24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-2404
STRUCTURAL TOPOLOGY OPTIMIZATION OF A FLUID-STRUCTURE SYSTEM WITH UNSTRUCTURED MESH

João Baptista Dias Moreira
Emanuel Moutinho Cesconeto
Ederval de Souza Lisboa
Fernanda Bichet Link
Walter Jesus Paucar Casas
Mechanical Engineering Department, Federal University of Rio Grande do Sul, 425 Sarmento Leite st., Porto Alegre, RS 90050-170, Brazil.
baptista.moreira@ufrgs.br
emanuelemc@hotmail.com
ederval.lisboa@yahoo.com.br
feulink@gmail.com
walter.paucar.casas@ufrgs.br

Abstract. *This paper presents an improved algorithm for the bi-directional evolutionary structural optimization (BESO) method for topology optimization problems, that will perform the optimization, or maximization, of the stiffness of a structure while its structural volume is gradually removed, with the goal of obtaining a structure that possesses a maximized stiffness while utilizing a fraction of its materials. This structure may be part of a fluid-structural system and be subjected to static external loads, as well as loads imposed by the fluid. The discretization can be done with a unstructured and irregular mesh, and the BEFSO (Bi-Directional Evolutionary Fluid-Structural Optimization) method is used. The developed algorithm was capable of optimizing cases found in the literature for its validation, and was also applied to cases not easily solved by conventional methods, which usually adopt regular meshes or commercial software.*

Keywords: *topology optimization, FEM, fluid-structure, unstructured mesh*

1. INTRODUCTION

Topology optimization methods have been frequently studied lately as a good alternative to the development of optimized structural concepts, especially in situations where the mass of the structure must be as low as possible, such as in the aerospace industry. The main methods are BESO (Bi-directional Evolutionary Structural Optimization), detailed in Huang and Xie (2010), and SIMP (Solid Isotropic Material with Penalization), first described by Bendsøe (1989).

The main difference between those methods is in the treatment of the optimized variable, in this case the density of the structural material: the BESO method uses discrete variables, where each element has a value that represents either its presence or absence, where the SIMP method uses continuous variables, where each element can have any density between 0% and 100%. A detailed analysis of the differences can be found in Huang and Xie (2010). A variation of the BESO method is the BEFSO method (Bi-directional Evolutionary Fluid-Structural Optimization), as shown by Vicente (2013). This paper is about a modification to the BEFSO method, as applied in fluid-structure systems, that allows it to be used with unstructured meshes. The main goal is to contribute with the development of the BEFSO method, increasing its scope by allowing it to be used in cases that have more complex geometries. To verify these developments and exemplify the capabilities of the extended method several comparisons are made with cases found in the literature.

The proposed extension consists in a change in the way the sensibility values are calculates, and in the way the filtering is applied. Several cases where we wish to find an optimized structure cannot be effectively discretized with regular meshes, such as airplane wings. In these cases it is still possible to divide the domain in a greater amount of regular elements, but with the proposed method we can divide it in fewer elements that better fit to the domain boundaries.

2. THEORETICAL FOUNDATION

2.1 Bi-directional Evolutionary Structural Optimization (BESO) method

The BESO method is an algorithm that seeks the topological optimization of a structure by way of the maximization or minimization of an objective function, bound to restrictions in the domain. This is accomplished with an iterative method that removes or includes elements in a discretized structure according to an analysis done on each iteration. It is composed of a series of modules, each having a specified goal but no fixed form; different implementations can use different methods to achieve similar goals. Implementation details can be found in Huang and Xie (2010), while here it will be presented a general review. The specific form used in this paper is described in item 3.

2.2 Domain discretization and optimized variables

The domain must be discretized with several elements, forming a mesh to be used in a finite element analysis. The mesh remains unchanged through all iterations of the method, what changes is the effective material of each element. In the BESO method, each element has a corresponding value that changes the material of each material, as shown Eq. (1), according to Huang and Xie (2010).

$$x_i = \begin{cases} 1: \text{if the element is present} \\ 0: \text{if the element is absent} \end{cases} \tag{1}$$

The x_i value shown is used to calculate the elastic modulus E_i for the i element, using the original coefficient E_i^0 for the material of the structure, as shown in Eq. (2):

$$E_i = (x_i)^p E_i^0 \tag{2}$$

Where p is an exponent called the penalty factor. When the element is considered absent its elastic modulus becomes effectively zero, and thus it does not participate in the FEM analysis. Alternatively, the elemental stiffness can be removed from the global stiffness matrix before solving the problem, and any DOFs belonging to a node not connected to any active elements can be removed from the system, reducing its complexity as the structure loses its volume. This method is used in the implementation described in this paper.

2.3 Problem formulation

In this paper we use the structural stiffness as the characteristic to be maximized, and thus the optimization seeks the minimization of the energy absorbed by the structure. We must then minimize the following objective function, according to Eq. (3).

$$\mathbf{C} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{u} \tag{3}$$

Where **u** is the displacement vector of the structure and **f** is its force vector. This vector can be written as a function of the displacement and stiffness of the system, **f** = **Ku**. The global stiffness matrix is composed by the elemental stiffness matrixes $\mathbf{k}_i$, which depend on the elastic modulus E_i of each structural element, which in turn depends on the x_i variable as shown in Eq. (2). As such, we can write the **K** matrix as shown Eq. (4).

$$\mathbf{K} = \sum_{i=1}^{n} (x_i)^p \mathbf{k}_i^{0e} \tag{4}$$

Where $\mathbf{k}_i^{0e}$ is the elemental stiffness matrix of the i element as a function of E_i^0 instead of E_i, and n is the number of elements in the mesh.

To optimize the structure we optimize the values of the vector **x**, and to do that we must take the derivative of the objective function in regards to each of the components of x. When we derivate **C** by a variable x_i we estimate how it behaves when x_i is changed, and with that we can determine which elements are more sensitive to the structure, structurally speaking. Due to this these derivatives of the objective function are called sensibilities.

If we differentiate **K** by a specific x_i element we get as a result $p = (x_i)^{p-1}\mathbf{k}_i^{0e}$, because $\partial x_j/\partial x_i = 0$ for $j \neq i$. Thus, if we differentiate the objective function with regards to x_i we get Eq. (5), which is the sensibility of the system in regards to element i, as follows:

$$\alpha_i = p\left(xi\right)^{(p-1)} \frac{1}{2}\left(\mathbf{u}_i^e\right)^{\mathrm{T}} \mathbf{k}_i^{0e}\mathbf{u}_i^e \tag{5}$$

Where the e superscript denotes that those are elemental values. The **u** vectors are not functions of x_i and as such remain unchanged when we differentiate the objective function, but all the components that do not refer to the DOFs of the *i* element are multiplied by zeros in the reduced stiffness matrix k_i^{0e} and thus can be discarded, with the elemental $\mathbf{u}_i^e$ displacement vector remaining. In the implementation used in this paper *p* is equal to one, and the equation is simplified. This is called hard-kill and is explained in details by Huang and Xie (2010).

2.4 Stabilization filter

This filtering scheme modifies the sensibility value of each element, as defined in Eq. (6), as follows:

$$\alpha_i^f = \frac{1}{2}\left(a_i + a_{(i-1)}^f\right) \tag{6}$$

Where α_i^f is the filtered sensibility value for the *i* iteration, α_i is the non-filtered value for the same element in the same iteration, and $\alpha^f_{(i-1)}$ is its filtered sensibility in the previous iteration. This way, the whole history of the sensibility values are taken in consideration during the calculation of the new value, with more distant iterations having a smaller influence. This is needed to smooth out sudden jumps in the objective function that could result from the removal or addition of an element, and that could lead to instabilities in the optimization process.

3. METHODOLOGY

For this paper a MATLAB™ software that implements a 2D hard-kill BEFSO method was programmed. This method was chosen over alternative ones due to its lower computational cost, even when using filtering methods necessary to bring its results to the same level as them. The implementation specifics are described in this section. The general program flow is shown in Fig. 1.

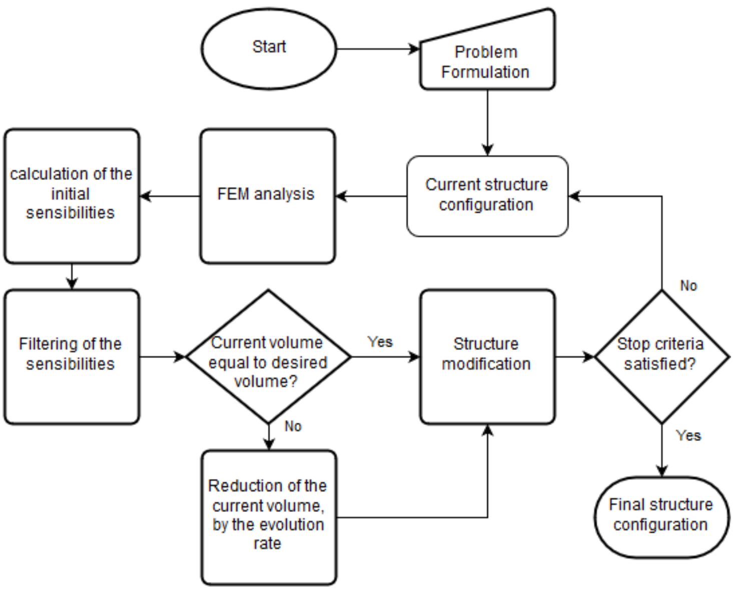

Figure 1. Flowchart showing the implemented BEFSO optimization procedure

3.1 Problem formulation

The analyzed problems consist of the structural optimization of fluid-structure systems, seeking the minimization of the flexibility, or compliancy, which is equal to the maximization of the stiffness. The mesh of the systems is not modified during the optimization. All the problems are first described by data that represent its mesh (node position and connectivity matrix), boundary conditions (forces and displacements, or prescribed pressures), structural material properties (Young modulus and Poisson coefficient), and initial fluid regions. The desired volume fraction and evolutionary rate are also supplied, as well as regions, if any, where the structure should not be removed.

3.2 Initial sensibility calculation

The calculation of the sensibilities begins with Eq. (5) with p = 1. The values are calculated with the elemental stiffness matrix and the components of the displacement vector that relate to the DOFs of the elements, but only if the element is currently part of the structure (active). This results in a different sensibility value for each element of the structure.

If an element is in the boundary of the structure with the fluid then the sensibility for it also takes into consideration the variation of the objective function due to the potential modification of the boundary. This variation is calculated based on the modification of the domain interface matrix, $\mathbf{L}_{sf}$, that would occur if the element were to be removed, and given by $(\mathbf{u}_s)^{\mathrm{T}}\,\mathbf{L}^i_{sf}\,\mathbf{p}_f$.

Equation (3.1) is used to calculate the sensibility for each element, based on the one used by Vicente (2013) but modified to consider the elemental volume according to Huang e Xie (2010). If the element is not in the fluid-structure boundary then $(\mathbf{u}_s)^{\mathrm{T}}\,\mathbf{L}^i_{sf}\,\mathbf{p}_f = 0$.

$$\alpha_i = \frac{\partial \mathbf{C}}{\partial x_i} = \begin{cases} \dfrac{\left(\dfrac{1}{2}\left(\mathbf{u}_s\right)^{\mathrm{T}}\mathbf{K}_s^i\mathbf{u}_s + \left(\mathbf{u}_s\right)^{\mathrm{T}}\mathbf{L}_{sf}^i\,p_f\right)}{V_e} & \text{for } x_i = 1 \\ 0 & \text{for } x_i \neq 1 \end{cases} \tag{7}$$

3.3 Sensibility filtering: mesh independency filter

A mesh independency filter must be used to prevent checkerboard patterns from emerging in the structure, which are patterns that depend on the mesh and that cannot be manufactured, which thus makes them undesirable. The filter must also be applied to allow for elements that do not belong to the structure to receive a sensibility value, which then allows them to be possibly re-included into the structure. This way, elements can be added to places that are very stressed, because the high sensibility values of the structural elements in those critical places are smoothed out over the empty elements around them.

For the calculation of the nodal sensibility first we must determine which nodes are within a certain radius r_{min} around the centroid of each element. The value of this radius should be chosen so as to include more than only the nodes that belong to the element, but it is arbitrary. For these nodes we determine the distance between them and the centroid, as exemplified in Fig 2.

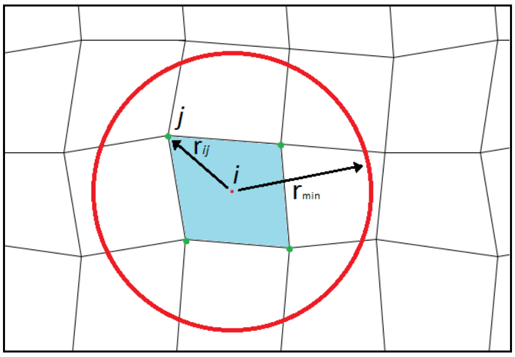

Figure 2. Process of determining the nodes within r_{min} of the centroid of an elements, and calculation of the distance between them.

We then attribute an abstract nodal sensibility value to each of those nodes, which is given by the weighted average of the element sensibility values of all elements that contain the node within a distance of r_{min}, according Eq. (8).

$$\alpha_j^n = \sum_{i=1}^{M} \omega_{ij}\,\alpha_i^e \tag{8}$$

Where ω_{ij} are the weights relating to the interaction between the element i and the node j, M is the number of elements near the node j, and α_i^e is the sensibility value for the element i. These weights should obey $\sum_{i=1}^{M} \omega_{ij} = 1$, and are given by Eq. (9), where r_{ij} is the distance between node j and element i.

$$\omega_{ij} = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right) \tag{9}$$

The nodal sensibility values are used only for calculating the filtered elemental sensibility values, as given by Eq. (10), Where W is the number of nodes close by element i and β_{ij} is equal to $r_{min} - r_{ij}$.

$$\alpha_i = \frac{\sum_{j=1}^{M} \beta_{ij} \alpha_j^n}{\sum_{j=1}^{W} \beta_{ij}} \tag{10}$$

It is then proposed a new way of applying the filter. After the weight calculations the filter is applied by way of simple multiplications and sums. As such, it is possible to write it in vector form, given by Eqs. (11) and (12).

$$\alpha^n = \omega \alpha^e \tag{11}$$

$$\alpha = \beta \alpha^n \tag{12}$$

Where α^n, α^e and α are the nodal, elemental and filtered elemental sensibilities respectively, ω contains the nodal weights and β contains the elemental weights. α^e and α consist of $N \times 1$ vectors, where N is the number of elements, α^n is a $Y \times 1$ vector, where Y is the number of nodes, ω is a $Y \times N$ matrix and β is a $N \times Y$ matrix.

Combining Eqs. (11) and (12) allows us to write, $\alpha = \beta\omega\alpha^e$, that can then be rewritten as in Eq. (13):

$$\alpha = \mathbf{H}\alpha^e \tag{13}$$

Where **H** consist of $N \times N$ matrix called smoothing matrix. This way there is no need to calculate or keep nodal sensibilities values, nor two separate matrixes. This matrix is still computationally expansive to generate, mas it depends only on the coordinates of the nodes and elements, and is thus unchanged throughout the optimization. It can then be computed on the first iteration only, being reused on each subsequent iteration.

As this filter does not consider volume differences between the volume of elements, its utilization with meshes with big volume differences between elements can cause undesired results. To minimize this effect we must divide the sensibility of each element by the square root of its volume before and after applying the filter.

3.4 Sensibility filtering: stabilization filter

As shown, this is a simple filter that calculates the average of the elemental sensibilities of iteration i and iteration $i - 1$, so as to stabilize the convergence. If written in expanded form, it is possible to see that all previous values influence the current one, according Eq. (14):

$$\alpha_i^f = \frac{\alpha_i}{2} + \frac{\alpha_i - 1}{4} + \frac{\alpha_i - 2}{8} + \cdots \tag{14}$$

Computationally, however, only the values for the current and the previous iterations are kept.

3.5 Restrictions and evolution rate

It is used a structural volume restriction during the optimization, implemented as a fraction of the initial volume. Throughout the optimization the structure must satisfy the Eq. (15), follows:

$$V_i = \begin{cases} V_0\left(1-iER\right) & \text{if } V_0\left(1-iER\right) > V^* \\ V^* & \text{if } V_0\left(1-iER\right) < V^* \end{cases} \tag{15}$$

Where V_i is the volume of the structure in the i iteration, computed as the sum of the volumes of all elements that make up the structure, V_0 is its initial volume, ER is the evolution rate, and V^* is the final desired volume, equal to $V_0 vp$ where vp is the final desired volume fraction. By way of this condition the structural volume is gradually reduced on each iteration, by the removal of structural elements, until the final volume is reached.

3.6 Structure modification

To implement the aimed optimization characteristics, first we must determine which elements previously removed can be included back into the structure. To this end, we use the bisection method to find a threshold to the sensibility values, such that removed elements with values above it can be considered re-included. We do this to limit the total volume that can be included in each iteration, to prevent the optimization from becoming unstable under certain circumstances. This threshold value x' is such that the sum of the volume of all removed elements with sensibility values above it must be equal to the allowed inclusion volume V_{ad}, as shown in the Eq. (15), below:

$$\begin{aligned} &\text{Find: } x' \mid y\left(x'\right) = 0 \\ &\text{Subject to: } y\left(x\right) = \sum_{j=1}^{N} V'^{e}_{j}\left(x\right) - V_{ad} \\ &\text{Where: } V^{e}_{j}\left(x\right) = \begin{cases} V^{e}_{j} & \text{if } j \text{ referes } \alpha_j > x \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{15}$$

Where N is the number of elements, V_j^e is the volume of element j and α_j is its filtered sensibility. The considered elements are in this case the all elements that do not belong to the structure but could be added back, that is, all elements that unable to become part of the structure due to imposed restrictions. The potential elements are the ones whose sensibility values are larger or equal than x'.

With the list of potential elements calculated we repeat the process, but this time the considered elements are all structural elements plus those potential elements and V_{ad} is replaced by V_i, which is the desired structural volume for iteration i. The structural elements for the next iteration are the ones with $\alpha_j > x'$, regardless if they are presently part of the structure or not. Some structural elements can be defined as fixed, however, and the sensibility values for these elements are considered to be infinite for the purposes of this test.

3.7 Stop criteria

To stop the optimization three criteria are considered. First the restrictive condition, which is applied gradually over the iterations, must have been fully applied before the optimization can be stopped. In other words, the optimization can only be stopped if the structural volume is equal to the desired volume.

After achieving the first condition, some iterations are still calculated until the change in the objective function, in this case the structural compliance, is below a tolerance value. During these iterations the volume of the structure does not change very much, but its configuration can change as elements are removed from some areas and added in others. This condition is necessary because as mentioned the methods used to modify the structure are only approximations, and although they are sufficient to guide the volume reduction during the first part of the optimization they are not capable of finding an optimum solution in only one iteration. Thus, during several iterations the structure stabilizes and converges to its final, optimized form.

As the same time as the second condition is checked, the current value of the objective function is compared to the best value found since the achievement of the first condition. If the present value is better than the previous best, it is kept as the new best, and the structural configuration is saved. If the optimization fails to produce a structure with a new best after a certain number of iterations it is stopped.

Mathematically, these conditions are represented by the following Eq. (16) and (17):

$$V_i = V^* \tag{16}$$

$$\left|\frac{C_i - C_{i-1}}{C_i}\right| < tol \vee C_{(i-K)\cdots i} > C_{min} \tag{17}$$

Where V_i is the structural volume in iteration i, V^* is the desired volume, C_i is the value of the objective function in iteration i, tol is the acceptable tolerance to consider the structure as having converged, and K is the number of straight iterations that are allowed to have C values that are worse than the best value so far, C_{min}, V^*, tol and K are inputs to the optimization method.

4. RESULT VERIFICATION

A comparison was made between the results obtained by the implemented algorithm and two cases presented by Vicente (2013). Figures 3 - 4 are present the meshes used in the algorithm for both cases. The first is a model of a tube joining, which was meshed with 12412 irregular elements, while the second is a model of a piston in two dimensions, meshed with 8375 irregular elements. The green area represents the initial fluid configuration, the developed algorithm does not befit from using regular or structured meshes over irregular, unstructured ones; it works equally with either type of mesh.

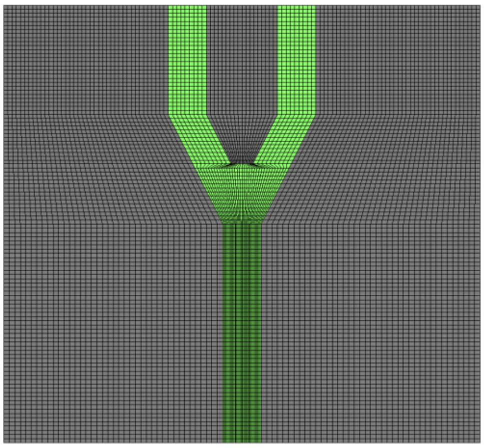

Figure 3. Irregular and unstructured meshes used by the developed algorithm: tube joint case, with 12412 elements.

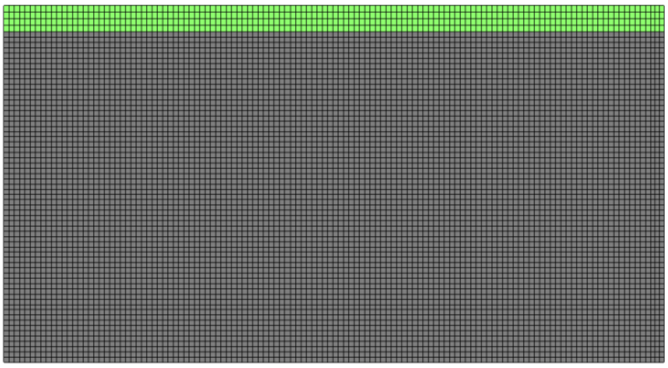

Figure 4. Irregular and unstructured meshes used by the developed algorithm: piston case, with 8375 elements.

Figure 5 show the result obtained for the tube joint case by the algorithm presented here is compared with the one found by Vicente (2013), that utilizes an irregular mesh with 33586 elements and a fixed fluid-structure boundary. There are also some structure elements that cannot be removed by the optimization, the colored areas represent regions occupied by the fluid, with the color representing pressure. The pressure difference is due to different boundary conditions at the input and output of the tubes.

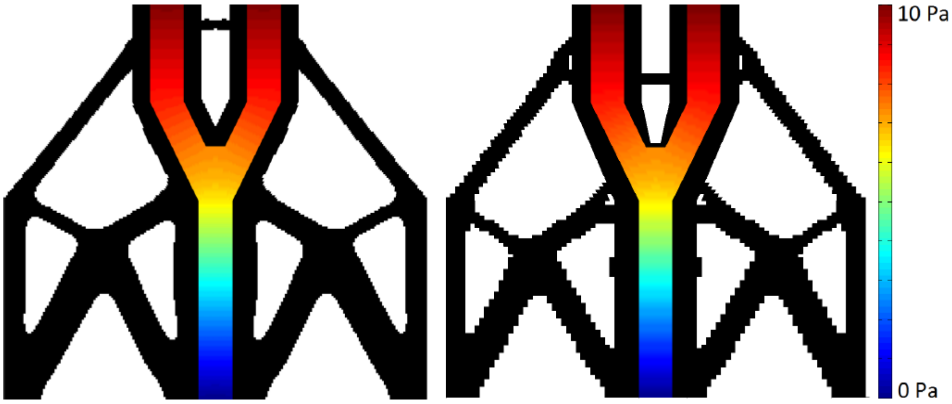

Figure 5. Comparison of the results found by Vicente (2013) with 33586 elements (left), and by the implemented algorithm with 12412 elements (right).

The comparison for the piston case is presented in Fig. 6, where the mesh used by Vicente (2013) is regular, with 34425 elements and movable interface. Even with significantly less elements the developed algorithm was capable of finding a very similar result to the ones found by Vicente (2013). The green area represents the fluid, with uniform pressure.

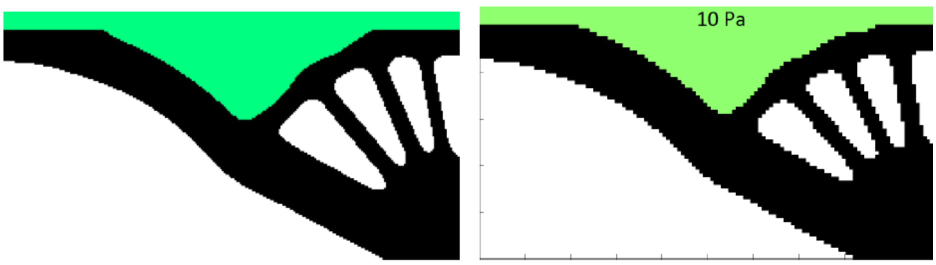

Figure 6. Comparison of the results found by Vicente (2013) with 33425 elements (left) and the implemented algorithm with 8375 elements (right).

5. CONCLUSIONS

The comparison of the cases in the verification shows that the BEFSO method is capable of optimizing cases with regular and irregular meshes, with and without movable interfaces. The modified algorithm contains an implementation of an FEM analysis as one of its parts, and thus does not depend on external software to run optimizations, needing only a MATLAB® console. It was verified with two cases found in the literature, and thus the goal of implementing an improvement to the BEFSO method was achieved. The proposed filtering implementation was also shown to be efficient.

6. ACKNOWLEDGEMENTS

The research described in this paper was financially supported by Brazilian national funding agencies, through the National Council of Scientific and Technological Development CNPq (grants 140081/2015-1, 148895/2016-6 and 148887/2016-3) from the Ministry of Science, Technology and Innovation; and the Coordination for the Improvement of Higher Education Personnel CAPES (grant 3686) from the Ministry of Education.

7. REFERENCES

Olhoff, N., Taylor, J. E. On structural optimization. Jounal of Applied Mechanics, v.50, pp.1139-1151, 1983.

Bendsøe, M. P.; Sigmund, O. Topology optimization: theory, methods, and applications. Berlin: Springer-Verlag, 2003, 370 p.

Xia, L., Xia, Q., Huang, X., Xie, Y. M. Bi-directional Evolutionary Structural Optimization on Advanced Structures and Materials: A Comprehensive Review. Archives of Computational Methods in Engineering. pp. 1-42. 2016.

Chang, C. "Design and Topology Optimization of Tissue Scaffolds". Tese de doutorado, University of Sydney, Australia, 2015.

Hansen, A. G., Sigmund, O., Haber, R. B. "Topology optimization of channel flow problems". Struct. Multidisc. Optim., USA, v.30, p.181–192, 2005.

Huang, X., Xie, Y. M. "Evolutionary Topology Optimization Of Continuum Structures: methods and applications". 1a edição.ed. Chichester, UK: John Wiley & Sons, Ltd, 2010.

237p.

Yoon, G. H., Jensen, J. S., Sigmund, O. "Topology optimization of acoustic–structure interaction problems using a mixed finite element formulation". International Journal for Numerical Methods in Engineerings, USA, v.70, p.1049–1075, 2007

Lisboa, E.S., Moreira, J. B. D., Cesconeto, E. M., Medeiros, J. E. G., Ribeiro, T. S. and Casas, W. J. P. (2017), Optimization of dynamic parameters with bidirectional evolutionary structural method of continuum structures, “XVII International Symposium on Dynamic Problems on Mechanics”, São Sebastião, Brazil, March.

Petyt, M. (1990), Introduction to Finite Element Vibration Analysis, Cambridge University Press, New York, NY, USA.

8. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.