inventions

MDPI

Article

Adaptive Variable Design Algorithm for Improving Topology Optimization in Additive Manufacturing Guided Design

Abraham Vadillo Morillas [1,*], Jesús Meneses Alonso [1,2], Alejandro Bustos Caballero [3], Cristina Castejón Sisamón [1,2] and Alessandro Ceruti [4]

[1] MAQLAB Research Group, Mechanical Engineering Department, University Carlos III de Madrid, 28911 Leganés, Spain; meneses@ing.uc3m.es (J.M.A.); castejon@ing.uc3m.es (C.C.S.)
[2] Pedro Juan de Lastanosa Research Institute, 28911 Leganés, Spain
[3] MAQLAB Research Group, Depaertment of Mechanics, Universidad Nacional de Educación a Distancia, 28040 Madrid, Spain; albustos@ind.uned.es
[4] Industrial Engineering Department, University of Bologna, 40136 Bologna, Italy; alessandro.ceruti@unibo.it
* Correspondence: abvadill@ing.uc3m.es

Abstract: CAD-CAE software companies have introduced numerous tools aimed at facilitating topology optimization through Finite Element Simulation, thereby enhancing accessibility for designers via user-friendly interfaces. However, the imposition of intricate constraint conditions or additional restrictions during calculations may introduce instability into the resultant outcomes. In this paper, an algorithm for updating the design variables called Adaptive Variable Design is proposed to keep the final design space volume of the optimized part consistently under the target value while giving the main algorithm multiple chances to update the optimization parameters and search for a valid design. This algorithm aims to produce results that are more conducive to manufacturability and potentially more straightforward in interpretation. A comparison between several commercial software packages and the proposed algorithm, implemented in MATLAB R2023a, is carried out to prove the robustness of the latter. By simulating identical parts under similar conditions, we seek to generate comparable results and underscore the advantages stemming from the adoption and comprehension of the proposed topology optimization methodology. Our findings reveal that the integrated enhancements within MATLAB pertaining to the topology optimization process yield favourable outcomes with respect to discretization and the manufacturability of the resultant geometries. Furthermore, we assert that the methodology evaluated within MATLAB holds promise for potential integration into commercial packages, thereby enhancing the efficiency of topology optimization processes.

Keywords: topology optimization; optimization software; performance optimization; manufacturability; discreteness

Citation: Vadillo Morillas, A.; Meneses Alonso, J.; Bustos Caballero, A.; Castejón Sisamón, C.; Ceruti, A. Adaptive Variable Design Algorithm for Improving Topology Optimization in Additive Manufacturing Guided Design. *Inventions* **2024**, *9*, 70. https://doi.org/10.3390/inventions9040070

Academic Editor: Joshua M. Pearce

Received: 30 May 2024
Revised: 26 June 2024
Accepted: 27 June 2024
Published: 1 July 2024

1. Introduction

Topology optimization (TO) started as a way to address structural challenges through mathematical formulations, pioneered by Mitchell, Schmit, and Dorn et al. [1–3]. However, its widespread adoption did not occur until the emergence of the Finite Element Method (FEM) and its application to engineering structures [4–7] (together with other later relevant advances in the application of FEM to structural optimization [8–10]), alongside the integration of optimization systems combining FEM and topology optimization, such as in [11,12]. Subsequently, with the computerization of the methodology pioneered by Bendsøe and Kikuchi [13], topology optimization achieved maximum efficiency. Recent years have witnessed further advancements in topology optimization applications, notably with contributions from Sigmund [14].

These advancements have enabled companies specialized in structural simulation, commercial software, and CAD-CAE packages to integrate topology optimization into their products. Consequently, designers can now utilize this methodology without needing

extensive knowledge of the underlying mathematics and physics involved in the calculation process. However, despite the accessibility afforded by commercial software, certain limitations persist, particularly in terms of result interpretation and adherence to strict manufacturing constraints.

Recent technological advancements in the Digital Era, such as Big Data, simulation, and, notably for this study, additive manufacturing [15], have sparked significant research interest. Three-dimensional printing allows for the manufacturing of more complex shapes (such as TO-resultant shapes), and the versatility of different TO methodologies enables the adaptation of the codes to the manufacturing method [16–23]. Research on additive manufacturing filters and adaptations to the design optimization are currently important topics. This study proposes an easy plug-and-play method for obtaining 3D printing parts through design optimization.

Several researchers have recently developed tools aimed at achieving better results, as evidenced by the works of Guest et al. [24], Andreassen et al. [25], Liu and Tovar [26], Schevenels and Sigmund [27], Pellens et al. [28], Langelaar [29,30], Fu et al. [31], and Lee et al. [32]. These tools have been tested and in some instances applied to real-world components. More recent studies and reviews evidence a trend to use and develop this method for future applications in the world of part design [22,23,33–38].

This paper conducts a comparative analysis of the structural optimization of a real part using four commercial software codes (PTC Creo Parametric 6.0.1.0, Altair HyperMesh 2023.1, Solidworks 2023, and Ansys Workbench 2023R1) and a self-implemented MATLAB code. The aim is to demonstrate the benefits of understanding and applying the core principles of the topology optimization process instead of using popular solvers, like Tosca, OptiStruct, Ansys, and PTC. While the existing literature includes other comparative studies (Choi et al. [38], Dalpadulo et al. [39], Tyflopoulos et al. [40], and Struz et al. [41]), these primarily focus on identifying the best software for practical use and aiding designers in selecting suitable tools for their projects, without suggesting further improvements to standard commercial codes or proposing a general-purpose method.

In this study, multiple implementations in MATLAB are undertaken to enhance partial optimization results, highlighting the development of an algorithm for updating the optimization parameters and, as a consequence, the design variables, while avoiding the use of robust formulation. We named it Adaptive Variable Design (AVD). Furthermore, to validate the developments presented in this work, comparisons are made between the results obtained from the MATLAB implementations and those provided by commercial packages.

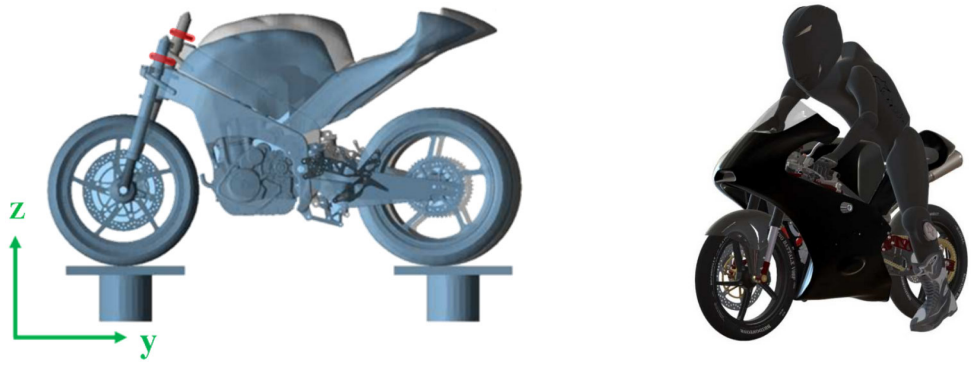
The comparison is facilitated through a case study which involves designing a racing motorcycle triple tree for fused deposition modelling (FDM). Identical loads, restrictions, and constraints are employed in all the commercial packages and MATLAB implementations. To prevent non-printable features in the result, Heaviside filters are used to derive a minimum length scale, adapted to this specific case. Additionally, a new algorithm is introduced and explained that addresses convergence without requiring a robust formulation.

The example of the racing motorcycle serves as a great illustration of the importance of research. In motorsports, a lightweight design is crucial for enhancing the speed of prototypes through the global vehicle weight. However, cost is another significant factor to consider. The methodology presented reduces the simulation time in comparison to commercial packages in terms of TO that is adapted to 3D printing, yielding similar results and eliminating the need for support material during manufacturing. This approach can lead to substantial cost reductions in simulation time, support material acquisition, and printing time. As a result, manufacturers can achieve an optimal design more efficiently and cost-effectively through design optimization geared towards 3D printing.

2. Case Study

The case study in this investigation involves the design of a racing motorcycle triple tree, specifically tailored for fused deposition modelling (FDM) manufacturing. The triple tree (highlighted in red in Figure 1a) is a critical structural component that links the

motorbike's front structure to the frame, facilitating steering. It undergoes significant loads during braking due to mass transfer and inertial forces, which are transferred to the front wheel.



(**a**) (**b**)

Figure 1. (**a**) MBD analysis result of the braking test with the triple tree highlighted in red and (**b**) rider in braking position utilized during the MBD analysis.

To identify the load conditions, a multi-body dynamics (MBD) analysis of a complete motorcycle model is conducted (Figure 1a). This analysis considers the most limiting negative acceleration during braking. A dummy, positioned as a braking rider (Figure 1b), is added to the MBD model, since the rider's weight significantly affects the model's behaviour. The resultant acceleration is analysed to determine the reaction forces in the triple tree. A small lateral load is also introduced to aid consistent shape generation. For static analysis in the topology optimization process, the triple tree is clamped in the direction of the axle hole, restricting all degrees of freedom, with forces applied at the front fork holes (see Figure 2).

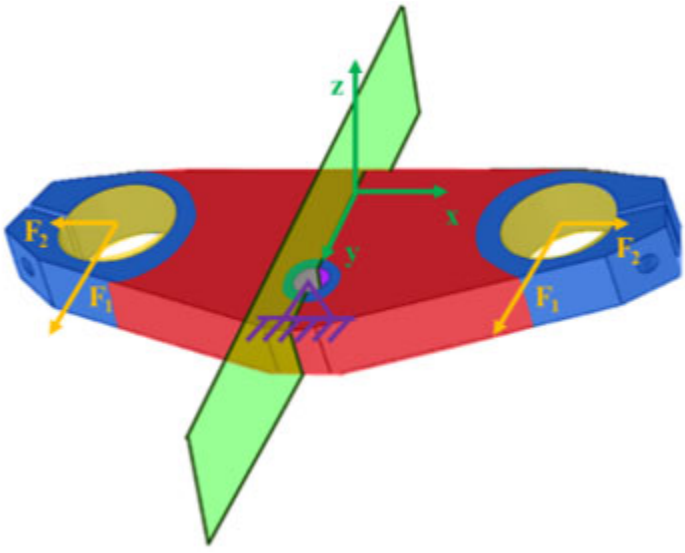


Figure 2. Motorcycle triple tree. In red, the design space is represented, while the non-design space is in blue. The symmetry plane YZ is coloured green. The applied force zones and directions are indicated in yellow, and the restricted areas are showcased in purple.

The primary objective of the optimization is to minimize the compliance of the part while adhering to a constraint that limits the final volume of the design space to below 30% of its original value. The design space is the volume where the design variables can be changed during the optimization process, i.e., the regions where the solver will change the

shape. The design domain is represented in red in Figure 2, while the non-design space is coloured in blue. Furthermore, manufacturing constraints are incorporated into all models, including the following:

- Minimum solid phase length of 2.4 mm: This constraint ensures that features that are smaller than six times the size of the extruded material are avoided, given that the typical nozzle pick diameter of an FDM printer is 0.4 mm.
- Symmetry in the YZ plane: While forces are symmetrical with respect to this plane, ensuring symmetric results is essential for consistent behaviour and stiffness across both sides of the front fork. This plane is represented in green in Figure 2.
- Extrusion in the Z-direction: To maintain comparability across different software packages, an extrusion constraint is applied as a 90° overhang constraint. Although an overhang angle is a common constraint in topology optimization for additive manufacturing, its implementation varies among software packages. In this case, the geometry of the part makes it suitable for this constraint due to its small thickness relative to other dimensions.

The mentioned loads F_1 and F_2 are 1500 N and 250 N, respectively. These forces and the clamped direction axle are represented in yellow and purple in Figure 2. The considered material is aluminium 7075T6, and its relevant mechanical properties are the density $\rho = 2810\ \mathrm{Kg/m^3}$, the Young modulus $E_0 = 72 \times 10^9$ Pa, and the Poisson ratio $v = 0.33$. The part limits are settled in a prismatic space of 288 × 102 × 25 mm. These values are used with all the cases in the commercial software packages and MATLAB implementations.

Results are evaluated based on manufacturability and assessed using software to verify compliance with manufacturing constraints. Additionally, the interpretability of the results is analysed, with a focus on the discreteness of the resultant density field. A more discrete density field (comprising only 0 and 1 densities) indicates a lower interpretability of the result.

3. MATLAB Implementation

The MATLAB topology optimization process is applied using well-known working codes [25,26,42,43] as the core of the implementations. As mentioned before, the objective of all the cases is to minimize the compliance (i.e., minimize the global strain energy) of the design space while keeping its final volume under 30% of the original one. This can be expressed as in Equation (1):

$$\begin{aligned} \mathrm{minc}(\rho) &= \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} \mathrm{E_e}(\rho_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e \\ \text{subjected to :} &\begin{cases} \mathrm{V}(\rho)/\mathrm{V}_0 \le \mathrm{f} \\ \mathbf{K}\mathbf{U} = \mathbf{F} \\ 0 \le \rho_i \le 1 \end{cases} \end{aligned} \tag{1}$$

where **K**, **U,** and **F** are the global stiffness matrix and the global displacement and forces vectors, respectively; N is the total number of elements, $\mathbf{u}_e$ is the e^{th} element displacement vector, $\mathbf{k}_0$ is the element stiffness matrix, ρ_e is the e^{th} element density, V_0 is the initial design volume, and $V(\rho)$ is the design volume, which can be described as presented in Equation (2); and f is the prescribed upper limit of the volume fraction constraint, set to 0.3 in all the cases.

$$\mathrm{V}(\boldsymbol{\rho}) = \frac{\sum_{e=1}^{n} \rho_e}{N}, n \in (1, N) \tag{2}$$

The used optimization algorithm is the Optimality Criteria (OC) [44,45]. This algorithm is used for updating the design variables according to the objective and restrictions. Additionally, the linear density filter [46,47] with the conic weights weighting technique is used for avoiding the formation of checkerboard structures [45]. The optimization algorithm and density filter formulation can be found in the scientific literature [25,26,42,43].

3.1. Geometry and Design Space Implementation

To improve performance in terms of calculation time and result manufacturability compared to commercial software, a 2D analysis is implemented, followed by an extrusion to the final thickness of the part. Symmetry conditions, considering only half the part, are applied to meet the described conditions.

For defining the initial mesh, the limits of the part in millimetres are translated into elements, so the initial mesh is 144 × 102 elements long. The design domain consists of three different spaces: the design space, represented in red in Figure 3; the non-design space, coloured in blue in Figure 3; and the void space, showcased in green in Figure 3. Active and passive elements [25,26] are used for defining the behaviour of each space, in a way that ensures the following:

- Void space: The density x_e of the elements is always 0. It defines the shape of the initial design of the triple tree and does not influence the structural calculations. It is formed by passive elements.
- Non-design space: This space encompasses volumes that must remain the same at the end of the optimization process. The density x_e of the elements inside these volumes is always 1, and they influence the structural calculations. It is constituted by active elements.
- Design space: The densities of the elements x_e vary from 0 to 1, with intermediate values adhering to a density-based approach to topology optimization. The volume constraint defined in Equation (2) is only applied to this space, as it is the only space that can vary its design variables in every iteration.

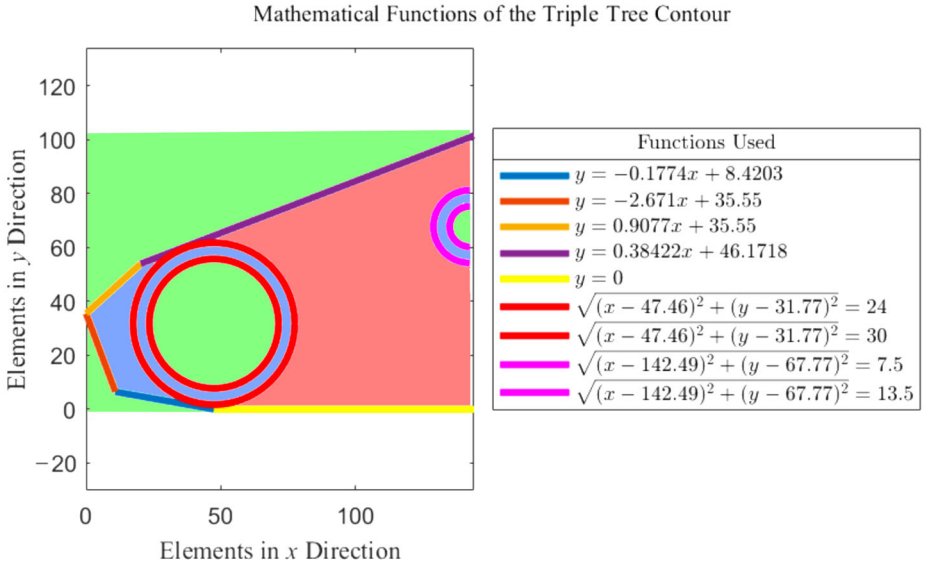


Figure 3. Mathematical functions of the triple tree contour. The functions are written in the legend "Functions Used". In shaded green, the void space is represented and in shaded red, the design space; and in shaded blue, the non-design space is represented.

3.2. Heaviside Step Function

Sigmund [48] introduced the morphology-based filters based on automatic inspection and restoration of image data [49]: the erode, dilate, close, open, close–open, and open–close filters. These filters strive to maintain the beneficial features of density filters while preventing the grey transition from the void phase to the solid phase. In doing so, intermediate densities are reduced to yield a more binary result. This binary result manifests as a less interpretable shape because the displayed shape does not obscure elements with intermediate densities that significantly affect the structural calculations.

The use of the close–open filter is especially interesting in this case, as a minimum length of the solid phase can be obtained. This minimum length is defined by the radius of the density filter, which is set to 2.4, as stated in epigraph 0. If "∧" represents the dilation operator and "∨" the erosion operator, the close–open filter can be expressed as in Equation (3) [48,50]:

$$\hat{\hat{\rho}}_e = \hat{\hat{\rho}}_e\left(\check{\hat{\rho}}_{i\epsilon N_e}\left(\check{\rho}_{j\epsilon N_i}\left(\hat{\rho}_{k\epsilon N_j}\left(\rho_{l\epsilon N_k}\right)\right)\right)\right) \tag{3}$$

In this case, the filter is applied to the design space, and e, i, j, k, and l are the elements inside the design space, being e ≡ i ≡ j ≡ k ≡ l. The used morphology operator is the smooth Heaviside function proposed by Wang et al. [51] and shown in Equation (4). When $\eta = 1$, the operator behaves as a min operator (erode) if β approaches ∞; and for $\eta = 0$, the operator behaves as a max operator (dilate) if β approaches ∞. For β approaching zero, the operator behaves as the linear density filter with conic weights.

$$\overline{\rho}_e = \frac{\tanh(\beta\eta) + \tanh\left(\beta\left(\widetilde{\rho}_e - \eta\right)\right)}{\tanh(\beta\eta) + \tanh(\beta(1-\eta))} \tag{4}$$

where $\widetilde{\rho}_e$ is the density field after the density filter, and $\overline{\rho}_e$ is the density field after the Heaviside filter. This expression can be derived for the calculation of sensitivities. This expression is shown in Equation (5):

$$\frac{\partial\overline{\rho}_e}{\partial\widetilde{\rho}_e} = \beta\frac{\operatorname{sech}^2\left(\beta\left(\widetilde{\rho}_e - \eta\right)\right)}{\tanh\beta + \tanh(\beta(1-\eta))} \tag{5}$$

Combining Equations (3) and (5) through a chain rule, the sensitivities of the density field $\partial\hat{\hat{\rho}}_e/\partial\rho_e$ can be calculated as in Equation (6):

$$\frac{\partial\hat{\hat{\rho}}_e}{\partial\rho_e} = \frac{\partial\hat{\hat{\rho}}_e}{\partial\check{\hat{\rho}}_e}\frac{\partial\check{\hat{\rho}}_e}{\partial\check{\rho}_e}\frac{\partial\check{\rho}_e}{\partial\hat{\rho}_e}\frac{\partial\hat{\rho}_e}{\partial\widetilde{\rho}_e}\frac{\partial\widetilde{\rho}_e}{\partial\rho_e} \tag{6}$$

The discreteness is analysed using the formula presented by Sigmund [48] in Equation (7), known as the measure of non-discreteness:

$$M_{nd} = \frac{\sum_{e=1}^{N} 4\hat{\hat{\rho}}_e\left(1 - \hat{\hat{\rho}}_e\right)}{N} \times 100\% \tag{7}$$

As the result becomes more discrete (binary, or 0–1), M_{nd} will approach 0; meanwhile, a density field formed solely by intermediate densities of 0.5 will yield a result of 100.

3.3. Parameter Update Algorithm

The addition of new constraints and filters to the optimization algorithm makes convergence more difficult due to the numerous conditions that the final design must meet and the limited design parameters that are available for modification in the end stages of the optimization. While the Heaviside step function with the morphology-based filter helps trend the design towards binary densities [50], the move limit at each step of the OC routine is often smaller than the density difference between a low-density and high-density element in the final optimization phase. This means that major changes are not made in the OC routine, and more intermediate densities are needed. Similar issues arise with other optimization algorithms, such as sequential quadratic programming or the method of moving asymptotes, which usually use smooth constraints to obtain convergence [51].

Some authors address the issue using robust formulation, which employs the worst-case scenario for each feature of the optimization problem (e.g., an eroded design for compliance or a dilated design for volume constraint) [52–56]. However, this often results

in a mathematically correct result, yet impractical outcomes for designs oriented towards the manufacturing of real parts. In this paper, the algorithm AVD is proposed to keep the final design space volume consistently under the target value while giving the main algorithm multiple chances to update the design variables and search for a valid design. This algorithm should result in not fully discrete designs, but with an M_{nd} between 0 and 5%, which will not affect the final interpretation of the optimization result.

First, the parameter β only increases by 1.2 every 25 iterations if the M_{nd} is larger than the prescribed M_{nd}^{ref}, which is initially set to 1 (although it can be changed within the algorithm). These changes are explained later in this epigraph. The penalization p follows a similar continuation scheme, starting at 2 and increasing by 1.01 up to a maximum value of 6.

The rest of the algorithm, where the filter parameters are modified to find an appropriate convergence, works as in the flowchart presented in Figure 4.

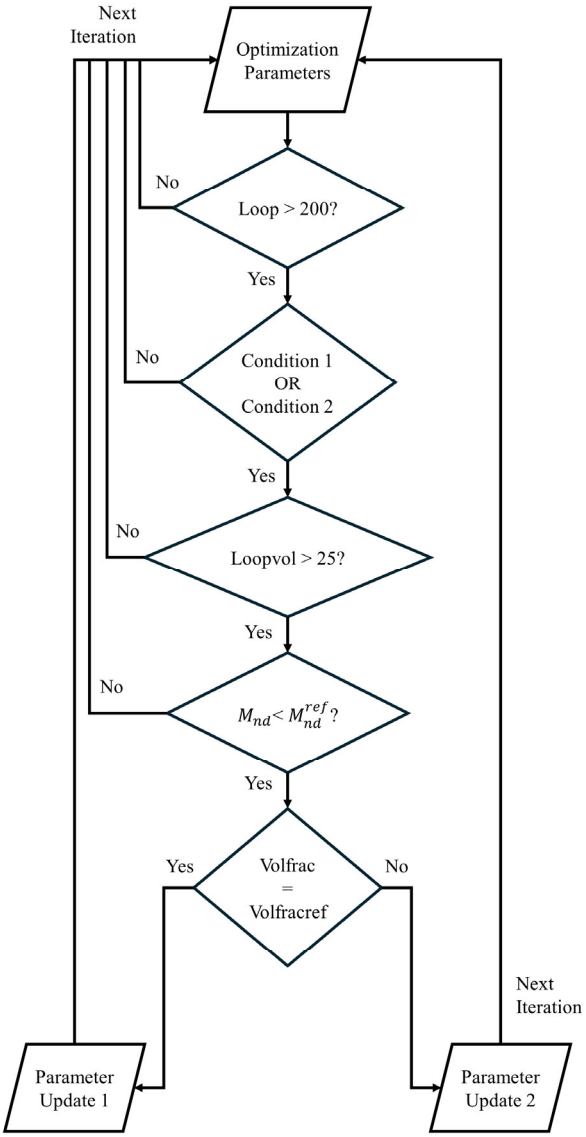


Figure 4. Flowchart of the design variable update algorithm.

In this process, the following apply:

- The algorithm begins operation in the 201st iteration, allowing the algorithm to create an initial design.

- Condition 1 compares the maximum density change of the last 50 iterations. If the median value is near the maximum permitted change, the condition is satisfied. This can be expressed as in Equation (8):

$$\overline{\mathrm{ch}(\mathrm{loop}-49:\mathrm{loop})} \geq \mathrm{move}-0.05 \tag{8}$$

  where ch is the maximum density change in every iteration, loop is the iteration number, and move is the target value for the maximum permitted change in every iteration.
- Condition 2 compares the same change value of the last 50 with the last 100 iterations. If the median change in both scenarios is similar, the condition is satisfied. This can be expressed as in Equation (9):

$$1.05 \geq \overline{\mathrm{ch}(\mathrm{loop}-49:\mathrm{loop})}/\overline{\mathrm{ch}(\mathrm{loop}-99:\mathrm{loop})} \geq 0.95 \tag{9}$$

- Loopvol is a counter that prevents consecutive parameter updates. It is restarted when the parameters are updated and must surpass 25 for the next parameter update to occur.
- M_{nd} must stay under 1 to maintain a relatively high discreteness.
- Volfrac is the actual iteration volume fraction following Equation (2).
- Volfracref is the prescribed volfrac upper limit described in Equation (1).
- In Parameter Update 1, the parameter β is set to 1, and a new parameter "Volfracobj" is introduced. Volfracobj is the value that Volfrac takes as a reference for the OC routine when Parameter Update 1 is activated. Initially matching Voldracref, it reduces Volfracobj whenever Parameter Update 1 is active until reaching a minimum of Voldracref-0.05. Additionally, all the loop counter variables are restarted, and the M_{nd}^{ref} is also restarted to a value of 1.
- In Parameter Update 2, Volfrac is expanded by 0.005. Note that this update occurs when Volfrac < Volfracref, so Volfrac increases until matching Volfracref, thereby triggering Parameter Update 1 instead of Parameter Update 2. Additionally, the M_{nd}^{ref} is expanded by 0.5 and the parameter β is halved.

4. Software Package Configuration

The setup of each software package for addressing the optimization and geometry constraints explained in Epigraph 0 is summarized below in Table 1 for the sake of clarity and reproducibility. As the density-based topology optimization is a mesh-dependant method, information about the mesh and the simulation times are also provided.

Table 1. Software packages' configuration.

Software	Mesh Information	Objective	Constraints	Geom. Restrictions	Sim.Time (hh:mm:ss)
HyperMesh 2023.1	491,809 CTETRA elements. Mesh size = 2.5.	Minimize compliance.	Volumefrac upper limit = 0.3.	Mindim = 2.4. Pattern grouping = 1 pln sym. No twist extrusion.	00:29:29
Parametric 6.0.1.0	433,488 Tetrahedrons. Mesh size = 1.98 mm.	Minimize strain energy.	Mass fraction upper limit 0.3.	Extrude along Z axis. Mirror about YZ plane. Minimum member size = 2.4.	00:25:45
SolidWorks 2023	Default mesh 2 mm size. 349,368 elements	Best stiffness-to-weight ratio.	70% mass reduction.	Thickness control—min = 4 mm. Symmetry control. De-mold direction—Stamping (pull direction only).	00:59:12

Table 1. *Cont.*

Software	Mesh Information	Objective	Constraints	Geom. Restrictions	Sim.Time (hh:mm:ss)
Workbench 2023 R1	467,009 Tetrahedrons. Mesh size = 2.2 mm.	Minimize compliance.	Volume constraint—30% retain.	Member size—Min Size = 2.4 mm. Symmetry design constraint. Extrusion.	00:52:57

The thickness control in SolidWorks is adjusted to 4 mm instead of 2.4 mm due to a software restriction: the minimum thickness must be twice the mesh size. For a 2.4 mm minimum thickness, the resulting mesh would be too large to compute.

All programs are configured to perform density-based topology optimization with SIMP as a weighting scheme, along with default parameters for the solver and the SIMP itself. Further information can be found in the developers' documentation [57–61].

5. Results

The resultant density fields are compared by representing three different density thresholds to evaluate the discreteness of the different results. Figure 5 illustrates the resulting density fields, where elements with a density $\rho_e > 0.1$ are represented in blue, elements with a density $\rho_e > 0.5$ are depicted in green, and in red, elements with a density $\rho_e > 0.9$ are represented.

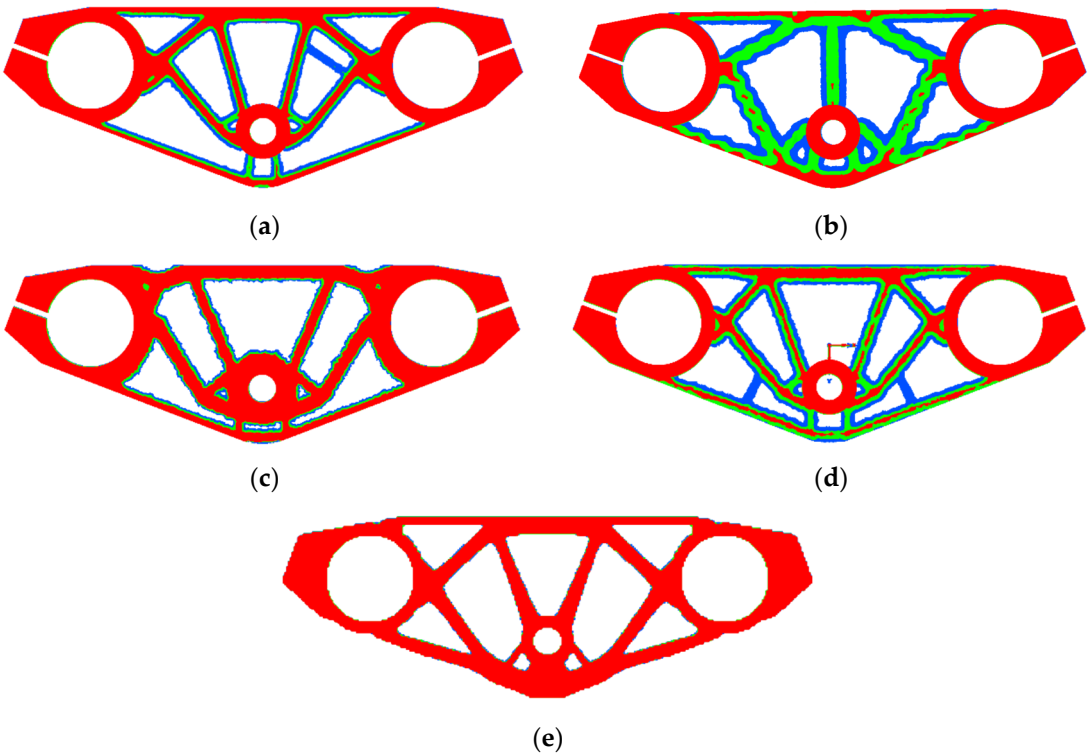


Figure 5. Results of the topology optimization process with the software packages and the MATLAB implementations. Coloured in blue, elements with a density over 0.1 are represented; in green, elements with a density over 0.5; and in red, elements with a density over 0.9. (**a**) HyperMesh 2023.1, (**b**) CREO Parametric 6.0.1.0., (**c**) SolidWroks 2023, (**d**) ANSYS Workbench 2023 R1, and (**e**) MATLAB.

The numerical results and comments regarding the resulting shapes and density fields are summarized below in Table 2.

Table 2. Summary of results.

Software	Volume Fraction	Max. Displacement	Advantages	Disadvantages
HyperMesh 2023.1	31.956%	4.055×10^{-2} mm	Accurate with objective and constraints. Good discreteness.	Symmetry constraint violated in low densities. Slight volume fraction constraint violation.
CREO Parametric 6.0.1.0	36.475%	3.825×10^{-2} mm	Well-defined and simple shape.	General instabilities in results. Lack of high-density elements. Volume fraction constraint violation.
SolidWorks 2023	46.117%	3.529×10^{-2} mm	Best discreteness of software packages.	Significant volume fraction constraint violation.
ANSYS Workbench 2023 R1	35.929%	3.629×10^{-2} mm	Best calculation performance of software packages.	Low-density features. Volume fraction constraint violation. Lack of continuity in high-density range.
MATLAB	29.540%	4.674×10^{-2} mm	Quasi − binary result (M_{nd} = 1.386). Did not violate volume fraction constraint. Simulation time significantly lower: 00:03:02.	Harder geometry introduction to the algorithm. Improved mesh definition, which would increase calculation time.

The only interpretation that satisfies the volume fraction constraint is the MATLAB case. This emphasizes the importance of achieving a resultant density field that is as low as possible. The visual differences can be observed in Figure 5. The objective function is very similar in all cases, with the MATLAB case being the worst scenario for compliance. Once again, the lack of intermediate density elements may be the cause of these results, which could be more similar if the interpretations had resulted in the parts having the same weight.

Table 1 shows that HyperMesh and PTC Creo achieve convergence in less time than ANSYS Workbench and SolidWorks. ANSYS has the best weight-to-compliance ratio, while SolidWorks provides the most discreet density field among all the commercial packages.

It can be seen that there were issues with the methods used. HyperMesh violated the symmetry constraint at low densities, while SolidWorks's interpretation violated the volume fraction constraint by 16.117%. This result was surprising considering the good results in terms of discreteness. The causes of these issues may stem from the representation system that the software uses to describe density fields with different thresholds. Instead of allowing for the direct input of numerical values, the software uses a floating bar that requires the user to estimate the density thresholds.

PTC Creo showed instabilities in the density field shape (see Figure 5b), with a "Hard Convergence" diagnostic. This indicates that the solver was unable to achieve smooth or high-quality convergence and had to force the design to converge by relaxing the convergence requirements. In ANSYS Workbench, a dispersed density field was observed, with discontinuities in both high-density and low-density features.

In the MATLAB case with AVD, the best results were achieved with a highly interpretable design. However, there was room for improvement in terms of a more user-friendly interface and more efficient calculation. Table 1 provides a summary of the advantages, disadvantages, and key findings for each case.

The optimization outcomes are interpreted into CAD designs with a density threshold of 0.5 and subsequently examined with Slicer software for FDM, Ultimaker Cura 5.4.0. In Figure 6, the CAD interpretations carried out in SolidWorks 2023 and the manufacturing

analyses performed in Cura 5.4.0 are showcased, with red arrows pointing to the critical features where the minimum length constraint has been invoked.

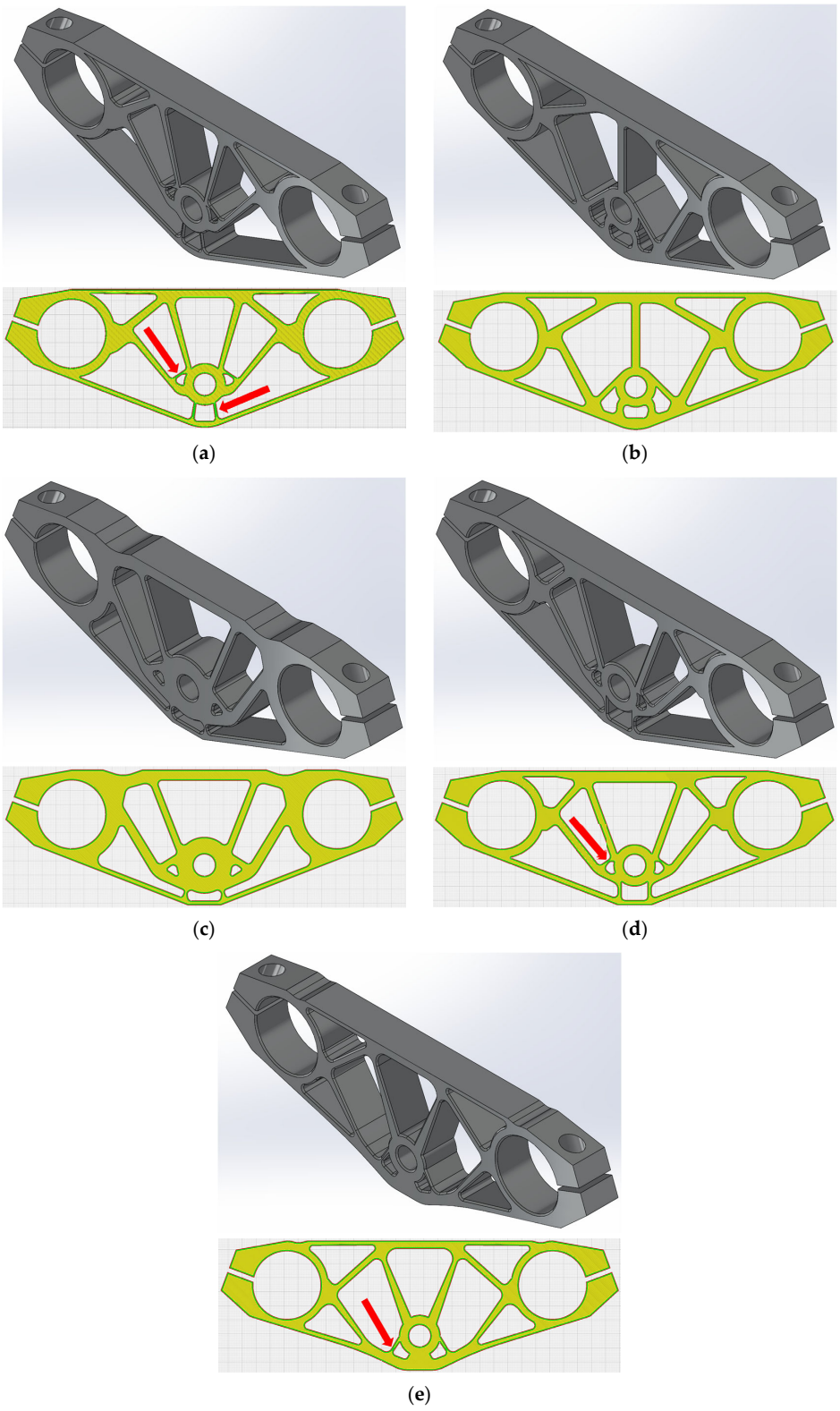
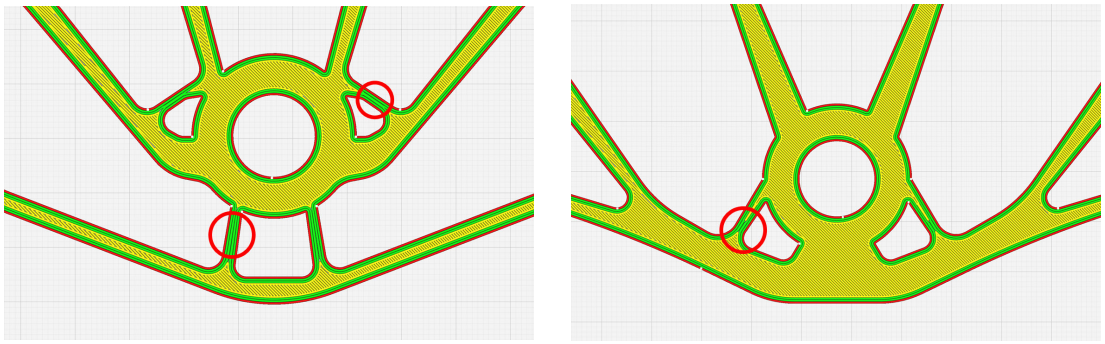


Figure 6. CAD interpretations and manufacturing analysis through Ultimaker Cura analysis of every optimization result. (**a**) HyperMesh 2023.1 CAD, (**b**) CREO Parametric 6.0.1.0, (**c**) SolidWorks 2023, (**d**) ANSYS Workbench 2023 R1, and (**e**) MATLAB. The critical features for checking the minimum length constraint are indicated using red arrows.

All the interpretations satisfy the minimum length constraint, with the slicer computing a minimum of six solid perimetral layers. The HyperMesh 2023.1 (Figure 6a) and MATLAB implementations (Figure 6e) are zoomed in on for better clarity on the minimum length constraints. These critical features are highlighted with red circles in Figure 7.



(**a**) (**b**)

Figure 7. Detail of the critical features for the minimum length constraint. (**a**) HyperMesh 2023.1 result. (**b**) MATLAB implementation result. The zones where the minimum length constraint is activated and only six solid perimetral layers are computed are highlighted with red circles.

6. Verification

In this subsection, the AVD is tested for different filter radii, resulting in varying minimum length scales and, consequently, different shapes. The AVD is also compared to other convergence strategies using Heaviside filters in density-based TO to assess its efficiency in the case of the racing motorcycle triple tree.

6.1. Filter Radii Check

For testing the AVD as a highly nonlinear and highly discrete TO convergence strategy, different filter radii are applied to the case study, ranging from small to large within the design domain scale. In Table 3, the used filter radii are listed, comparing them to the maximum and minimum dimension of the design space for use as a reference in readers' studies.

Table 3. Used filter radii for AVD verification and their percentages with respect to the maximum and minimum dimensions of the design domain.

Filter Radius	% Maximum Direction	% Minimum Direction
1.2 mm	0.417	1.176
1.6 mm	0.556	1.569
2.4 mm	0.833	2.353
3 mm	1.042	2.941
4 mm	1.389	3.922
6 mm	2.083	5.882
8 mm	2.778	7.843
12 mm	4.167	11.765

The data presented in Table 3 are visualized in Figure 8. The case study described in Section 2 was analysed using the specified filter radii. In Figure 8, the data are arranged from left to right and top to bottom, following the order of the information presented in Table 3. Figure 8 is presented in the same format as Figure 5, where elements coloured in red are elements with a density over 0.9; in green, elements with a density over 0.5 are showcased; and in blue, elements with a density over 0.1 can be observed.

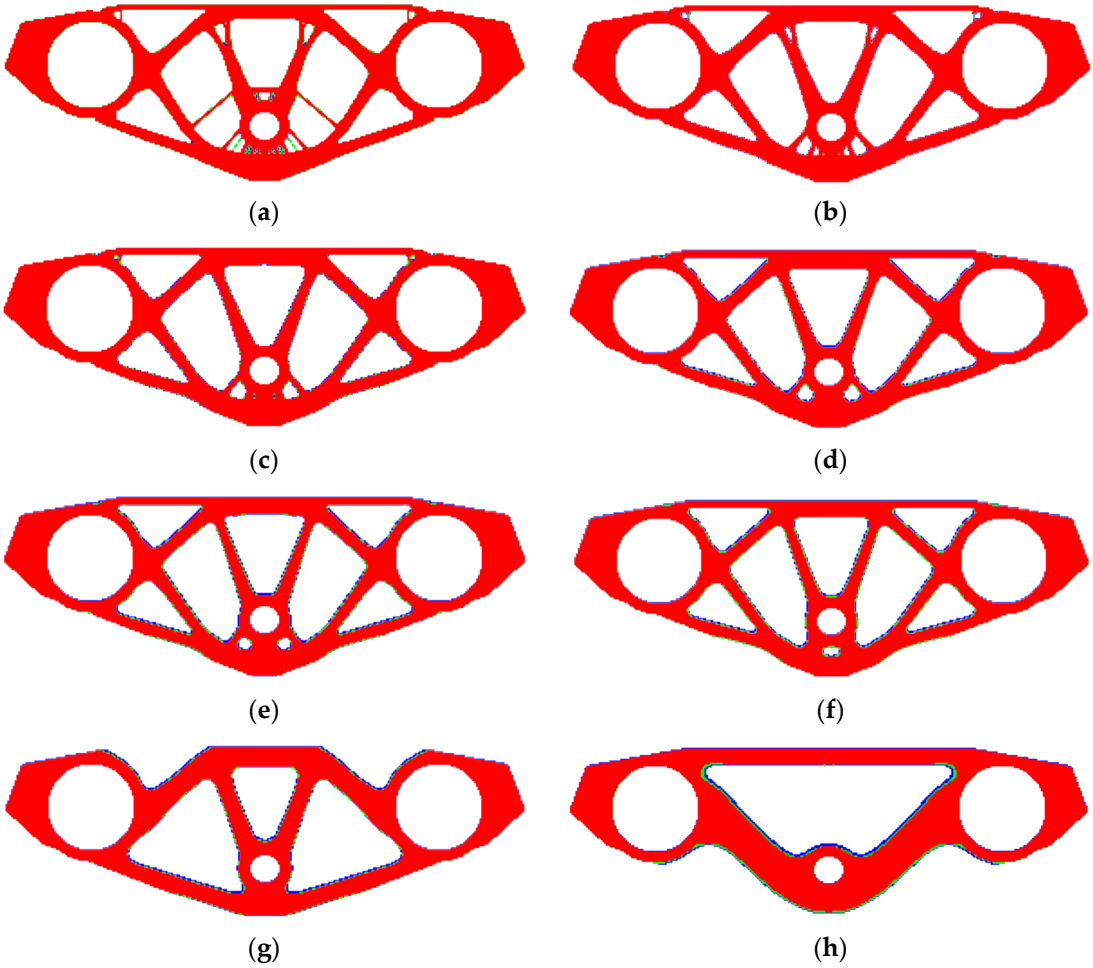


Figure 8. Results of the case study for different filter radii. Elements with a density over 0.1 are coloured in blue; in green, elements with a density over 0.5; and in red, elements with a density over 0.9. (**a**) R = 1.2 mm. (**b**) R = 1.6 mm. (**c**) R = 2.4 mm. (**d**) R = 3 mm. (**e**) R = 4 mm. (**f**) R = 4 mm. (**g**) R = 8 mm. (**h**) R = 12 mm.

To reinforce the conclusions of this verification of the AVD, Table 4 showcases the most relevant data from the results: the final volume fraction, a measure of non-discreteness, and the simulation time and iterations required for convergence.

Table 4. Different filter radii check for AVD.

Filter Radius	Volfrac	M_{nd}	Sim.Time (hh:mm:ss)	Iterations
1.2 mm	0.3	1.536	00:02:04	140
1.6 mm	0.3	1.336	00:02:22	180
2.4 mm	0.3	1.385	00:03:21	281
3 mm	0.3	2.377	00:06:40	680
4 mm	0.3	2.405	00:10:06	797
6 mm	0.3	2.769	00:10:43	883
8 mm	0.295	2.221	00:09:06	719
12 mm	0.3	2.940	00:14:28	831

From Figure 8a–c, it can be observed that the minimum length of Heaviside filter works properly, as features below the specified filter radii disappear. The same can be seen in Figure 8g,h, where different shapes with respect to the other solutions are achieved for fulfilling the minimum length condition. The AVD algorithm facilitates the modification of

the initial shape within movement limits and volume fraction evolution. It also adjusts β to smoothen the density field when needed.

Based on the data in Table 3, it is evident that the volume fraction consistently reaches its upper limit constraint. However, it converges at a slightly smaller value when R = 8 mm. The M_{nd} remains low in all cases, as seen from the minimal differences between the 0.9, 0.5, and 0.1 (red, green, and blue, respectively) designs in Figure 8. It shows a slight increase with larger filter radii, which aligns with the typical behaviour of the density filter.

When it comes to simulation time and iterations, there are two different sets of results: one for low filter radii (up to 2.4 mm) and another for values above that range. For the first set, quick convergence is achieved using the continuation scheme without the need to update parameter 1 of the AVD. The other set requires more iterations and involves updating parameter 1, as well as carrying out several parameter 2 updates, to converge. However, the simulation times are still significantly lower than the results obtained from commercial software packages, as shown in Table 1.

6.2. Comparison to Similar Convergence Strategies

In this section, we compare four different convergence strategies for Heaviside step filters from the literature to the proposed AVD strategy. These strategies are utilized when solving highly nonlinear systems. They involve starting with smoother parameters in the initial iterations and gradually increasing their impact during the optimization process. A robust convergence strategy can help avoid undesired local minima solutions and lead to more optimal designs.

The strategies discussed in this excerpt are used by authors in density-based TO using OC or the Method of Moving Asymptotes (MMA) [62] as optimization algorithms. Strategies ranging from simple continuation schemes to algorithms that are similar to AVD are now discussed and compared.

In their study, Qie et al. [63] propose a continuation scheme based on a previous work by Li and Khandelwal [64], where the following strategies are proposed:

- Penalization increments by 0.1 every 50 iterations or when the convergence criteria are achieved, from a starting value of 1 to a maximum value of 3. This is used for hat-shaped filters.
- Increments of β from a starting value of 0.5 to a maximum value of 200, doubling its value every 50 iterations or when the convergence criteria are achieved. Penalizations are fixed at a value of 3. This is used for Heaviside filters.
- Fix β with a value of 50 and following the first continuation scheme described in this list for penalization. This is also used for Heaviside filters, and it was presented by Guest and Asadpoure in [65].
- Here, the penalization increases as in the first listed continuation scheme, and when it reaches the maximum value of penalization, a β continuation scheme is proposed: it is increased by 1.5 factor every 50 iterations or when the convergence criteria are achieved; and when M_{nd} falls below 0.5, β is increased by 5.

In all of these algorithms, final convergence occurs when the updated parameters reach their maximum prescribed values and the convergence criteria are met.

In his study [48], Sigmund proposed a simple continuation scheme for β from a value of 0.2 to a maximum value of 200, doubling its value every 50 iterations or when the convergence criteria are achieved.

Langelaar proposed a different β continuation scheme in his study [29], where it has an initial value of 0.5 and is doubled every 125 iterations or when the convergence criteria are achieved. This is an interesting example, as Langelaar uses it for his additive manufacturing filter.

Gaynor and Guest proposed an interesting strategy for additive manufacturing constraints, where penalization and beta increase when convergence occurs. Penalization starts at a value of 10 and increases to a maximum of 18 in increments of 2; β starts at a value of 5

and increases in increments of 5 to a maximum of 25. Final convergence is achieved when both parameters reach their maximum values and the convergence criteria are met.

The strategies were tested using the same case study as the AVD, with the parameters described in Section 2. A comparison of the strategies with the AVD was not possible because the strategies did not reach final convergence. Their processes were stopped after four hours of calculation, with identical results: a very low M_{nd} value, the maximum parameter value incremented with the continuation schemes, and the maximum move limit of the OC routine being reached in every iteration. In all cases, the objective function became extremely large, demonstrating convergence difficulties.

7. Conclusions

This paper presents an algorithm for updating the design variables of a topology optimization process. The algorithm is conceived to keep the final design space volume consistently under the target value, while it is possible to update the design variables and search for a valid design.

Additionally, a comparison of four commercial software packages with self-implementations in a MATLAB topology optimization code was carried out through the example of a racing motorcycle triple tree. The aim was to illustrate the benefits of understanding the topology optimization process and having the ability to control the whole process, focusing on a design case. Based on this, the following conclusions are drawn:

Commercial software packages offer a straightforward method for conducting topology optimization, but they fall short of yielding a quality density field for post-processing. As the software's resulting discreteness is poor, the latter CAD interpretation leads to a violation of the initial optimization constraints and, in the long term, a non-optimal design. Getting a low M_{nd} value ensures that the interpretation is consistent with the optimization constraints.

The implementations presented in Section 3 are effective. The geometry introduction through active and passive elements works correctly, but it takes more time to introduce the geometry manually with formulas than introducing a CAD. Future implementations could introduce a macro for CAD part introduction. The Heaviside step function written as a close–open filter achieves the desired minimum length scale, as demonstrated in Section 5. The presented convergence algorithm has provided a result with low interpretability (this is a low M_{nd} value), leading to a CAD design interpretation in line with the optimization constraints.

The effectiveness of the strategy was demonstrated in Section 6, where different filter radii were used to evaluate the AVD efficiency. Promising results were obtained for various AVD behaviours. The strategy was compared to similar methodologies found in the literature. However, due to the high nonlinearity of the case study, involving the application of four Heaviside step filters along with symmetry conditions and enforced solid/void areas, these strategies did not lead to convergence. It was observed that these strategies were developed for simpler cases, and AVD proved to be more efficient in dealing with real part design processes, despite the challenges it faced in converging.

In broader terms, the authors believe that the implementations developed in this study result in a more design-friendly algorithm for the optimization of real parts and can easily be implemented in the existing topology optimization processes. This could be a good tool for designers who search for a good relation between the resultant density field of their studies and the design CAD interpretation.

8. Future Work

The findings of this study have implications for future guidelines. It is evident that the emergence of additive manufacturing technologies, along with other Industry 4.0 technologies, has made new features more accessible and created new needs in the modern industry. This is where TO adapted to additive manufacturing becomes especially important, helping designers achieve optimal results in terms of both the optimization

process itself and the manufacturing process. It is like a mixture of Simulation-Driven Design and Manufacturing-Driven Design.

In this study, we examined a case involving a racing motorcycle triple tree to demonstrate this situation. We found that using MATLAB for extrusion constraint, minimum length control through open–close filters, and AVD was efficient in producing printable parts without the need for support material, while ensuring a good interpretation of the resultant density field of the TO. However, despite MATLAB showing better performance than commercial software packages, there are still potential areas for improvement. These improvements can be made in two fields: the preprocessing and the AVD itself.

During preprocessing, the MATLAB code uses a structured mesh with one element per millimetre for clarity and computational efficiency. The design domain is created by enforcing solid/void phases, adding nonlinearity to the process. Possible improvements include introducing a macro for CAD treatment, using an unstructured mesh to adapt it to the prescribed CAD, and translating the code to a more efficient programming language such as C++, Python, or Julia. These languages can take advantage of the computer's capabilities, increasing the code's efficiency and allowing for a higher definition in the mesh.

Improvements can be achieved in the AVD. Prescribing parameters that update in a concrete number of iterations or when convergence is achieved can sometimes result in enlarging the number of iterations for converging. A good strategy can be to store the discreteness and objective function and move the limit results and adapt them to a differentiable mathematical function for optimization in each iteration [66–68]. This, along with speedups, can make the code work with more demanding problems. Another option is to introduce artificial intelligence to choose the AVD configuration after extensive training [69–71].

Author Contributions: Conceptualization, A.V.M. and A.C.; methodology, A.V.M., J.M.A., A.B.C. and A.C.; software, A.V.M.; validation A.V.M., J.M.A., A.B.C. and A.C.; investigation, A.V.M. and A.C.; resources, C.C.S.; writing—original draft preparation, A.V.M., J.M.A. and A.B.C.; writing—review and editing, A.V.M., J.M.A. and A.B.C.; supervision, J.M.A., A.B.C. and A.C.; project administration, C.C.S.; funding acquisition, C.C.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research work described in this paper is part of the R&D and Innovation projects MC4.0 PID2020-116984RB-C21 and MC4.0 PID2020-116984RB-C22, supported by the MCIN/AEI/10.13039/501100011033.

Data Availability Statement: The original data presented in the study are openly available in Dropbox at https://www.dropbox.com/scl/fi/q436i732ldwzmuekxse0i/TijasPaperSym.m?rlkey=fgra5ae64nwvs2b3pgn2i3plm&st=t6ec8rdp&dl=0 (accessed on 29 May 2024). Readers can also write to the corresponding author (Abraham Vadillo Morillas—abvadill@ing.uc3m.es) for solving any questions about the configuration of the code for obtaining different results.

Acknowledgments: The research work described in this paper is part of the R&D and Innovation projects MC4.0 PID2020-116984RB-C21 and MC4.0 PID2020-116984RB-C22, supported by the MCIN/AEI/10.13039/501100011033. The authors especially thank the Department of Industrial Engineering of the Alma Mater Studiorum Università di Bologna and their research group for allowing the collaboration and for the stay on their campus.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Michell, A.G.M. LVIII. The limits of economy of material in frame-structures. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1904**, *8*, 589–597. [CrossRef]
2. Schmit, L.A. Structural design by systematic synthesis. In Proceedings of the 2nd Conference on Electronic Computation, Pittsburgh, PA, USA, 8–9 September 1960; ASCE: New York, NY, USA, 1960; pp. 105–122. Available online: https://www.vrand.com/resources/publications/archive/structural-design-by-systematic-synthesis/ (accessed on 29 May 2024).
3. Dorn, W.S.; Gomory, R.E.; Greenberg, H.J. Automatic design of optimal structures. *J. Mécanique* **1964**, *3*, 25–52.

4. Argyris, J.H. Energy Theorems and Structural Analysis: A Generalized Discourse with Applications on Energy Principles of Structural Analysis Including the Effects of Temperature and Non-Linear Stress-Strain Relations. *Aircr. Eng. Aerosp. Technol.* **1954**, *26*, 347–356. [CrossRef]
5. Synge, J.L.; Rheinboldt, W.C. The hypercircle in mathematical physics. *Phys. Today* **1957**, *10*, 45–46. [CrossRef]
6. Clough, R.W. The finite element in plane stress análisis. In Proceedings of the 2nd ASCE Conference on Electric Computation, Pittsburgh, PA, USA, 8–9 September 1960. Available online: https://www.semanticscholar.org/paper/The-Finite-Element-Method-in-Plane-Stress-Analysis-Clough/035536cf1b0157b3cc7a6a19ed1b66638b388553 (accessed on 29 May 2024).
7. Argyris, J.H. Continua and discontinua, Matrix methods in structural mechanics, Opening address. In Proceedings of the Conference, Wright-Patterson Air Force Base, Dayton, OH, USA, 26–28 October 1965; pp. 26–28.
8. Logan, D.L. *A First Course in the Finite Element Method*; Thomson Canada Limited: Toronto, ON, Canada, 2007.
9. Cook, R.D. *Concepts and Applications of Finite Element Analysis*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2007.
10. Bathe, K.-J. *Finite Elements Procedures in Engineering Analysis*, 1st ed.; Prentice-Hall Inc.: Upper Saddle River, NJ, USA, 1996.
11. Park, G.-J. *Analytic Methods for Design Practice*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
12. Haftka, R.T.; Gürdal, Z. *Elements of Structural Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 11.
13. Bendsøe, M.P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.* **1988**, *71*, 197–224. [CrossRef]
14. Sigmund, O.; Maute, K. Topology optimization approaches. *Struct. Multidiscip. Optim.* **2013**, *48*, 1031–1055. [CrossRef]
15. Pfeiffer, S. The Vision of 'Industrie 4.0' in the Making—A Case of Future Told, Tamed, and Traded. *Nanoethics* **2017**, *11*, 107–121. [CrossRef]
16. Kruth, J.-P.; Van Den Broucke, B.; Van Vaerenbergh, J.; Van Vaerenbergh, J.; Mercelis, P. Benchmarking of Different SLS/SLM Processes as Rapid Manufacturing Techniques. PMI Paper 525. 2005. Available online: https://research.utwente.nl/files/5676701/Wa1021.pdf (accessed on 5 May 2024).
17. Kranz, J.; Herzog, D.; Emmelmann, C. Design guidelines for laser additive manufacturing of lightweight structures in TiAl6V4. *J. Laser Appl.* **2015**, *27*, S14001. [CrossRef]
18. Wang, D.; Yang, Y.; Yi, Z.; Su, X. Research on the fabricating quality optimization of the overhanging surface in SLM process. *Int. J. Adv. Manuf. Technol.* **2013**, *65*, 1471–1484. [CrossRef]
19. Thompson, M.K.; Moroni, G.; Vaneker, T.; Fadel, G.; Campbell, R.I.; Gibson, I.; Bernard, A.; Schulz, J.; Graf, P.; Ahuja, B.; et al. Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints. *CIRP Ann.* **2016**, *65*, 737–760. [CrossRef]
20. Boothroyd, G.; Dewhurst, P.; Knight, W.A. *Product Design for Manufacture and Assembly*; CRC Press: Boca Raton, FL, USA, 2010.
21. Dangal, B.; Jung, S. The Impact of Additive Manufacturing Constraints and Design Objectives on Structural Topology Optimization. *Appl. Sci.* **2023**, *13*, 10161. [CrossRef]
22. Zhu, J.; Zhou, H.; Wang, C.; Zhou, L.; Yuan, S.; Zhang, W. A review of topology optimization for additive manufacturing: Status and challenges. *Chin. J. Aeronaut.* **2021**, *34*, 91–110. [CrossRef]
23. El Khadiri, I.; Zemzami, M.; Nguyen, N.-Q.; Abouelmajd, M.; Hmina, N.; Belhouideg, S. Topology optimization methods for additive manufacturing: A review. *Int. J. Simul. Multidiscip. Des. Optim.* **2023**, *14*, 12. [CrossRef]
24. Guest, J.K.; Prévost, J.H.; Belytschko, T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int. J. Numer. Methods Eng.* **2004**, *61*, 238–254. [CrossRef]
25. Andreassen, E.; Clausen, A.; Schevenels, M.; Lazarov, B.S.; Sigmund, O. Efficient topology optimization in MATLAB using 88 lines of code. *Struct. Multidiscip. Optim.* **2011**, *43*, 1–16. [CrossRef]
26. Liu, K.; Tovar, A. An efficient 3D topology optimization code written in Matlab. *Struct. Multidiscip. Optim.* **2014**, *50*, 1175–1196. [CrossRef]
27. Schevenels, M.; Sigmund, O. On the implementation and effectiveness of morphological close-open and open-close filters for topology optimization. *Struct. Multidiscip. Optim.* **2016**, *54*, 15–21. [CrossRef]
28. Pellens, J.; Lombaert, G.; Lazarov, B.; Schevenels, M. Combined length scale and overhang angle control in minimum compliance topology optimization for additive manufacturing. *Struct. Multidiscip. Optim.* **2019**, *59*, 2005–2022. [CrossRef]
29. Langelaar, M. An additive manufacturing filter for topology optimization of print-ready designs. *Struct. Multidiscip. Optim.* **2017**, *55*, 871–883. [CrossRef]
30. Langelaar, M. Topology optimization for multi-axis machining. *Comput. Methods Appl. Mech. Eng.* **2019**, *351*, 226–252. [CrossRef]
31. Fu, Y.-F.; Ghabraie, K.; Rolfe, B.; Wang, Y.; Chiu, L.N.S. Smooth Design of 3D Self-Supporting Topologies Using Additive Manufacturing Filter and SEMDOT. *Appl. Sci.* **2020**, *11*, 238. [CrossRef]
32. Lee, H.Y.; Zhu, M.; Guest, J.K. Topology optimization considering multi-axis machining constraints using projection methods. *Comput. Methods Appl. Mech. Eng.* **2022**, *390*, 114464. [CrossRef]
33. Berrocal, L.; Fernández, R.; González, S.; Periñán, A.; Tudela, S.; Vilanova, J.; Rubio, L.; Martín Márquez, J.M.; Guerrero, J.; Lasagni, F. Topology optimization and additive manufacturing for aerospace components. *Prog. Addit. Manuf.* **2019**, *4*, 83–95. [CrossRef]
34. Fritz, C.; Fischer, L.; Wund, E.; Zaeh, M.F. Inner design of artificial test bones for biomechanical investigations using topology optimization. *Prog. Addit. Manuf.* **2023**, *8*, 427–435. [CrossRef]

35. Noordman, B.; Ton, Y.; van den Toorn, J.; de Smit, M.; Haagsma, R.; Koenis, T.; van den Brink, W. Topology optimization for the design of a 3D-printed rotating shaft balance. *Prog. Addit. Manuf.* **2023**, *8*, 19–25. [CrossRef]
36. Berce, P. Advances in Additive Manufacturing and Their Applications. *Metals* **2024**, *14*, 165. [CrossRef]
37. Prathyusha, A.L.R.; Babu, G.R. A review on additive manufacturing and topology optimization process for weight reduction studies in various industrial applications. *Mater. Today Proc.* **2022**, *62*, 109–117. [CrossRef]
38. Choi, W.; Kim, J.; Park, G.-J. Comparison study of some commercial structural optimization software systems. *Struct. Multidiscip. Optim.* **2016**, *54*, 685–699. [CrossRef]
39. Dalpadulo, E.; Pini, F.; Leali, F. Assessment of Computer-Aided Design Tools for Topology Optimization of Additively Manufactured Automotive Components. *Appl. Sci.* **2021**, *11*, 10980. [CrossRef]
40. Tyflopoulos, E.; Steinert, M. A Comparative Study of the Application of Different Commercial Software for Topology Optimization. *Appl. Sci.* **2022**, *12*, 611. [CrossRef]
41. Struz, J.; Hruzik, L.; Klapetek, L.; Trochta, M. Comparative analysis of different softwares in terms of parameters optimized by topological optimization. *MM Sci. J.* **2023**, *2023*. [CrossRef]
42. Sigmund, O. A 99 line topology optimization code written in Matlab. *Struct. Multidiscip. Optim.* **2001**, *21*, 120–127. [CrossRef]
43. Ferrari, F.; Sigmund, O. A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D. *Struct. Multidiscip. Optim.* **2020**, *62*, 2211–2228. [CrossRef]
44. Prager, W. Optimality criteria in structural design. *Proc. Natl. Acad. Sci. USA* **1968**, *61*, 794–796. [CrossRef] [PubMed]
45. Sigmund, O.; Petersson, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.* **1998**, *16*, 68–75. [CrossRef]
46. Bruns, T.E.; Tortorelli, D.A. Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput. Methods Appl. Mech. Eng.* **2001**, *190*, 3443–3459. [CrossRef]
47. Bourdin, B. Filters in topology optimization. *Int. J. Numer. Methods Eng.* **2001**, *50*, 2143–2158. [CrossRef]
48. Sigmund, O. Morphology-based black and white filters for topology optimization. *Struct. Multidiscip. Optim.* **2007**, *33*, 401–424. [CrossRef]
49. Pratt, W.K. *Digital_Image_Processing*, 1st ed.; Course Notes from the Learning Tree; Wiley-Interscience: Michigan, MI, USA, 1991.
50. Wang, F.; Lazarov, B.S.; Sigmund, O. On projection methods, convergence and robust formulations in topology optimization. *Struct. Multidiscip. Optim.* **2011**, *43*, 767–784. [CrossRef]
51. Driouch, A. Approximations Based on the Method of Moving Asymptotes. Ph.D. Thesis, Université de Pau et des Pays de l'Adour, Pau, France, Université Ibn Tofail, Kénitra, Morocco, 2020.
52. Fernández, E.; Yang, K.; Koppen, S.; Alarcón, P.; Bauduin, S.; Duysinx, P. Imposing minimum and maximum member size, minimum cavity size, and minimum separation distance between solid members in topology optimization. *Comput. Methods Appl. Mech. Eng.* **2020**, *368*, 113157. [CrossRef]
53. Trillet, D.; Duysinx, P.; Fernandez, E. Analytical relationships for imposing minimum length scale in the robust topology optimization formulation. *Struct. Multidiscip. Optim.* **2021**, *64*, 2429–2448. [CrossRef]
54. Wang, F.; Jensen, J.; Sigmund, O. Robust topology optimization of photonic crystal waveguides with tailored dispersion properties. *JOSA B* **2011**, *28*, 387–397. [CrossRef]
55. Jog, C.S. A robust dual algorithm for topology design of structures in discrete variables. *Int. J. Numer. Methods Eng.* **2001**, *50*, 1607–1618. [CrossRef]
56. Zhang, X.; Takezawa, A.; Kang, Z. Robust topology optimization of vibrating structures considering random diffuse regions via a phase-field method. *Comput. Methods Appl. Mech. Eng.* **2019**, *344*, 766–797. [CrossRef]
57. Abbey, T. PTC Creo Blogs. Available online: https://www.ptc.com/en/blogs/cad/what-is-topology-optimization (accessed on 16 May 2024).
58. Thompson, E. Design Exploration via Topology Optimization. Available online: https://www.ansys.com/content/dam/amp/2023/october/quick-request/design-exploration-via-topology-optimization-et.pdf (accessed on 16 May 2024).
59. Altair Engineering, Inc. Practical Aspects of Structural Optimization, A Study Guide, 1st ed.Altair Engineering, Inc.: Troy, Michigan, USA, 2021; Volume 1. Available online: https://www.advanced-eng.cz/wp-content/uploads/2021/06/ebook_Practical_Aspects_of_Optimization_with_Altair_OptiStruct_2021.pdf (accessed on 16 May 2024).
60. Altair Engineering, Inc. Altair OptiStruct Help Guide. Available online: https://2021.help.altair.com/2021/hwsolvers/os/topics/solvers/os/topology_opt_design_variables_r.htm (accessed on 16 May 2024).
61. Dassault Systèmes. SolidWorks Help Page. Available online: https://help.solidworks.com/2021/english/SolidWorks/cworks/c_simp_method_topology.htm (accessed on 16 May 2024).
62. Svanberg, K. The method of moving asymptotes—A new method for structural optimization. *Int. J. Numer. Methods Eng.* **1987**, *24*, 359–373. [CrossRef]
63. Qie, L.; Jing, S.; Lian, R. Heaviside-Based Morphological Filters for Topology Optimization. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *472*, 12034. [CrossRef]
64. Li, L.; Khandelwal, K. Volume preserving projection filters and continuation methods in topology optimization. *Eng. Struct.* **2015**, *85*, 144–161. [CrossRef]
65. Guest, J.K.; Asadpoure, A.; Ha, S.-H. Eliminating beta-continuation from heaviside projection and density filter algorithms. *Struct. Multidiscip. Optim.* **2011**, *44*, 443–453. [CrossRef]

66. Karush, W. Minima of Functions of Several Variables with Inequalities as Side Conditions. 2014. Available online: https://api.semanticscholar.org/CorpusID:117180997 (accessed on 14 May 2024).
67. Venkiteswaran, V.K.; Turkkan, O.; Su, H.-J. Speeding Up Topology Optimization of Compliant Mechanisms with a Pseudorigid-Body Model. *J. Mech. Robot.* **2017**, *9*, 041007. [CrossRef]
68. Li, W.; Suryanarayana, P.; Paulino, G.H. Accelerated fixed-point formulation of topology optimization: Application to compliance minimization problems. *Mech. Res. Commun.* **2020**, *103*, 103469. [CrossRef]
69. Kim, Y.Y.; Yoon, G. Multi-resolution multi-scale topology optimization—A new paradigm. *Int. J. Solids Struct.* **2000**, *37*, 5529–5559. [CrossRef]
70. Nie, Z.; Lin, T.; Jiang, H.; Kara, L.B. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *J. Mech. Des.* **2021**, *143*, 031715. [CrossRef]
71. Luo, J.; LI, Y.; Zhou, W.; Gong, Z.; Zhang, Z.; Yao, W. An Improved Data-Driven Topology Optimization Method Using Feature Pyramid Networks with Physical Constraints. *Comput. Model. Eng. Sci.* **2021**, *128*, 823–848. [CrossRef]