Applying beam sizing concepts along with topology optimization on the design of continuum structures

Tarcísio L. de Oliveira[1], Jonas de Carvalho[1]

[1]*Dept. of Mechanical Engineering, Sao Carlos School of Engineering, University of Sao Paulo*
Av. Trabalhador San-carlense, 400, 13566-590, Sao Carlos, Sao Paulo, Brazil
tarcisio.oliveira@usp.br, prjonas@sc.usp.br

Abstract. Using beams as a modeling and design tool in structural design has long been displaced by more recent numerical methods, such as finite element analysis and structural optimization, while those concepts became more restricted to the design of trusses and shafts. But is there still room for it to be applied in contemporary design of continuum structures? This research investigates some possible aplications of beam theory and beam sizing concepts when used along with contemporary technologies such as topology optimization, additive manufacturing and numerical methods, and how it could impact the structural design process.

Keywords: Mechanical design, Topology optimization, Design for additive manufacturing, Design automation

1 Introduction

The term "Design Automation" is a broad term which can be used to describe the development of techniques, algorithms and technologies which automate the design process, and which can also used to describe the development of automated systems, such as in ASME's Design Automation Conference [1], as defined by Ragsdell [2].

Design automation is more well developed in Electronic Engineering, in the form of Electronic Design Automation, a field dedicated the automation of the design of integrated circuits using micro- and nanometric transistors, which according to Lavagno et al. [3] is the main factor behind the increasing processing speeds of contemporary digital devices.

In Mechanical Engineering, design automation is not so well developed, especially for continua. Although there are optimization algorithms which are commonly used to aid the design process (namely size, shape and topology optimization), the process of designing a structure or part is still mostly manual and iterative. There is however some research on the field, such as by use of generative adversarial networks (in works of Oh et al. [4] and Shu et al. [5]), machine learning (Sharpe et al. [6] and Camburn et al. [7]), generative design (Oh et al. [8]), among others. Therefore, the field would highly benefit from a method which could streamline the structural design process from its inception.

2 Algorithm

This research is investigating the application of beam sizing methods as means to automatically generate continuum structures which are suited to withstand the loads that they are required to support by design. The sizing method consists of obtaining the minimum cross-section dimensions which satisfy the relation:

$$J(\mathbf{X}) = \frac{L(\mathbf{X})}{S(\mathbf{X})}. \tag{1}$$

Where L is the internal load at point $\mathbf{X}$, S the maximum stress on the cross-section, and J a geometric function related to the cross-section and the type of stress being measured. For normal stresses, for example, it is the area of the cross-section A, and for flexural stresses, the resistance moment W_f.

For usual beam sizing, this is used to determine the cross-section of the entire beam or set of beams. However, for a continuum application, it can be used to determine the cross-section at every point. Applying it to a 2D case, for example, for a beam with rectangular cross-section of thickness t, assuming an isotropic material, the height $h(\mathbf{X})$ can be obtained by:

$$h(\mathbf{X}) = \max\left\{\sqrt{\frac{6|M_z(\mathbf{X})|}{t\sigma_{max}}}, \frac{|P_n(\mathbf{X})|}{t\sigma_{max}}, \frac{3|P_s(\mathbf{X})|}{2t\tau_{max}}\right\}. \tag{2}$$

Where $M_z(\mathbf{X})$ is the internal bending moment, P_n, the internal normal force, P_s, the internal shear force, and σ_{max} and τ_{max} are maximum design stresses.

In order to size the beam, it needs to have a ”skeleton” first, that is, one must define its beginning, end, and the ”path” it takes between those two points. While that could be done manually, an automatic algorithm would be preferred. To do this, it was opted to use a *pathfinding* algorithm. Those kinds of algorithms are used to connect each load to each support, forming a beam for each pair.

For a discretized environment, such as a finite element mesh, the A* algorithm can used, as developed by Hart, Nilsson and Raphael [9], as it is simple to be implemented, especially for regular square grids. It consists of:

1. Obtaining the position which has the lowest cost among the collected ones;
2. Obtaining the neighboring positions which can be traversed to and calculating their costs;
3. Adding those positions to the collection, linking them to current one, which is then removed from the collection.

That sequence is repeated until the objective is reached, and the path is obtained by tracing it back from the final position. The cost function, used to obtain the cost for each node, is a heuristic function used to guide the path towards the objective. In this case, it is a distance function, such as Euclidian distance or L^1 distance.

For a continuum setting, one could use a visibility graph algorithm in order to obtain the pathfinding nodes, such as the one developed by Yu [10], and find the path using A*.

While the geometry obtained from the sizing step is designed to be *sufficient* for the loads imposed on it, it is also unoptimized. In the case of problems that consist of predominantly bending loads, for example, regions closer to the center of the cross-section will be much less stressed than the beam's boundary. Consequently, the geometry should also be further improved by use of topology optimization algorithms.

It should be noted, however, that the use of topology optimization will affect the cross-sections of the beams, and consequently are likely to increase the stress levels. Therefore, the sizing algorithms may also have to be tuned to account for that, such as by introducing a constant to increase the final dimensions of each cross-section, striving for a balance between dimensions and volume.

3 Implementation

A proof-of-concept was implemented in Python, building upon ToPy, made by Hunter [11, 12], a Python implemention of topology optimization for minimal conformity, heat conduction and mechanism synthesis using the linear finite element method.

The implementation developed is a simplification of the algorithm described, designed to work using beams that are a single element wide and using the resulting element stresses. It can be found at: `https://github.com/TarcisioLOliveira/topy`. An article based on this implementation is currently under review.

Another more complete implementation is under active development, being based on constructive solid geometry. For finite element analyses, GT9 elements are beng used, as described by Wen, Long and He [13], in order to decrease computational load while maintaining precision comparable to quadratic elements, and for topology optimization, minimal volume with global stress constraints and MMA, based on work by Bendsøe and Sigmund [14], Le et al [15] and Svanberg [16].

4 Possible applications being researched

Preliminary results obtained from the proof-of-concept implementation raised some possible applications for this algorithm, which are currently under research. Those possibilities are described in the following subsections.

4.1 Design automation in general

Design automation is the main possible use of this algorithm, considering that it is able to automatically generate and roughly size a structure based solely on a design space and the boundary conditions. Some challenges are involved in this, such as determining the necessity of and possible optimal values of the cross-section expansion constant previously described, as well as handling multiple beams within the same design space and their crossing and touching one another, cases which are not accounted for in usual beam theories such as Euler-Bernoulli and Timoshenko.

4.2 Heuristics for topology optimization

Another possible application is as a heuristic method to improve the speed of topology optimization.

Being an iterative algorithm which requires information of every element of a geometry at every iteration, topology optimization naturally requires considerable time to generate results for large topologies and fine meshes, especially considering it needs fine meshes in order to reach well-defined results, as described by Bendsøe and Sigmund [14].

As the beams are created and shaped according to the limitations defined by the input geometry, the algorithm could also be interpreted as a *design space restrictor*, restricting the ground structure to a topology which is designed to withstand the design loads. Being a smaller structure, it would result in a smaller amount of elements for the same meshing settings. On the other hand, it would also restrict the *solution space*, meaning it may not reach the most optimal solution it could when using the original ground structure, in terms of volume reduction and stress levels.

Tests done on the first implementation reveal that this design space restriction can considerably reduce optimization time depending on the amount of elements removed, even accounting for the time spent generating and sizing the beams. Figure 1 shows the result of the sizing step for an L-bracket problem, made using the new implementation. Final dimensions of the sized beams are of course dependent on the boundary conditions, especially load intensity.

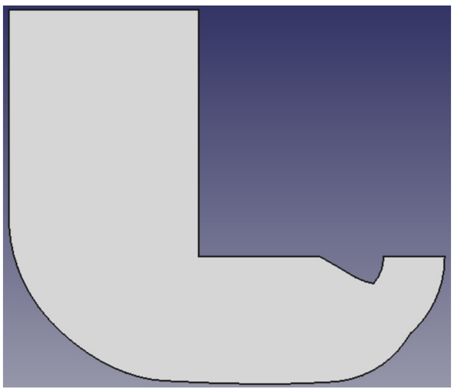


Figure 1. Result of the sizing step for an L-bracket problem using the new implementation.

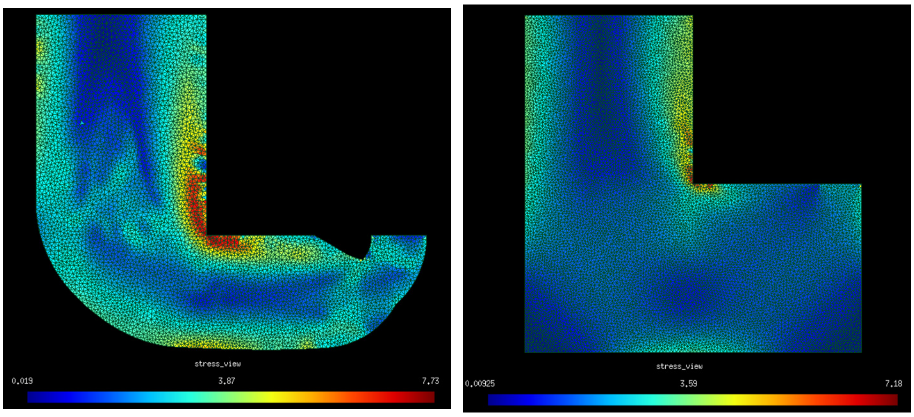
4.3 Build time reduction for additive manufacturing

The algorithm could also be used as a heuristical optimization of structures for additive manufacturing.

Build time is one of the main problems of additive manufacturing, as small pieces can take hours to be fabricated, even when using consumer 3D printing machines and not aiming for maximum geometrical accuracy. According to Teitelbaum [17] and Zhang et al. [18], some of the factors which increase build time are piece height, base area, surface area, generation of support structures and height of support structures, being the piece's height the most important factor.

Therefore, in the case of useful parts fabricated by additive manufacturing, reducing their dimensions while ensuring that their function is maintained would allow them to be fabricated faster, and could allow a greater quantity of them to be fabricated in a single batch.

The algorithm proposed, due to the beam sizing process, is able to reduce the dimensions of the object while also retaining function, and the geometry can then be further improved by topology optimization in order to reduce total volume, resulting in both material and time savings. Figure 2 displays a comparison of the results of topology optimization with and without beam sizing for the L-bracket problem. The implementation of the optimization step is still under development, and the results displayed most likely are not representative of global optima.



(a) (b)

Figure 2. Comparison of preliminary results from topology optimization. Starting volume was 2400000 mm^2. (a) Optimization of the topology shown in Fig. 1, with a final volume of 1086890 mm^2. (b) Optimization of the initial rectangular topology, resulting in a volume of 2119300 mm^2.

5 Conclusions

From the tests done using the proof-of-concept implementation, the results look promising. While the simplifications used in the first implementation result in oversized topologies considering the loads they are subject to, it is already enough to both decrease final dimensions and optimization time. The new implementation, which does not feature such simplifications, is able to display much greater reduction, due to the quadratic relationship between bending stress and height of cross-section (as seen on Fig. 1 and 2). Some challenges still remain, such as correctly handling beam intersections and evaluating the need to purposefully oversize the beams in order to be able to reduce the final volume, but those matters are currently being investigated.

Acknowledgements. This work was done with support from the CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil [process number 131344/2020-0].

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] New perspectives on design automation: Celebrating the 40th anniversary of the ASME Design Automation Conference. *Journal of Mechanical Design*, vol. 137, n. 5. 050301, 2015.

[2] K. M. Ragsdell. Design and automation. *Journal of Mechanical Design*, vol. 102, n. 3, pp. 424–429, 1980.

[3] L. Lavagno, I. Markov, G. Martin, and L. Scheffer. *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology*. CRC Press, 2016.

[4] S. Oh, Y. Jung, I. Lee, and N. Kang. Design automation by integrating generative adversarial networks and topology optimization. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V02AT03A008, 2018.

[5] D. Shu, J. Cunningham, G. Stump, S. Miller, M. Yukish, T. Simpson, and C. Tucker. 3D design using generative adversarial networks and physics-based validation. *Journal of Mechanical Design*, vol. 142, pp. 1–51, 2019.

[6] C. Sharpe, T. Wiest, P. Wang, and C. Seepersad. A comparative evaluation of supervised machine learning classification techniques for engineering design applications. *Journal of Mechanical Design*, vol. 141, pp. 1–38, 2019.

[7] B. Camburn, Y. He, S. Raviselvam, J. Luo, and K. Wood. Machine learning based design concept evaluation. *Journal of Mechanical Design*, vol. 142, pp. 1–48, 2019.

[8] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang. Deep generative design: Integration of topology optimization and generative models. *Journal of Mechanical Design*, vol. 141, pp. 1, 2019.

[9] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, n. 2, pp. 100–107, 1968.

[10] K. Yu. Finding a natural-looking path by using generalized visibility graphs. In *PRICAI 2006: Trends in Artificial Intelligence*, volume 4099, pp. 170–179, 2006.

[11] W. Hunter and others. Topy - topology optimization with python. `https://github.com/williamhunter/topy`, 2017.

[12] W. Hunter. Predominantly solid-void three-dimensional topology optimisation using open source software. Master's thesis, University of Stellenbosch. Faculty of Engineering. Dept. of Mechanical and Mechatronic Engineering, 2009.

[13] X.-Z. Wen, Y.-Q. Long, and F.-L. He. Generalized conforming triangular membrane elements with rotational degree of freedom. *Finite Elements in Analysis and Design*, vol. 19, pp. 11–15, 2002.

[14] M. P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[15] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli. Stress-based topology optimization for continua. *Struct Multidisc Optim*, vol. 41, pp. 605–620, 2010.

[16] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, vol. 12, pp. 555–573, 2002.

[17] G. A. Teitelbaum. *PROPOSED BUILD GUIDELINES FOR USE IN FUSED DEPOSITION MODELING TO REDUCE BUILD TIME AND MATERIAL VOLUME*. PhD thesis, Department of Mechanical Engineering. University of Maryland (College Park, Md.), 2009.

[18] Y. Zhang, A. Bernard, J. M. Valenzuela, and K. Karunakaran. Fast adaptive modeling method for build time estimation in additive manufacturing. *CIRP Journal of Manufacturing Science and Technology*, vol. 10, pp. 49–60, 2015.