Universitat Politècnica de València
Departamento de Ingeniería Mecánica y de Materiales

Master Thesis in Mechanical Engineering

2 LEVEL CONTINUOUS TOPOLOGY OPTIMIZATION IN STRUCTURAL MECHANICS

Presented by: D. Mikel Barral Poveda
Supervised by: Dr. D. Enrique Nadal Soriano
Co-supervised by: Dr. D. Juan José Ródenas García

Valencia, September 2018

Abstract

In the current industry, component development that satisfies the customer requirements evolves quickly. Therefore, the companies are asked for quick and efficient solutions for the optimization of their component models. In this sense, it is very useful to use a technique called Topology Optimization, which has been extensively used to optimize, at a structural level, the geometry of many components. By using this technique, a given amount of material is distributed in the best possible way in order to maximize the component's stiffness. The results obtained by this method allow the user to change the topology in a simple way, by removing or adding material (appearance or collapse of holes). However, this type of optimization entails a greater difficulty than other types of optimization. That has led to the development of specific methods, considering that the main goal of this optimization is the manufacturing of components using the additive manufacturing process. In order to alleviate this difficulty, the work proposes the combination of two techniques. The first one subdivide the whole component in cells and generates an optimized material structure in each cell. The second one is a balancing technique that ensures the continuity of tractions between cells, thus ensuring the material continuity after the cell optimization process. In this way the additive manufacturing is only required for the manufacturing of those cells and not for the whole component, thus being able to manufacture larger components. The results obtained by using this method prove its viability and the continuity at cell level.

Resumen

En la industria actual, el desarrollo de componentes que satisfagan las necesidades de los usuarios evoluciona rápidamente. Por ello, las empresas se ven obligadas a generar modelos óptimos para los componentes de sus productos de manera rápida y eficaz. En este sentido, es de gran utilidad el uso de una técnica denominada Optimización Topológica, que ha sido usada para optimizar, a nivel estructural, la geometría de diversos componentes. En esta técnica se busca la distribución óptima de una cantidad de material prescrita por el analista de modo que se maximice la rigidez del componente. Los resultados son mucho más ricos permitiendo cambios topológicos importantes (aparición o colapso de agujeros) de manera sencilla. No obstante, el proceso de optimización, es decir, la búsqueda de la distribución de material óptima, conlleva una mayor dificultad que otros tipos de optimización, lo que ha forzado el desarrollo de métodos específicos. Además, las estructuras resultantes requieren ser fabricadas mediante fabricación aditiva debido a su complejidad, que si la estructura es grande puede resultar costoso. Para paliar esta dificultad, en este trabajo se propone un método que, en vez de obtener la estructura optimizada del componente en su totalidad, subdivide el componente en celdas de tamaño manejable y son éstas las que son optimizadas y se plantea un método que tiene como objetivo garantizar la continuidad de la estructura entre las celdas basado en técnicas de equilibrado utilizadas en procesos de acotación del error de discretización. De este modo, la fabricación aditiva sólo es necesaria para la construcción de las celdas y no para el componente entero, pudiendo así llegar a fabricar componentes de mayores dimensiones. Los resultados obtenidos demuestran la viabilidad del método y la continuidad de la estructura.

Resum

En la indústria actual, el desenvolupament de components que satisfacen les necessitats dels usuaris evoluciona ràpidament. És el motiu pel qual les empreses es veuen obligades a generar models òptims per als components dels seus productes de manera ràpida i eficaç. En aquest sentit, és de gran utilitat l'ús d'una tècnica denominada Optimització Topològica, que ha sigut usada per a optimitzar, en l'àmbit estructural, la geometria de diversos components. En aquesta tècnica es busca la distribució òptima d'una quantitat de material prescrita per l'analista de manera que es maximitze la rigidesa del component. Els resultats són molt més rics permetent canvis topològics importants (aparició o col·lapse de forats) de manera senzilla. No obstant això, el procés d'optimització, és a dir, la cerca de la distribució de material òptima, comporta una major dificultat que altres tipus d'optimització, la qual cosa ha forçat el desenvolupament de mètodes específics. A més, les estructures resultants requereixen ser fabricades mitjançant fabricació additiva a causa de la seua complexitat, si l'estructura és gran, pot resultar costós. Per a pal·liar aquesta dificultat, en aquest treball es proposa un mètode que, en comptes d'obtindre l'estructura optimitzada del component íntegrament, subdivideix el component en cel·les de grandària manejable i són aquestes les que són optimitzades i es planteja un mètode per a garantir la continuïtat de l'estructura entre les cel·les fonamentat en tècniques d'equilibrament utilitzades en processos d'acotació de l'error de discretització. D'aquesta manera, la fabricació additiva només és necessària per a la construcció de les cel·les i no per al component sencer, podent així arribar a fabricar components de majors dimensions. Els resultats obtinguts demostren la viabilitat del mètode i la continuïtat de l'estructura.

Summary

Abstract I

Resumen III

Resum V

Summary VII

1 Introduction 1

2 Problem Statement 4

2.1 Linear Elastic Problem 4

2.2 Topology optimization problem 5

3 The proposed 2-level topology optimization technique 7

3.1 Traction equilibrium between elements 8

3.1.1 Internal nodes 10

3.1.2 Nodes with Dirichlet boundary conditions 13

3.1.2.1 Corner node 13

3.1.2.2 Standard case 14

3.1.3 Nodes with Neumann boundary conditions 15

3.1.3.1 Corner node 15

3.1.3.2 Standard case 16

3.2 Developed program structure 18

4 Implementation and numeric results 19

4.1 Problem 1 20

4.2 Problem 2 25

5 Conclusions 30

References 32

Appendix. Programmer's manual. 35

1 Introduction

From the industrial point of view, the optimization of mechanical components has been, and currently is, a challenge for the engineers. In this work the optimization should be understood as an appropriate distribution of certain amount of material in a given domain, to maximize the stiffness of the component, considering predefined loads. Since ancient times, engineers have successfully found solutions to an enormous amount of problems generated by the needs of the constant evolution of the society. Most outstanding structures can be found on bridges (Figure 1) and cathedrals, also the nature has developed very efficient structures, like the internal structure of the bones (Figure 2). All the optimized structures designed by the human being are the result of experience among years and development of new technologies.

Figure 1: Lattice beam of a bridge

Nowadays, numerical simulation of structural problems considerably helps in decision making along the design phase of complex structures. These numerical methods, e. g. the Finite Element Method (FEM), allow us to know the structural response to given external loads in a very detailed way. In that way the decisions can be made when talking about material types and its own design before the construction. It is easy to find out that the FEM has helped the development of new increasingly complex structures, by streamlining the design phase. However the distribution of the material is not yet optimized, thus it depends on the decisions made by the engineer. This is why a third step in this regard is the use of automated structural optimization techniques. These optimization techniques use numeric simulation (FEM), not only evaluating the structural behavior, but to propose new modifications in the material distribution. Therefore a complete new range of possibilities opens in front of the engineer. By exploring and studying these different new structures the engineer will be capable of making a better decision.

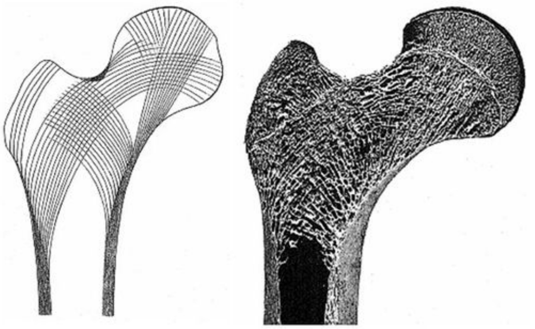


Figure 2: Internal structure of a femour bone [1]

In this sense there are currently two clearly differentiated optimization techniques. The first one defines a base structure parametrically (dimensions, shape,...) [2]. The algorithm will decide the value of each parameter of the structure in order to get the best stiffness with the minimum required material, while preserving the topology of the structure. The second type of optimization technique, known as Topology Optimization was mainly developed by Professor O. Sigmund [3], precursor of the SIMP method, (Solid Isotropic Material with Penalization), which even allowed to obtain topologically optimized structures for an aircraft wing [4].

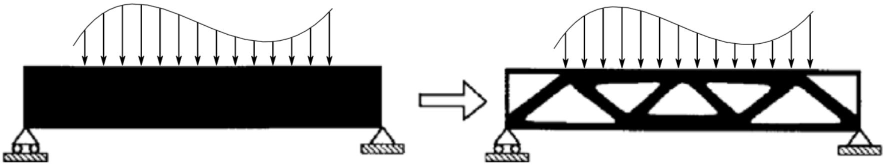


Figure 3: Topology optimization example

However, the previously mentioned methods are based on a single scale study. The SIMP method produces an optimal topology by means of an iterative process that involves the use of the FEM to evaluate the results needed to update the material distribution at each step. Nevertheless, the work done by Professor Oliver [5] and his co-workers deals with a new dimension in this problem. In this case the topology optimization is done in two different levels. This means that the microstructure is the one that is optimized and not the macrostructure as in Figure 3. Precisely, the information obtained from a stress analysis in each of the integration points of the macro scale is used to define a local (micro) problem that is solved satisfying the stress requirements of the global (macro) problem. The topology obtained at the microscale is processed by a homogenization process and the properties of the equivalent material are used to define the material at each integration point for solving the macro problem. This iteration continues until a convergence criteria is fulfilled. This process indeed has a high computational cost, this is why the authors of this method propose a new method based on precalculating all the problems in the micro scale. However, the solution obtained by this method is difficult to manufacture because the method doesn't take care of the continuity between the cells.

Therefore, the objective of this work is to develop a two level topology optimization process that provides continuous material distributions across the cells of the global level.

The proposed method initially solves the global optimization problem using the SIMP technique which distributes the material in a soft way through the analyzed area. After that, also using the SIMP technique the optimization problem in a local (micro) scale is solved. The boundary conditions of this problem are obtained considering a stress equilibration technique, introduced by Professor Ladevèze and Professor Maunder [6], that is used to guarantee the continuity of the topology between local cells.

2 Problem Statement

In this section, the two main problems presented in this work will be considered, the linear elastic problem and the topology optimization problem.

2.1 Linear Elastic Problem

This work is devoted to solve the 2D linear elasticity problem by means of the FEM. The notation used all along the contribution is settled in this section. The Cauchy stress field is denoted as $\boldsymbol{\sigma}$, the displacement field as $\mathbf{u}$, and the strain field as $\boldsymbol{\varepsilon}$, all these fields being defined over the domain $\Omega \subset \mathbb{R}^3$, with boundary denoted by $\partial\Omega$. Prescribed tractions denoted by $\mathbf{t}$ are imposed over the part Γ_N of the boundary, while displacements denoted by $\bar{\mathbf{u}}$ are prescribed over the complementary part Γ_D of the boundary. Body loads are denoted as $\mathbf{b}$.

The elasticity problem takes the following form. We seek $(\boldsymbol{\sigma}, \mathbf{u})$ satisfying:

- Static admissibility:

$$\mathbf{L}^T\boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \qquad \text{in } \Omega \tag{1}$$

$$\mathbf{G}\boldsymbol{\sigma} = \mathbf{t} \qquad \text{on } \Gamma_N, \tag{2}$$

where $\mathbf{L}$ is the differential operator defined as:

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{3}$$

and $\mathbf{G}$ is the projection operator that projects the stress field into traction over the boundary. The operator $\mathbf{G}$ is the matrix form of Cauchy's law considering the unit normal $\mathbf{n} = \{n_x \; n_y\}^T$ to Γ_N such that:

$$\mathbf{G} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix}. \tag{4}$$

- Kinematic admissibility:

$$\mathbf{u} = \bar{\mathbf{u}} \qquad \text{on } \Gamma_D. \tag{5}$$

- Constitutive relation:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}), \text{ with } \boldsymbol{\varepsilon}(\mathbf{u}) = \mathbf{L}\mathbf{u} \qquad \text{in } \Omega, \tag{6}$$

  where the matrix $\mathbf{D}$ contains the elasticity coefficients of the usual linear isotropic constitutive law relating the stress field with strain field.

The problem above takes the primal variational form:

$$\begin{aligned}
&\text{Find } \mathbf{u} \in (V + \{\mathbf{w}\}) : \forall \mathbf{v} \in V \\
&a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \qquad \text{where} \\
&a(\mathbf{u}, \mathbf{v}) = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{v}) \, \mathrm{d}\Omega \\
&l(\mathbf{v}) = \int_\Omega \mathbf{b}^T \mathbf{v} \, \mathrm{d}\Omega + \int_{\Gamma_N} \mathbf{t}^T \mathbf{v} \, \mathrm{d}\Gamma,
\end{aligned} \tag{7}$$

where $V = \{\mathbf{v} \mid \mathbf{v} \in \left[H^1(\Omega)\right]^3, \mathbf{v}|_{\Gamma_D} = \mathbf{0}\}$ and $\mathbf{w}$ is a particular displacement field satisfying the Dirichlet boundary conditions. The problem raised in (7) is solved using FEM after the discretization of the domain $\boldsymbol{\Omega}$, in elements of characteristic size h.

2.2 Topology optimization problem

The topology optimization problem is based on getting the best distribution of a given amount of material in a defined domain Ω, so that the stiffness of the component obtained is maximized. With this statement, a new variable is defined, $\rho(\mathbf{x}) \in [0, 1] \forall \mathbf{x} \in \Omega$, which will determine the material distribution inside the domain. This means that the elements with value equal to 1 will be filled with material, on the other side, the elements with value equal to 0 will be empty. Initially the variable ρ would take 0 or 1 values, but the SIMP method allows the variable to be continuous and therefore to acquire values between 0 and 1, even though the intermediate values are penalized. Considering that, the optimization problem can be stated as follows:

$$\begin{aligned} &\min \psi(\rho, \mathbf{u}) = \int_\Omega \rho^\nu \varepsilon(\mathbf{u}(\rho))^T \mathbf{D} \varepsilon(\mathbf{u}(\rho)) \, \mathrm{d}\Omega \\ &\text{Subject to: } \; V = \int_\Omega \rho \, \mathrm{d}\Omega \; \leq \; V_0 \end{aligned} \tag{8}$$

With ν defined as the penalty coefficient used to penalize intermediate values of ρ, $\mathbf{u}$ the displacement solution form problem (7) for a determined ρ field value, V_0 the material volume to be distributed. The problem in (8) is a nonlinear problem that requires advanced methods like the SIMP method [3] for solving it. The SIMP method follows the following steps:

1. An initial value is set for ρ_0, fulfilling the volume restriction.
2. The problem in (7) is solved considering the defined density, by redefining $a(\mathbf{u}, \mathbf{v}; \rho) = \int_\Omega \rho^\nu \varepsilon(\mathbf{u})^\mathbf{T} \mathbf{D} \varepsilon(\mathbf{v}) \, \mathrm{d}\Omega$.
3. The sensibility of the functional ψ with respect to the variation of ρ is evaluated $\frac{\partial \Psi(\rho; \mathbf{u})}{\partial \rho_e}$. For the topology optimization problem (8) to have solution, the sensibility problem must be filtered as described in [3], affecting the element value by the surrounding ones to a maximum distance of r_{min}.
4. The value of ρ is updated in each element e with the following criteria:

$$\rho_e^{new} = \begin{cases} \max(\rho_{min}, \rho_e - m) \text{ if } \rho_e B_e^\eta \; \leq \; \max(\rho_{min}, \rho_e - m) \\ \rho_e B_e^\eta \text{ if } \max(\rho_{min}, \rho_e - m) \; < \; \rho_e B_e^\eta \; < \; \min(1, \rho_e + m) \\ \min(1, \rho_e + m) \text{ if } \min(1, \rho_e + m) \; < \; \rho_e B_e^\eta \end{cases} \tag{9}$$

   Being $m = 0.2$ and $\eta = 0.5$ standard values of the parameters used to control the method. B_e is known as the factor multiplying the current density of the element in order to obtain a new density, so that the initially defined material volume is maintained:

$$B_e = -\frac{\frac{\partial \Psi(\rho; \mathbf{u})}{\partial \; rho_e}}{\lambda \frac{\partial V(\rho)}{\partial \rho_e}}, \tag{10}$$

   where λ is the Lagrange multiplier, common for all elements, whose value is found by the bisection method.
5. With the new density values in each element the procedure returns to step 2 and the whole process is repeated until a convergence criteria is satisfied.

3 The proposed 2-level topology optimization technique

As shown in the method, a certain material value V_0 is defined into the domain. However the analyst does not know a-priori the minimum amount of material required to withstand the loads applied on the component. In this work we consider that the maximum von Mises stress must be less than the considered material's yield stress, $\sigma_{vm}^{max} \leq S_y$. The value of V_0 can be calculated using the iterative process shown below.

1. Two different density values are assumed, commonly $V_1 = 1$ and $V_2 = 0.5$, and the topology optimization program gets the material distribution for each of them. The maximum value of Von Mises stress will be obtained, as shown below (represented by black dots) in Figure 4, for each of these two cases.

2. With those two values and by intersecting the function obtained with the yield stress (red line), a new value can be obtained. The new density value is then used to optimize the topology and again the maximum stress for the solution obtained is calculated, this allowing to correct the position of the value of V that would provide a material distribution where $\sigma_{vm_{max}} = S_y$.

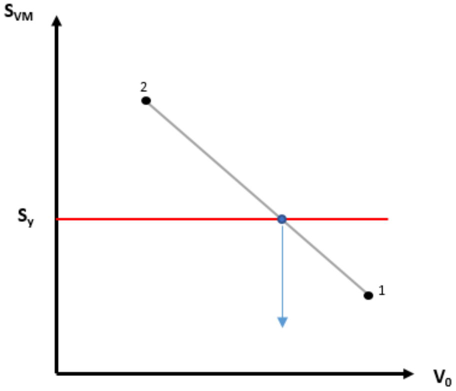


Figure 4: Linear interpolation for initial material volume.

3. Once the third point is completely defined, by intersecting the parabolic function created with those three points and the yielding stress value, a fourth value is obtained. After that, the process for correcting the position of the new value is applied as explained in step 2.

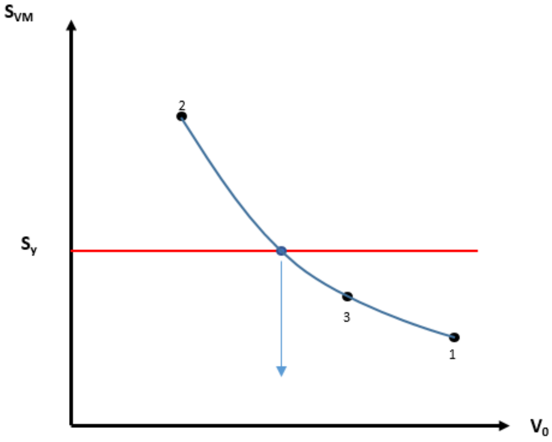


Figure 5: Quadratic interpolation for initial material volume.

4. The loop continues using the last density value and the two closest values to that last one, to get a new density value as explained before and, therefore, repeating step 3 until a convergence criteria is satisfied.

This process allows to obtain the optimum topology using the minimum required material volume that satisfies the Von Mises stress criteria, under given load situations.

For all the calculations performed with the FE method a cartesian mesh is used to discretize the domain, using quadrilateral elements.

3.1 Traction equilibrium between elements

As explained in the introduction, a topology optimization process is done in the Ω global domain, but another one is also done in each of the elements used to discretize the domain Ω. The loads for each of the local topology optimization problems are obtained from the global solution. The fastest way to do that would probably be to use the obtained stress from the FEM analysis, as presented in [5]. However, this method present many difficulties if a continuous stress field between elements is desired, and also causes local stress concentration due to non-equilibrated stress conditions. This thesis presents the alternative of generating an equilibrated traction field, continuous along the edges between elements, as shown in Figure 6. This way, and only imposing constraints to prevent rigid body motions, the topology optimization problem of each element can be solved.

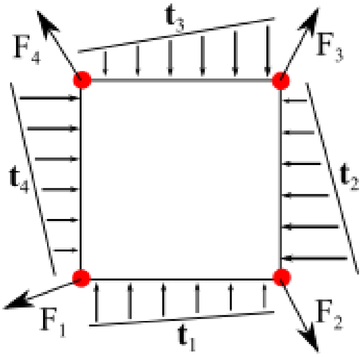


Figure 6: Isolated element with forces resulting from the global problem (F) and equilibrated traction field (**t**) equivalent to nodal forces.

The method used to obtain the traction field was introduced in [6], where an equilibrated traction field and continuous field in an element was originally used to find a statically admissible stress field to obtain an upper bound of the FE discretization error in energy form. The method presented in [6] uses the Maxwell diagram to obtain an equivalent traction field. The method followed in this project is explained below.

This method first analyzes the type of node to work on, that will result on two main groups. The first one considers nodes located on the domain's boundary and divides them in three groups: nodes with Dirichlet boundary conditions, nodes with Neumann boundary conditions and free nodes. The second main group contains the internal nodes. Once determined the node type to be analyzed, we take the elements surrounding the node and calculate the interface force.

3.1.1 Internal nodes

The nodes where no boundary conditions are applied, i.e., nodes without any external force or reaction directly acting on them, are considered as *free nodes.* The most common case is the internal node, which are those surrounded by 4 elements. In this case all four elements will have no other forces than the internal forces to be considered for the equilibrium. This case produces a particular Maxwell diagram, that is the simplest one to be analyzed.

The known variables are the corner node force (F_n) from the FE solution, equation (11), for each element and side.

$$F^e = [K^e] \cdot \{U^e\} \tag{11}$$

The goal is to obtain the interface forces (P_{in}), to finally complete the traction field (F_{Elm}) matrix for each node and side of the element.

Since there are many solutions to the problem of evaluating the equilibrated traction field due to having two undetermined degrees of freedom, this method uses a graphical solution known as the Maxwell diagram [7] [8]. The main polygon of the diagram is built using the force vectors ($-\hat{F}_n^E$) for node n taking in an anticlockwise sequence. This will always produce a closed polygon because of these force vectors form a balanced set. Each of the vertices represents the side of the elements, therefore the 4 elements surrounding a node will form a 4 vertice polygon and so on. Then, a pole point P is positioned inside the polygon. Adding this point will define a unique solution for the problem and create a triangle for each element built with the force vector ($-\hat{F}_n^E$) and two components that represent the interface forces $\hat{P}_{in}^E$ and $\hat{P}_{jn}^E$.

As an example to explain how the system of equations is built, the case of 4 elements surrounding a node will be used.

The diagram shown in Figure 8 is built following the base diagrams that [6] proposes. With this in mind, a series of equations are written based on the relations seen in the diagram.

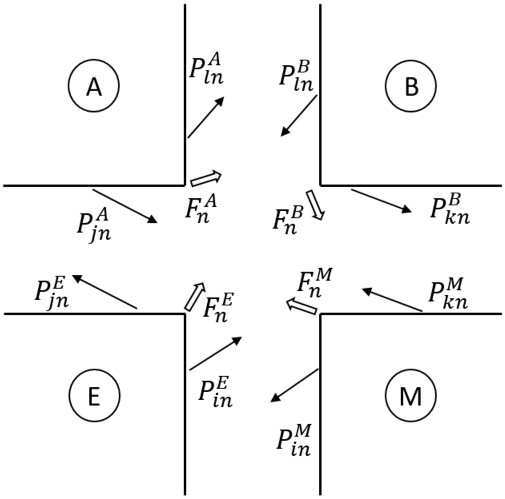


Figure 7: Isolated internal node surrounded by 4 elements.

$$\begin{aligned}
P_{in}^{E} + P_{jn}^{E} &= F_n^E \\
P_{jn}^{A} + P_{ln}^{A} &= F_n^A \\
P_{ln}^{B} + P_{kn}^{B} &= F_n^B \\
P_{kn}^{M} + P_{in}^{M} &= F_n^M \\
P_{in}^{E} + P_{in}^{M} &= 0 \\
P_{jn}^{E} + P_{jn}^{A} &= 0 \\
P_{ln}^{A} + P_{ln}^{B} &= 0 \\
P_{kn}^{B} + P_{kn}^{M} &= 0 \\
P_{in}^{E} &= -\frac{F_n^M}{2}
\end{aligned} \tag{12}$$

These equations are then placed in a matrix format for programming purposes. Note that the last equation of the system is not directly obtained from the diagram, this equation is added to the system to define the pole point (P). Therefore it will constrain the whole diagram in the 2D space. This is accomplished when the equations for both x and y directions are implemented, meaning that for each direction a constrain

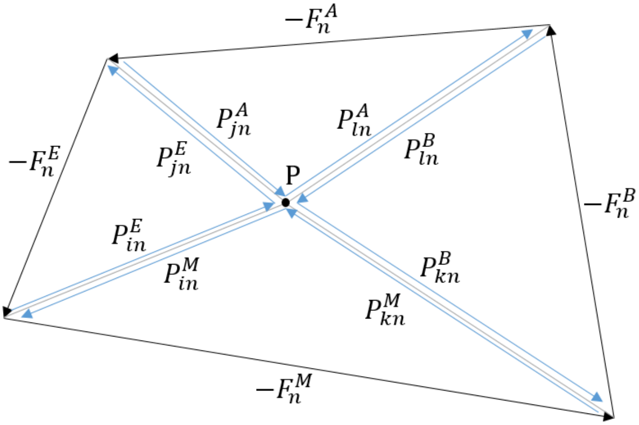


Figure 8: Maxwell diagram for four element surrounding a free node.

equation must be placed. Here the general case is presented, thus this system of equations derives into two systems, one for each direction.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_{in}^E \\ P_{jn}^E \\ P_{jn}^A \\ P_{ln}^A \\ P_{ln}^B \\ P_{kn}^B \\ P_{kn}^M \\ P_{in}^M \\ \lambda \end{pmatrix} = \begin{pmatrix} -\hat{F}_n^E \\ -\hat{F}_n^A \\ -\hat{F}_n^B \\ -\hat{F}_n^M \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{\hat{F}_n^M}{2} \end{pmatrix} \tag{13}$$

Once the relations are established and the equations are written the program will follow the following steps to obtain the desired equilibrated traction field:

1. Build the relations matrix between the interface forces and the nodal forces resulting from the FE analysis, depending on each case, as explained in the sections below.

2. Get each element's stiffness matrix and calculate the resulting nodal forces with the known nodal displacements from the global FE analysis.
3. Solve the system of equations.
4. Save the obtained interface forces in a predefined order according to the problem configuration.
5. Knowing that the integration of T ($T = N_i \cdot T_i + N_j \cdot T_j$) in one side must be equivalent to the interface force P_n on each side, T_i and T_j will define the traction field used in the micro scale topology optimization.

Equation (13) represents equilibrium of one component of the forces. A similar system of equations will be needed to represent the full 2D equilibrated forces. The resolution of the full system of equations will provide us with the interface forces of each side of an element (remember that each element will have a total of 8 internal force component per axis (x and y), two per side of each element).

The case with nodes surrounded by one or two elements is considered below, as they will be a special case of the nodes with Neumann boundary conditions.

3.1.2 Nodes with Dirichlet boundary conditions

The nodes on the a Dirichlet boundary are studied in this section. In the case of 2D Cartesian grids these nodes will be surrounded by one or two elements.

3.1.2.1 Corner node

The first case corresponds to corner with a Dirichlet boundary condition. The following figure represents this case, where the reaction **R**, resulting from the Dirichlet condition, is applied on the node.

In this particular case, the reaction's module is equal to the node force, but with opposite sign. This means that there is no force left to add, as this node will be

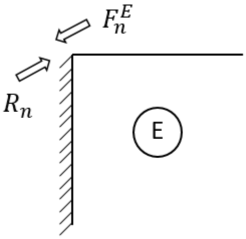


Figure 9: Isolated corner node where a Dirichlet boundary condition is applied.

equilibrated. Therefore the forces applied on this node are obtained straight away and are introduced in the F_{Elm} vector.

3.1.2.2 Standard case

The second case is defined by two elements surrounding a node on the Dirichlet boundary. In this case, the reaction force will be divided into two components that are calculated together with the interface forces (P_{in}).

The Maxwell diagram shown in Figure 10 is then converted in a system of equations that allows us to get the previously mentioned forces.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} P_{in}^E \\ R_n^E \\ P_{in}^M \\ R_n^M \\ \lambda \end{pmatrix} = \begin{pmatrix} \hat{F}_n^E \\ \hat{F}_n^M \\ R_n \\ 0 \\ 0 \end{pmatrix} \tag{14}$$

This system of equations represents equilibrium along each direction on space. In practical terms, the program will take into account both x and y component of each variable, in this case resulting in a system of equations with 10 unknowns. This specific case will introduce the P_{in} and R_n values in the F_{Elm} matrix.

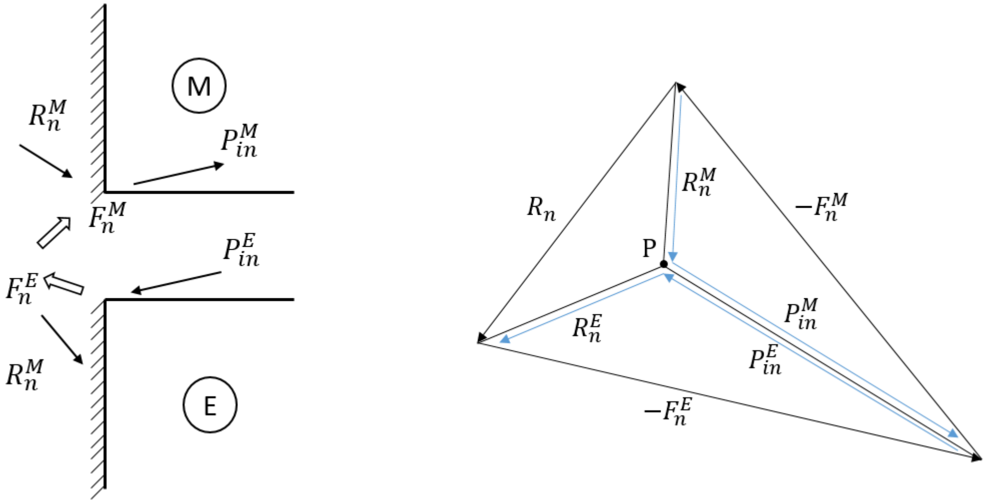


Figure 10: Isolated contour node where a Dirichlet condition is applied and its Maxwell diagram.

In (14), the last row will fix the solid body translation of the Maxwell diagram along one of the dimensions of the 2D space.

3.1.3 Nodes with Neumann boundary conditions

The way to work with nodes under Neumann boundary conditions is quite similar to the one used in 3.1.2. However, the external forces will all be known as the program calculates previously in the FEM analysis of the optimization process, where a given load has to be distributed along the specified nodes.

3.1.3.1 Corner node

The case of corner nodes is as simple as the one with Dirichlet condition, but here instead of using the reaction force, the external force (T_n) is applied.

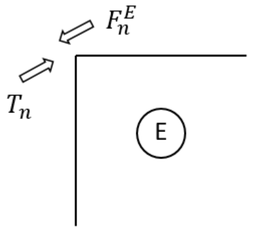


Figure 11: Isolated corner node where a Neumann condition is applied.

Once again the resulting force, that is afterwards entered in F_{Elm}, is obtained straightaway as the two forces are equilibrated.

When no fores are applied but the same conditions are fulfilled, we are facing the previously described *free node.* This node will only be affected by the corner node force (F_n), thus being the force introduced in the traction field (F_{Elm}) matrix.

3.1.3.2 Standard case

For the second case, where a boundary node, surrounded by two elements, is under Neumann boundary conditions, the external forces applied in each element of the same node are obtained previously when the load distribution is calculated for each node. This way the resulting equation system is much smaller and thus, faster to calculate.

As in the second case of nodes with Dirichlet boundary conditions, the Maxwell diagram in Figure 12 is then materialized in the system of equations (15).

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{in}^E \\ P_{in}^M \\ \lambda \end{pmatrix} = \begin{pmatrix} \hat{F}_n^E - T_n^E \\ \hat{F}_n^M - T_n^E \\ 0 \end{pmatrix} \tag{15}$$

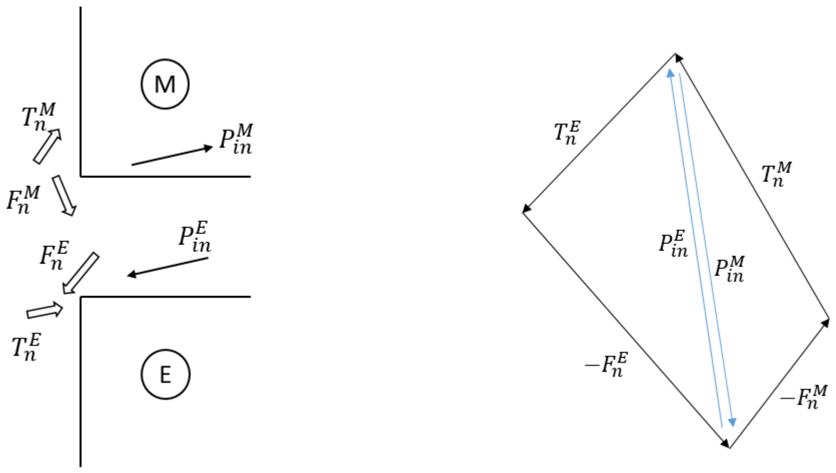


Figure 12: Isolated contour node where a Neumann condition is applied and its Maxwell diagram.

As well as in 3.1.3.1, the standard case considers also *free nodes* where no external forces are applied. The Figure 13 shows which forces take part in this case.

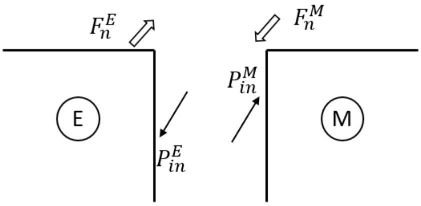


Figure 13: Isolated contour node surrounded by 2 elements.

The resulting diagram of this case does not help in any way, as $F_n^E + F_n^M = 0$ and $P_{in}^E + P_{in}^M = 0$, resulting on the following system of equations obtained directly from Figure 13.

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{in}^{E} \\ P_{in}^{M} \\ \lambda \end{pmatrix} = \begin{pmatrix} \hat{F}_{n}^{E} \\ \hat{F}_{n}^{M} \\ 0 \end{pmatrix} \tag{16}$$

3.2 Developed program structure

This section aims to present how the developed program in MatLab^R is structured and therefore, offer a better understanding on how the previously explained methods are implemented and work together to obtain the results that are shown below.

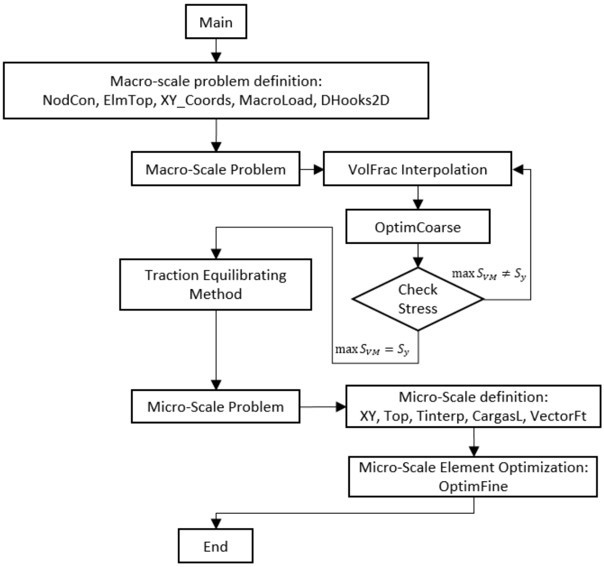

Main
Macro-scale problem definition:
NodCon, ElmTop, XY_Coords, MacroLoad, DHooks2D
Macro-Scale Problem
VolFrac Interpolation
OptimCoarse
Traction Equilibrating Method
Check Stress
$\max S_{VM} \neq S_y$
$\max S_{VM} = S_y$
Micro-Scale Problem
Micro-Scale definition:
XY, Top, Tinterp, CargasL, VectorFt
Micro-Scale Element Optimization:
OptimFine
End

Figure 14: Structure of the developed program, represented as a block diagram.

4 Implementation and numeric results

As this project has shown until now, the main objective of the present work is the topology optimization of an structural component by topologically optimizing each of the cells contained in the problem domain. In the current implementation, each of the defined cells corresponds to an element in the component mesh. This restriction is not mandatory, but helps to create a first implementation of the technique proposed in the previous section. In the present section, the topology optimization process is described using two numerical examples. Both problems are based on the classic cantilever beam problem, where one of the edges is fixed, and the other one is free.

In both problems the same material properties are used: Young's modulus $E = 210GPa$, Poisson's coefficient $\nu = 0,3$ and Yield strength $S_y = 400MPa$. The applied load will be distributed as a parable representing a tangential load, where $5 \cdot 10^7 Pa$ is the maximum value. Also the same mesh dimensions are used in both problems: Macro mesh size 30x20 elements, micro mesh size 50x50 elements and 2D bilinear quadrilateral elements in all cases.

4.1 Problem 1

In the first example the applied load is defined as a parabolic distribution of a pure tangential load in the free edge of the beam, as shown in Figure 15.

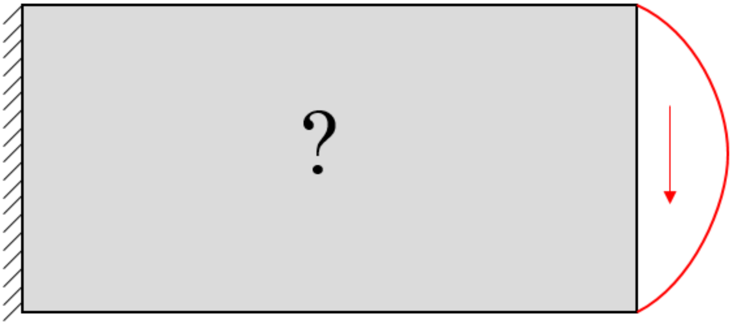


Figure 15: Scheme used to represent the first numeric problem.

The algorithm presented in this work solves the global topology optimization problem in the first place as described in the point 2.2 applied to the global problem shown in Figure 16, using a penalty value of $\nu = 1$ and $r_{min} = 0.1003$ calculated as $r_{min} = \sqrt{h_x^2 + h_y^2}$ being h_x and h_y the element dimensions. This combination of values generates a very smooth distribution along the elements in the global problem domain, caused mainly because of the low value defined for the penalty parameter that will not penalize intermediate density values. The iteration to reach the optimal volume fraction is shown below in Table 1.

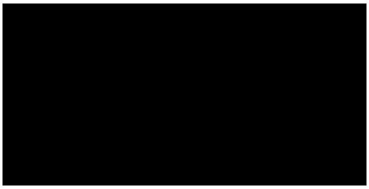
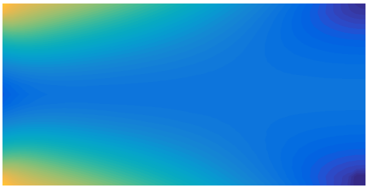
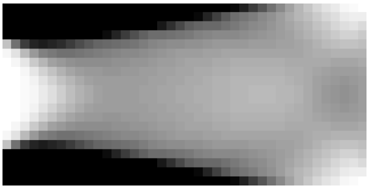
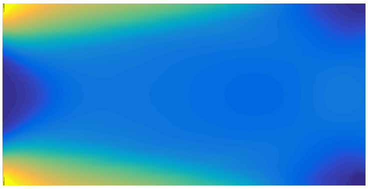
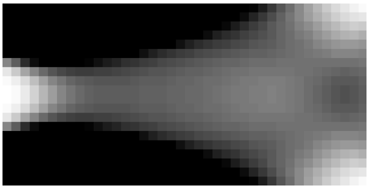
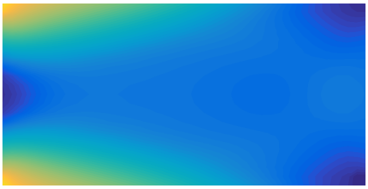
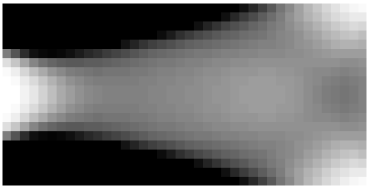
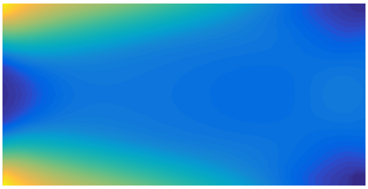
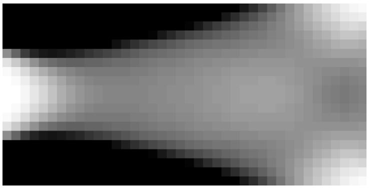
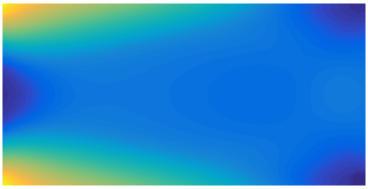
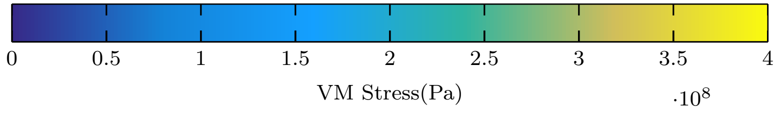
<table>
<tr><td colspan="2">It.1 - Vol. fraction = 1</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">It.2 - Vol. fraction = 0.5</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">It.3 - Vol. fraction = 0.73</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">It.4 - Vol. fraction = 0.62</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">It.5 - Vol. fraction = 0.60</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2"> </td></tr>
</table>

Table 1: First problem optimization iteration results.

The material distribution of the last iteration (Figure 16) and its FE results are saved in order to proceed with the traction equilibrating process afterwards.

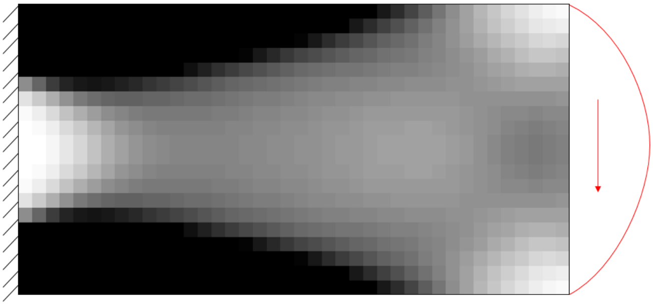


Figure 16: Material distribution proposed by the topology optimization algorithm in the first problem.

As described above, the material distribution obtained in Figure 16 will be used as input for the local (cell level) topology optimization. This means that the density (ρ) obtained in each element from the global topology optimization will be used as the volume fraction at the cell level, meaning that it will be a fixed value in each one of the optimizations in the micro level. Prior to the topology optimization on each cell, the boundary conditions to be applied in each one need to be evaluated. With that purpose, the results obtained from the global FEM analysis (from the last optimized geometry) are postprocessed in order to obtain, as described in section 3.1, the equivalent equilibrated traction field in each of the cells. Once the traction field is defined for each cell in the problem, the micro level topology optimization problem is solved also as shown in 2.2, but this time there is no need to optimize the amount of material as it has been assigned before. This last topology optimization algorithm is run using the following parameters: $\nu = 3$ and $r_{min} = 1.5$, these allow a higher penalty for the intermediate density value areas.

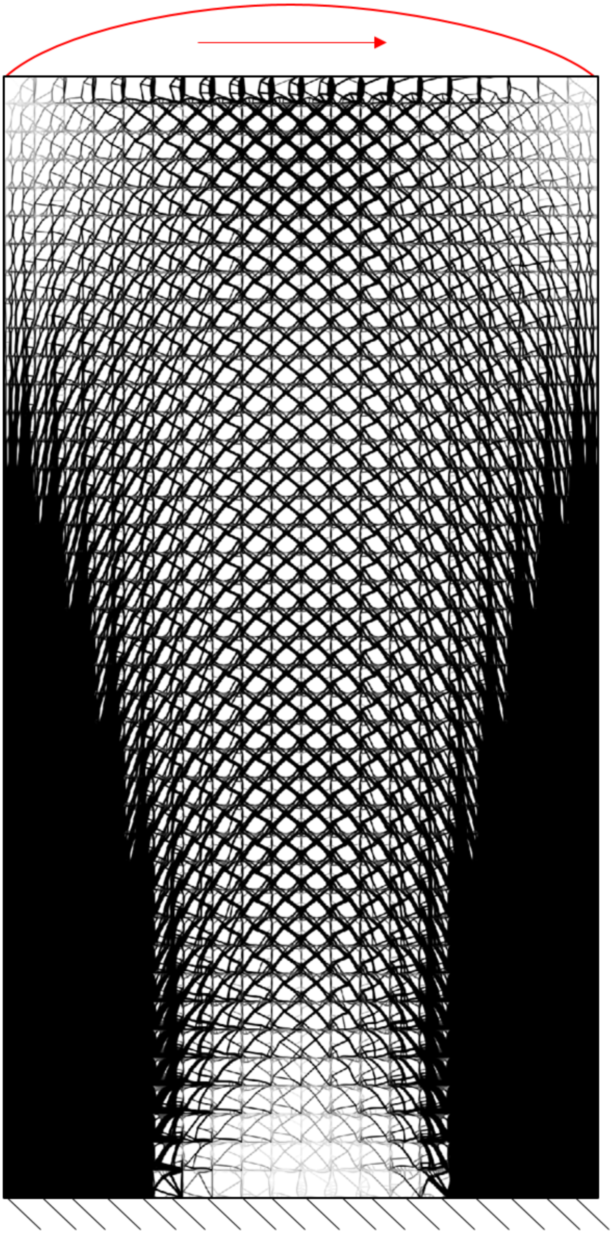


Figure 17: End solution of the second problem.

This problem, where a parabolic load typed is applied, produces a symmetrical optimization about an xz plane that can be easily identified looking to Figure 17. If right corner top and bottom elements are thoroughly analyzed and compared looking for symmetry, differences can be seen as shown in Figure 18.

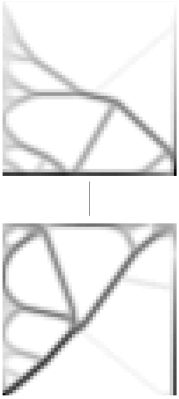


Figure 18: Top and bottom left elements compared to show lack of symmetry.

As the loads and volume fraction are exactly symmetrical, the only variable that could have altered the symmetry is the imposed constraints to eliminate rigid body motions in the micro scale topology optimization, as they are always the same. Two vertical displacement restrictions on both bottom corner nodes and one horizontal displacement restriction on left bottom corner node. This difference leads to non symmetrical elements.

4.2 Problem 2

In order to obtain a different and non symmetric solution, a second problem is proposed as a modification of the first one, adding a linear load in x direction as shown in Figure 19, which will show how the program adapts to the new load state.

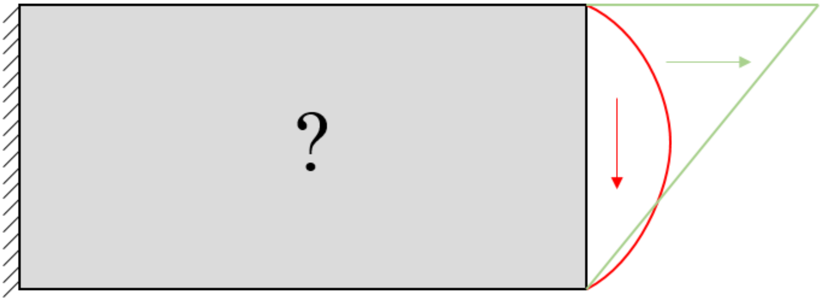


Figure 19: Scheme used to represent the second numeric problem.

Following the same procedure explained for the first problem, the program will optimize the material distribution for the given loads resulting on the table shown below. The optimization parameters for the macro scale problem are $\nu = 1$ and given the fact that the element size is the same as the first problem $r_{min} = 0.1003$.

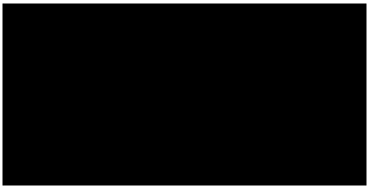
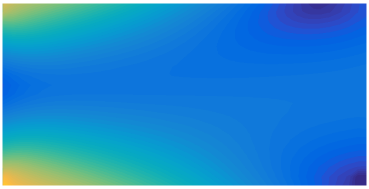
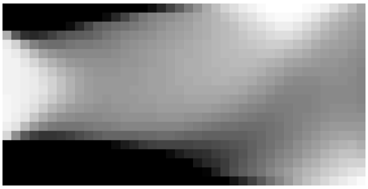
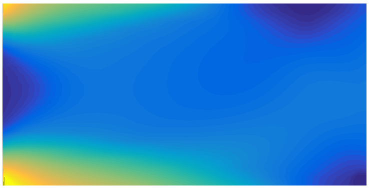
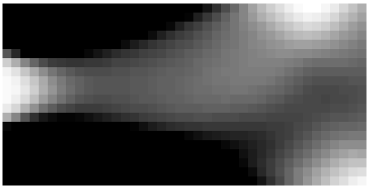
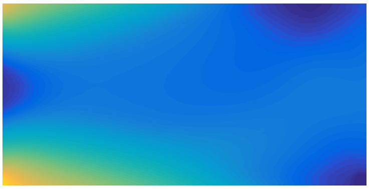
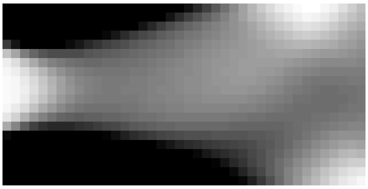
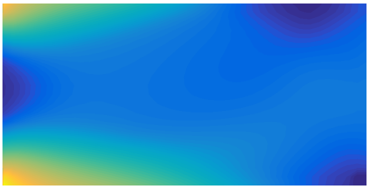
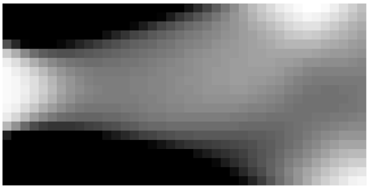
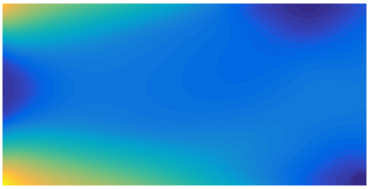
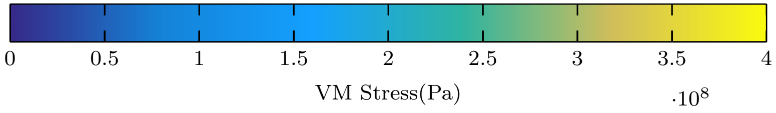
<table>
<tr><th colspan="2">It.1 - Vol. fraction = 1</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">It.2 - Vol. fraction = 0.5</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">It.3 - Vol. fraction = 0.727</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">It.4 - Vol. fraction = 0.617</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">It.5 - Vol. fraction = 0.606</th></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">
</td></tr>
</table>

Table 2: Second problem optimization iteration results.

Once the iteration is finished and the criteria is reached, the last iteration's material distribution and FE results are saved for the traction equilibrating process.

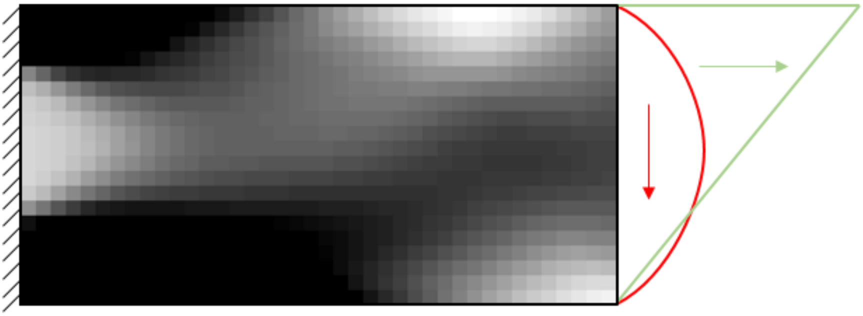


Figure 20: Material distribution proposed by the topology optimization algorithm in the second problem.

The last step provides the topology optimization code for each of the cells of the lower level. Figure 21 shows the end result of the whole optimization process.

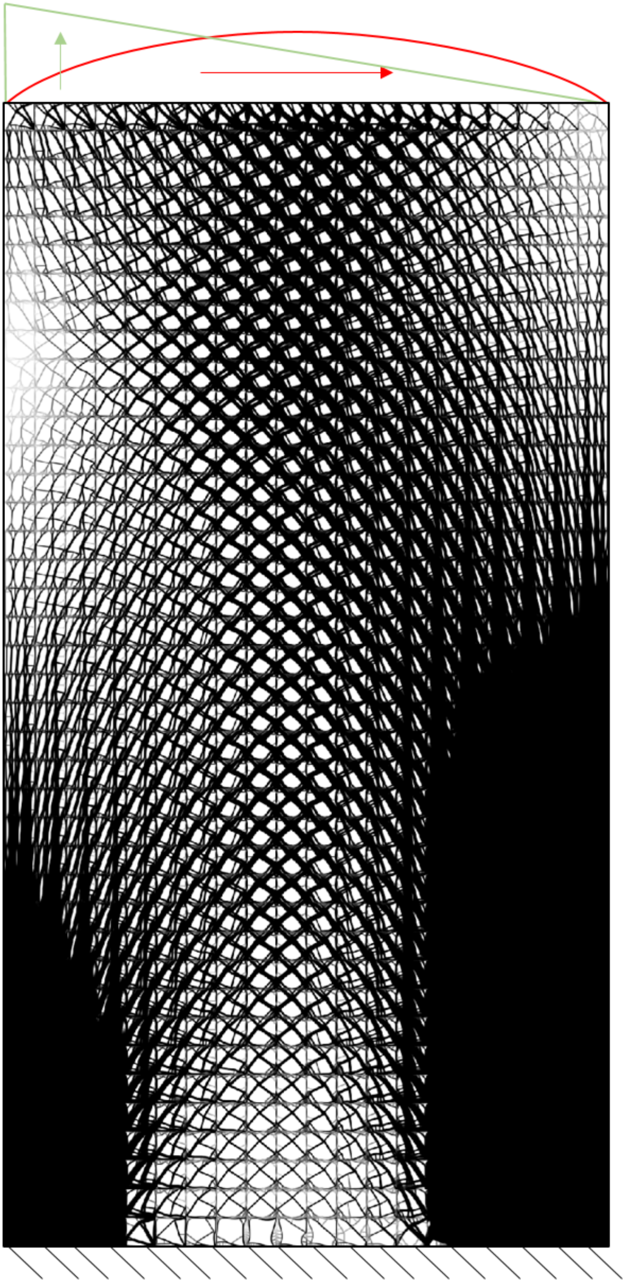


Figure 21: Final solution of the second problem.

A detailed section of the optimization result is shown in Figure 22, where the continuity between cells can be observed. Finally, each of these cells could be materialized by additive manufacturing techniques and place them in the established order to build the desired component.

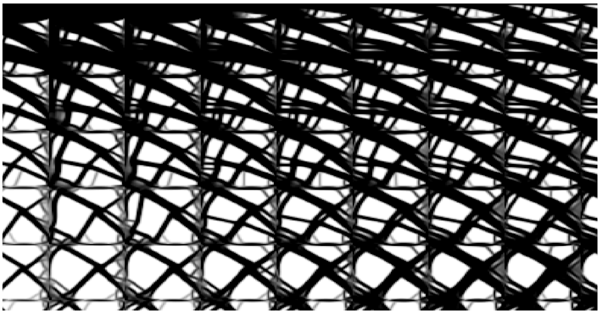


Figure 22: Various cell detail of the final result.

One of the main advantages of this technique, compared to classic topology optimization [9][10], is that it allows greater tolerance to unexpected variation of applied load [11].

5 Conclusions

In the present work, a two level topology optimization technique based on the SIMP method is purposed. In the first level, a global topology optimization problem is solved, which will fix the needed material distribution for the component. In the second level and after evaluating the boundary conditions in each cell using a traction equilibrating method, a new topology problem is solved for each cell, resulting in a continuous material distribution along the cells according to the previously established requirements. This is a very interesting result as this methodology allows to adequately decouple the different levels of multi-level analysis techniques. This continuity allows a posterior manufacturing of each cell, with the benefit of a greater tolerance to unexpected load variations. All this leads to the proposed method being of interest to the industry and justifies continuing its development and implementation in a 3D environment, as well as parallel processing of each cell's optimization allowing a significant reduction computational cost.

On the other hand, the results show that the continuity is not strictly satisfied. There are a number of causes for this lack of continuity that should be taken into account in future work:
- Each of the elements in the 1st level is assigned a constant value of relative density ρ, resulting in a non continuous relative density distribution that would lead to material discontinuities between adjacent cells.
- Intermediate values of relative density have not been fully eliminated in the cells. We believe that eliminating these intermediate values would lead to a higher material continuity. To do this we could follow the process developed by G. Cortes at the DIMM [12] who proposed to refine the elements of the resulting topology optimization with intermediate values of relative density and run a new topology optimization loop, repeating the process until a prescribed element size is reached.
- The individual optimization of each cell could also lead to this type of problems. The method could be improved if each optimized cell could somehow take into account its adjacent cells.

This work has considered that each cell of the lower level is obtained from only one element of the higher level, but the size of the cells of the lower level should be prescribed by manufacturing constrains. This could lead to too big elements in the higher level and the corresponding high discretization error of the solution. To solve this, it would be possible to reduce the size of the elements to warranty the required accuracy level of the solution and then to create cells defined by groups of elements from the higher level.

References

[1] Wolff's law on bone growth. http://3diy.cc/post/48784403668/topology-optimization-or-ribbon-candy-doodles. Accessed: 2018-09-12.

[2] Onofre Marco, Juan José Ródenas, José Albelda, Enrique Nadal, and Manuel Tur. Structural shape optimization using cartesian grids and automatic h-adaptive mesh projection. *Structural and Multidisciplinary Optimization*, pages 1–21.

[3] Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001.

[4] Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84, 2017.

[5] Alex Ferrer, Javier Oliver, Juan Carlos Cante, and Oriol Lloberas-Valls. Vademecum-based approach to multi-scale topological material design. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):23, 2016.

[6] P. Ladevéze and E.A.W. Maunder. A general method for recovering equilibrating element tractions. *Computer Methods in Applied Mechanics and Engineering*, 137(2):111–151, 1996.

[7] J Clerk Maxwell. Xlv. on reciprocal figures and diagrams of forces. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(182):250–261, 1864.

[8] Sven Olof Asplund. Structural mechanics: classical and matrix methods. 1966.

[9] Luca Dedè, Micheal J Borden, and Thomas JR Hughes. Isogeometric analysis for topology optimization with a phase field model. *Archives of Computational Methods in Engineering*, 19(3):427–465, 2012.

[10] Grégoire Allaire, Frédéric De Gournay, François Jouve, and A-M Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and cybernetics*, 34(1):59, 2005.

[11] Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. Infill optimization for additive manufacturing-approaching bone-like porous structures. *IEEE transactions on visualization and computer graphics*, 24(2):1127–1140, 2018.

[12] Guillem Josep Cortes Carbonell. Desarrollo de un programa de optimización de forma de componentes mecánicos mediante optimización topológica adaptativa. 2011.

Appendix. Programmer's manual.

Introduction

The main objective of the programmer's manual is the creation of a document that details the most important variables and functions used in the developed program. In this way anyone that is interested on the improvement or implementation of this program will have a basic reference.

The program will be initialized from the file `Main.m`, where also some of the functions are written.

The following sections will describe some of the more important variables and functions used, in order to understand the program's behavior.

Structure Variables

This section analyzes the main variables used in the program, those which the user can change to modify the component and its analysis. Most of the `Param.` variables are used for FEM related programs developed in the Department of Mechanic Engineering, and the structure is kept in case this program is improved.

Variable	**Description**
`NElem_xy_coarse`	1x2 vector that defines the coarse problem mesh size as the number of elements per axis [x y].
`NElem_xy_fine`	1x2 vector that defines the fine problem mesh size as the number of elements per axis [x y].
`XIni, XEnd, YIni, YEnd`	Variables that define the component's edge coordinates.
`Param.NGP`	Variable that defines the number of integration points.
`Param.NSpE`	Variable that defines the number of sides per element.
`Param.NNpE`	Variable that defines the number of nodes per element.
`Param.Dim`	Variable that defines the problem's dimensions, this program is only implemented in a 2D space.
`Param.Mode`	Variable (0/1) that defines whether plane strain or plane stress state is being analyzed.
`Param.ElmType`	Variable that defines the element type, in this program 1 is selected, which corresponds to linear elements.
`Param.LoadCase`	Variable (1/2) that defines the load state to be applied. The first load case applies a vertical load with a parabolic distribution. The second load case, adds a horizontal linearly distributed to the first load case.
`E, nu, Sy`	Variables that define the component's properties. Young's Modulus (E), Poisson's ratio (ν) and Yield strength (S_y).
`Py, Px`	Variables that define the horizontal and vertical force values.

The following variables are calculated from the ones defined before.

Variable	Description
`hx, hy`	Variable that defines the mesh element dimensions.
`Lx, Ly`	Variable that defines the components dimensions, assuming the components is rectangular shaped.
`Param.NNod_xy`	Variable that contains the total number of nodes per dimension (x and y), calculated adding 1 to the number of elements of each dimension (`NElem_xy_coarse + 1`).
`Param.NumElm`	Variable that contains the total number of elements, calculated from the product of both dimensions element numbers (`NElem_xy_coarse`).
`Param.NumNodes`	Variable that contains the total number of nodes, calculated from the product of both dimensions element numbers (`Param.NNod_xy`).
`Param.NDoF`	Variable that contains the degree of freedom, calculated from the problem dimensions.

Ending with the variable list, the table below shows the variables calculated in the main functions, and which are the most used and larger ones.

Variable	Description
`q`	This variable is found in the main program and will be an output of the `Macro_Load` function, it contains the resulting nodal forces calculated from the traction field defined in the beginning.

`TLoad`	This variable is also an output of the `Macro_Load` function and contains the force in the node of each element separately, thus there will be 2 forces in each direction per element, that are later required by the traction equilibrating function.
`xcoarse`	This variable is the main output of the `CoarseProblem` function and contains the material distribution of the coarse level problem.
`F`	Output variable of the `CoarseProblem` function that contains the nodal forces of the coarse problem.
`U`	Output variable of the `CoarseProblem` function that contains the nodal displacements of the coarse problem.
`TElm_x, TElm_y`	These variables are the unique output of the traction equilibrating function (`EvaluateForces`), and each of them contains the values of the equilibrated forces grouped by element in both x and y directions.

The same structure of variables is also used in the micro scale topology optimization, the explanation for those is therefore omitted.

Main Functions

This section will provide to the reader with a quick guide of the main functions. The order they are presented is the one used in the program's code, so the reader can have a quicker reference to what is looking for.

`NodalCoords_2D`

This function is found in the main program and evaluates the nodal coordinates of a 1D bar for a given number of elements `NumElms` of a given degree `FEDegree`. The bar is defined form the coordinate `X_Ini` to the coordinate `X_End`. The returning variable `XY_Nod` contains coordinates of each node in the mesh.

`ElmTopology_2D`

This function, found in the main program, evaluates the topology of a 1D bar for a given number of elements `NumElms` of a given degree `FEDegree`. The returning variable `ElmTop` contains each element's nodes.

`NodalConectivity_2D`

This function, found in the main program, evaluates the nodal connectivity of each node for a given topology `ElmTop`. Returning the variable `NodCon` contains the elements connected to each node, always anticlockwise starting from the bottom left element.

`Macro_Load`

This function, found in the main program, evaluates the applied load for a given load state `Param.LoadCase` and loads `(qx,qy)`.

`DHooks2D`

This function, found in the main program, evaluates the D matrix for given material properties `E` and ν. And for a given stress mode `Param.Mode`, defined for both plane stress and plane strain.

`CoarseProblem`

This function, found in the main program, evaluates the first level (macro) topology optimization and will return the last optimized geometry and forces of the iteration. As explained in **section**, this function will study the different geometries resulting from the iterations where the density and the maximum Von Mises stress is analyzed.

`OptimCoarse`

Found inside of the `CoarseProblem` function, this one will apply the topology optimization code from [3]. The finite element solver is changed to one used in the Mechanical Department of the UPV, to alleviate the different element numeration.

`Sigma`

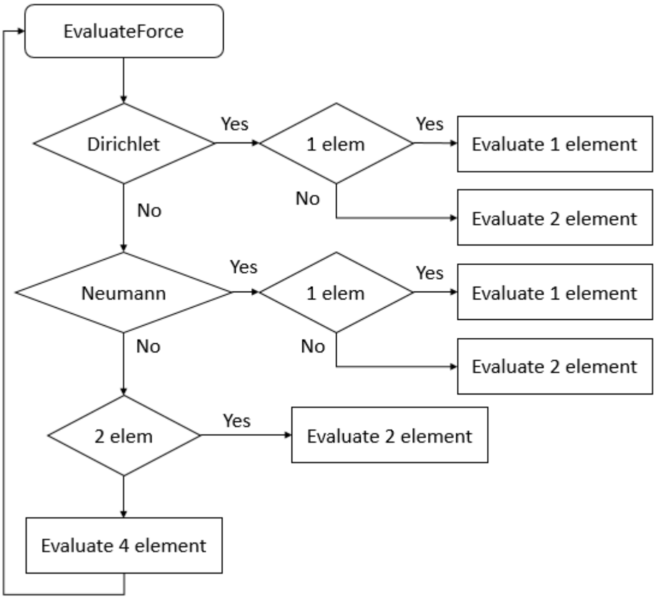
This function will be launched after each optimization (`OptimCoarse`) with the displacement results amongst other variables needed. It will calculate the Von Mises nodal stress and discard the edge values as they will be affected by the fix constrain and generate a much bigger values than they actually are. Specifically the top left and bottom left nodal stresses will be discarded.

This function will also be able to represent the nodal stress values in a figure when needed by changing the `fig.stress` value to 1 in the `Main` function.

`EvaluateForces`

The purpose of this function is to evaluate the forces in each of the nodes of the coarse mesh and get an equilibrated and continuous distribution of forces between elements, keeping the same load applied in each element. This way program will later be able to optimize the geometry of all the micro elements without being affected by the constrains applied in the topology optimization.

This function is executed after the coarse problem is solved, allowing to provide the function with the results from the stress/material optimized distribution and forces/displacements. The first part of the function will define some variables that are needed afterwards, after that the main loop starts and will run over each and everyone of the nodes. In order to differentiate the nodes depending on where are placed and what conditions are applied to them, a series of `if` statements are used, and each node will fall under one of the group as follows:



The evaluation boxes refer to whether is a 1, 2 or 4 element surrounded node case.

TopologyOpt

This function in charge to send information of each cell to the FineElm function as well as generating the end result figure. With that purpose, a loop is defined to travel around the mesh and send each cell's information to the FineElm function, which is defined below. The main information to be sent is, on the one hand the equilibrated forces in that element's nodes. On the other hand, the previously obtained density for each cell is now used as the input volume fraction accordingly.

In order to represent the end result, a matrix X is defined with the required size to insert each of the optimized cell next to each other.

FineElm

This function will gather the information for each of the cells coming from the TopologyOpt function and adapt it for the topology optimization. As the topology optimization program will use a different element numeration and the tractions gathered from the equilibrating function are applied in the nodes, some functions are created.

1. XYCords

   Function that generates new coordinates for the micro scale mesh.

2. TopoElem

   Function that generates new element topology for the micro scale mesh.

3. TInterp

   This function will interpolate the tractions applied in nodes coming from the equilibrating process along the nodes of the micro scale mesh side nodes.

4. VectorCargas

   This function is in charge to generate a matrix that will be used afterwards by the Vector_Ft function to obtain the equivalent forces in each side node. This matrix will contain the traction values of an element's side defined by the node numbers and both x and y tractions.

5. `Vector_Ft`

   As mentioned before, this function will use the `CargasL` variable generated in the `VectorCargas` function to obtain the equivalent forces in each node of the mesh's side.

6. `OptimFine`

   Finally the topology optimization program is launched with all the previously generated information.

The resulting topology will be then sent back to the `TopologyOpt` and this process will repeat for each and everyone of the cells in the macro scale mesh.