6** Resulted topologies, objective function and run time values for different values of the density filter radius parameter

**Penalty coefficient investigation**

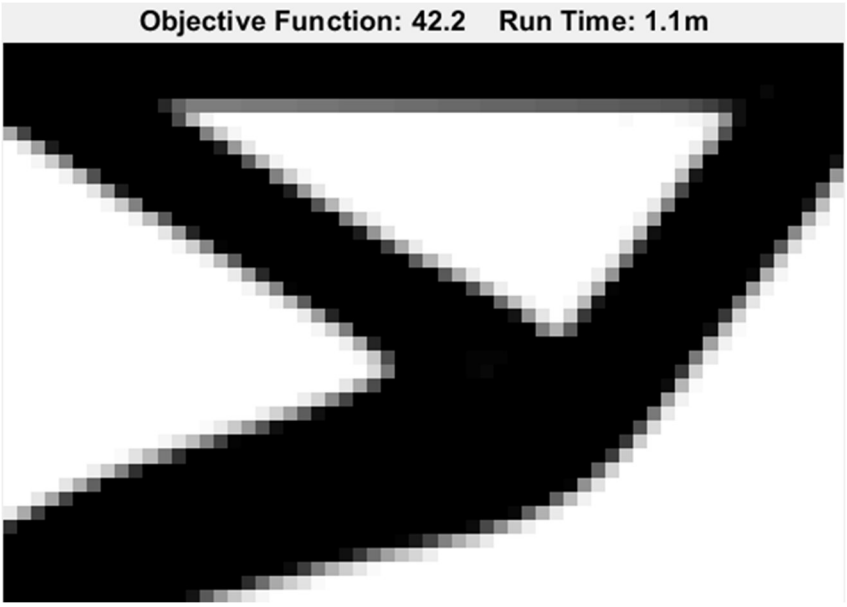
The final parameter we investigate is the penalty coefficient. We performed 3 optimizations for penalty coefficients of 6,3 and 10. As we have previously stated, penalty coefficient in SIMP dictates how much intermediate densities are penalized. Consequently, for greater values of penalty coefficient we expect topologies without many holes. On the contrary, for smaller values we anticipate topologies with wholes and plenty of truss-equivalent members. This is exactly what we obtained from the optimizations we performed. For the example with penalty coefficient equal to 10, our structure has 4 truss-equivalent members. For penalty coefficient equal to 6, our structure has 5 truss-equivalent members. Finally, for penalty coefficient equal to 3, our structure has 6 truss-equivalent members. Furthermore, the lower penalty coefficient was, the more elements received intermediate density values.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2




- Volume fraction: 0.5
- Penalization coef.: 3
- Density filter radius: 2




- Volume fraction: 0.5
- Penalization coef.: 10
- Density filter radius: 2

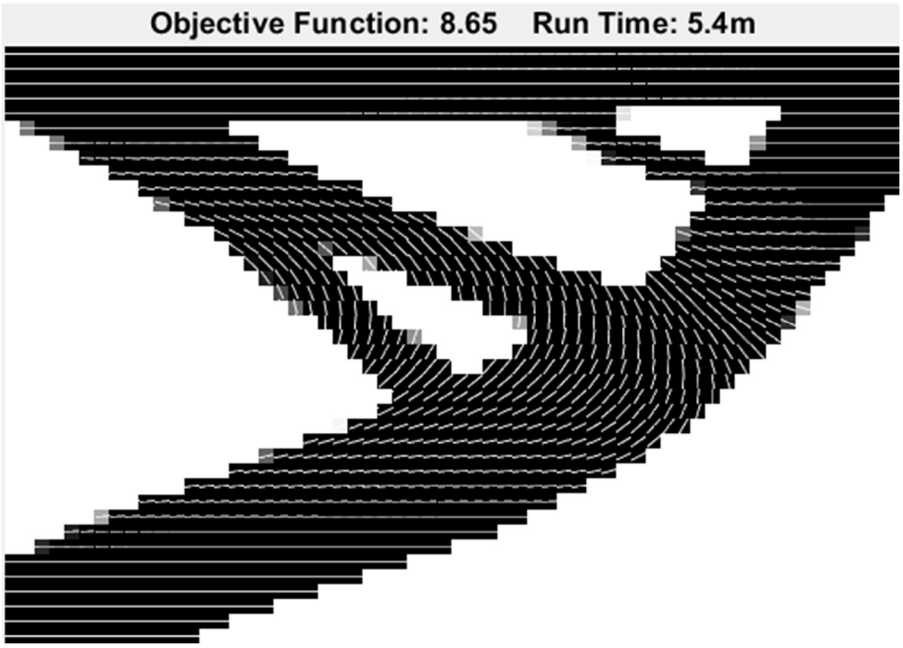
**Figure 5.7** Resulted topologies, objective function and run time values for different values of the penalty coefficient parameter

## 5.3. Multiscale Topology Optimization results

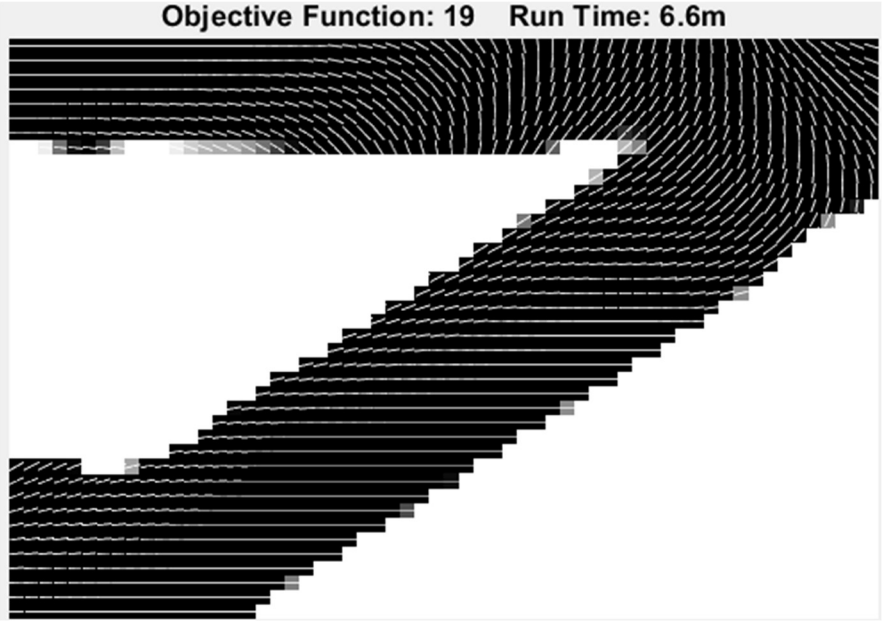
Before comparing the results of the proposed scheme to simple topology optimization, it is significant to discuss our scheme's sensitivity to problem's parameters. It is rational for our algorithm's behavior to differ from the one of Topology Optimization, as we have a different formulation, we perform adaptive filtering, we eliminate intermediate densities and we have multiple convergence criteria. Furthermore, we have an additional set of design variables (orientations) which distribution needs to be discussed for the variation of each parameter.

**Density filter radius investigation**

In order to investigate density filter radius' impact, we performed 2 different optimizations with density filter radiuses of 2 and 10. We observe that for greater values of density filter radius we obtain simple topologies without many truss-equivalent members, which means that the structure is easily constructible. However, the problem is that the great value of density filter radius does not permit elements to receive their optimal density value and therefore structure's compliance is much higher than the one of a more complicated structure with plenty of truss-equivalent members. This behavior of the objective function in respect to various values of density filter radius is observed in simple topology optimization as well.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2
- Orientation filter radius: 15



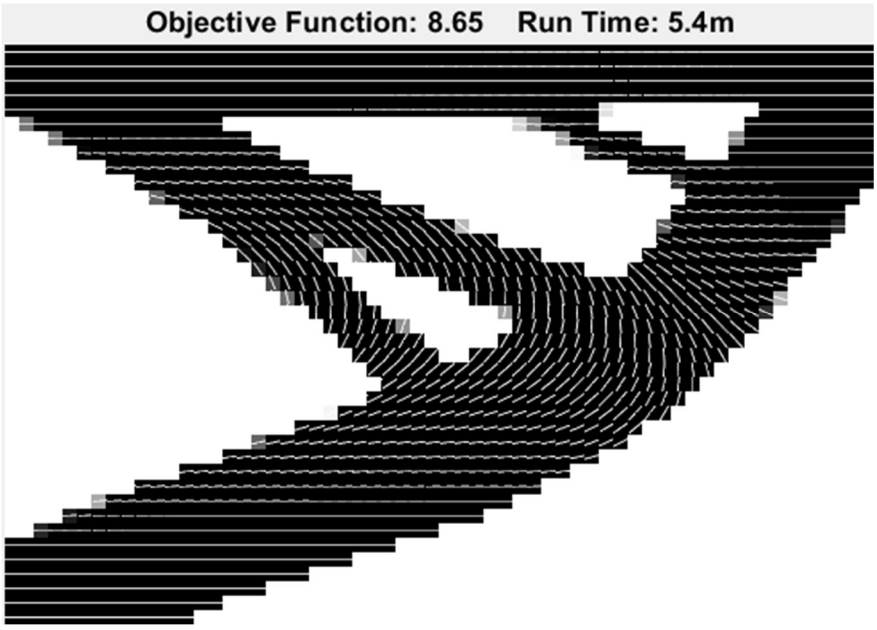

- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 10
- Orientation filter radius: 15

**Figure 5.8** Resulted topologies and orientations distribution, objective function and run time values for different values of the density filter radius parameter.

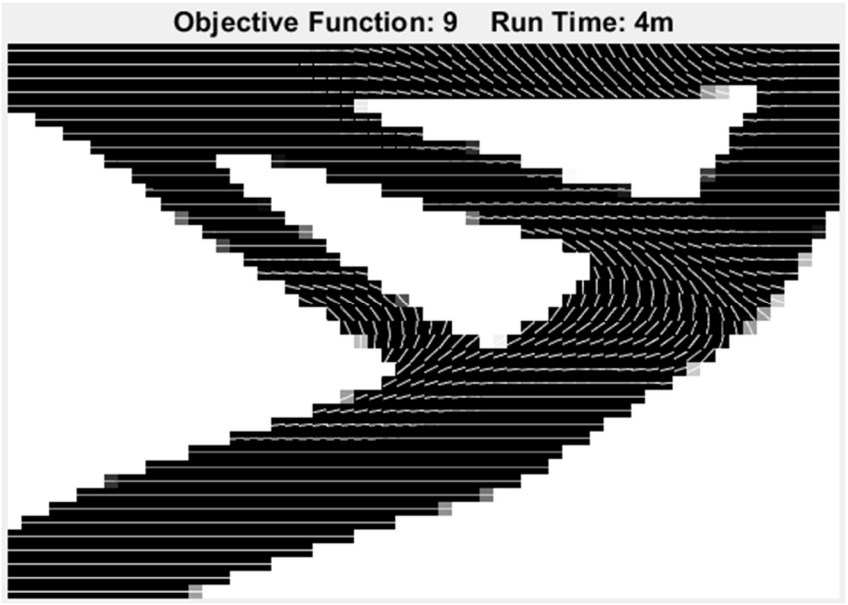
**Orientation filter radius investigation**

In order to investigate orientation filter radius' impact, we performed 2 different optimizations with orientation filter radiuses of 8 and 15. One of the first conclusions we reach from our investigation, is that much greater values are required for orientation filter radius to avoid orientation discontinuities than those required for density filter radius to avoid checkboard patterns. For the relatively low orientation filter radius value of 8, we have no orientation discontinuities, but we observe great relative change of orientations on some occasions (in the beginning and the end of truss-equivalent members). For greater values, we obtain a smooth orientation distribution which coincides with the slope of the truss-equivalent members in which the finite elements are located. A very interesting result is that for orientation filter radius of 15 we obtain a topology with an additional (small though) truss-equivalent member in comparison to the example with orientation filter radius of 8. It is important to mention that for greater values of both orientation and density filter radius the algorithm converges slower to the optimal solution. This happens because of the adaptive filtering procedure (see chapter

4.5). For example, in the occasion of orientation filter of 15 the algorithm converges at least to 7 additional local minima in comparison with the optimization with orientation filter of 8.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2
- Orientation filter radius: 15

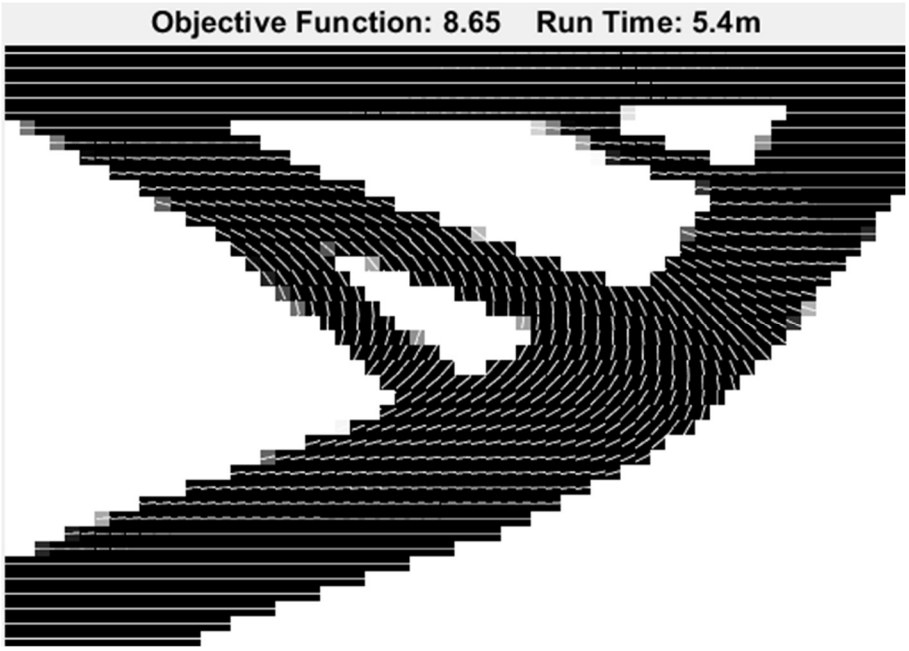



- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2
- Orientation filter radius: 8

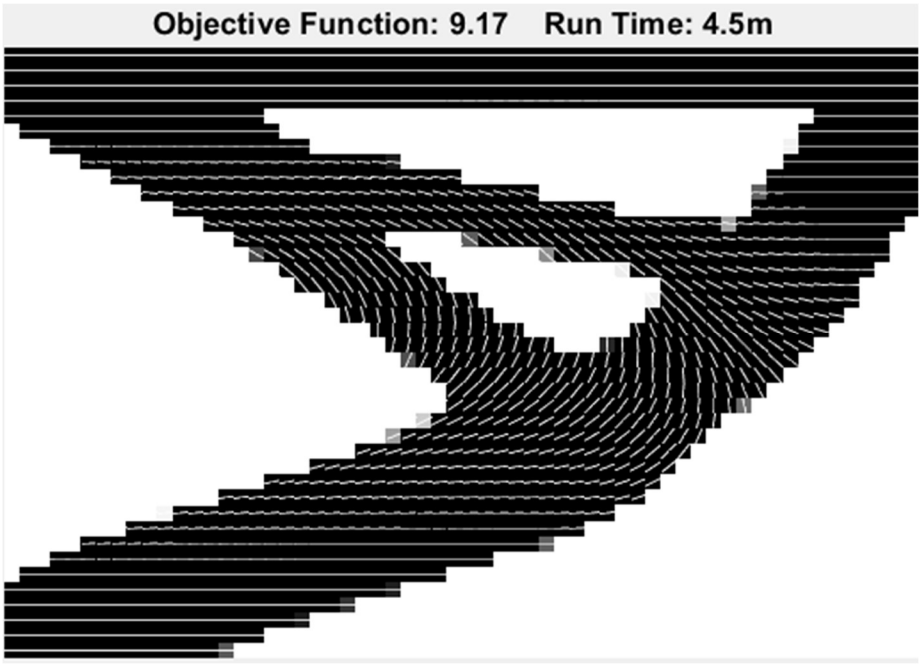
**Figure 5.9** Resulted topologies and orientations distribution, objective function and run time values for different values of the orientation filter radius parameter.

**Penalty coefficient investigation**

In order to investigate penalty coefficient's impact, we performed 2 different optimizations with penalty coefficients of 6 and 12. As we observed in the occasion of the Topology optimization, for greater values of penalization coefficient we obtain simpler topologies (less truss-equivalent members) and the convergence is faster. Orientations are identical in the two examples. Even though according to our adaptive filtering methodology our algorithm converges to at least 6 more local minima in the first optimization (penalty coefficient 12) in comparison to the second one (penalty coefficient 6), the first optimization is excecuted approximately 1 minute faster than the second one. This happens as a result of the faster convergence in every local minimum for higher values of the penalty coefficient.




- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2
- Orientation filter radius: 15




- Volume fraction: 0.5
- Penalization coef.: 12
- Density filter radius: 2
- Orientation filter radius: 15

**Figure 5.10** Resulted topologies and orientations distribution, objective function and run time values for different values of the penalty coefficient parameter.

## 5.4. Multiscale Topology Optimization vs Topology Optimization

It is very significant to compare the results we obtain by performing multiscale topology optimization to those of simple topology optimization in composite material structures. This way we can understand the value of the proposed scheme and the compromissions we need to make to use of it. Considering the results from previous investigations, we chose the values of the design parameters in order to use both algorithms in a very efficient way. We set constant values for density and orientation filter radius, penalty coefficient and we compared the two algorithm's results' objective functions for various values of volume fraction. We select the design parameters as follows:

Topology Optimization

- Penalty coefficient: 6
- Density filter radius: 2

Multiscale Topology Optimization

- Penalty coefficient: 12
- Density filter radius: 2
- Orientation filter radius: 15

Considering this and the previous investigations we reached to the following conclusions:

- Multiscale topology optimization uses composite material much more efficiently than simple topology optimization. For the same value of volume fraction (0.5), multiscale topology optimization resulted in a structure with objective function value of 9.17, while topology optimization's resulted structure's objective function valued 42.2. This means that for the same amount of material we obtain a much stiffer structure. To further support this observation, multiscale topology optimization's resulted structure's objective function, for a volume fraction of 0.3, has half the value of the objective function of the structure obtained by simple topology optimization for a volume fraction of 0.5. The following diagram of objective function value in respect to volume fraction depicts the efficiency of the proposed scheme.

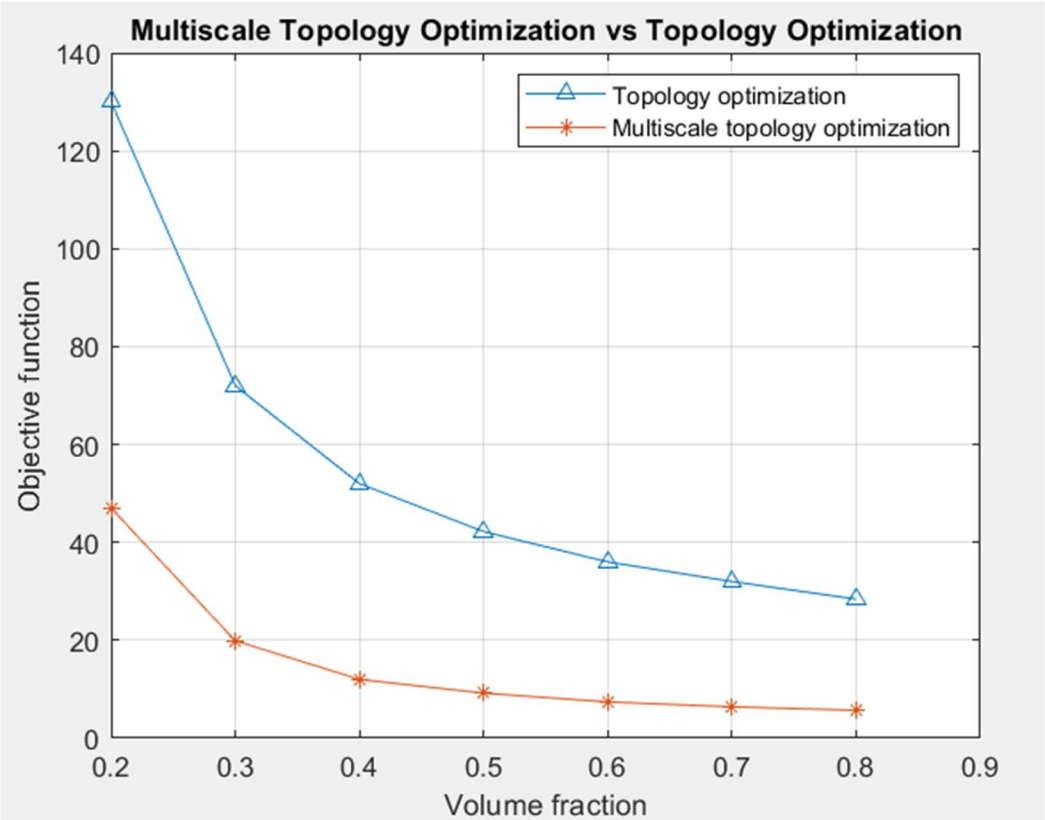
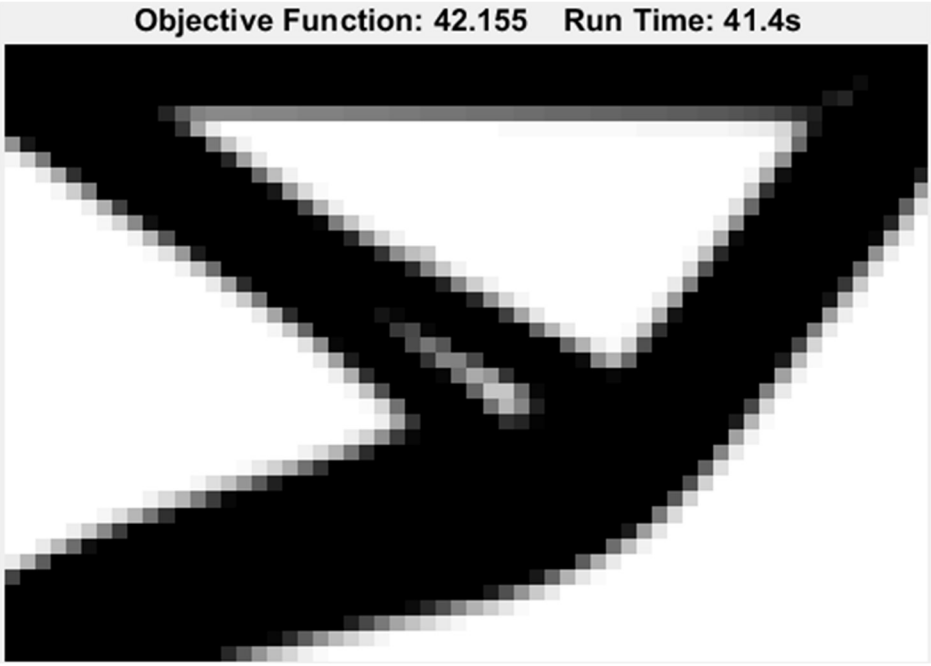


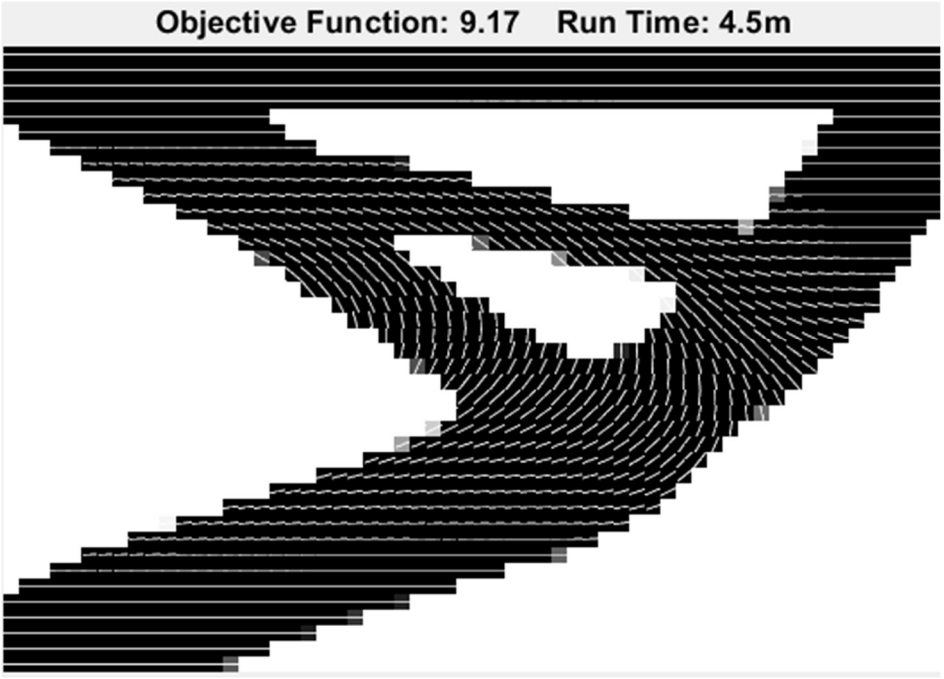

**Figure 5.11** Objective function values for different values of the volume fraction for multiscale topology optimization and topology optimization.

- Both methodologies demonstrate the same behavior in respect to the selection of the design parameters.

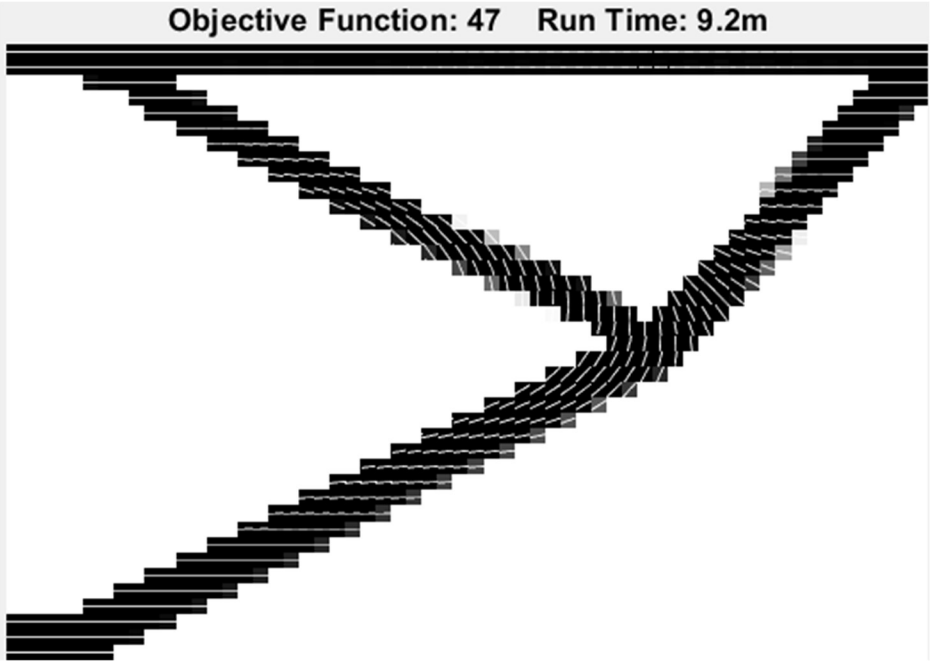
- Topology optimization algorithm is much faster than the multiscale topology optimization. For the same value of volume fraction (0.5), multiscale topology optimization converged in the optimal solution in 4.5m, while topology optimization resulted in the optimal solution in 41.4s. Obviously, multiscale topology optimization is much slower than topology optimization as a result of being a much more complicated procedure resulting to much stiffer structures.



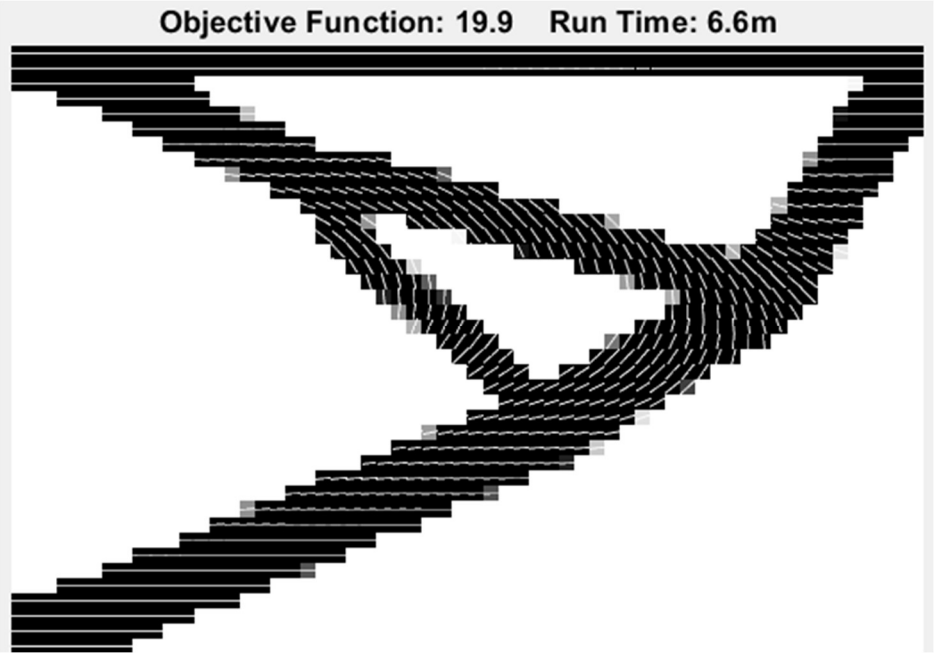

- Volume fraction: 0.5
- Penalization coef.: 6
- Density filter radius: 2




- Volume fraction: 0.5
- Penalization coef.: 12
- Density filter radius: 2
- Orientation filter radius: 15




- Volume fraction: 0.2
- Penalization coef.: 12
- Density filter radius: 2
- Orientation filter radius: 15




- Volume fraction: 0.3
- Penalization coef.: 12
- Density filter radius: 2
- Orientation filter radius: 15

**Figure 5.12** Comparison between multiscale topology optimization and topology optimization for different values of the volume fraction.

# Chapter 6

## 6. Conclusions, future work

We have developed an algorithm to optimize the topology and orientations for a fiber-reinforced composite structure. A two-level multiscale technique was used to derive the material elasticity tensor for horizontal fibers. The derived elasticity tensor was then rotated to simulate different fiber orientations. The optimization problem was described using the SIMP formulation enriched with the fiber orientations and was solved using the gradient-based Method of Moving Asymptotes. A convolution filter was used to eliminate localized discontinuities and checkerboard patterns. The optimization parameters were adapted throughout the optimization, in order to accelerate convergence, avoiding local minima at the same time. Finally, post-processing the densities and orientations eliminated grey material and orientation discontinuities in the optimized structure.

The optimization scheme was both stable and efficient, yielding truss-like structures with no checkerboard patterns, and smooth spatial distributions of the fiber orientations, that are easy to manufacture. Our algorithm's results showed that it is important to take fiber orientations into account when designing fiber-reinforced structures. Even for the small value of the fiber volume fraction that was studied, the objective function's value is significantly reduced. The proposed optimization scheme successfully integrated optimization with the multiscale methodology. It therefore opens the area of concurrent multiscale optimization of structures made of heterogeneous materials. Using a multiscale methodology significantly reduces the computational cost inherent in such analyses, and the proposed scheme showcases that optimization on multiple scales is possible. Using the multiscale methodology enabled us to describe the problem of optimizing the fiber orientations by introducing only one additional variable per macroscopic element.

As it was previously stated, our algorithm opens the area of concurrent multiscale optimization of structures made of heterogeneous materials. Therefore, many adjustments or different approaches could lead to more efficient composite structures. Firstly, our description of the problem can be enriched to optimize microstructural properties other than the orientation. For example, an optimization with respect to the nanotubes' structural properties for a nanotube-

reinforced structure. Moreover, considering that fiber in the RVE could be a curve instead of a straight line, we could optimize the curvature of the fiber in each finite element of the macrostructure. We may also consider that there are 2 fibers in the RVE optimizing the orientation of both fibers. It would be interesting to examine whether both fibers will have the same (just adding material to the fiber in the optimized direction) or different (stiffening another direction of the RVE) orientations. Finally, we may consider fiber volume fraction in the RVE as an additional design variable. Yet, it is rational that the algorithm will converge to the maximum value of volume fraction allowed from our problem formulation, because the higher the volume fraction of the stiffer material is, the stiffer the RVE will be. However, adding fiber material in some areas of the RVE's is more effective than in others. Consequently, we need to detect the optimal to distribute the fiber material in the RVE. This is the problem topology optimization addresses to, so we may perform topology optimization in the microstructure (RVE) to detect the optimal fiber material distribution. So, in this $(\text{Topology Optimization})^2$ formulation two optimizations are performed simultaneously, one in the macroscopic level distributing the material in a beam and one in the macroscopic level distributing the fiber material in the RVE.

# References


1. L. Siva Rama Krishna, Natrajan Mahesh and N. Sateesh, *'Topology optimization using solid isotropic material with penalization technique for additive manufacturing'*, (2017)

2. Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S. Lazarov, Ole Sigmund *'Efficient topology optimization in MATLAB using 88 lines of code'*, (2010)

3. Ole Sigmund, *'A 99-line topology optimization code written in MATLAB'*, (2001)

4. C. Miehe, A. Koch, *'Computational micro-to-macro transitions of discretized microstructures undergoing small strains'*, (2002)

5. Vissarion Papadopoulos, Maria Tavlaki, *'The impact of interfacial properties on the macroscopic performance of carbon nanotube composites. A FE2-based multiscale study'* , (2002)

6. Savvas Triantafyllou, *'Finite Element Methods course notes'*, (2021)

7. Gao X, Ma H, *'A modified model for concurrent topology optimization of structures and materials.'*, (2015)

8. Safonov AA, *'3D topology optimization of continuous fiber-reinforced structures via natural evolution method.'*, (2019)

9. Esposito L, Cutolo A, Barile M, et al. '*Topology optimization-guided stiffening of composites realized through Automated Fiber Placement.'*, (2018)

10. Garland A, Fadel G, '*Optimizing Topology and Gradient Orthotropic Material Properties Under Multiple Loads.'*, (2018)

11. Groen JP, Sigmund O., '*Homogenization-based topology optimization for high-resolution manufacturable microstructures.*', (2018)

12. Jonathan B. Russ and Haim Waisman, *'A novel elastoplastic topology optimization formulation for enhanced failure resistance via local ductile failure constraints and linear buckling analysis'*, (2021)

13. Krister Svanberg, *'The Method of Moving Asymptotes – A new method for structural optimization'*, (1987)

14. Vicente WM, Zuo ZH, Pavanello R, Calixto TKL, Picelli R, Xie YM. *'Concurrent topology optimization for minimizing frequency responses of two-level hierarchical structures'*, (2015)

15. Bendsøe MP, Sigmund O (Ole). '*Topology Optimization: Theory, Methods, and Applications'*, (2003)

16. Deng J, Chen W. '*Concurrent topology optimization of multiscale structures with multiple porous materials under random field loading uncertainty.'*, (2017)

17. Sivapuram R, Dunning PD, Kim HA. '*Simultaneous material and structural optimization by multiscale topology optimization.'*, (2016)

18. Kato J, Yachi D, Kyoya T, Terada K. '*Micro-macro concurrent topology optimization for nonlinear solids with a decoupling multiscale analysis.'*, (2018)

19. Kumar T, Suresh K. '*Simultaneous Optimization of Topology and Pointwise Constitutive Properties through Multi-Variable SIMP*', (2017)