

Please refer to the following:

Daróczy L, Jármai K

From a quasi-static fluid-based evolutionary topology optimization to a generalization of BESO
ENGINEERING OPTIMIZATION 47:(5) pp. 689-705. (2015)

From a quasi-static fluid-based evolutionary topology optimization to a generalization of BESO

László Daróczy

PhD. student, University of Magdeburg "Otto von Guericke", Germany

E-mail: laszlo.daroczy@ovgu.de

Károly Jármai

Professor, University of Miskolc, H-3515 Miskolc, Egyetemváros, Hungary

E-mail: altjar@uni-miskolc.hu

1. Abstract

The current article proposes a new algorithm for topology optimization based on a fluid dynamics analogy. The new algorithm possesses characteristics similar to most well-known methods, as the ESO/BESO (Xie and Steven 1993) method working with discrete values and the SIMP method (Bendsøe 1989; Zhou, Rozvany, 1991) (using OC or MMA) working with intermediate values, as it is able to work both with discrete and intermediate densities, but always yields to a solution with discrete densities. It can be proven mathematically that the new method is a generalization of the BESO method and when using appropriate parameters it will operate exactly as the BESO method. The new method is less sensitive to rounding errors of the matrix solver as compared to the BESO method and is able to give alternative topologies to well-known problems.

The article presents the basic idea, the optimization algorithm and compares the results of three cantilever optimizations to the results of the SIMP and BESO method. Finally, conclusions are drawn.

2. Introduction

Structural optimization mainly deals with geometric optimization of problems, e.g., sizing and finding the best shape of a structure, while topology optimisation, which became very popular in the recent years, optimizes material distribution on a given spatial discretization. The mechanical problem is solved usually by employing a Finite Element Analysis (FEA), the given set of loads and given boundary conditions play an essential role in the resulting layout. The most important quantities for

designers is the stress (Cheng & Jiang 1992), but the stability and vibrations can be crucial in different studies as well as well (Cinquin et al. 1997). If we go into deep, the different types of materials and the material defects can also play an important role (Bruggi & Cinquini 2011, Burczyński et al. 2002). In some cases, the displacement of a structure is the most-valuable (Qiao & Liu 2013), or the energy dissipation (Cheng & Yan 2013), or building topology (Phan et al. 2013). Reliability-based topology optimization shows another direction (Torii et al. 2012).

In our case, we use the well-known classical problem of compliance minimization to present the idea of the new method as it is the easiest to compare to different algorithms due to the huge amount of available data with this objective function. According to the well-known formulation:

$$\begin{aligned} \min_{\mathbf{x}} \quad & C = \frac{1}{2} \mathbf{f}^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{K} \mathbf{u} = \mathbf{f} \\ & 0 < x_{\min} \leq x_i \leq 1 \quad , \\ & \sum x_i V_i - V_0 \cdot f = 0 \end{aligned} \quad (1)$$

where \mathbf{K} is the global stiffness matrix, V_0 is the volume of the full domain, \mathbf{f} is the external force acting on the structure, \mathbf{x} is the design variable (density of the elements), \mathbf{u} is the displacement, f is the volume ratio to be satisfied, x_i is the density of cell i , x_{\min} is the minimal density of the cells, V_i is the volume of the finite element cell i .

To calculate the elemental stiffness matrix the SIMP (Solid Isotropic Material with Penalization) method is used, where the Young's modulus is calculated according to (Zhou & Rozvany 1991, Bendsoe & Sigmund 1995) as

$$E(x_i) = (x_i)^p E_0, \quad (2)$$

where E_0 is the Young's modulus of the solid material, and p is a penalty factor.

The sensitivity number is calculated according to the formulation of BESO (Bidirectional Evolutionary Structural Optimization) method to be consistent with the definitions (Xie & Steven 1993). Please note that there is no fundamental difference compared to the definition of the sensitivity number used by SIMP algorithm as the equation is changed only by a constant value $(-1/p)$:

$$\alpha_i = -\frac{1}{p} \frac{\partial C}{\partial x_i} = -\frac{x_i^{p-1}}{2} \mathbf{u}_i^T \mathbf{K}_i^0 \mathbf{u}_i \quad (3)$$

In this formulation the sensitivity is proportional to the increase of the mean compliance resulting from the removal of element i . If we want to minimize the compliance (C) we have to maximize the sensitivity number of the solid elements. From here on the optimized variable will be denoted by α , as with the use of sensitivities, we are actually maximizing it instead of the minimization of the compliance.

In the followings, the BESO method will be reviewed shortly to be able to present the important similarities and differences between the BESO and the new algorithm. As all topology optimization algorithms, the BESO algorithm starts with the definition of the problem as well, followed by the spatial discretization of the domain and by the definition of boundary conditions. Afterwards, the iterative algorithm performs either the defined number of cycles, or until convergence criteria are reached. In each cycle, we perform a finite element analysis followed by the calculation of sensitivity numbers. At this point, it is very fundamental to ensure a mesh-independent solution, for which reason mesh-independent filter and historical stabilization filter are applied to the sensitivities (Huang, Xie 2010b). As BESO uses a different approach compared to SIMP method, a different method has to be used for handling the volume constraint. Opposed to the SIMP method, the volume constraint is not immediately applied, but rather step by step and in each cycle elements are removed or added to ensure the volume constraint of the actual step ($\sum x_i V_i - V_0 \cdot f_j = 0$, where j is the iteration number). Finally, the new densities are evaluated. After this step, the current cycle is finished, and a new one begins. The detailed description of the algorithm can be found in the literature (Xie and Steven 1993).

3. Quasi-static quasi-fluid approach

Our approach is based on a resemblance taken from the nature (as e.g. simulated annealing algorithm). Fluids usually tend to move away from high-pressure regions to lower pressure regions, or from higher values of a potential field to lower levels (e.g. waterfall) to create equilibrium. This behaviour can be used for an optimization process. If we want to minimize a scalar-field, we simply need to define the pressure of the fluid to be higher in regions with higher scalar values, so it will move away from it. However, in the case of maximization the rule is the inverse. In the followings the basic idea of the algorithm will be introduced, using the standard notation of the fluid dynamics. After this chapter we will return to the notation of the accepted notation of topology optimization in order to avoid confusion.

After solving the equation $\mathbf{Ku}=\mathbf{f}$ for an intermediate solution of the topology optimization process, a quasi-static quasi-fluid simulation step will be performed (further on called as QSQF). The following concepts need to be defined:

Density of the fluid continuum (the design variable itself): At the beginning of the QSQF step $\rho_f \equiv x$, therefore $\rho_f \in [0,1]$ must be satisfied. However, following the QSQF step $x \neq \rho_f$, instead we will introduce a *historical density-damping* scheme, where

$$x^{new} = H_D x^{old} + (1 - H_D) x^{calculated}, \quad (4)$$

where H_D is a *historical density-damping* coefficient for stabilizing the solution (which must be within the range of $[0,1]$). Usually 0.5 will be an appropriate choice for this parameter.

The idea behind this formulation is that the fluid continuum can move extremely quickly in the presence of huge pressure differences. However, the optimized solid structure should be updated in a coupled manner with the fluid continuum, if the density of fluid would be exactly the same after and before the iterations as well. As this would require vast computational resources, we apply a quasi-static approach instead, where the optimized solid structure and fluid continuum is updated in a segregated way. However, to do this, we need to make sure that no sudden change can happen inside the continuum. This is actually similar to the averaging scheme applied to sensitivities by BESO method, but this time it is applied to the densities.

The potential field: $U(\alpha)$ acting on the fluid continuum is a function of the sensitivities. This function defines whether we are maximizing or minimizing. However, without the loss of generality from here on we will only consider minimization. E.g., if high values of the sensitivity function correspond to high values of the potential field, the fluid will try to move away from these regions, i.e., minimize the sensitivities.

Equation of state for the fluid continuum: $p_f(\rho_f)$. This function defines the connection between the pressure and density of the fluid (compressible fluid). To prevent fully empty regions, the density must be between the defined values (x_{min} and 1). Additionally, the pressure must be positive.

Equilibrium equation for the fluid:

$$p_f(\rho_f) + U(\alpha) = const. \quad (5)$$

This equation means that the sum of the energy stored by the potential field and the energy resulting from the pressure of the fluid is constant in every point (see **Figure 1**). The term quasi-fluid comes from the fact that this is not an equation for a real fluid, but rather for a continuum behaving *similarly* to fluids.

(It is worth noting that the hydrostatic equation, well-known by everyone, is highly similar to the previous form, hence the analogy:

$$p_f(\rho_f) + U_{gravity} = p_f + \rho g z = const. \quad (6)$$

In Eqn. (6) p is the pressure of the fluid, ρ is the density of the fluid, g is the gravitational acceleration and z is the height.)

In the followings, we will denote $const.$ in **Eqn. 5** by *EquilibriumLevel*, as it represents a significant parameter of the method.

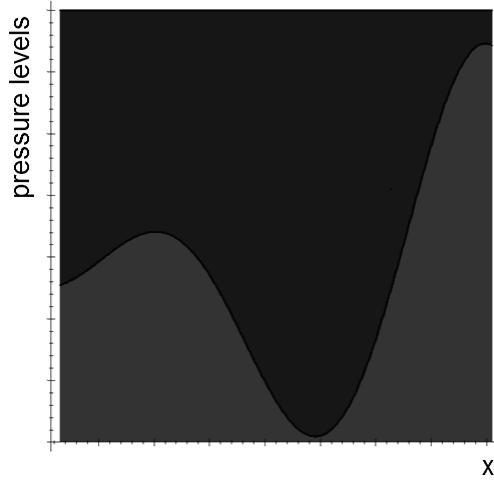


Figure 1

*Distribution of different energy levels in a quasi-static equilibrium state
Grey=Potential energy, Black=Pressure of the fluid continuum*

Although we could define directly the previous functions, but instead a computationally more efficient way is chosen. **Eqn. 5** can be reorganized into the following form:

$$\rho_f = p_f^{-1}(\text{EquilibriumLevel} - U(\alpha)), \quad (7)$$

where p_f^{-1} is the material rule for the fluid continuum. We can see that the density of the fluid continuum depends only on the optimized variable and on the parameter. Instead of calculating both values we can further merge the two steps into the following form:

$$\rho_f = \rho_f(\alpha, \text{EquilibriumLevel}). \quad (8)$$

This way, we can eliminate the need to mathematically define and calculate the potential field and material rule, instead, a single function is to be found. The critical point of the optimization process is not the calculation of the density of the fluid continuum, but rather finding an appropriate material rule, for which we can satisfy the volume constraint as well (see **Eq. 1**):

$$\int_{V_0} \rho_f(\alpha) dV = V_0 \cdot f_j. \quad (9)$$

From this equation we can determine the value of *EquilibriumLevel*. For further simplifications, we define the non-dimensionalized sensitivity using normalization to avoid difficulties with mixed sign problems:

$$\bar{\alpha} = \frac{\alpha - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}}. \quad (10)$$

Here α_{\min} and α_{\max} are the minimum and maximum values of the sensitivities. Remembering **Eq. 1**, the density of the fluid continuum can be defined as:

$$x^{\text{calculated}} \equiv \rho_f = x_{\min} + (1 - x_{\min}) \cdot \rho_f^s(\bar{\alpha}). \quad (11)$$

where $\rho_f^s(\bar{\alpha})$ is the scaled material rule, which has a domain and range of [0,1].

This function defines how “full” a cell is, in a continuous manner. This kind of logic is called fuzzy logics in mathematics (as opposed to the 0/1 classical logic), and these functions are called fuzzy functions. However, such a function cannot take any form in our problem. It is possible, that for a specific function, we cannot find an appropriate *EquilibriumLevel* to satisfy **Eq. 9** so that no negative pressures would be present in the fluid continuum, e.g. see **Figure 2**. However, negative pressures are meaningless in physics.

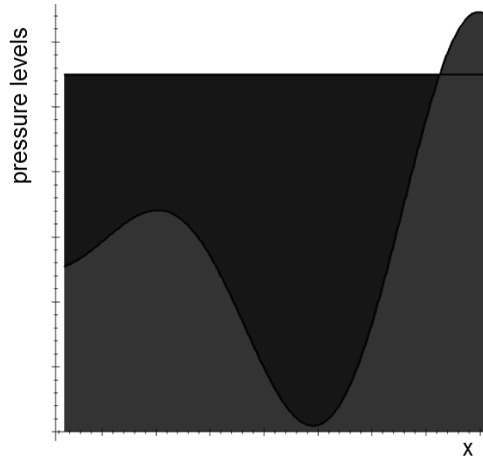


Figure 2

*Unsuitable choice of a function for ‘EquilibriumLevel’ resulting in invalid energy distribution
Grey=potential energy, Black=Pressure of the fluid continuum*

As one can see, the analogy leads actually a very simplified form. In this form one has to find the appropriate function from a set of functions to satisfy **Eq. 9**. After finding this function, the new densities can be calculated simply by using **Eq. 11**, and the density of the cells in the Finite Element discretization can be calculated by using the proposed density dampening scheme. In this way, the method actually becomes very similar to BESO, ESO or SIMP, where after determining the sensitivities, the densities have to be calculated based on some kind of solution methods.

4. Suitable functions for optimization

From here on, we will return to the standard notation of structural mechanics and topology optimization. The only notation to be kept is the scaled material rule ($\rho_f^s(\bar{\alpha})$), using which the new densities can be calculated, in a similar fashion to BESO or SIMP. To ensure the solvability of **Eqn. 9** we introduce the Possible Optimization Fuzzy Function (further POFF), which is a parametric function. This function is parametrized by *threshold*, and an appropriate *threshold* value must always exist, for which **Eq. 9** can be satisfied:

$$\mathbb{F}(\bar{\alpha}, threshold). \quad (12)$$

Please note that the material rule was already parametrized by *EquilibriumLevel* in **Eq. 8**, the only change is that instead of *EquilibriumLevel* we use a different parameter.

\mathbb{F} must satisfy that only one *threshold* value exists, for which the volume constraint can be solved (and this value must be additionally in the [0,1] range). This ensures that only one unique solution exists for **Eq. 9**.

Additionally, \mathbb{F} must give 0 as output at 0, and 1 at 1. The function has to be “flat” at the ends (i.e., the first derivative has to be smaller at the ends of the domain, as inside the domain) and must be monotonically increasing. These constraints ensure the correct fluid-like behaviour and minimization. Please note that the parameter *threshold* is only a parameter to this family of functions, not to the optimization itself. The end-user never has to deal with it, the value is to be determined internally by the software (similar to B_E in the SIMP algorithm Sigmund, O. (2001)).

To be able to control the behaviour of the fluid we introduce one more concept, the CPOFF (Controllable Possible Optimization Fuzzy Function), an extension to the POFF. Aside *threshold* we introduce the β parameter, which controls the slope of the function at *threshold*, mathematically formulated:

$$\frac{d\mathbb{F}(\bar{\alpha}, threshold = threshold_1, \beta_1)}{d\bar{\alpha}} > \frac{d\mathbb{F}(\bar{\alpha}, threshold = threshold_1, \beta_2)}{d\bar{\alpha}}, \quad (19)$$

if and only if $\beta_1 > \beta_2$

meaning that lower β values will result in smaller values of the first derivative of \mathbb{F} at *threshold*.

Some proposed functions can be seen in **Table 1**, and functions can be seen in **Figure 3**.

$\mathbb{F}_{pow}(\bar{\alpha}, threshold, \beta)$	$= \begin{cases} \frac{1}{threshold^{1/\beta} + (1 - threshold)^{1/\beta}} \left(-(threshold - \bar{\alpha})^{1/\beta} + threshold^{1/\beta} \right), & \text{if } 0 \leq \bar{\alpha} < threshold \\ \frac{1}{threshold^{1/\beta} + (1 - threshold)^{1/\beta}} \left(1 + threshold^{1/\beta} \right), & \text{if } \bar{\alpha} = threshold \\ \frac{1}{threshold^{1/\beta} + (1 - threshold)^{1/\beta}} \left((\bar{\alpha} - threshold)^{1/\beta} + threshold^{1/\beta} \right), & \text{if } 1 \geq \bar{\alpha} > threshold \end{cases}$
$\mathbb{F}_{pow.lin.}(\bar{\alpha}, threshold, \beta)$	$= \begin{cases} 0, & \text{if } 0 \leq \bar{\alpha} < threshold - \frac{1}{2\beta} \\ \beta(\bar{\alpha} - threshold) + 0.5, & \text{if } \bar{\alpha} = threshold \\ 1, & \text{if } 1 \geq \bar{\alpha} > threshold + \frac{1}{2\beta} \end{cases}$
$\mathbb{F}_{sinh}(\bar{\alpha}, threshold, \beta)$	$= \int_{x=0}^{\bar{\alpha}} \left(\frac{\beta \cdot \left(e^{-\beta^2(x - threshold)^2} - e^{-\beta^2 threshold^2} \right) - \frac{\text{sgn}(x - threshold) + 1}{2} \cdot \frac{(x - threshold)^2}{(1 - threshold)^2} \cdot \beta \cdot \left(e^{-\beta^2(1 - threshold)^2} - e^{-\beta^2 threshold^2} \right)}{\int_{x=0}^1 \left(\beta \cdot \left(e^{-\beta^2(x - threshold)^2} - e^{-\beta^2 threshold^2} \right) - \frac{\text{sgn}(x - threshold) + 1}{2} \cdot \frac{(x - threshold)^2}{(1 - threshold)^2} \cdot \beta \cdot \left(e^{-\beta^2(1 - threshold)^2} - e^{-\beta^2 threshold^2} \right) \right) dx} \right) dx$

$$\mu_{f_{dis.}}(\bar{\alpha}, \text{threshold}) = \begin{cases} 0, & \text{if } 0 \leq \bar{\alpha} \leq \text{threshold} \\ 1, & \text{if } \text{threshold} \leq \bar{\alpha} \leq 1 \end{cases}$$

Table 1: Proposed CPOFF functions

Without full proof we can see that

$$\begin{aligned} \lim_{\beta \rightarrow \infty} \left(\mu_{f_{inv.pow.}}^{\beta}(\bar{\alpha}, \text{threshold}, \beta) \right) &= \lim_{\beta \rightarrow \infty} \left(\mu_{f_{pcw.lin.}}^{\beta}(\bar{\alpha}, \text{threshold}, \beta) \right) = \\ \lim_{\beta \rightarrow \infty} \left(\mu_{f_{sub.}}^{\beta}(\bar{\alpha}, \text{threshold}, \beta) \right) &= \mu_{f_{dis.}}^{\beta}(\bar{\alpha}, \text{threshold}) \end{aligned} \quad (20)$$

Using the value of β the user can dynamically influence the optimization: as the value is increases, the algorithm will eliminate the intermediate densities step by step. This way QSQF will work with intermediate densities at the beginning but converge to discrete densities in the end, i.e., it is the best of two worlds. **Eq. 9** shows that the analogy at $\beta=\infty$ yields to the BESO method, thus it is a generalization or extension to the BESO method. We find it an important result to show that BESO is based on real physical phenomena, i.e. it is imitating nature. Moreover, in QSQF the most significant difference compared to BESO is the use of fuzzy functions instead of discrete ones. This can improve the stability due the lack of sudden changes in densities.

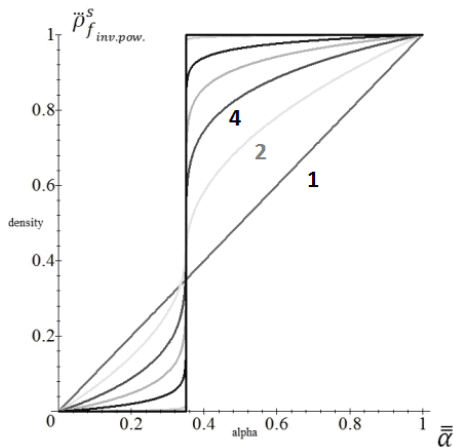


Figure 3/a

Inverse-power function (Table 1/1)
threshold=0.35, $\beta=1,2,4,8,20,200,10000$

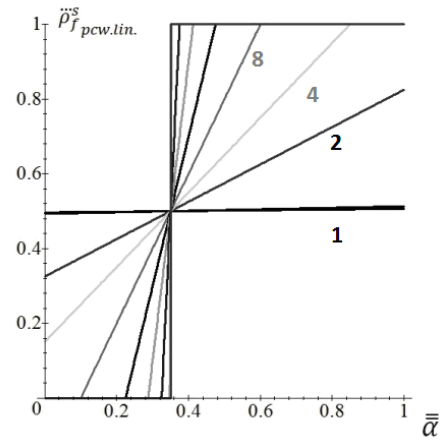


Figure 3/b

Piecewise-linear fuzzyfication function (Table 1/2)
threshold=0.35, $\beta=1,2,4,8,20,200,10000$

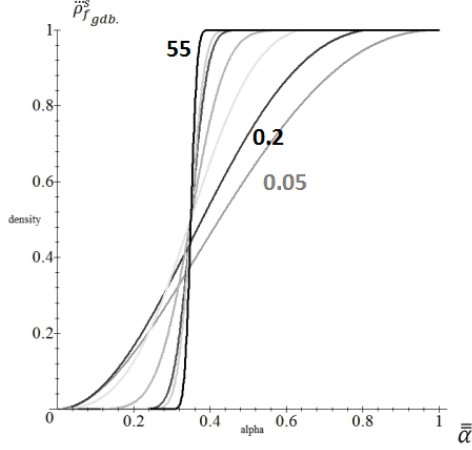


Figure 3/c

Gauss-distribution based fuzzyfication func. (Table 1/3)
 threshold=0.35, $\beta=0.05, 0.2, 0.5, 1, 2, 5, 10, 20, 25, 55$

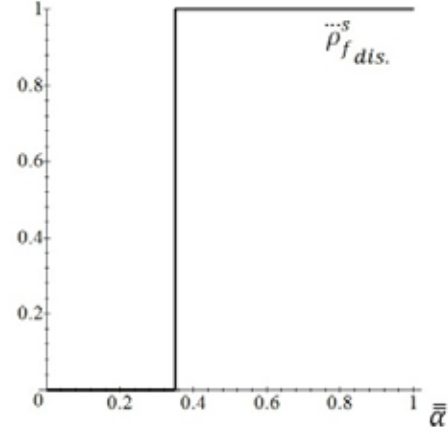


Figure 3/d

Discrete linear fuzzyfication function (Table 1/4)
 threshold=0.35

5. Workflow of the algorithm and the applied filters

Although the main idea is easy to understand now, presenting the workflow of the optimization process is still necessary. The comparison of the BESO and QSQF method's workflow is summarized on **Figure 4**.

Input: Definition of the problem (boundary conditions, volume constraint, etc.).

Step 0: Finite Element Discretization.

Step 1: Finite Element Analysis (FEA).

Step 2: Calculation of the sensitivities (according to **Eq. 3**.)

Step 3: Mesh independence filter: Before starting the QSQF step mesh-independent filters must be applied. Instead of inventing a new method we choose a well-documented and well-researched mesh-independent filter. Although several versions of the mesh-independent filters are available, in the current implementation the BESO method's filtering technique was chosen (Sigmund & Petersson 1998). The mesh independency filter in BESO and QSQF serves the same purpose as by SIMP. Moreover, at hard-kill BESO the local averaging has to work, even if the neighbouring densities are zero. This results in a slightly modified convolution operator:

$$\alpha_i^{filtered} = \frac{1}{\sum_{i=1}^N w_{ij}} \sum_{i=1}^N w_{ij} \alpha_j, \quad (21)$$

where the weight factors are determined as

$$w_{ij} = r_{\min} - r_{ij}, \text{ if } r_{ij} < r_{\min} \quad (j = 1, 2, \dots, N), \quad (22)$$

and r_{ij} is the distance between the centers of cells i and j . This smoothing is computed in the neighbourhood of cell i , in a circle with radius r_{min} (mesh independency filter radius).

Step 4: Historical sensitivity stabilization: For stabilization of the process the following scheme is used:

$$\alpha^{new} = H_s \alpha^{old} + (1 - H_s) \alpha^{calc}, \quad (23)$$

i.e. the new sensitivity will be the weighted sum of the old and newly calculated value. In BESO method $H_s = 0.5$ is used (Huang & Xie 2007). We also recommend using the same value.

Step 5: Refresh volume constraint (if necessary). Although the prescribed volume for the structure could be immediately turned on if using appropriate parameters for the algorithm, a sequential cutting method will help in the convergence, as at BESO (ER_{max}):

$$V_{j+1} = \max(V_f, V_j (1 - ER / 100)) \quad (24)$$

Step 6: Calculate new densities (QSQF step): To update the density field we must first determine the value of the *threshold* parameter and then we can calculate the new densities using **Eq. 11**. To prohibit fast movement we propose another scheme despite the fact, that during the tests the historic density-damping scheme worked correctly. This method is able to control changes in the fluid continuum both globally and locally.

The idea is that not all regions of the fluid have to change during the iterations, but only the ones with *appropriate driving force* (this definition is *close* to the behaviour of the Bingham continuum, where below a certain shear stress level the fluid does not start moving, and thus it is actually not a real fluid). For this reason, we introduce the additional AR_{max} criteria (with same purpose as in BESO) to limit the maximum amount of density-increase in the whole fluid region.

The density-increase of a single cell is

$$dx_{inc} = \frac{(x^{calculated} - x^{old}) + |x^{calculated} - x^{old}|}{2}, \quad (22)$$

and thus the whole density-increase

$$x_{inc}^{total} = \int \frac{(x^{calculated} - x^{old}) + |x^{calculated} - x^{old}|}{2} dV, \quad (23)$$

If during the update the

$$x_{inc}^{total} > AR_{max} V \quad (24)$$

condition is violated, then instead of the *threshold* parameter we determine a $threshold_{inc}$ parameter, for which the criterion will not be violated, and we update only the densities which would increase.

In the next step, we calculate the $threshold_{dec}$ parameter, for which the volume constraint can be satisfied if only updating elements, which would decrease. This way there is a zone of threshold values in which no movement will happen (as in Bingham continuum). In fully discrete cases ($\beta = \infty$), this is

the same definition as AR_{max} at BESO (the total number of elements allowed to be switched from 0/ x_{min} to 1 defined as a percentage of the volume), and $threshold_{inc}$ and $threshold_{dec}$ correspond to α_{add}^{th} , α_{del}^{th} . These values does not have to be specified by the user but are temporary, internal parameters determined by the algorithm.

Step 7: Historical density-dampening: see **Eq. 4**.

Step 8: Refresh β values: The value of β can be increased to generate a solution closer to the discrete solution (see **Eq. 9**).

Step 9: If the convergence criteria is not yet met, then we start the next optimization step (go to Step 1), otherwise go to Step 10.

Step 10: The solution is acquired.

One can see that although we used a completely different approach and analogy from nature, the two algorithms have very similar workflow, with the only difference being that the new algorithm is capable of working with intermediate densities. However, at $\beta=\infty$ it yields to the BESO algorithm, thus it is a generalization of BESO method.

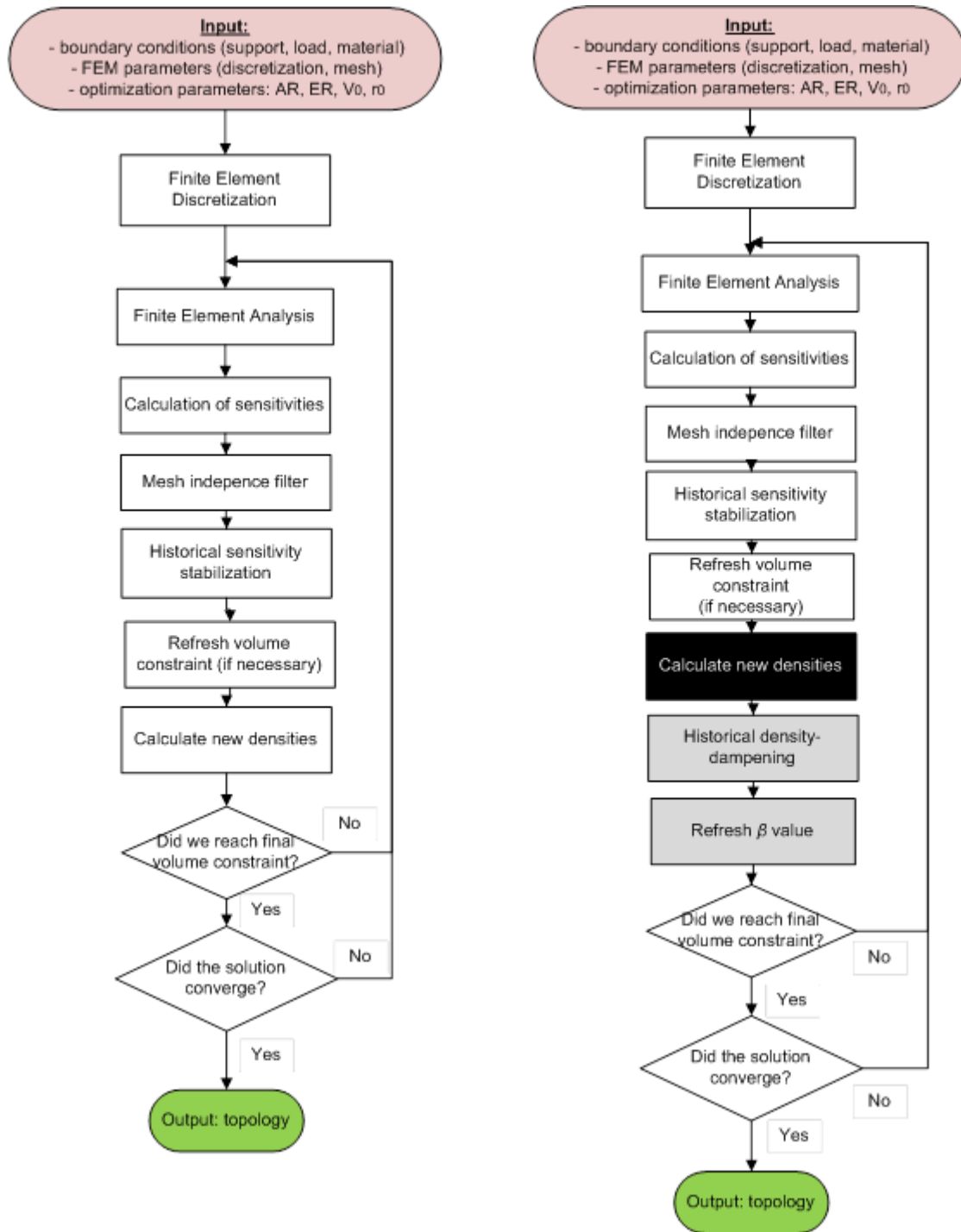


Figure 4

Proposed workflow of QSQF (quasi-static quasi-fluid) optimization (right side) compared to BESO method (left side), new steps are highlighted with grey, steps with different computations are highlighted with black

6. Test examples

In the followings, the results achieved with the new method are presented and compared to the results of the previous methods, SIMP&OC and BESO, using some classical problems. The FEA was solved using preconditioned conjugate gradient method with the final residual error always in the range of 10^{-6} - 10^{-10} N (3D problems) and 10^{-8} - 10^{-10} N (2D problems), depending on the problem. The FEA code was successfully validated against an example of a bent cantilever using ADINA R&D Inc. ADINA®.

Although the chosen problems are well-known and well-researched basic examples of optimization and thus we cannot expect to achieve huge improvement, we found it fundamental to validate the algorithm against these tests. For the presented cases, the new method has achieved the well-known solutions without error and for two cases it could even provide, although only slightly, but better solutions with different topologies compared to the literature. We consider this a major achievement as these problems have been examined for decades.

All tests were run for 200 steps to ensure that no sudden change happens in the later iterations, and C_{200} is presented along with $j_{1\%}$ and/or $j_{2\%}$, where C_j is the compliance in step j and

$$j_{k\%} = j, \text{ for which } \left| \frac{C_j - C_{200}}{C_{200}} \right| \leq k/100 \quad (26)$$

Putting it into other words: convergence criterion is reached when the compliance is within ± 1 or $\pm 2\%$ of the compliance at step 200. This criterion was proposed in order to ensure that the final solution was really reached and not only slow convergence occurred. Except for the benchmarking of the software normal convergence criterion should be used (e.g. $\left| \frac{C_j - C_{j+1}}{C_{j+1}} \right| \leq k/100$). Performing the FEA (i.e., solving a large sparse system) requires with orders of magnitudes larger computational time as compared to the actual computations performed by QSQF (which has the same time complexity as BESO). As a result, more than 99% of the CPU time is consumed by the linear algebra solver. As different research groups use very different methods (e.g., PCGM, DPCG, direct solvers), the efficiency of the algorithm is better presented by the required number of iterations, as by the CPU time.

6.1 A BENT CANTILEVER

As on the field of linear elasticity, where the examples were tested, the resulting structure does not depend on the magnitude of the load or Young's modulus, publications and books use many times small but easily comparable load and Young's modulus values. Here in *Example 1* we will use $E=1$ MPa, $\nu=0.3$, 160×40 mm domain with 160×40 discretization. The load is $F= -1$ N, the volume constraint $V_f=0.5$, while $x_{\min}=0.001$, $r_{\min}=3.0$ mm (**Figure 5**). The calculated values corresponded to the values given by Huang & Xie (2010a) for both SIMP&OC and BESO method with the current in house code (**Figure 6**). The C_{200} , $j_{1\%}$ are given for all cases and $j_{2\%}$ values for the examples with convergence history. It is important to point out that all algorithms reach the $j_{1\%}$ state at almost the same speed, but they need significantly more time to reach the C_{200} value: the SIMP method reaches the presented value only at $j=188$, the BESO at $j=46$, and the QSQF at $j=93$. However, it is essential to note that the result given by QSQF is 0.22% lower than the result of BESO method, but with a different topology! The convergence history is given for **Figure 6**, while **Figure 6d** only presents that using a different β value the QSQF method is also able to give the previously known topology. Depending on the supports and loadings, there can be asymmetric solutions as well Cheng & Liu (2011).

Note: At SIMP the values are higher due to the presence of intermediate densities.

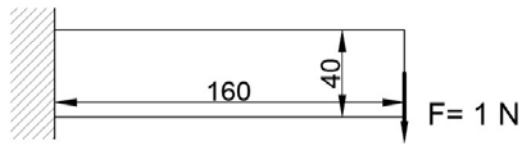


Figure 5: Boundary conditions of Problem 1

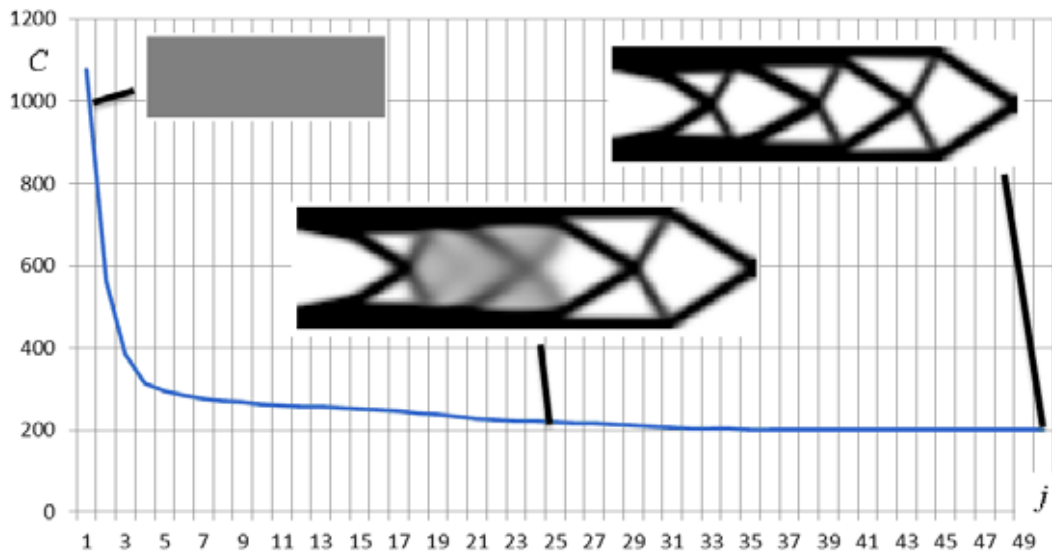


Figure 6a: Solution obtained with SIMP&OC ($C_{200}=201.2$ Nmm; $j_{1\%}=33$; $j_{2\%}=33$) for Problem 1

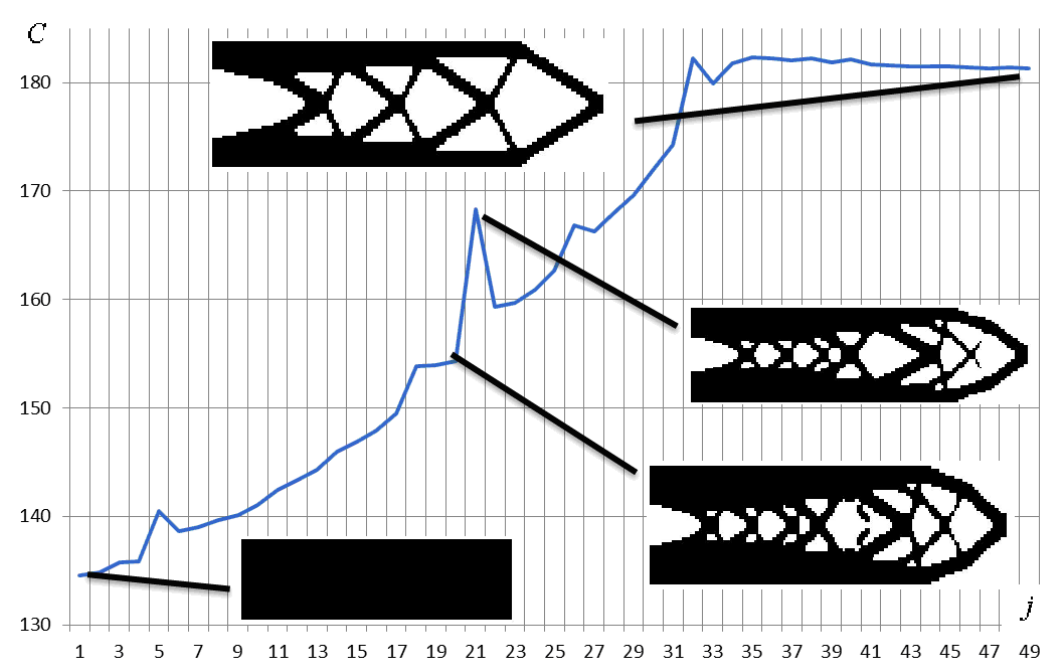


Figure 6b: Solution obtained with BESO ($C_{200}=181.4$ Nmm; $j_{1\%}=32$; $j_{2\%}=32$) for Problem 1

$ER=2.0\%$; $AR_{\max}=50.0\%$

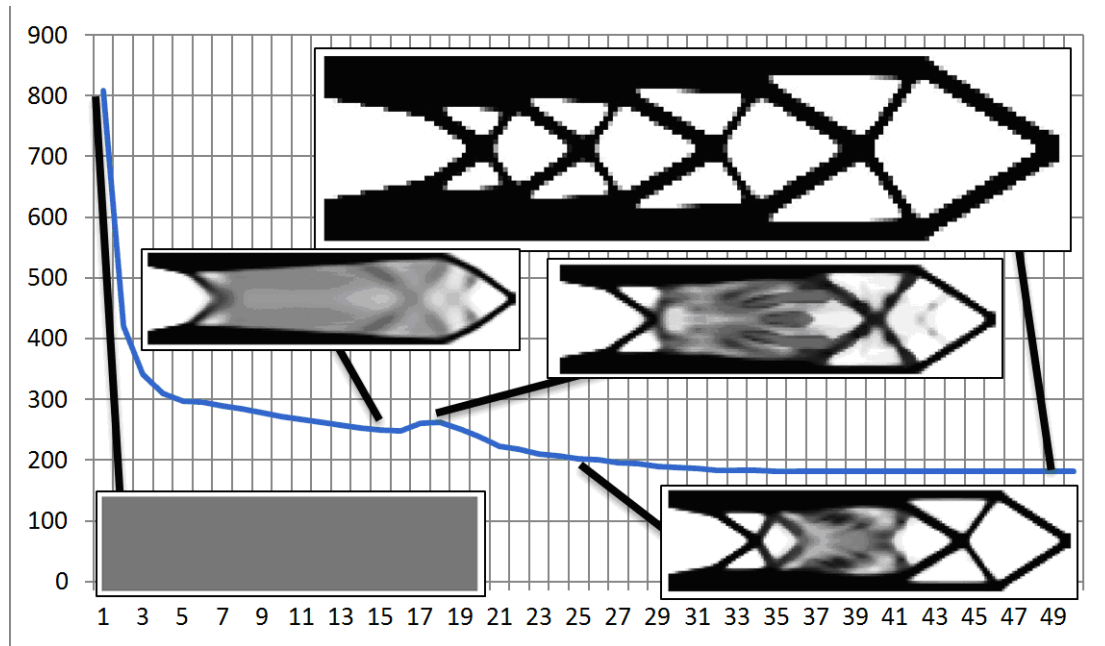


Figure 6c: Solution obtained with QSQF ($C_{200}=181.0$ Nmm; $j_{1\%}=35$; $j_{2\%}=32$, pcw.) for Problem 1
 $V_0=0.55$; $ER=1.5\%$; $H_s=0.5$; $H_d=0,5$ (it.<30); $\beta=4,6,8\dots$



Figure 6d: Solution obtained with QSQF ($C_{200}=184$ Nmm; $j_{1\%}=52$, inv.pow) for Problem 1
 $V_0=0.7$; $ER=1.5\%$; $H_s=0.5$; $H_d=0,4$ (it.<40); $\beta=4,5,6\dots$

Comparing Figure 6b and 6c one can see, that while at BESO sudden changes occur in the topology, the QSQF method rather improves the different areas in a continuous manner. While in such a simple case it is not obvious, in difficult real life problems the sudden change can cause numerical instability, and BESO will fail to converge to a solution. This effect is less likely to happen at QSQF due to the lack of sudden removal of elements. In presence of rounding errors and precision problems (present to all solvers, even more well pronounced at iterative solvers), these effects are even more likely to occur. Leveraging the convergence criteria to 10^{-8} (which is still very strict) leads to **Figures 7** and **8**. This shows, that as a result of numerical errors the BESO converges to an asymmetric solution with slightly higher compliance. This asymmetry is clearly only the result of rounding errors of the machine.



Figure 7: BESO (solution if using 10^{-8} as convergence criteria at iteration 50; $C_{50}=182.8$ Nmm)

$$ER=2.0\%; AR_{\max}=50.0\%$$



Figure 8: *QSOF* (solution if using 10^{-8} as convergence criteria at iteration 50 $C_{50}=182.6$ Nmm)
 $V_0=0.55$; $ER=1.5\%$; $H_s=0.5$; $H_d=0,5$ (it.<50); $\beta=4,6,8\dots$

6.2 OPTIMIZATION OF A BRIDGE STRUCTURE

$E=200$ GPa, $\nu=0.3$ material was used with a domain of 240 mm x 40 mm size and 240x40 mesh, and a load of $F= -100$ N, while the volume constraint was $V_f=0.5$, and $x_{\min}=0.001$, $r_{\min}=6.0$ mm. At soft-kill BESO the calculated results were close to Huang & Xie (2010a) (it was only calculated for the right hand side of the structure). The defined problem and results can be seen on **Figures 9-12**. C_{200} and $j_{1\%}$ values are presented for all cases.

It can be seen that all three methods yield to the same result. Although the iteration numbers are high for the new algorithm, this is rather the result of poor choice of the parameters.

Higher compliance of SIMP comes from the presence of intermediate densities, which is the characteristics of this type of mesh independence filter.

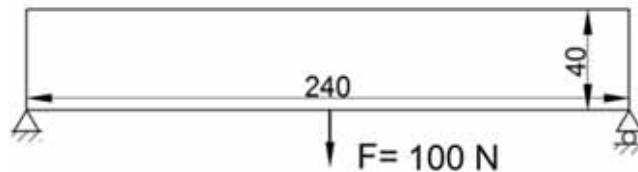


Figure 9: Boundary conditions of Problem 2



Figure 10: Solution obtained with SIMP& OC ($C_{200}=2.551$ Nmm; $j_{1\%}=30$) for Problem 2
 $r_{\min}=3.0$ mm



Figure 11. Solution obtained with BESO ($C_{200}=2.365$ Nmm; $j_{1\%}=37$) for Problem 2
 $ER=5\%$; $AR_{\max}=5\%$



Figure 12. Solution obtained with QSQF ($C_{200}=2.41$ Nmm; $j_1\%=93$, pcw.) for Problem 2
 $V_0=0,7$; $ER=5\%$; $H_s=0,5$; $H_d=0,2$ (it.<30); $\beta=5,6\dots$

6.3 3D CANTILEVER

$E=10$ GPa, $\nu=0.3$ material was used with 100 mm x 20 mm x40 mm domain and 100x20x40 mesh, the load was $F= -1000$ N, and $V_f=0.1$, $r_{\min}=3.0$ mm. In contrast with the previous cases, here we used $x_{\min}=0.0025$ for larger speed. C_{200} and $j_1\%$ values are presented for all cases.

The problem definition and results can be seen on **Figures 13-18**. The new method was able to find a new, alternative and 1.2% better topology. For the BESO method, we present two different results to show that the results of BESO method depend also on the parameters.



Figure 13: Boundary conditions for Problem 3

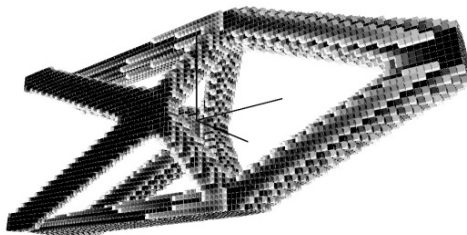


Figure 14: Solution obtained with SIMP&OC ($C_{200}=2089.6$ Nmm; $j_1\%=160$) for Problem 3

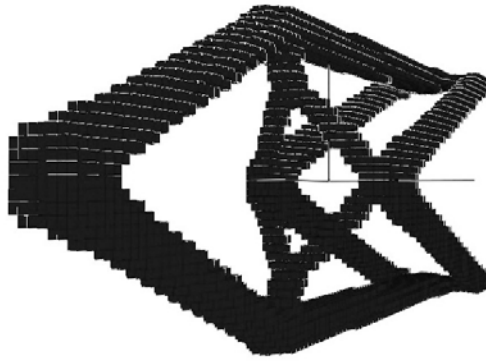


Figure 15: Solution obtained with BESO ($C_{200}=1272$ Nmm; $j_{1\%}=90$) for Problem 3
 $ER=2.5\%$; $AR_{\max}=10.0\%$

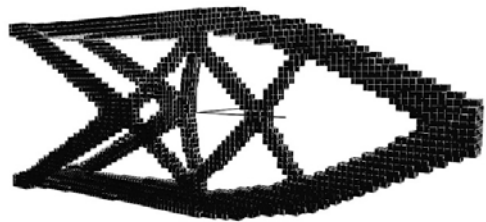


Figure 16: Solution obtained with BESO ($C_{200}=1303$ Nmm; $j_{1\%}=168$) for Problem 3
 $ER=3.0\%$; $AR_{\max}=50.0\%$

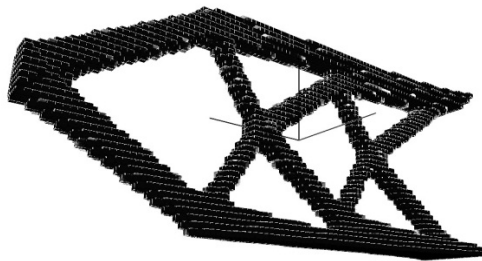


Figure 17: Solution obtained with QSQF ($C_{200}=1255$ Nmm; $j_{1\%}=144$; pcw.) for Problem 3
 $V_0=0.3$; $ER=3.0\%$; $\beta=4,5,\dots$; $H_s=0.5$, $H_d=0.5$ (it.<140)

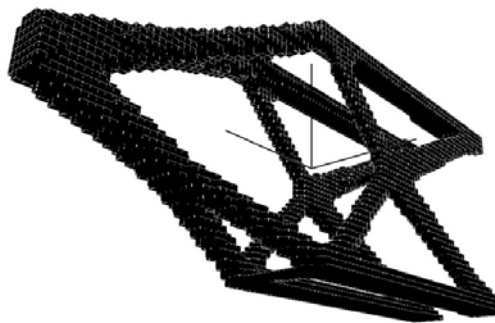


Figure 18: Solution obtained with QSQF ($C_{200}=1417$ Nmm; inv. Pow.) for Problem 3
 $ER=2.5\%$; $H_s=0.55$, $H_d=0.55$ (it.<60)

7. Future work

Although extensive testing is still needed, the method can already be applied to many cases successfully. In our future work, we would like to extend the analysis of the QSQF method to not only compliance minimization problems, and perform an extensive comparison to BESO and SIMP method. We would like to find other analogies as well taken from nature (e.g. the use of gravitational fields). An ambitious plan would be to find a generalization, which includes ESO, BESO, SIMP and QSQF algorithms too, but the existence of such an algorithm remains an open question.

8. Results and conclusions

In this article, we have proven that the BESO method can be derived using an analogy from nature. Moreover, we proposed a new algorithm, which is generalization or extension of the BESO method as it is able to work both with intermediate densities, but in special cases ($\beta=\infty$, $H_D=0$) it behaves like the BESO method. Moreover, due to the introduced new parameters, it provides flexibility and more options than BESO or SIMP. Although at first it seems that the introduction of new parameters makes the decision maker's task more difficult, but we want to emphasize, that most parameters are only present to help advanced users in their research. Usually, $H_D=0.5$; $H_s=0.5$; $\beta_0=4$ and *piecewise linear fuzzyfication functions* are the recommended, so we only have to choose ER_{max} , AR_{max} , V_f and β_{inc} . The first three of these serve the same purpose as by BESO, the last can be used to converge to discrete density values (usually $4+(1.0-2.0)i$ is a good choice, where i is the iteration counter).

The main advantage of QSQF is not that it is much faster compared to SIMP or BESO method, but the different path to the solution. As the algorithm converges through designs containing intermediate densities, it is less sensitive to rounding errors (i.e. with the use of iterative solvers for the FE model) and instead of immediately deleting bars, it slowly makes them disappear, which also removes the local peaks in the convergence history (see **Figure 7-8**). Moreover, as topology optimization at the moment only serves as a starting tool for the design, in the real world applications engineers usually use them as an intuition, and therefore they may need more alternatives for the same problem. Just note, that the QSQF method was able to find a slightly better but different topology on Problem 1 (see **Figure 8** compared to **Figure 6**) while at Problem 3, it was able to find (see **Figure 17** compared to **Figure 14-16**) a better, but significantly simpler (lower number of beams). We can also look on **Figure 19** and see the difference between the solution given by SIMP and QSQF method, where QSQF gives four almost equally distributed main columns.

We would like to point out again, that the main advantages of the algorithm are the generalization of the BESO method and the possibility to provide many different, but equally good solutions to the same problem, thus giving alternatives to the engineers, or the possibility to choose more aesthetic solutions, which is a more and more urging need in the field of civil engineering and mechanical

engineering, where attractiveness of a product is defined more and more by its design and aesthetics, rather than structural simplicity and simple functionality.

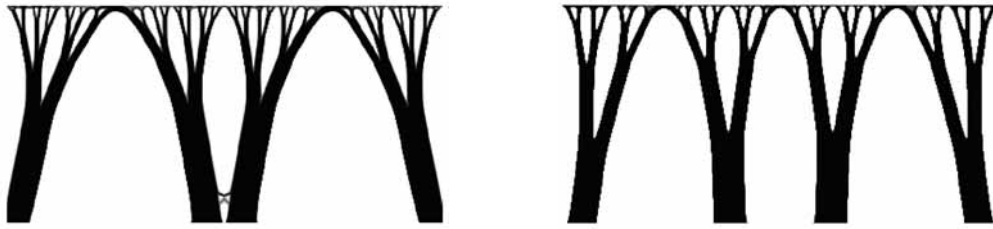


Figure 19: Different solution obtained with SIMP (left hand side) and QSOF (right hand side)
Load: distributed load on the upper surface; Dirichlet boundary on the lower surface

Acknowledgements

The research was supported by the TÁMOP 4.2.4.A/2-11-1-2012-0001 priority project entitled ‘National Excellence Program - Development and operation of domestic personnel support system for students and researchers, implemented within the framework of a convergence program, supported by the European Union, co-financed by the European Social Fund. The research was supported also by the Hungarian Scientific Research Fund OTKA T 109860 projects and was partially carried out in the framework of the Center of Excellence of Innovative Engineering Design and Technologies at the University of Miskolc.

References

-
- Bendsøe MP (1989): Optimal shape design as a material distribution problem. *Structural Optimization*, Vol 1, pp. 193–202.
- Bendsoe, M. P., Sigmund, O. (1995): Topology Optimization – Theory, Methods and Applications. Springer.
- Burczyński, T., Bonnet, M., Fedeliński, P. & Nowakowski, M. (2002) Sensitivity Analysis and Identification of Material Defects in Dynamical Systems, *Systems Analysis Modelling Simulation*, Vol. 42, No. 4, pp. 559-574.
- Bruggi M., Cinquni C., (2011) Topology optimization for thermal insulation: an application to building engineering. *Engineering Optimization*, Vol. 43. No. 11, pp. 1223-1242.
- Cheng, G. & Yan, K. (2013) A hybrid approach to structural topology optimization of vehicle for crashworthiness, WCSMO10, 2013 May, Orlando, USA, Abstract book 5188, E-Proceedings 5188.pdf, 10 p. ISBN-13:978-0-615-85240-9

- Cheng, G. & Jiang, Z. (1992) Study on Topology Optimization with Stress Constraints, *Engineering Optimization* Vol. 20, No. 2, pp. 129-148, DOI:10.1080/03052159208941276
- Cheng, G. & Liu, X. (2011): Discussion on symmetry of optimum topology design, *Structural and Multidisciplinary Optimization*, Vol. 44, No. 5, pp. 713–717, DOI 10.1007/s00158-011-0686-z
- Cinquini, C., Mariani, C., & Venini, P. (1997) Optimal Robust Design of Novel Materials: Problems of Stability and Vibrations, *Engineering Optimization*, Vol. 29, No. 1-4, pp. 323-245
- Huang, X. & Xie, Y.M. (2007): Convergent and mesh independent solutions for bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, Volume 43, Issue 14, 1039-1049.
- Huang, X. & Xie, Y-M. (2010a): A further review of ESO type methods for topology optimization, *Structural and Multidisciplinary Optimization*, Vol. 41, No. 5. pp. 671–683, DOI 10.1007/s00158-010-0487-9
- Huang, X. & Xie, Y. M. (2010b): *Evolutionary Topology Optimization of continuum Structures – Methods and Applications*. Wiley.
- Phan,D.T., Lim, J.B.P., Sha, W., Siew, C.Y.M., Tanyimboh, T.T., Issa, H.I. & Mohammad, F.A. (2013) Design optimization of cold-formed steel portal frames taking into account the effect of building topology, *Engineering Optimization* Vol. 45, No. 4, pp. 415-433.
- Qiao, H. & Liu, S. (2013) Topology optimization by minimizing the geometric average displacement, *Engineering Optimization*, Vol. 45, No. 1, pp. 1-18.
- Querin, O.M., Steven G.P., Xie, Y-M. (1998): Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations*, Vol. 15, No. 8, pp. 1031-1048.
- Sigmund, O. & Petersson, J. (1998): Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh independencies, and local minima. *Structural and Multidisciplinary Optimization*, Vol. 16, No. 1, pp. 68-75.
- Sigmund, O. (2001): A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*. Vol. 21, No. 2, pp. 120-127.
- Torii, A.J., Lopez, R.,H. & Biondini, F. (2012) An approach to reliability-based shape and topology optimization of truss structures, *Engineering Optimization*, Vol. 44, No. 1, pp. 37-53.
- Xie, Y.M. & Steven. G.P. (1993): A Simple Evolutionary Procedure for Structural Optimisation, *Computers and Structures*, Vol. 49, No. 5, pp. 885-896.
- Zhou, M. & Rozvany, G.I.N. (1991): The COC algorithm, part II: Topological, geometry and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, Vol. 89. pp. 309-336.
- Zhou, M. & Rozvany, G.I.,N. (2001): On the validity of ESO type methods in topology optimization. *Structural and Multidisciplinary Optimization*, Vol. 21, No. 1, pp. 80-83.