Available online at www.eccomasproceedia.org
Eccomas Proceedia EUROGEN (2023) 315-323

ECCOMAS
Proceedia

EUROGEN 2023
15th ECCOMAS Thematic Conference on
Evolutionary and Deterministic Methods for Design, Optimization and Control
N. Gauger, K. Giannakoglou, M. Papadrakakis, J. Periaux (eds.)
Chania, Crete Greece, 1–3 June 2023

VOLUME CONSERVING BOUNDARY SMOOTHING FOR 2D TOPOLOGY OPTIMIZATION SOLUTIONS

Nikhil Singh[1] **and Anupam Saxena**[1]

[1]Indian Institute of Technology Kanpur
Mechanical Engg. Dept., IIT Kanpur, Kanpur, Uttar Pradesh, India, 208016
e-mail: {singhn, anupams}@iitk.ac.in

Abstract. *Solution in topology optimization is a connected set of discrete polygonal /polyhedron finite elements. As a consequence, boundary of the continuum contains numerous notches undesirable from a manufacturing/3D printing view point, thus motivating the development of boundary smoothing techniques. Noting that volume constraint plays an essential role in topology optimization, we propose a novel volume conserving boundary smoothing method for a generic 2D tessellation. Herein, boundary smoothing is expressed as an optimization problem. The formulation proposes a boundary smoothing function based on local angels and develops an area constraint using the shoelace formula. Aforementioned evaluations are accompanied by evaluation of sensitivities for both objective and constraint(s). Finally, the optimization process takes boundary nodes as input and optimizes their location for the boundary smoothing function subject to the constraint that the area/volume of the structure remains unchanged. The method's ability is demonstrated by performing boundary smoothing on solutions to many well known topology optimization problems.*

Keywords: Boundary smoothing, Topology optimization, Volume conservation, Boundary optimization.

Peer-review under responsibility of the organizing committee of EUROGEN 2023.
doi: 10.7712/140123.10207.18921

1 INTRODUCTION

Topology optimization is a diverse design tool used to obtain optimal topologies for a variety of objectives/purposes. The method usually employs FE analysis to evaluate objectives and thus, the final solution is expressed as a set of discrete finite elements. Boundary of the optimized structure is usually defined via straight edges (for 2D cases) or plane surfaces (for 3D cases). This leads to sharp notches, presence of which is undesirable because (a) they lead to regions of stress concentration and (b) are inconvenient for manufacturing purposes. Additionally, topology optimization involving contact requires evaluation of normals at the boundary. In such cases, sharp notches lead to jump in normals which interferes with contact modeling, thus necessitating smooth boundaries.

Boundary smoothing for topology optimization solutions is previously proposed in various works. Kumar and Saxena [1] construct a fictitious contour which passes through the mid point of boundary edges. Boundary nodes are then relocated on this contour. This, smoothing method is implemented to evaluate objectives for intermediate topologies as well. Kumar et al. [2] implement the same boundary smoothing technique to design contact aided compliant mechanisms. Li et al. [3] suggest conducting boundary smoothing using a predefined table of possible configurations material can take within a boundary element. This approach is confined to rectangular elements. Nana et al. [4] propose reconstruction of the 3D topology solution using bar elements. Given a topology, the method identifies a skeleton like structure, and based on various parameters proposes constructing bars surrounding the skeleton.

Owing to the fact that volume constraint plays an important role in topology optimization, herein, we propose a novel boundary smoothing method which conserves volume. This approach presents the smoothing process an optimization problem. Section 2 discusses various steps involved in boundary recognition. Next, a boundary smoothing function is constructed (section 3), minimizing which produces smoother boundaries. This is accompanied with an area constraint which conserves volume (section 4). Eventually, independent optimization problems are established (section 5) and solved for each void boundary. Boundary smoothing of various topologies is conducted in section 6, followed by a discussion on effectiveness of the method. Conclusions and future scope related to the subject are shared in section 7

2 BOUNDARY IDENTIFICATION

The first step for any boundary smoothing process is to identify the boundary nodes and arrange them in order of connectivity. For a 2D topology, the boundary is defined by a piece-wise linear curve and thus each boundary node is connected to two other boundary nodes. Special cases may arise at locations where the structure has unit cell thickness or displays point connectivity. To this end, a boundary identification algorithm specific to requirements of the problem is developed.

Topology optimization ideally outputs a 0 or 1 density within each element of the domain, providing information on weather an element/cell is void or solid respectively. Voids is structure have independent boundaries, that is, each void creates an independent closed loop. This makes it easier to identify and work with void boundaries instead of a single solid boundary. We identify all voids, that is, outer and inner loops and their boundaries. To ensure that all voids are closed loops, nodes which do not lie on the solid-void interface but lie on the domain boundary and are surrounded by void elements are also considered part of the void boundary. Void boundary nodes which also lie on the domain boundary are deemed *dormant boundary nodes* while those on the solid-void interface (and not on the domain boundary) are deemed as *active*

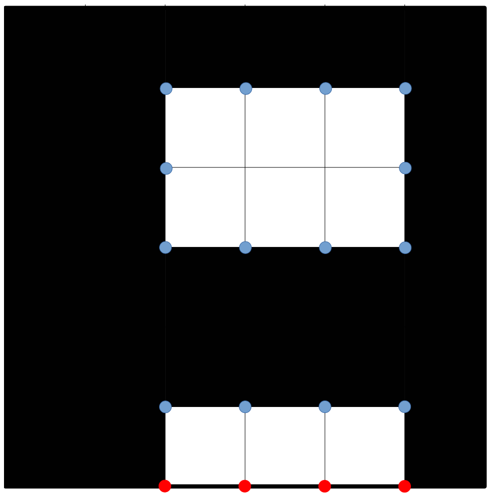


Figure 1: Active (in blue) and dormant (in red) void boundary nodes

boundary nodes. Fig. 1 highlights void boundary nodes for the corresponding topology. Active and dormant boundary nodes are colored using blue and red respectively. Note that, because the boundary is defined using piece-wise linear segments and nodes, and voids are independent closed loops, each void is an n sided polygon where n is the number of rectilinear edges along its boundary. Various steps/stages involved in identifying void boundaries are discussed below.

Obtaining the required boundary information has two steps: The first step is to identify nodes at the solid-void interface and those at the domain boundary which are surrounded by void elements. This information can be extracted using density distribution and the connectivity matrix used in performing topology optimization. The second step involves identifying independent loops. We commence with a random node, i, and identify neighboring nodes which also lie on the void boundary. This gives two nodes, h and j. One of these nodes is picked at random, say j, and the same process is repeated on this node, which again leads to two nodes one of which is node i. As node i is already identified, the other node is chosen and the same process is repeated until one circles back to node i. This provides boundary nodes for one loop, ordered according to connectivity. The same process is repeated until all boundary nodes are accounted for.

Once all the void boundaries, i.e., inner and outer loops are obtained, we construct an independent optimization problem for each void. These problems optimize location of nodes to give smoother boundaries. A constraint is imposed to ensure volume conservation. The construction of objective function (section 3) and constraint (section 4) for the optimization problem is discussed next.

3 BOUNDARY SMOOTHING FUNCTION

Solutions in 2D topology optimization is a collection of discrete polygons, with boundaries of the final structure defined by edges of these polygons. Due to the discreteness involved with the boundaries, an idea of smoothness at a boundary point, needs to be established. Correspondingly, a measure for smoothness at a point is also required. This section focuses on establishing the required definitions and measures, and implementing them to construct the Boundary Smoothing Function (BSF) which when minimized smoothens the structural boundary. Additionally, some properties of the aforementioned BSF are discussed.

3.1 Smoothness at a point

A measure for smoothness at a boundary node is required to construct the Boundary Smoothing Function (BSF). A 2D curve is said to be smooth at a point if the left and right derivatives at that point are equal. As the boundary is piece-wise discrete, the left and right derivatives at a node are slopes of the edges connected to that node (see Fig. 2). For construction of the Boundary Smoothing Function (BSF), we look at local properties of edges and angles at each boundary node.

Fig. 3(a) and (b) demonstrate two cases: In both, slope of the lines intersecting at point

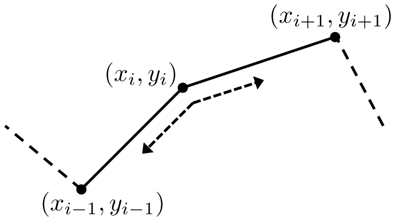
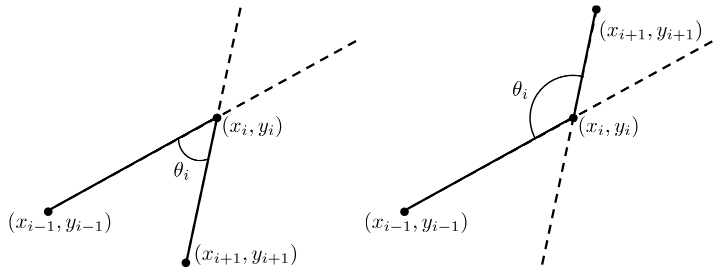


Figure 2: Slope at a point



(a) (b)

Figure 3: Different local angles for lines of same slope

$\mathbf{X}_i \equiv (x_i, y_i)$ is the same but from observation, the connection in case (a) leads to a notch while that in (b) leads to a smoother curve. Thus, a BSF constructed using only the slopes of lines will be unable to distinguish between the two scenarios. Therefore, we utilize the local angle between the two boundary edges to construct the BSF.

The boundary is considered smoother as the local angle approaches π radians. Thus, smoothness at a point is measured via the function

$$S_i = (1 + \cos\theta_i)^2 \quad (1)$$

where θ_i is the local angle at point $\mathbf{X}_i$. Note that, S_i is minimum when $\theta_i = \pi$ and maximum for $\theta_i = 0$. Additionally, S_i gives different values for the two cases described above and therefore, is able to distinguish between the two. We evaluate $\cos\theta_i$ using the following expression,

$$\cos\theta_i = \frac{(\mathbf{X}_{i+1} - \mathbf{X}_i) \cdot (\mathbf{X}_{i-1} - \mathbf{X}_i)}{\|\mathbf{X}_{i+1} - \mathbf{X}_i\| \times \|\mathbf{X}_{i-1} - \mathbf{X}_i\|}. \quad (2)$$

Employing the above measure of smoothness the BSF is constructed as,

$$\text{BSF} = \sum_{i=1}^{n} (1 + \cos\theta_i)^2 \quad (3)$$

where n is the number of *active boundary nodes* associated to a given void.

3.2 Equal angle property of BSF

Herein, we justify the choice of BSF by showing that, for a set of n points, which when connected in order forming an n-sided polygon, the BSF is minimized when all local angles are equal. This shows that, in an unconstrained system, minimizing BSF gives the closest approximation of a closed, smooth curve one can achieve using n discrete points. Let $\{\mathbb{N}\}$ be a set of n ordered nodes such that connecting them in the ordered manner gives a closed polygon. The optimization function for this system is,

$$\text{Minimize} \qquad F(\theta_i) = \text{BSF} \equiv \sum_{i\in\{\mathbb{N}\}} (1+\cos\theta_i)^2. \tag{4}$$

As the n nodes when connected form a polygon,

$$\sum_{i\in\{\mathbb{N}\}} \theta_i = (n-2)\pi. \tag{5}$$

This gives,

$$\theta_k = (n-2)\pi - \sum_{\substack{i\in\{\mathbb{N}\}\\ i\neq k}} \theta_i \tag{6}$$

implying the last angle is a function of its $n-1$ predecessors. Thus, the objective function is modified to,

$$\text{Minimize} \qquad F(\boldsymbol{\theta}) = \sum_{\substack{i\in\{\mathbb{N}\}\\ i\neq k}} (1+\cos\theta_i)^2 + (1+\cos\theta_k)^2 \tag{7}$$

where $\boldsymbol{\theta}$ is the vector of local angles, $\theta_i,\ i\in\{\mathbb{N}\} \mid i\neq k$, and expression for θ_k is given in eqn. 6. To find the local minima, we evaluate gradients of $F(\boldsymbol{\theta})$ and equate it to 0. This gives,

$$\frac{\partial F(\boldsymbol{\theta})}{\partial \theta_i} = 2(1+\cos\theta_i)\sin\theta_i - 2(1+\cos\theta_k)\sin\theta_k = 0$$

$$\implies (1+\cos\theta_i)\sin\theta_i = (1+\cos\theta_k)\sin\theta_k \qquad \forall\, i\in\{\mathbb{N}\} \mid i\neq k \tag{8}$$

$$\text{This implies,} \qquad (1+\cos\theta_i)\sin\theta_i = c \qquad \forall\, i\in\{\mathbb{N}\} \tag{9}$$

where c is a constant. This allows for two values θ_i in the interval $(0,\pi)$. Thus, the local angles should satisfy eqns. 5 and 9. This is only possible when

$$\theta_i = \frac{(n-2)\pi}{n} \quad \forall\, i\in\{\mathbb{N}\}. \tag{10}$$

Thus, BSF is minimized when all local angles in the polygon are equal. Constraint evaluation is discussed next.

4 AREA CONSTRAINT

This section constructs an area constraint to ensure that volume of the continuum remains conserved through the boundary smoothing process. As the domain is divided in solid and void regions,

$$A_\Omega = A_s + A_v \tag{11}$$

where A_Ω is the area of design domain, A_s is the area of solid region and A_v is the area covered by the void region. As will be discussed ahead, the smoothing process does not alter the location of domain boundary nodes, thus A_Ω remains constant. Therefore, conserving area of the void region ensures area conservation for the solid region.
As discussed in section 2, boundary smoothing for each void is done independently. We therefore conserve the volume of each void. Utilizing the fact that each void is a polygon, we implement the shoelace formula to evaluate the area enclosed within a void. The shoelace formula evaluates the area of a polygon as,

$$A = \frac{1}{2}\left\{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + \dots \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix} + \dots \begin{vmatrix} x_n & y_n \\ x_1 & y_1 \end{vmatrix}\right\} \tag{12}$$

where (x_i, y_i) are the coordinates of vertex i and vertices are numbered in order of connectivity, that is, vertex i is connected to vertex $i-1$ and $i+1$. The method for obtaining ordered vertices for void is discussed in section 2. The area constraint is given as,

$$A^k - A_v^k = 0 \tag{13}$$

where A_v^k is the original area of void k and A^k is the area of void at any given optimization iteration. Both A_v^k and A^k are evaluated using Eqn. 12.

5 PROBLEM FORMULATION AND SENSITIVITY ANALYSIS

This section develops the optimization formulation for boundary smoothing. As stated in section 2, an independent optimization problem is solved for boundary smoothing of each void. The smoothing problem for a void is formulated as; optimize location of *active boundary nodes* to

$$\text{Minimize} \qquad F(\boldsymbol{\theta}) = \sum_{i\in\{\mathbb{N}^k\}} (1+\cos\theta_i)^2 \tag{14}$$

$$\text{such that} \qquad A^k - A_v^k = 0$$

$$\text{and} \qquad x_{0i} - a \le x_i \le x_{0i} + a$$

$$y_{0i} - a \le y_i \le y_{0i} + a$$

where $\cos\theta_i$ is evaluated from node locations as in eqn. 2, $\{\mathbb{N}^k\}$ is the set of active boundary nodes for the k^{th} void, A^k and A_v^k are the current and original area of voids respectively, (x_{0i}, y_{0i}) and (x_i, y_i) are the original and current location of node i, and a is the edge length of the square region within which the node is bound.
The optimization problem is solved using a gradient based method. Sensitivity analysis of the objective and constraint are conducted as follows.

5.1 Objective and constraint gradients

Using eqn. 2 and 14, the objective gradient with respect to design variables x_j and y_j, $j \in \{\mathbb{N}^k\}$ are evaluated as,

$$\frac{\partial F(\boldsymbol{\theta})}{\partial x_j} = \sum_{i\in\{\mathbb{N}^k\}} 2(1+\cos\theta_i)\frac{\partial \cos\theta_i}{\partial x_j} \tag{15}$$

$$\text{and} \qquad \frac{\partial F(\boldsymbol{\theta})}{\partial y_j} = \sum_{i\in\{\mathbb{N}^k\}} 2(1+\cos\theta_i)\frac{\partial \cos\theta_i}{\partial y_j} \qquad \text{respectively.}$$

As the nodes are numbered in ordered manner, nodes adjacent to node j are numbered $j+1$ and $j-1$, thus,

$$\frac{\partial \cos\theta_i}{\partial x_j} = 0 \quad \text{and} \quad \frac{\partial \cos\theta_i}{\partial y_j} = 0 \quad \text{for} \quad i \in \{\mathbb{N}^k\} \ | \ i \neq \{j-1, j, j+1\}. \tag{16}$$

Substituting eqn. 16 into eqn. 15 gives,

$$\frac{\partial F(\boldsymbol{\theta})}{\partial x_j} = \sum_{i=j-1}^{j+1} 2(1+\cos\theta_i)\frac{\partial \cos\theta_i}{\partial x_j} \tag{17}$$

and
$$\frac{\partial F(\boldsymbol{\theta})}{\partial y_j} = \sum_{i=j-1}^{j+1} 2(1+\cos\theta_i)\frac{\partial \cos\theta_i}{\partial y_j}$$

where
$$\frac{\partial \cos\theta_i}{\partial x_{i-1}} = \frac{(x_{i+1}-x_i)}{d_{(i+1,i)}d_{(i-1,i)}} - \frac{(x_{i-1}-x_i)}{d^2_{(i-1,i)}}\cos\theta_i,$$

$$\frac{\partial \cos\theta_i}{\partial x_{i+1}} = \frac{(x_{i-1}-x_i)}{d_{(i+1,i)}d_{(i-1,i)}} - \frac{(x_{i+1}-x_i)}{d^2_{(i+1,i)}}\cos\theta_i \tag{18}$$

and
$$\frac{\partial \cos\theta_i}{\partial x_i} = -\left[\frac{\partial \cos\theta_i}{\partial x_{i+1}} + \frac{\partial \cos\theta_i}{\partial x_{i-1}}\right]$$

where $d_{(j,i)}$ is the distance between (x_j, y_j) and (x_i, y_i) given by

$$d_{(j,i)} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}.$$

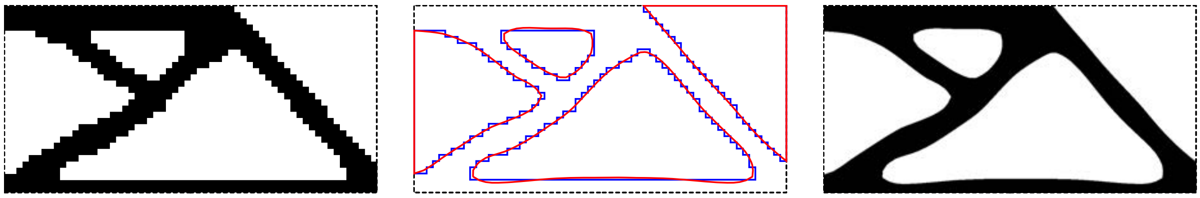
Expressions for $\frac{\partial \cos\theta_i}{\partial y_{i-1}}$, $\frac{\partial \cos\theta_i}{\partial y_i}$ and $\frac{\partial \cos\theta_i}{\partial y_{i+1}}$ can be evaluated in a similar fashion.
Next we look at the constraint gradient. Gradients for the equality constraint in eqn. 14 can be evaluated using the formula in eqn. 12 as,

$$\frac{\partial A^k}{\partial x_j} = \frac{y_{j+1} - y_{j-1}}{2} \quad \text{and} \quad \frac{\partial A^k}{\partial y_j} = \frac{x_{j+1} - x_{j-1}}{2}. \tag{19}$$

Note that A^k_v is the initial area of the void and thus, is a constant. Therefore, it does not participate in gradient evaluation. Boundary smoothing can be performed using the above formulation.

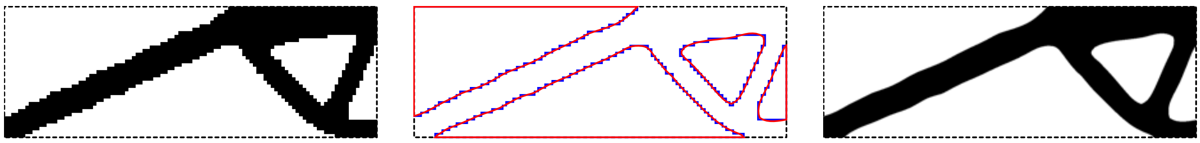
6 EXAMPLES AND DISCUSSION

We demonstrate boundary smoothing, using the above formulation, on solutions to some well known topology optimization problems. Fig. 4c, 5c, 6c present boundary smoothed for solutions of the cantilever beam, mid-load beam and displacement inverter problems presented in Fig. 4a, 5a, 6a respectively. Fig. 4b, 5b, 6b show the initial and final void boundaries. We highlight the initial and final void boundaries as blue and red respectively, while boundary of the design domain is highlighted by dashed black lines. Solutions to the topology optimization problems are obtained using the nFP method proposed in Singh and Saxena[5]. As the objective is to demonstrate intricacies of boundary smoothing, topology optimization is conducted on a rather course mesh.



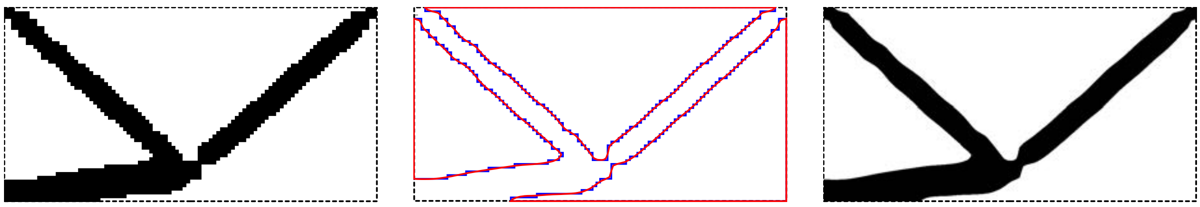
(a) Original solution (b) Initial and final boundaries (c) Smoothed solution

Figure 4: Boundary smoothing on the solution for Cantilever beam problem



(a) Original solution (b) Initial and final boundaries (c) Smoothed solution

Figure 5: Boundary smoothing on the solution for Mid-load beam problem



(a) Original solution (b) Initial and final boundaries (c) Smoothed solution

Figure 6: Boundary smoothing on the solution for displacement inverter problem

Observing the smoothing of solution in Fig. 4a, it is realized that, bottom edge of the void which is initially straight, is now curved which from a manufacturing point of view is undesirable. This, arch in the void boundary is a consequence of the area constraint. Such undesirable consequence of the smoothing process can be avoided by excluding the nodes on straight edge from the optimization process, that is, those nodes can be fixed by excluding them from the set of design variables. Alternatively, bounds on nodal coordinates can be tightened. Thus, a method for identification of desirable and undesirable features in a structure is required, and is left for a future work. Another approach to resolve the issue would be to propose an alternate BSF which does not interfere with straight edges. On the other hand, solution in Fig. 6c shows that the smoothing process counteracts the local thinning in Fig. 6a. The objective attempts to reduce sharp changes in gradients at the points of local thinning thus adding material to produce a smoother boundary. Such alterations in boundary are considered desirable from a manufacturing point of view.

The method as described above is confined to 2D topologies. Developing an extension of the method for a generic case of 3D topologies is not apparent. This is because, the above method depends on local angles which is hard to define for boundary surfaces. Additionally, the method requires nodes be stored in the order of connectivity which in a 3D case is not apparent as each node on the surface is connected to more than 2 nodes. Thus, a different approach needs to be developed, to conduct volume conserving boundary smoothing in 3D.

7 CONCLUSION

A volume conserving boundary smoothing method for 2D topology optimized solutions is established. Boundary smoothing is presented as an independent optimization problem for each void, wherein, objective is a function of local angles at the edges and an equality constraint is established to conserve volume. Effectiveness of the method is demonstrated via conducting boundary smoothing of solutions to some well known problems. The method is shown to counteract local thinning but is also shown to alter some straight edges into curves. Thus, the approach presents desirable, as well as, undesirable behavior. The presented methodology even though relevant and effective, has some drawbacks and presents scope of further improvement.

REFERENCES

[1] Kumar Prabhat, and Anupam Saxena, *On topology optimization with embedded boundary resolution and smoothing.* Structural and Multidisciplinary Optimization 52 (2015): 1135-1159.

[2] Kumar, Prabhat, Roger A. Sauer, and Anupam Saxena. *On topology optimization of large deformation contact-aided shape morphing compliant mechanisms.* Mechanism and Machine Theory 156 (2021): 104135.

[3] Li, Zhi, et al. *Smoothing topology optimization results using pre-built lookup tables.* Advances in Engineering Software 173 (2022): 103204.

[4] Nana, Alexandre, Jean-Christophe Cuillière, and Vincent Francois. *Automatic reconstruction of beam structures from 3D topology optimization results.* Computers and Structures 189 (2017): 62-82.

[5] Singh, Nikhil, and Anupam Saxena. *Normalized Field Product method for Topology Optimization.* arXiv preprint arXiv:2208.10879 (2022).