

TO3DPGS

The Future Of Glass

Pim Brueren

2023 | 2024

*'Development of an Algorithm for **T**opology **O**ptimized **3D P**rinted **G**lass **S**tructure'*

TU Delft

Department MSc. Architecture, Urbanism & Building Sciences
Track Building Technology Graduation Studio
Mentors Dr. Faidra Oikonomopoulou
Dr. Charalampos Andriotis
Student Pim Brueren - 4658752

Abstract

This thesis explores the application of topology optimization for large-scale glass structures in architecture, addressing the limitations of traditional casting methods and exploring the potential of 3D printed glass. Previous research emphasized the importance of manufacturing methods, but highlighted the lack of transparency in topology optimized cast glass. This study uses 3D printing to overcome these challenges, focusing on the unique properties and manufacturing techniques of 3D printed glass, including the limitations of its brittle nature and different tensile and compressive strengths.

An extensive literature review provides a foundation for glass properties, manufacturing techniques and the principles of topology optimization. The research advances the use of SIMP methodology and adapts it to 3D printed glass constraints such as overhang, path continuity and nozzle size. Therefore, it incorporates an adapted layer-to-layer overhang filter and addresses the island effect and path control using advanced computing techniques.

The implementation of these methodologies is described in detail, with specific adjustments to the overhang angles and connection strategies for island structures. Testing within a predefined design domain evaluates the limitations and capabilities of the proposed solutions, culminating in the selection and 3D printing of a feasible design.

The results validate the approach and highlight the need for further research, especially regarding the anisotropy of glass layers.

This research is concluded with a discussion of the broader academic implications and future research opportunities, supported by additional insights and findings presented in the appendix. This work demonstrates significant advances in structural glass architecture and paves the way for innovative applications of 3D printed glass.

Extended abstract

This thesis builds on ongoing research into the possible application of topology optimization for large-scale glass structures in architecture. These earlier works emphasized the importance of manufacturing methods and suggested that casting could be a viable option. However, as shown in this thesis, the casting process for topology-optimized shapes still has several limitations. Previous research emphasized the importance of manufacturing methods. But highlighted the lack of transparency and form freedom limitations in topology-optimized cast glass. TThis study addresses these challenges by utilizing 3D printing, which allows for the creation of fully freeform glass structures with inherent translucency and zero waste. Therefore, this master's thesis investigates and implements the new potential of 3D printed glass. Several studies have highlighted the significant potential of using 3D printing techniques to create such structures.

The limitations in creating a monolithic glass bridge stem directly from the brittle nature of glass, leading to disparate behavior in terms of maximum allowable tensile and compressive limits. The use of commercial software to calculate these stresses is no longer possible due to the difference in compressive and tensile strength. Previous research has focused mainly on developing algorithms that can optimize structures within these constraints, initially in 2D and later in 3D. This thesis builds on these foundations and includes considerations regarding the manufacturing limitations of structural glass elements, with the limitations of 3D printing glass chosen for this thesis.

The research starts with a literature review which compasses a wide range of imporant topics for this master thesis.

Firstly, it delves into the properties of glass and its specific characteristics, establishing a fundamental understanding of the challenges involved in crafting a monolithic glass structure.

Secondly, it explores existing fabrication techniques for glass, particularly focusing on producing intricate and customizable shapes. This section underscores the potential of utilizing 3D printed glass for creating complex structural forms. Various fabrication methods such as float, casted, and printed glass are examined, with limitations identified for each. Ultimately, it is concluded that 3D printed glass presents significant opportunities for glass structures. Where the main limitations for structural elements created with 3D printing are: overhang, path continuity, and path with (due to the nozzle size).

The third section delves into the principles and applications of topology optimization, laying the groundwork for further research. Initially, it discusses the working principles of topology optimization before evaluating necessary extensions to accommodate the limitations of 3D printing glass. This establishes the criteria for the type and principle of algorithm required to implement these limitations effectively. It was determined that the SIMP methodology is the most suitable approach for this project. Initially, a 2D model was tested and employed for the structural analysis. Subsequently, this model was scaled up to 3D, utilizing 8-noded hexahedral elements. The implementation of a layer-to-layer overhang filter was necessary to adhere to the overhang constraints inherent in 3D printing glass. The islanding effect was addressed using a method identified in the literature, employing the Matlab islanding function. Path control involved configuring the medial axis, followed by establishing shape boundaries based on a minimum multiplication of two times the nozzle width.

These elements are incorporated into the algorithm in the following section, where initial results and findings are elaborated upon. Specific topics are addressed, and after testing, a decision is made to adjust the overhang angle from the standard 45 degrees to greater than 60 degrees by modifying the size of the 8-noded hexahedral. This adjustment results in the node becoming twice as high as its length / depth, creating elements with a midpoint angle of 63.4 degrees, meeting the required overhang of 60 degrees. Initially, islanding was attempted using the Matlab connection function, but final testing revealed an excess of material filling the structure beyond the designated amount. Consequently, after further investigation and implementation, the Dijkstragrid was integrated with the function, facilitating the connection of islands in each layer. The nozzle offset was ultimately achieved using Python, where boundaries were defined, and the medial axis was established. Interconnection points formed boundary regions, later connected by doubling the nozzle size boundary. Subsequently, outcomes are sliced using existing algorithms to ensure alignment with elements generated by the algorithm.

Following the implementation of these changes, they are tested within the predetermined design domain. Subsequently, the limitations and opportunities afforded by these implementations are evaluated. Afterwards, the final outcome is discussed.

The results are further assessed in the subsequent chapter, presenting the final outcomes for the selected design domains. These outputs are carefully evaluated, leading to the selection of a desired shape based on specific criteria of the desing domain.

The final chapter involves testing the feasibility of the selected outcome using 3D printing techniques, providing tangible validation. This includes generating a G-code file suitable for printing by a robot, as well as fabricating a glass variant using the lost wax technique.

Conclusions regarding the entire thesis and potential avenues for future research are discussed in the conclusion section. While the potential of the created object is acknowledged, further research is deemed necessary to ensure its structural integrity, particularly regarding the incorporation of glass layering anisotropy, which was not addressed in this thesis.

A reflection on the broader academic implications within the field of the built environment concludes this master thesis.

In the appendix, several extensions of the main research are presented, delving deeper into various facets and implications of the study's findings. These extensions offer valuable insights and augment the comprehensiveness of the research, providing a more nuanced understanding of the subject matter. Additionally, the appendix includes findings that were ultimately not incorporated into the thesis but were nevertheless subjected to testing for potential applications within its scope.

Acknowledgements

The completion of this master's thesis ends my time at the TU-Delft. My understanding in the realm of architecture and building technology has significantly broadened. Never could I have anticipated, seven years ago, that this journey would contribute to a bachelor's degree in architecture and a double degree in both architecture and building technology. Particularly noteworthy is the focus of this building technology graduation, which centered on developing an algorithm for 3D printed topology optimized glass. If someone had told me seven years ago that this would be my subject, I would have found it hard to believe, given the high level of complexity inherent in the realm of coding / topology optimization.

Hence, I extend my gratitude to all those who contributed to the realization of this project.
First and foremost, I express my appreciation to Faidra Oikonomopoulou for her support and trust throughout the graduation project. Her expertise, and undivided attention througout the project especially was immensely beneficial. Additionally, Charalampos, as my secondary mentor, for guiding me through the principles of the algorithm and providing essential strategies for working with MATLAB and strategies for coding. Helping me to get to use DelftBlue was benneficail for this master thesis.

In the realm of topology optimization, I am especially grateful to Matthijs Langelaar for several insightful discussions and for his guidance on implementing the layer-by-layer filter into the optimization algorithm. As a guest teacher, his generosity with his time and patience was invaluable, especially considering my beginner status in topology optimization just six months ago. I also express gratitude to Can Ayas for initiating my journey into topology optimization by providing essential books, even though Langelaar eventually became my primary guide due to his expertise with the filter implementation.

I must also recognize Josephine and Jackson for their assistance and encouragement in navigating the challenges of topology optimization. Mainly their passion for their respective fields was truly inspiring. Understanding and mastering topology optimization proved to be a gradual process, and their guidance was needfull.

Moreover, I extend my thanks to Eva Schoenmaker for her assistance with understanding her code and topology optimization as a whole. Prateek for setting up the supercomputer for the calculations. Special mention goes to my friends Chiel and David, who aided in troubleshooting the overhang filter and setting up the 2D topology optimization algorithm.

In terms of material aspects, I appreciate the support of Kaithlyn and Daniel, who patiently addressed my inquiries regarding the manufacturing process, particularly regarding the printing of glass. Their insights into the feasibility and material-specific considerations were extremely usefull.

Lastly, I express my gratitude to Telissia and Maindres for their explanations of the lost-wax technique, which proved essential in creating a cast prototype from glass.

To conclude, I would like to thank my roommates (de ongekenede hoogte) and friends (Roelof, Raymen, Koen) whom I have become very close with over the past two years in the Building Technology program.

As I stand on the threshold of graduation from TU Delft.
I reflect on the incredible journey this has been.

"Glass as a material, has properties and characteristics we are only just beginning to understand."

Adam Savage

Contents

1. Research frame (14 | 19)
1.1 Research introduction
1.2 Research goal
1.3 Research question
1.4 Methodology
1.5 Time planning
1.6 Graduation schedule

2. Case study (20 | 27)
2.1 Case study 1
2.2 Case study 2

3. Glass the material science (28 | 43)
3.1 Glass
3.2 The history of glass
- 3.2.1 Fabrication history
- 3.2.1.1 Fabrication of float glass
- 3.2.1.2 Fabrication sand moulds for casting glass
- 3.2.1.3 Initiating 3D Printing with Glass

3.3 Structural glass
3.4 Strength of glass
3.5 Chemical composition of glass
3.6 Properties of glass
3.7 Annealing process
- 3.7.1 Approximating the Annealing time

3.8 Flaws and defects in glass
3.9 Recycling of glass

4. Fabrication of glass (44 | 71)
4.1 Fabrication of glass
4.2 Float glass
- 4.2.1 Float glass production
- 4.2.2 Cutting of float glass
- 4.2.3 Lamination
- 4.2.4 Structural considerations

4.3 Casting
- 4.3.1 Sand Mold
- 4.3.2 Lost-Wax Technique

4.4 3D printing
- 4.4.1 Fused deposition modelling
- 4.4.2 Stereolithography and digital light processing
- 4.4.3 Selective laser sintering (SLS) and Selective laser Melting (SLM)
- 4.4.4 Direct ink writing
- 4.4.5 Comparison 3D printing of glass

4.5 Comparison of laminated/casted & printed
4.6 FDM - printer G3DP2
4.7 Limitations of glass FDM printing - incorpeartion in the algorithm

5. Algorithms – topology optimization (72 | 107)
5.1 Topology optimization
5.2 Structural optimization problems
5.3 Topology optimization methods
- 5.3.1 Gradient based methods
- 5.3.1.1 Homogenization
- 5.3.1.2 SIMP – Solid isotropic material with penalization
- 5.3.1.3 LSM – Level Set Method
- 5.3.2 Gradient – free | direct search method
- 5.3.2.1 ESO – evolutionary structural optimization
- 5.3.2.2 BESO – bidirectional evolutionary structural optimization
- 5.3.2.3 SKO – soft kill option
- 5.3.2.4 Other direct search methods

5.4 Comparison
- 5.4.1 Gradient vs Direct search
- 5.4.2 Combination
- 5.4.3 Conclusions

5.5 Checkerboards, point flexures and islanding
5.6 Finite elements
- 5.6.1 1D elements
- 5.6.2 2D elements
- 5.6.3 3D elements

5.7 Formulation of the optimization problem
- 5.7.1 Compliance-based minimization
- 5.7.2 Volume-based minimization
- 5.7.3 Stress-based minimization

5.8 Topology optimization algorithms
- 5.8.1 99-line for 2D topology optimization
- 5.8.2 88-line for 2D topology optimization
- 5.8.3 200-line for 2D topology optimization
- 5.8.4 2D topology optimization for cast glass structures (Koniari)
- 5.8.5 169-Line 3D topology optimization algorithm (top3d)
- 5.8.6 3D topology optimization for cast glass structures (Schoenmaker)

5.9 Constraints
- 5.9.1 Filters
- 5.9.1.1 Sensitivity filter
- 5.9.1.2 Density filter
- 5.9.1.3 Density filter with a Heaviside step function
- 5.9.2 overhang filters
- 5.9.2.1 Overhang filters overview
- 5.9.2.2 front propagation-based overhang-filter
- 5.9.2.3 Layer by layer overhang filter
- 5.9.3 Structural constraints

5.10 Solver
5.11 Path finding
- 5.11.1 Direction-parallel paths
- 5.11.2 Contour-parallel paths
- 5.11.3 Space filling paths
- 5.11.4 Adaptive paths

5.12 Nozzle offset generation in topology optimization
5.13 Path continiuty

Contents

6. Development of the algorithm (108 | 147)
6.1 Introduction
6.2 Introduction to design domain specifics
6.3 Introduction to setup complete step by step Algorithm
6.4 2D overhang
- 6.4.1 2D overhang constrant
- 6.4.2 2D overhang constraint adaptation 60°

6.5 3D overhang
- 6.5.1 3D overhang constrant
- 6.5.2 3D overhang constrant adaptation 60°
- 6.5.3 3D overhang constrant adaptation 63.435°
- 6.5.4 3D overhang constrant adaptation 26.6°
- 6.5.5 3D overhang constrant adaptation 75°
- 6.5.6 3D overhang constrant adaptation 15°
- 6.5.7 3D overhang addition
- 6.5.8 3D overhang conclusion

6.6 Path continuity
- 6.6.1 Path continuity
- 6.6.2 Path continuity Dijkstra
- 6.6.3 Path contnuity conclusion & discussion

6.7 Nozzle offset
6.8 STL conversion
6.9 Mirror of the element
6.10 Grasshopper and visualisation steps
6.11 Conclussions & other implementations

7. Algorithm testing (148 | 189)
7.1 Introdcution
7.2 Algorithm performance 2D
7.3 Design - case study (boundary conditions)
7.4 Glass addition in topology optimization
7.5 Layer by Layer filter
- 7.5.1 Layer by layer
- 7.5.2 Layer by layer orientations
- 7.5.3 Layer by layer orientations front side
- 7.5.4 Layer by layer objective

7.6. Islanding problem speed and comparison
- 7.6.1 First comparison
- 7.6.2 Second comparison
- 7.6.3 Third comparison
- 7.6.4 Fourth comparison (big domain)
- 7.6.5 Islanding problem orientation reconnecting (print-direction)
- 7.6.6. Islanding problem conclusion & discussion

7.7 Nozzle offset
7.8 STL conversion time and performance
7.9 Final outputs and results
7.10 Conclusions and comparison
7.11 Conclusions and comparison (case study 1)
7.12 Conclusions and comparison (case study 2)
7.13 Discussion

8. Final geometry (190 | 219)
8.1 Final geometry 480
8.2 Final geometry 600
8.3 Final geometry 800
8.4 Final geometry
8.5 Ansys
8.6 Final detailing
8.7 Different material visualization
8.8 Render bridge (case study 1)
8.9 Render stepping stones (case study 2)

9. Prototype production (220 | 275)
9.1 Small scale plastic
9.2 RoboDK
9.3 Glass prototype
9.4 Final pictures glass model
9.5 Waterjet cut glass
9.6 Final pictures waterjet cut glass
9.7 Final pictures PET-G scale maquette 1:10

10. Conclusion (276 | 281)
- Material properties of glass
- Fabrication of topology optimized glass structures
- Fabrication limitations
- Algorithms
- Fabrication
- Concluding

11. Discussion (282 | 287)
- Evolution of the algorihm
- Limitations for the overhang
- Islanding Dijkstra and connecting the dots
- Nozzle offsetting
- Combining variables for printing
- 3D printing properties
- Post processing
- Post processing additions
- 3D printing and testing
- Process

12. Reflection (288 | 293)
1. What is the relation between your graduation project topic, your master track (A, U, BT, LA, MBE), and your master programme (MSc AUBS)?

2. How did your research influence your design/ recommendations and how did the design/ recommendations influence your research?

3. How do you assess the value of your way of working (your approach, your used methods, used methodology)?

4. How do you assess the academic and societal value, scope and implication of your graduation project, including ethical aspects?

5. How do you assess the value of the transferability of your project results?

6. How did the existing research differ/extend from the findings and development of the project?

7. If the project would develop further what would be your recommendations?

13. References (294 | 299)

14. Appendix (300 | 324)
- Getting toplogy optimmization & overhang filter to work
- Printed TO in PET-G
- Path offset nozzle trials with Matlab
- Printed objects wax model for glass model
- Printed candel wax in PLA model
- Chocolate in silicon mold
- Blender (rendersoftware)
- DelftBlue enviroment personal setup

01

Research framework

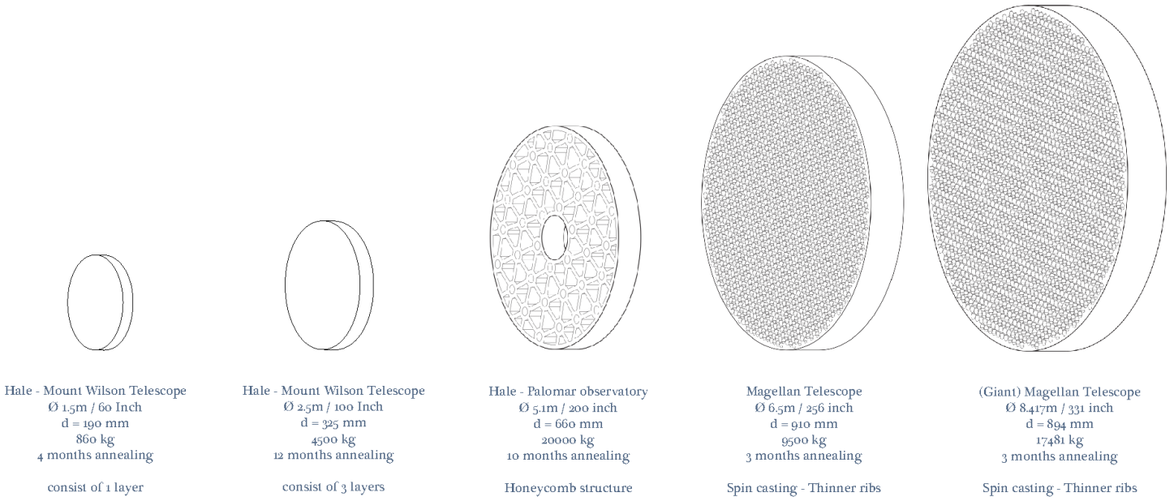
Introduction, Background information, framekwork, research goal, research question and methodology

1.1 Research introduction

The significant design potential of glass, combined with its exceptional compressive strength that exceeds that of traditional structural materials such as concrete or structural steel, underlines its importance as a building material. Moreover, its advantages in transparency, sustainability and recyclability increase its appeal. Nevertheless, significant challenges arise during the annealing process. The longer time period and significant energy requirements for the annealing process make it inefficient in terms of both time and cost for the industry. Various factors influence the annealing time of the glass components. These are mainly their mass and the general shape of the geometry, but also the composition of the glass (Oikonomopoulou, 2019).

To reduce the mass and minimize the geometric radii to significantly shorten the annealing time, topology optimization has been used as a possible solution. The goal of topology optimization is to reduce mass while maintaining the allowable strength of the freeform structural component. These optimizations are tailored for both 2D and 3D scenarios. However, the resulting optimized designs often exhibit complicated and complex shapes, which pose manufacturing challenges. Float glass by its nature tends to limit structures to two-dimensional, planar elements. As a result, architects often face limitations in realizing their creative visions within the constraints imposed by conventional float glass manufacturing processes. Cast glass, on the other hand, offers an attractive alternative that offers the opportunity to create completely transparent structures with virtually unlimited design freedom. Recent experiments have shown the feasibility of exploiting this freedom when shaping complex and precise interlocking elements. However, the milling potential of topology optimization is limited due to the intensive post-processing of the cast glass elements. This mainly takes into account different coatings or polishing methods. An alternative approach that shows promise is the use of FDM 3D printing technology with glass. Glass is accurately applied into complex shapes using a computer-controlled 3D printer. The domain of 3D printing with glass remains largely unexplored. Thus, incorporating the limitations associated with 3D printing glass into the topology optimization process could reveal potential structures in which the material's unique properties excel.

By integrating the parameters specific to glass 3D printing, into the optimization algorithm, it becomes possible to generate designs that are not only structurally optimized but also feasible for fabrication using 3D printing techniques. This approach not only addresses the challenges posed by conventional annealing processes but also opens up new avenues for the efficient and customizable production of glass structures with intricate geometries.
This thesis endeavours to overcome the challenges posed by 3D printing of glass by devising a custom algorithm for topology optimization tuned to the distinctive structural properties of both glass and the printer. The algorithm will seamlessly integrate manufacturing criteria, thereby establishing a comprehensive tool for the design of glass topology optimized components.



Evolution of cast mirror blanks used for telescopes
Figure 1, based on: F. Oikonomopoulo et al.; The university of Tokyo; Mtwilson



Figure 2: https://www.brecorder.com/news/462681

1.2 Research goal

This thesis strives to advance current research by creating an algorithm specifically tailored for designing topology-optimized 3D-printed glass components. This algorithm will not only consider the structural properties of glass but also address manufacturing constraints of 3D printing. Contributing to the broader exploration of 3D printing in topology optimization in the context of glass structures.

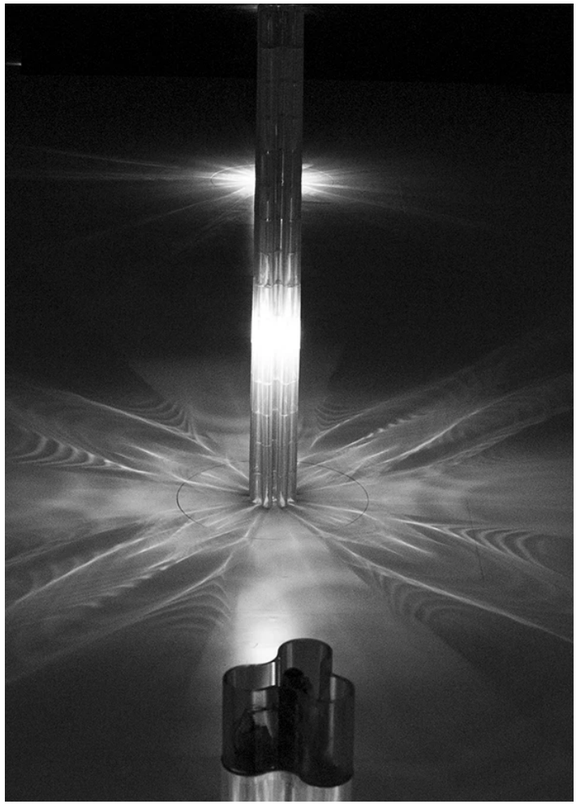


Figure 3: https://catarinensepharma.com.br

1.3 Research question

Given the research goal, the main research question is defined as follows:

- How can the manufacturing process be tailored to fabricate a topology-optimized monolithic glass structure while ensuring effective production constraints?

Secondary goals, supporting the exploration of the method's potential, include:

- What properties of glass, should be tailored to create a topology-optimized structure using monolithic glass?
- What fabrication options exist for constructing freeform giant glass structures, and what are the associated advantages, disadvantages, and potential applications in conjunction with topology optimization?
- What are the constraints inherent in this favored fabrication technique, and how must they be adapted to suit a topology-optimized structure?
- Which algorithms (implementations) are essential for producing a topology-optimized glass structure, particularly in the context of 3D printing?

Additionally the fabrication of the structure can be tested for the feaseability of the algorithm:
- Can the integration of glass constraints be achieved in the creation of a topology-optimized structure using additive manufacturing, mimiced with the use of large-scale plastic additive manufacturing?

1.4 Methodology

The project is divided into phases, each leading to outcomes contributing to the final result.
The thesis begins with a literature review covering scientific studies and recent theses at TU Delft related to glass material features, topology optimization methods, characteristics of the fabrication methods. Information derived from each theme is used in the next phase to formulate the optimization problem.

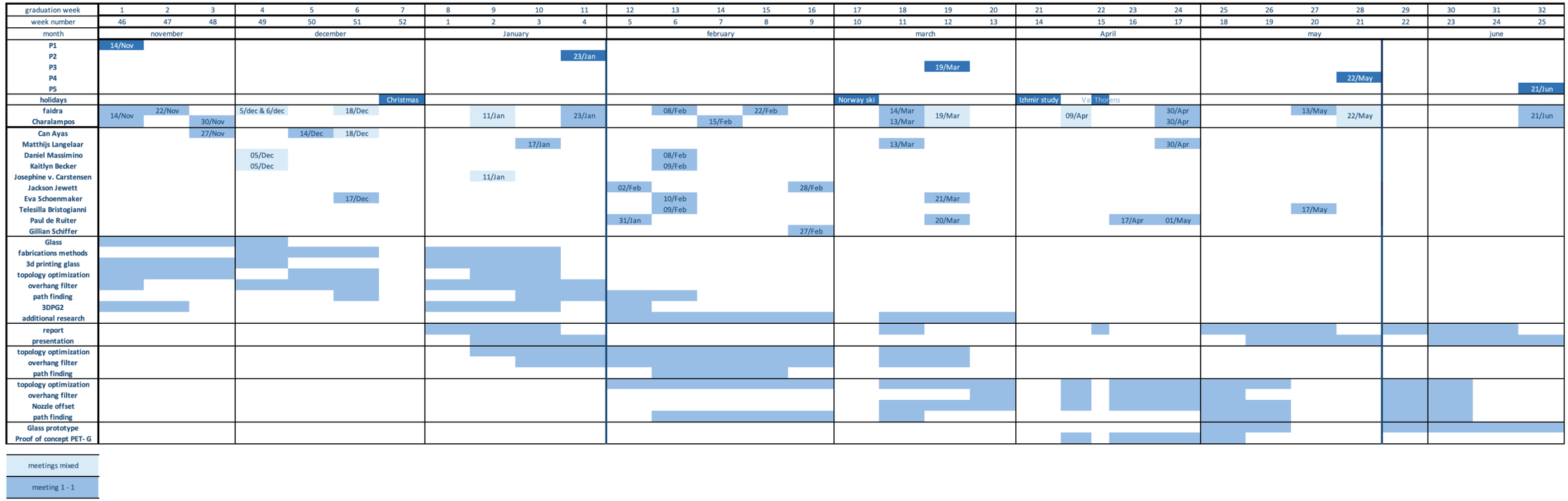
In the design process, we tailor the optimization problem to fit the printer's limitations. The code is then applied to the design, and we compare the results. Initially, we use standard domains for optimization, like the MBB beam, and then consider the constraints posed by the printer. Subsequently, the problem can be adapted to suit more unique structures, such as bridges or roofs.

Once the design is completed, the shape will be refined to achieve the desired outcome through tweaking of the parameters. The algorithm's performance will be closely evaluated. Following this, a proof of concept is conducted by testing the design on a model created through additive manufacturing. This will be done with the use of petg printers as the availebilty and complexity of the glass printer is for this stage out of the scope. However when these prints seem promissing test can be conducted in glass. This process not only helps ensure the design's viability but also generates new insights for refining the algorithm, making it user-friendly for individuals with specific requirements.

1.5 Time planning

The building technology graduation officially started on November 13th. Initially the focus of the research was directed towards reviewing relevant literature to establish boundaries for the specific research questions. My aim was to complete this graduation within the originally designated timeframe, which allowed for approximately 29 weeks to complete the thesis. Effective time management was crucial, especially considering the time required for printing and refining the topology optimizations. Algorihme testing is scheduled to start immediately after P2, with the plan to initiate the first trial production of the topology optimizations following P3. The algorihm of dijkstra was finished at the time of P3 only the visualization for the end product needed to be conducted. Until P4 work was done on the overhang filter & the nozzle offset tool. Hereby creating time to apply changes to the final optimization. Testing of the algorihm started 2 weeks previous to P4. thereby having all elements and chosing which optimization fits best to the specific design cases. But most importantley production was layed within casting of glass of the 3D printed part. This production process took a lot of time between P4 and P5. After completing the production, the final steps involved rendering and finishing the booklet and preparing the presentation, which would conclude the graduation project.

1.6 Gradution schedule



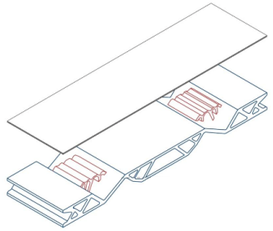
02

Case study

This chapter is dedicated to introducing the chosen case study, which involves the redesign of a traditional glass structure. Later the case study appliciable to the current chosen glass printer is discussed. The objective is to evaluate the effectiveness of the topology optimization algorithm in practical application.

2.1 Case study 1

To give a good comparison between the development of the algorithms the first case study that is selected is the bridge in the British museum in London. This case study is previously used by Koniari (2022) and Schoenmaker (2023). Therefore the comparison in final shape production of the topology optimization algorithm can be seen. The redesign of the structural part of the bridge is redesigned as the following:



Anna Maria Koniari 2022



Eva Schoenmaker 2023

To gather an insight in the bridge that is chosen and transformed to a monolithic glas bridge a small introduction about the bridge itself.

Forster + Partners' overhaul of the Great Court at The British Museum featured the addition of a pedestrian bridge connecting the Reading Room to other exhibition areas. This bridge is notable for its glass bottom, constructed from float glass sheets supported by a metal substructure anchored to adjacent walls (Koniari, 2022). With dimensions measuring approximately 2.30m x 4.20m x 0.20m (width, length, height).

Koniari (2022) undertook a redesign of the bridge, employing a two-dimensional topology optimization tailored for cast glass structures. This method integrates parameters such as maximum annealing time and specific material properties of glass. Schoenmaker (2023) took this same bridge and made a redesign using a three-dimensional topology optimization tailored for cast glass structures. Where she also incorperated the maximum annealing and material specific parameters of glass.

In both cases the redesign primarily focuses on the monolithic slab, which constitutes the structural core of the design. Notably, the algorithm does not encompass the railing, which is incorporated post-slab design completion.

For the supports, it is assumed that the slab will be firmly affixed to neighboring walls at the short side of the domain, ensuring full fixation. Within this new algorithm design the slab will be considerd as simply imposed. therefore changes in the final shape are fairly great at the place of affixation. The two layers of float glass will be placed atop the monolithic structure to shield it from accidental damage and local contact stresses. This will also be used in the making of this redesign.

Although the bridge narrows in width, for computational simplicity, the domain is consistently modeled as rectangular. Consequently, the slightly curved edges are treated as straight lines during calculations.



Figure 4: https://www.agefotostock.com/age/en/details-photo/ionic-columns-at-the-great-court-by-norman-foster-of-the-british-museum-london-england-uk/B20-476327

To ensure the structural integrity meets the intended loads and usage requirements, various load cases must be considered. One scenario involves a combination of dead load and distributed live load. The dead load encompasses the floor's weight, including any finishes, while the distributed live load represents the weight of individuals occupying the floor. For museums (Category C3), the live load typically ranges from 3.0 to 5.0 kN/m2.

Another scenario entails asymmetric loading, where the live load is applied solely on one side of the structure. Additionally, all loads must be multiplied by a safety factor; permanent loads have a factor of 1.2, while short-term loads, such as live loads, require a factor of 1.5.

Regarding allowable deformation, two distinct norms are relevant. One pertains to component deflection during construction and installation, while the other, stricter norm governs allowable deformation during use (Koniari, 2022). Only the norms for use are therefore considerd as they are twice as strict.

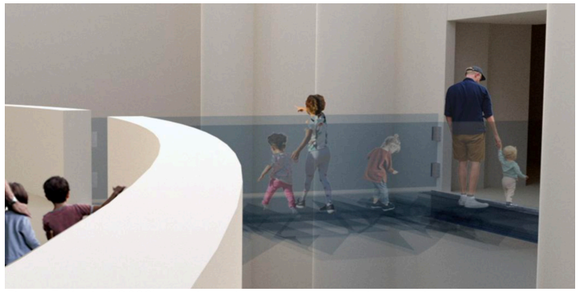


Figure 5: Koniari (2022) two-dimensional topology optimizated bridge

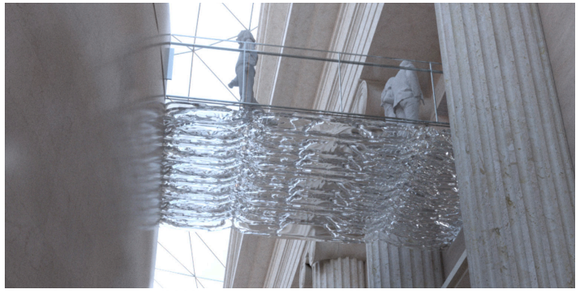
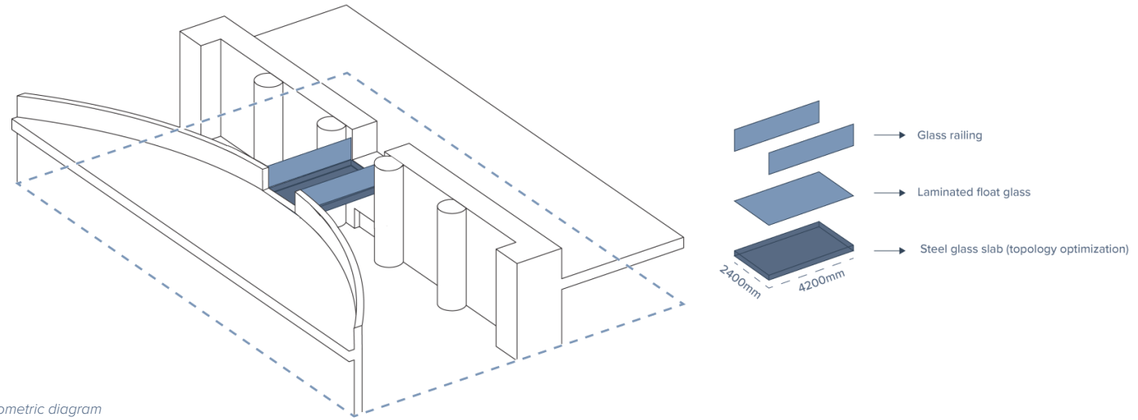


Figure 6: Schoenmaker (2023) three-dimensional topology optimized bridge



Isometric diagram



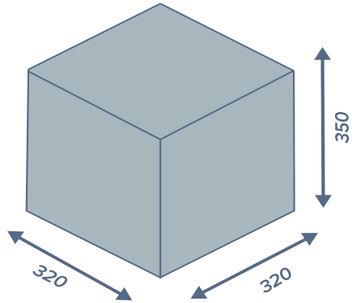
Figure 7: https://www.flickr.com/photos/darrellg/7172296625



Figure 8: https://www.archivibe.com/great-court-at-the-british-museum-by-foster-partners/

2.2 Case study 2

This thesis selects another case study, opting for the G3DP2 3D printer for architectural purposes, as it will be demonstrated in this report. However, the current printer exhibits certain limitations, notably its relatively small print bed/chamber. With a maximum area of 320mm x 320mm on the baseplate and a height limit of 350mm, design possibilities are constrained. The objective of this case study is to create an object prioritizing compressive load while adhering to these spatial constraints.
Steppingstones emerge as an ideal solution, where the design will primarily focus on ensuring compression connection to the bottom component which in this case is underwater. To enhance impact resistance, a double laminated float glass piece will be incorporated on top, dispersing forces and reinforcing the topology optimized loading structure. Connection points on the underside will be formed through the convergence of various design outputs, with the most aesthetically pleasing and feasible option selected for production.
The chosen stepping stones are the pathway through a park in Dorking, England.



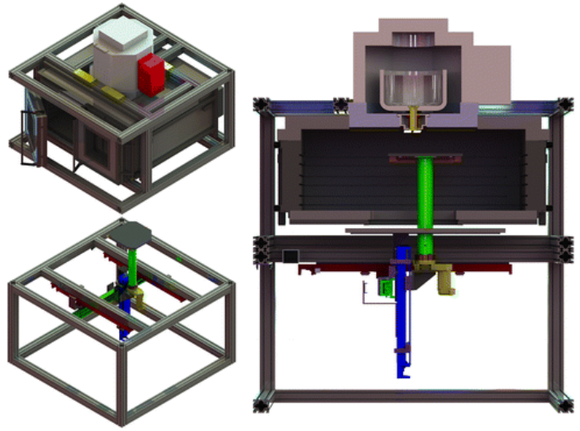


Figure 9: https://www.borntoengineer.com/mit-engineers-are-now-3d-printing-glass

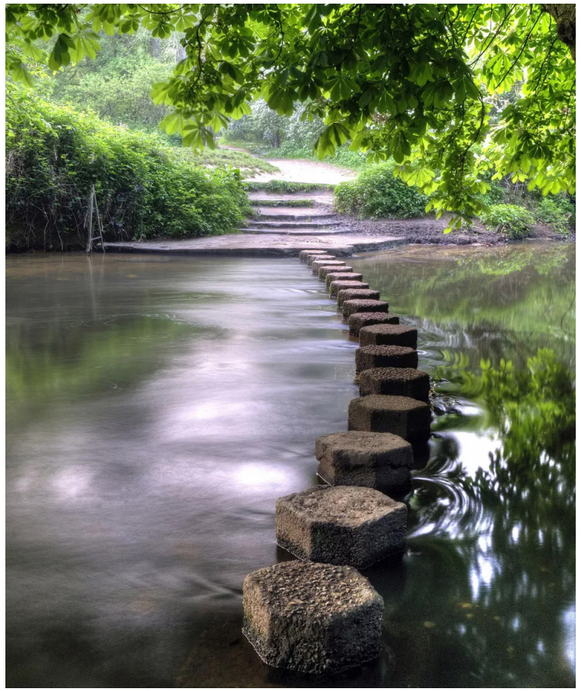


Figure 10: https://www.nationaltrust.org.uk/visit/surrey/box-hill/box-hill-stepping-stones

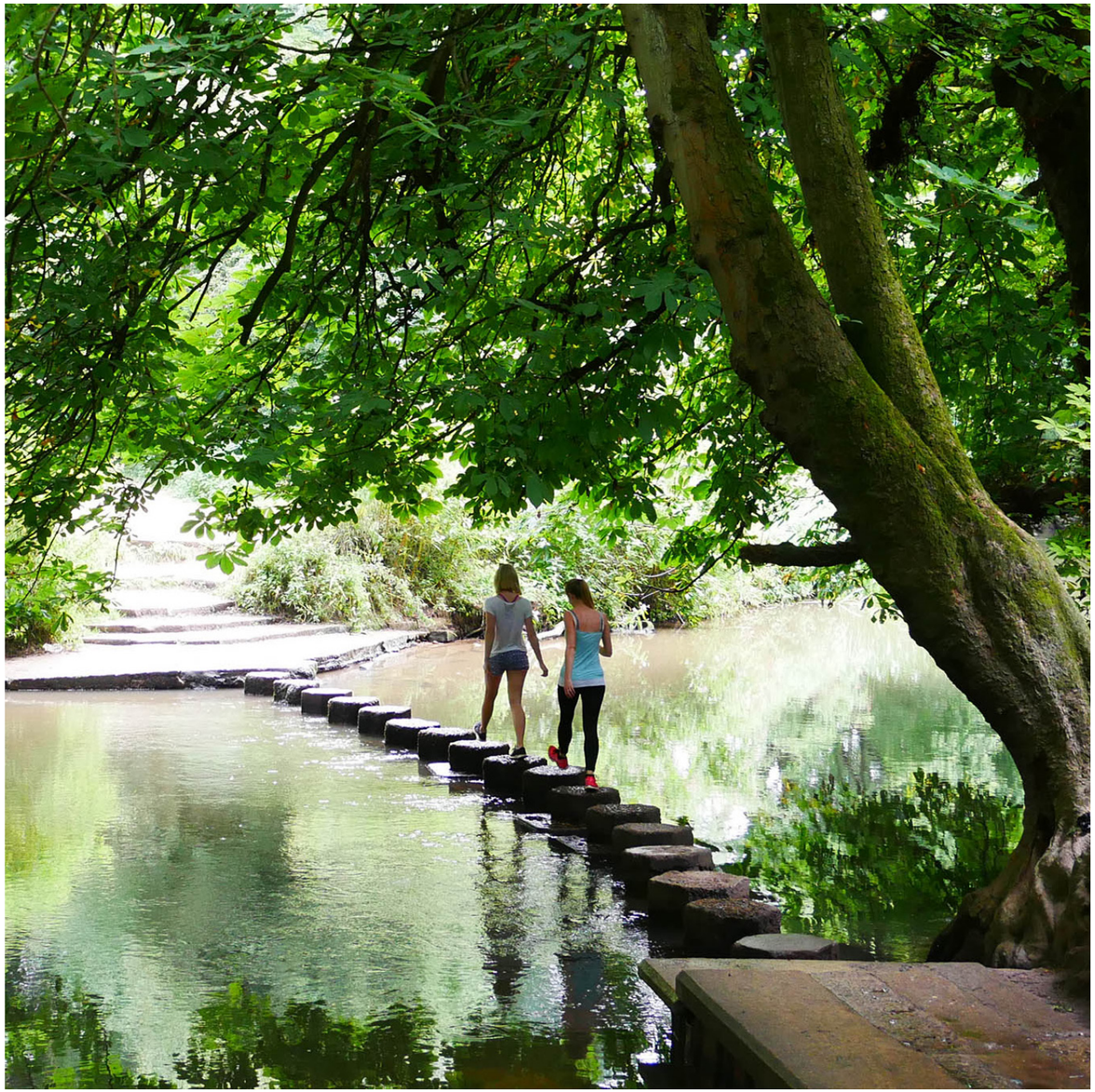


Figure 11: https://www.nationaltrust.org.uk/visit/surrey/box-hill/box-hill-stepping-stones-walk



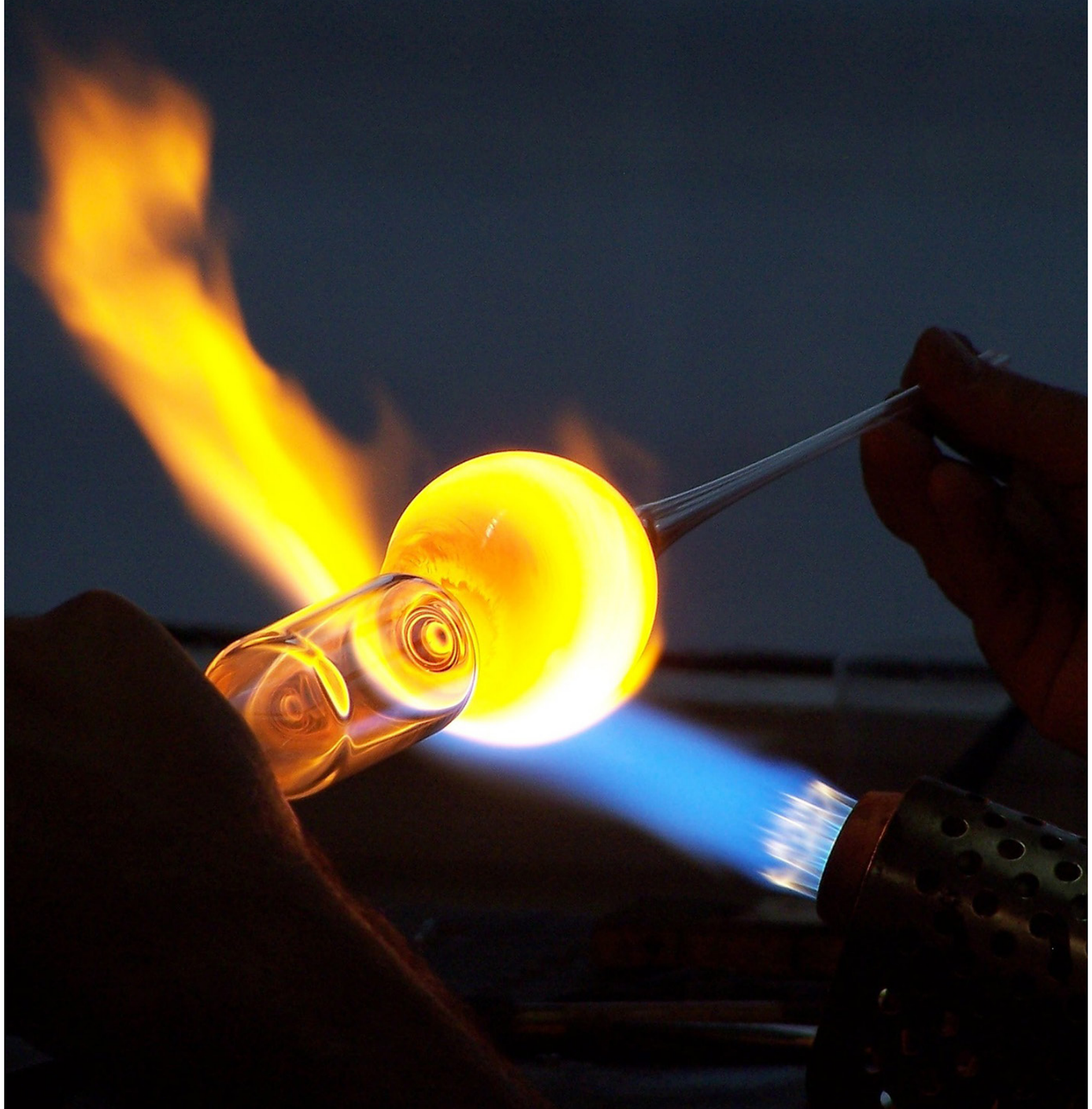
03
Glass

Summarizing the features and historical development of glass production, this introduction explores modern applications of structural glass while acknowledging the constraints posed by its properties and current manufacturing methods for use with topology optimization.

3.1 Glass

Glass has exceptional and distinctive characteristics, is a highly sought-after material across various industries, including construction, automotive, boating, interior design, optics, glassware, art, tableware, and data cables. Its transparency, strength, recyclability, and longevity make it an ideal choice for a wide range of applications. Depending on its composition, glass can exhibit various features such as transparency, translucency, coloration, or clarity (Musgraves et al., 2019). its mechanical and thermal properties can also vary based on the different compositions.

Formed by solidifying a liquid with a consistent composition, glass achieves transparency through the complex and random arrangement of its molecules, lacking a crystal lattice formulation. Unlike crystals, glass exhibits properties that remain consistent regardless of the measurement direction, known as amorphous isotropy (Schittich & Lenzen, 2007). Glass does not corrode easily and doesn't react to most acids or alkaline solutions. This makes it great for structures that don't need a lot of maintenance.



3.2 A brief history of glass

Glass, an ancient material originating in Mesopotamia and Egypt approximately 4500 years ago, has undergone diverse manufacturing techniques and recipes throughout the centuries (Klein, 2015; Musgraves et al., 2019). One of the earliest methods, glass casting, dates to the 15th century BC (Trentinella, 2003). Key milestones in the glass manufacturing timeline include the invention of glass blowing in the 1st century BC, the cylinder method around the 2nd century BC, the crown or disk method in the 4th century BC, plate glass casting in the 17th century, and the float glass process in the mid-20th century (Musgraves et al., 2019).

The initial use of glass was in jewellery and art, with Roman times seeing its emergence as glazing to admit light into buildings. Until the Middle Ages, glass glazing remained a privilege for the upper classes and churches, particularly in the form of stained glass. In the 17th century, the preference shifted to clear glass over stained glass, influenced by the rationalism movement. In the mid-19th century, the concept of architectural glass materialized with the construction of the Crystal Palace (1851) by Sir Joseph Paxton (Musgraves et al., 2019). The Crystal Palace revolutionized production and construction methods through innovative changes. Although it marked the first application of architectural glass, the glass itself was not structural, relying on a lightweight cast iron skeleton for support.

Figure 12: https://berngallery.com/blog/how-to-blow-glass/

3.1.1 Fabrication methods

A brief overview of the initiation of various fabrication processes is provided here. For a comprehensive comparison, readers are directed to Chapter 4, where a detailed explanation of each type of glass production is presented.

3.1.1.1 Fabrication of float glass

The introduction of the float glass manufacturing process by Pilkington in 1959 (Pilkington company, n.d.), marked a significant turning point that enabled the utilization of glass in all-glass building envelopes and, subsequently, as structural glass façades. Throughout its history, glass has evolved from being a fragile and opaque material to becoming a durable, optically transparent, and structural substance with considerable potential in future architectural and structural applications (Pilkington company, n.d.).

3.1.1.2 Sand moulds for glass

Cast glass, the oldest method of glassmaking, has a rich history dating back to more than 2000 B.C. in Mesopotamia, where beads and small objects made of molten glass were cast in molds (The Corning Museum of Glass, 2001). In recent years, glass manufacturing has undergone a transformative through specialized 3D printing techniques. The procedure can be implemented trough 2 direct ways, one includes employing binder jetting with sand molds, the other, adopting the lost wax technique. In the lost wax technique a mold is formed by burning the wax model within a sand bed to generate the reverse of the desired object. (Giesecke & Dillenburger, 2022; Oikonomopoulou, 2019; Bhatia, et al., 2020). Moreover, the specific principles of these techniques can be found at chapter 4.2

3.2.1.3. 3D printing with glass

Over the past two decades, the utilization of 3D printing technology for directly producing glass has played a significant role in its implementation. Numerous researchers have explored this avenue, although the size of the manufactured glass objects has generally remained limited. Notably, the research conducted by the Mediated Matter Group, specifically the G3DP and G3DP2 projects, has yielded promising outcomes in glass printing using beads with a maximum width of 10mm (Inamura et al., 2018). This achievement marks a milestone in architectural applications, representing the first successfully scaled project in this domain. For a more in-depth exploration of the topic, readers are directed to Chapter 4.3.



Figure 13: https://news.mit.edu/2015/3-d-printing-transparent-glass-0914

3.3 Structural Use of glass

The structural application of glass is a relatively recent development, spanning only a few decades, as opposed to materials like steel or concrete, which have been in use for approximately 200 and 100 years, respectively. Due to its distinctive and sometimes contradictory properties, it is anticipated that glass will find broader use as a structural material in the coming years (Nijsse, 2003). Presently, there is a growing body of research and projects exploring the structural applications of glass. Glass is favoured over other materials for its transparency, durability, and favourable mechanical properties, especially its compressive strength. Float glass pieces are laminated together with a transparent interlayer to create structural elements. Processes like tempering, heat treatment, and chemical strengthening have significantly expanded the applications and research associated with structural elements crafted from laminated glass. A notable instance is the Apple Store on Fifth Avenue in New York (Bohlin Cywinski Jackson, 2006). In this case, fins composed of multiple layers of float glass are laminated together to create the columns and beams of the structure. These structural elements interconnect, resulting in a glass cube structure that is free of structural steel. The design was revisited in 2011, with a reduction in the number of fins and beams, further enhancing the design and transparency of the structure. The reduction went from 106 panels with 250 primary fittings to 15 panels and 40 fittings. (Eckersley O'Callaghan, n.d.).

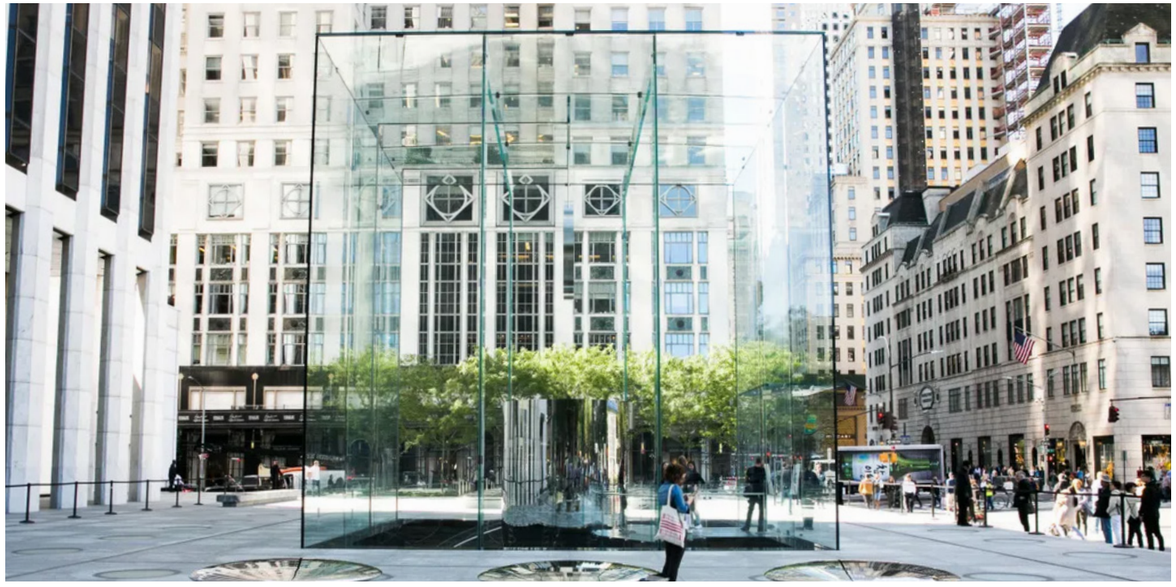


Figure 14: https://www.cnet.com/AppleStoreFifthAvenuereopening

3.3 Structural Use of glass

However, this principal underscores not only the rapid advancement of the evolving industry but also the tangible manifestations of innovation, as evidenced by recent ground-breaking projects like the Beijing city sub-central library, the Apple store in Brussels, and the Steve Jobs Theatre. The city sub-central library, where the glass configuration consists of 7 layers of 15mm ultra-clear glass, each with a dimension of 15050mm x 2500mm (Jinjing Group, 2021).
Similarly, the Apple store in Brussels stands out with its five-layer double-curved façade, pushing the boundaries of architectural design with float glass. These are just a few examples that highlight the ongoing improvements in the structural applications of float glass elements, illustrating the significant growth and potential for innovative structural glass applications.



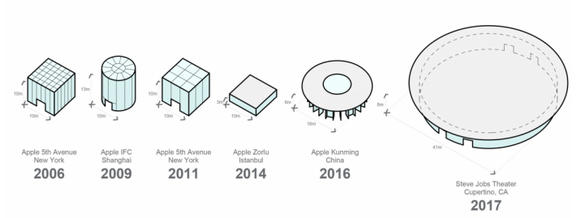


Figure 15: https://breezewiki.com/apple/wiki/Steve_Jobs_Theater



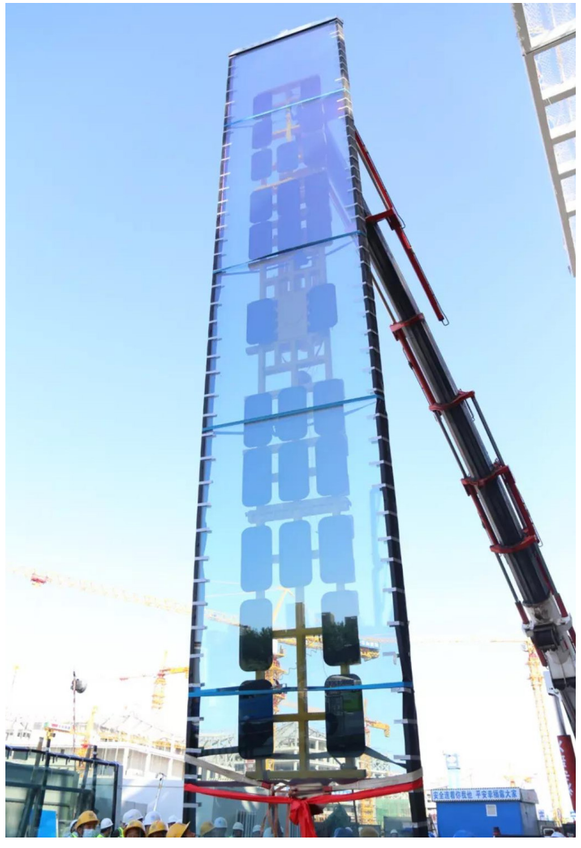


Figure 16: https://www.jinjing-glass.com/news/jinjing-ultra-clear-glass-on-beijing-sub-central-library/

3.4 Strength of glass

The unique characteristics of glass, marked by its amorphous structure without a crystalline arrangement and a lack of a specific chemical formula, result in a gradual transition from solid to liquid when subjected to heat. These distinctive properties, including a high theoretical compressive strength and recyclability, contribute to its durability and versatile applications. The compressive strength for float soda-lime glass has been stated 1000 MPa by Saint Gobain 2016; Weller et al. 2008; Ashby, Jones 2006.

Despite its compressive strength, glass is deemed brittle because it cannot undergo plastic deformation after the breakage of covalent bonds. The amorphous structure, a consequence of rapid cooling after melting, provides transparency but limits its ability to exhibit plastic behaviour (Veer, 2007; Oikonomopoulou, 2019). Glass, being an isotropic material, experiences slight variations in characteristic properties based on its ingredient composition. Overall, its properties, such as Young's modulus and compressive strength, are comparable or even higher to those of other conventional building materials like aluminium and stainless steel respectively (Oikonomopoulou, 2019). However, it diverges from them in that it cannot yield (O'Regan, 2014), leading to brittle failure at normal temperatures. Consequently, its behaviour is nearly perfectly elastic, and its comparison is more aptly made with other brittle materials, such as unreinforced concrete (Oikonomopoulou, 2019).

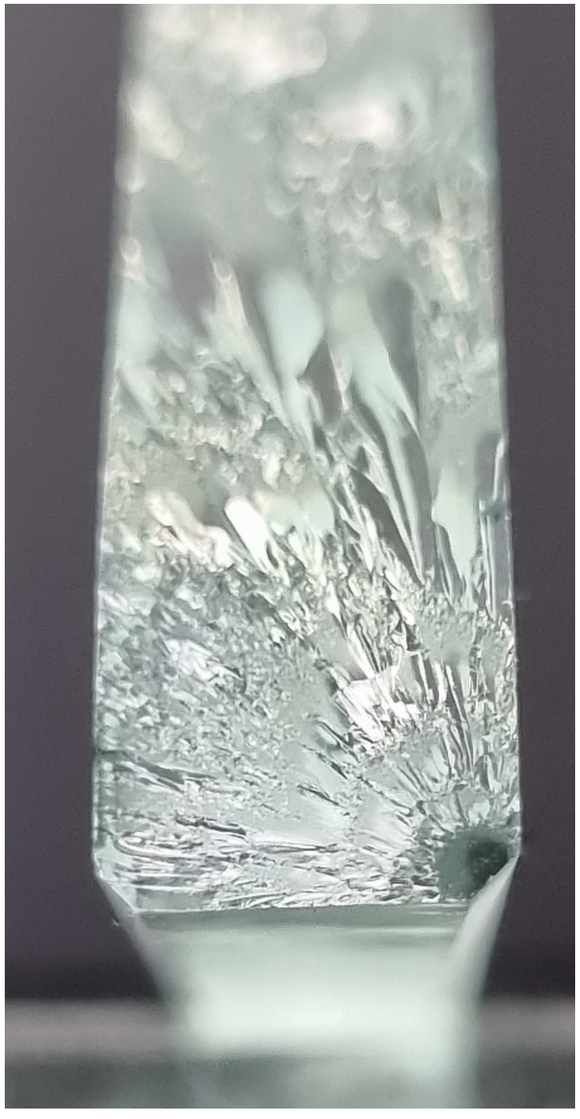


Figure 17: https://www.tu-darmstadt.de/glass-cc/forschung/forschungsgebiete/glass_qualitaet/glas_qualitaet.en.jsp

3.5 Chemical composition of glass

The chemical composition of glass, primarily represented by pure silica oxide, determines its properties. Introduction of various elements alters behaviours like thermal expansion and viscosity. Soda-lime glass, economically advantageous but susceptible to thermal shock, contrasts with borosilicate glass, resistant to thermal shock and suitable for variable temperatures. Lead silicate glass, softer but prone to scratching, is used in artistic pursuits. Other types like aluminosilicate, 96% silica, and fused silica face limitations in structural applications due to high production costs (Oikonomopoulou, 2019).



Figure 18: https://www.glassstudiosupplies.co.uk/Frit-Powder-Blue

Glass Type Symbol	Mean melting point at 10 Pa. s*	Softening point	Annealing Point	Strain Point	Density	Coefficient of Expansion 0°C - 300°C	Young's modulus
	°C	°C	°C	°C	Kg/m3	10-6/°C	GPa
Soda - Lime	1350 – 1400	730	548	505	2460	8.5	69
Borosilicate	1450 – 1550	GPa	525	480	2230	3.4	63
Lead silicate	1200 – 1300	626	435	395	2850	9.1	62
Aluminosilicate	1500 - 1600	915	715	670	2530	4.2	87
Fused - silica	>> 2000	1667	1140	1070	2200	0.55	69
96% Silica	>> 2000	1500	910	820	2180	0.8	67

*These values are only given as a guideline of the differences between the various glass types. In practice, for each glass type there are numerous of different recipes resulting into different properties. Properties based on the different glass composition (Oikonomopoulou, 2019).

3.6 Properties of glass

When evaluating building materials, two prominent options are soda-lime and borosilicate glass. Soda-lime glass, known for its cost-effectiveness and durability, finds widespread use in various applications. It is the most produced glass type, extensively employed in glazing, bottles, and jars. The composition of soda-lime glass contributes to a lower working temperature, resulting in reduced manufacturing expenses, making it a highly economical choice for numerous purposes. However, it does have limitations in its ability to withstand abrupt temperature changes, potentially leading to thermal shock. Additionally, soda-lime glass requires a longer annealing time compared to alternative glass compositions like borosilicate glass (Oikonomopoulou, 2019). On the other hand, borosilicate glass, despite having a higher working temperature and cost, is preferred for cast glass art because of its softer nature, facilitating easier grinding and polishing processes (Thwaites, 2011). The research conducted on glass printers for both the G3DP and G3DP2 systems utilized the identical soda-lime glass formula, referred to as Spectrum Glass system 96 (Daniel Lizardo, 2018).

	Symbol	Units	Soda-Lime	Borosilicate Glass
Density	ρ	Kg/m3	2500	2200 – 2500
Hardness (Knoop)	$HK_{0,1/20}$	Gpa	6	4.5 – 6
Young's modulus	E	Gpa	70	60 – 70
Poisson's ratio	v	-	0.22 – 0.24	0.2
Thermal expansion coefficient	αT	10-6/K	9	Class 1: 3.1-4 Class 2: 4.1-5 Class 3: 5.1-6
Specific thermal capacity	c_P	$J \cdot kg^{-1} \cdot K^{-1}$	720	800
Fracture tensile strength	$f_{gl,t}$	Mpa	30 – 35 (45)	v22 – 32
Fracture compressive strength	$f_{gl,c}$	Mpa	300 – 420 (1000)	260 – 350 (500)

Properties of standardized soda-lime and borosilicate glass according to EN572-1:2004, Strength values of standardized float/cast soda-lime and borosilicate glass by CES Edupack Program (Granta Design Limited 2015), Blue variables: the characteristic tensile strength of soda-lime glass is 45 MPa (O' Regan 2014). According to (Saint Gobain 2016; Ashby & Jones 2006; Weller et al. 2008) float soda-lime glass has a compressive strength of 1000 MPa, whereas (Oikonomopoulou et al. 2017b) has conducted experiments with borosilicate glass with a nominal compressive failure stress of 500 MPa. data from (Oikonomopoulou,2019).

3.7 Annealing process

Regardless the manufacturing method in which the glass is created: float, casted or printed, a consistent process must be followed to ensure efficient cooling. This involves transitioning through various characteristic temperature points determined by the glass recipe's composition. Each of these points plays a distinct role in forming the final product, and the cooling rate between them is closely tied to their functions. The key characteristic temperature points are as follows:

• Melting Temperature: Signifying the point where the melt's viscosity allows for homogeneity and fining in a reasonable time (<10^3 Pa).

• Working Point: Denoting the viscosity when the melt is delivered to a processing device (10^3 Pa), allowing it to flow under reasonable stresses while maintaining its shape.

• Softening Point: The temperature at which the glass melt has sufficient viscosity ($10^{6.6}$ Pa) to resist deformation under its own weight.

• Annealing Point & Strain Point: Establishing the annealing temperature range critical for the final quality of the cast glass product. The annealing process, initiated below the softening point, eradicates existing strains. The product is held at the annealing point for adequate time before being slowly cooled to prevent the generation of additional internal residual stresses. Beyond the strain point, stress becomes permanent, and subsequent cooling can occur at a faster yet cautious rate to avoid breakage.

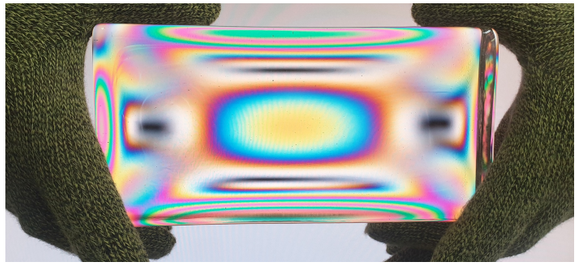


Figure 19: https://www.tu-darmstadt.de/glass-cc/forschung/forschungsgebiete/glass_qualitaet/glas_qualitaet.en.jsp

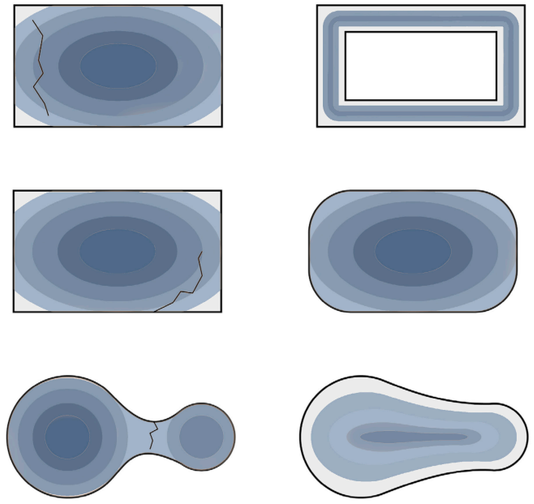


Figure 20: rework of Damen - Three primary design principles to reduce annealing times of glass components. From top to bottom: Weight reduction, rounded shapes, equalised mass distribution

The annealing phase, constituting the most time-consuming part of the cooling process, significantly impacts the overall process duration. This step requires a considerable delay to allow for strain release and prevent residual stress generation, lasting from several days to months or even years (Oikonomopoulou 2019). The duration of the annealing process is strongly linked to the maximum cross-sectional size, favouring smaller annealing times for larger elements with honeycomb structures, as there is simply more glass that can radiate heat outwards. The shape of an element directly correlates with internal stresses during annealing. It is essential to achieve an approximately equal distribution of mass within the component, ensuring that the cross-sectional size is consistent across all parts. Conversely, if fewer sides are exposed to cooling, the annealing time is increased. Furthermore, it is important to maintain a limited range for the section seize in all parts of the component to prevent significant differences in size. This precaution is essential because variations in mass distribution can lead to disparate cooling times, resulting in local stress concentrations that may lead to fractures. as this primary important for casting glass, attention should be given in all forms of glass making to the corners of the shape, as sharp and thin edges tend to cool faster than the rest of the component, increasing the risk of cracks. Consequently, it is advisable to avoid thin elements and sharp edges, opting instead for elements with minimal dimensions and rounded edges. Hence, it can be inferred that to minimize annealing duration and mitigate internal stresses, design guidelines involve steering clear of uneven volume distribution and sharp edges, instead favouring round or ellipsoidal shapes with uniform mass distribution.

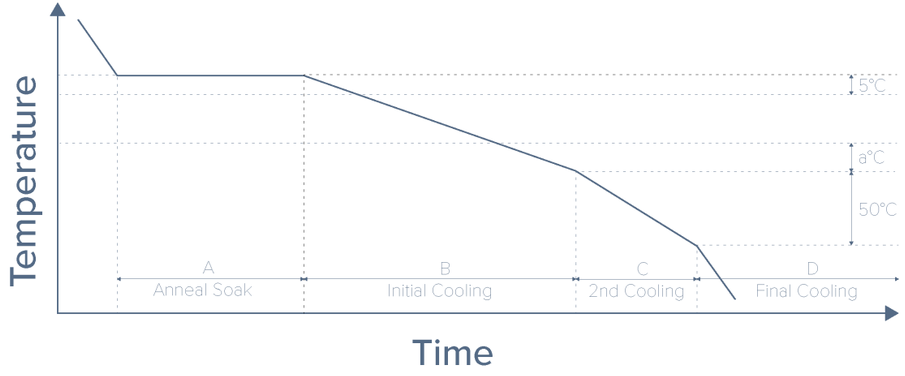


Figure 21: Based on F. Oikonomopoulou (2019)

3.7.1 Approximating the annealing time

The time required for proper annealing depends on various factors, including element size, glass type's thermal properties, and component geometry. Formulas attempt to approximate annealing time, but they cannot consider all factors, such as the number of sides exposed and the furnace's configuration. One formula, adapted by Koniari (2022) from (Shand & Armistead, 1958), calculates the cooling rate (h), incorporating material properties and a shape factor (b). This formula, originally for annealing in a lehr, is versatile across different glass compositions and shapes.

$$h = \frac{\sigma}{\frac{E a_{ex} p c_p}{(1-\mu)\lambda} * d^2 * b} \qquad (1)$$

In this formula, σ represents the maximum permissible permanent stress in the object. The characteristic dimension, denoted as (d), is calculated accordingly. (b) is the shape factor. Additionally, E, α_{ex}, and ρ correspond to the Young's modulus, thermal expansion coefficient, and density of the material, respectively. The other material properties influencing the annealing process include Poisson's ratio (μ), specific heat (c_p), and thermal conductivity (λ). This formula accounts for various material properties and cooling configurations, making it adaptable to different scenarios. The significance of the cross-sectional parameter becomes apparent when considering its impact. If the cross-sectional area doubles, the value of (d) within the formula scales to the power of 2.

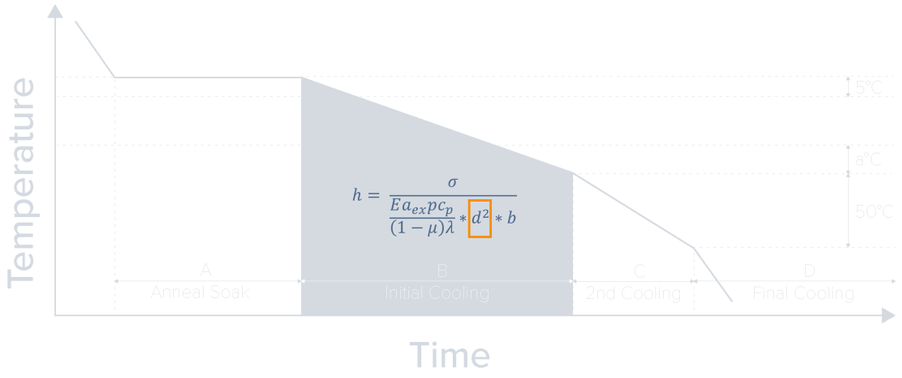


Figure 22: Based on F. Oikonomopoulou (2019) & Koniari (2022)

3.8 Flaws and defects in glass

The flaws and defects in glass are mainly caused during the casting process and belong to the following categories (Bartuška 2008 & Oikonomopoulou, 2019):
- gaseous inhomogeneities (bubbles or seeds)
- crystalline inclusions (stones)
- glassy inhomogeneities (cords or striae).

Nevertheless, the mentioned flaws and defects are the most common shortcomings. however, glass can have various other flaws and defects. Many of these issues can be significantly reduced through well-executed processing procedures. (Louw, 2015):
- Cracks or Checks: Internal cracks, known as checks, can develop during the cooling process if the glass cools unevenly or too rapidly. External cracks can occur due to mechanical stress or impact.
- Devitrification: This defect involves the formation of crystalline structures on the surface of the glass, usually due to improper annealing or reheating. It affects the transparency of the glass.
- Waviness or Distortion: This defect may occur during the manufacturing process, resulting in uneven thickness or a distorted shape in the glass.
- Scratches: Surface scratches can occur during handling, transportation, or processing, affecting the aesthetic appeal of the glass.

The presence of flaws is crucial as it directly impacts the tensile strength of glass. As it is the critical factor in glass design. Flaws act as stress concentrators, elevating local tensile stress values and hastening the likelihood of surpassing allowable limits, ultimately resulting in fracture (Oikonomopoulou, 2019). In contrast, geometric flaws don't impact compressive strength because they don't spread in a compressive stress field (Damen, 2019).

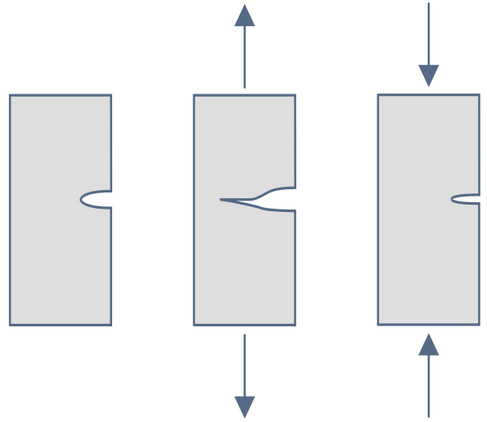


Figure 23: Diagram illustrating the propagation of a crack under tension while closing or remaining unaffected during compression (based on: Damen, 2019).

3.9 Recycling of glass

Glass recycling is the process of gathering discarded glass and transforming it into new glass products. It's a sustainable practice since glass can be recycled endlessly without compromising its quality or purity. Recycling glass helps preserve natural resources, cuts down on energy usage, and lowers greenhouse gas emissions compared to manufacturing glass from raw materials.

Certain countries excel in glass recycling, such as Belgium, the Netherlands, and Sweden, where respectively 96%, 92%, and 90% of container glass is recycled. However, the United States lags behind with only 33% of glass waste being recycled. By implementing different techniques and embracing innovative glass structures, the recycling industry could experience significant growth. Utilizing recycled glass (cullet) for glass production requires far less energy than using raw materials.



Nevertheless, glass recycling does come with challenges, particularly when dealing with materials like PVB found in laminated glass. PVB is a plastic layer that enhances the safety of laminated glass by holding it together upon breakage. Separating the glass from the PVB layer is crucial but can be complex and costly, requiring specialized equipment and processes.

The recycling options for laminated glass may be limited due to these challenges. However, specialized companies and facilities have emerged to tackle this issue, employing advanced technologies to effectively separate the glass from the plastic interlayer. These facilities can recover and recycle both components, reducing waste and environmental impact (Swain et al., 2015).

Despite these obstacles, initiatives are underway to enhance the recycling of windows and laminated glass, aiming to promote sustainability and minimize waste. This involves ongoing research into innovative recycling technologies and efforts to raise awareness and encourage the incorporation of recycled materials in glass manufacturing (Dyer, 2014).

In this master thesis, an assessment reveals the promising potential of utilizing recycled bottle glass for 3D printing applications in glass. This innovation opens up avenues for creating structures where recycled glass serves as a fundamental material, thus circumventing the need for gathering raw materials and significantly reducing emissions. This advancement holds the promise of not only environmental sustainability but also the efficient utilization of resources in manufacturing processes. By leveraging recycled cullets as a base material, we pave the way for a more eco-conscious approach to construction and fabrication methods.

Figure 24: https://cen.acs.org/materials/inorganic-chemistry/glass-recycling-US-broken/97





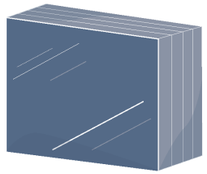
04

Fabrication of glass

the different fabrication methods for building components with topology optimization shapes made of glass.

4.1 Fabrication of glass

The subsequent chapter introduces three primary methods for crafting free-form architectural glass structures. Beginning with float glass, the predominant method of glass production, the research delves into two main approaches to cutting float glass panes. Subsequently, to cast glass, where recent research has notably enhanced both the possibilities and quality of glass components. The most comprehensive investigation then focuses on 3D printing, representing the central emphasis of this overall research endeavour. Following a comparative analysis, conclusions will be drawn regarding which fabrication method is preferable for various approaches to creating topology-optimized glass structures.









4.2 Float glass

As previously mentioned, the introduction of Pilkington's float glass manufacturing method in 1959 marked a transformative milestone, revolutionizing the application of glass in construction, especially in the creation of all-glass building envelopes and structural glass facades. Presently, the float glass technique accounts for 90% of flat glass production (Bourhuis, 2014). As of 2022, there are 320 flat glass manufacturers operating on 560 sites, collectively producing approximately 106 million tons of flat glass (82nd Conference on Glass Problems, Volume 270, 2022).

The shaping of free form float glass planes can be accomplished through Abrasive Water Jet (AWJ) cutting or the use of laser cutting, which integrates subtractive and additive manufacturing techniques. The lamination of AWJ-cut / Laser-cut elements involves three primary stages: float glass production, AWJ /laser - cutting, and lamination.

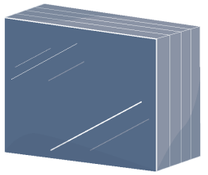


Figure 25: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass

4.2.1 Float glass production

The float glass production process commences with a glass batch, typically composed of sand, lime, dolomite, soda, and sulfate, with the possible addition of glass cullet to expedite melting. The batch undergoes melting in a furnace at temperatures ranging from 1550 to 1600°C, and the molten glass is then poured onto a float tank containing molten tin. The glass spreads evenly, forming a flat sheet, which is subsequently annealed and gradually cooled in a lehr. Following inspection and trimming, the glass is ready for use without the need for grinding or polishing (Saint-Gobain, n.d.), resulting in significant time and cost savings compared to casted glass.

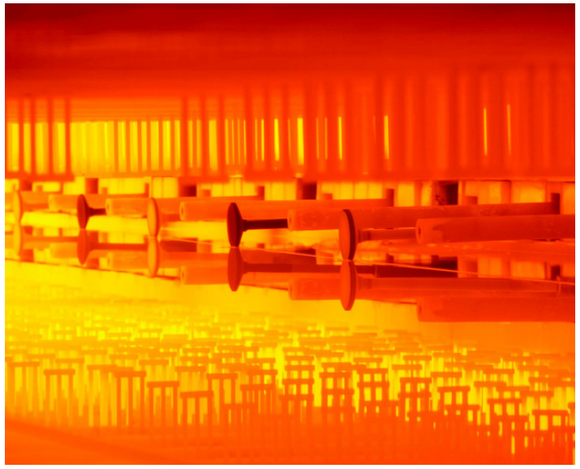


figure 26: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass

4.2.2 Cutting of float glass

AWJ cutting harnesses the power of high-pressure water jets infused with abrasive materials (such as garnet or aluminium oxide), deftly directed by computerized heads and nozzles. Typically, glass undergoes AWJ cutting prior to lamination, apart from tempered glass, which proves unsuitable for this method. The incorporation of 5-axis AWJ machines elevates precision, cutting speed, versatility, and diminishes the kerf angle (Jet Edge, n.d.).
In contrast, laser cutting unfolds its own narrative. A concentrated laser beam is slicing through the glass. This intense beam, directed onto the glass surface, induces the glass to melt or vaporize along the designated cutting line. In the case of both AWJ and laser-cut float glass panels, the outcome is an immaculate, flawless cut, boasting superior edge quality compared to conventional mechanical tools. (Zhimalov et al., 2006; Shivaji Rao & Satyanarayana, 2020).

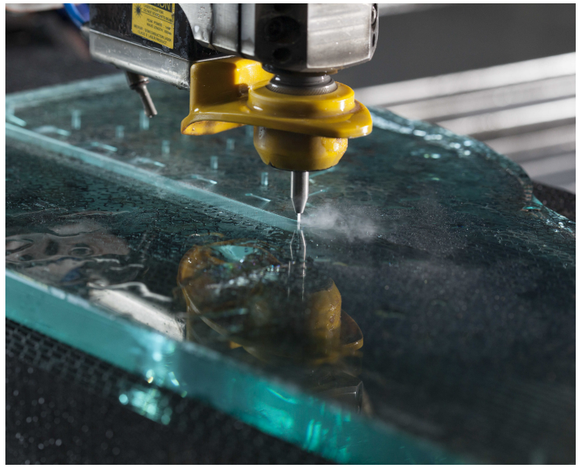


figure 27: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass

Various laser cutting principles can be applied, but when it comes to cutting glass, CO2 lasers are the most prevalent. This is attributed to their cost-effectiveness and the capability to cut thicker glass panes. CO2 lasers operate on the principle of amplifying light in the infrared spectrum using a gas mixture, primarily consisting of carbon dioxide, nitrogen, and sometimes helium. When an electric current is applied to the gas mixture, it produces a high-energy laser beam. When cutting through glass, the laser cutting process produces heat. To maintain the structural integrity of the cut edges, it is sometimes necessary to revisit the cutting line multiple times or cool the area, often with water, after the laser has been applied (Kang et al., 2002).
Yet, one of the main challenges with float glass production is achieving a high-quality finish on the cut edges of the glass sheets. When the glass is cut, it creates a rough edge that requires polishing to achieve the desired smoothness and clarity. Moreover, the precision required in the polishing process adds complexity to the production line, as even minor imperfections can compromise the structural integrity and visual appeal of the glass. Achieving uniformity in edge quality across large quantities of glass sheets demands attention to detail and precize machinery. Enhancing the efficiency and consistency of edge polishing remains a focus point in the refinement of float glass manufacturing techniques.

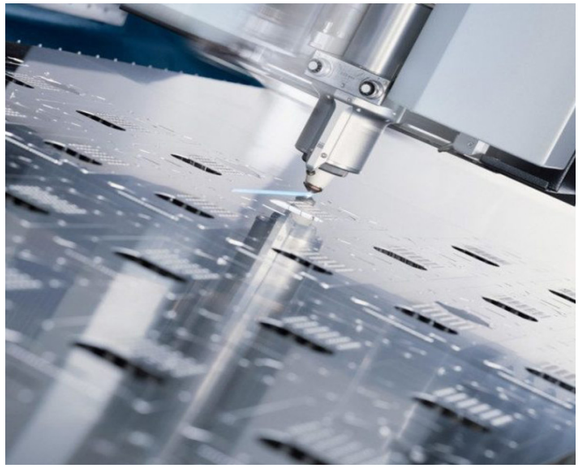
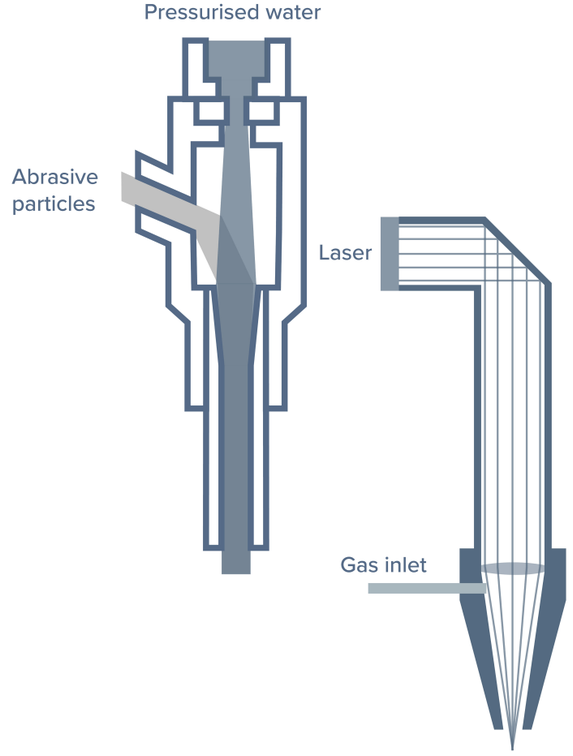


figure 28: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass



4.2.3 Lamination

In the lamination phase, polyvinyl butyral (PVB) serves as the most common interlayer due to its high light transmittance, mechanical strength, good glass adhesion, and cost- effectiveness. Alternative interlayers such as SentryGlass® and polyurethane may be chosen based on design preferences. Glass plies, along with interlayers, are stacked, heated, pressed to remove air, and sealed in an autoclave (Yang, 2021). The more interlayers the harder it is to properly laminate them. Laminated components demonstrate documented mechanical properties and failure behaviour, ensuring enhanced safety and predictable post-failure responses. Transparency varies with layer orientation and viewing angles, while the high finishing quality facilitates seamless connections with other glass or structural elements.

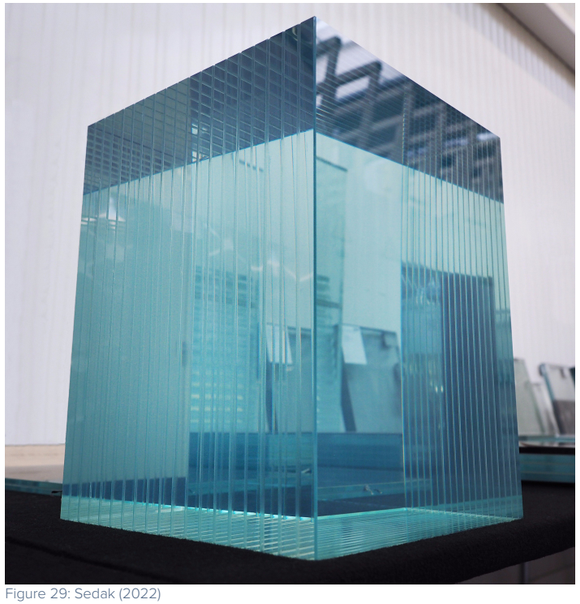


Figure 29: Sedak (2022)

4.2.4 Structural considerations

Nevertheless, challenges persist in the sustainability of laminated building components. Recycling is complex due to interlayers, and AWJ / Laser cutting generates substantial glass waste. A significant challenge in employing AWJ / Laser - cutting with float glass for 3D glass objects is the restriction on lamination layers within structural applications. Research indicates that industrially fabricated examples often limited to 5-6 layers, due to practical constraints associated with the autoclave lamination process. Issues such as bubble formation due to uneven pressure can compromise optical quality and escalate costs (Eckersley O'Callaghan, 2018). Despite glass panes being able to stretch over 24m in length, their maximum standardized thickness in float production does not exceed 25mm (Oikonomopoulou et al., 2023). In various instances, such as with the Glass Sphinx and the vertically layered laminate house, the decision was made to forego lamination.

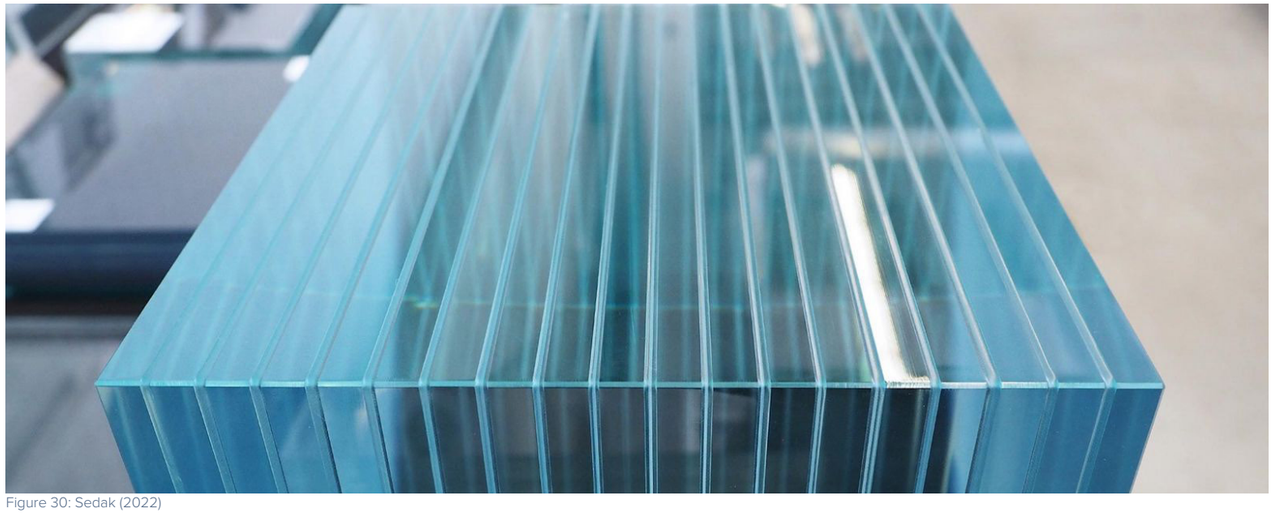


Figure 30: Sedak (2022)

This choice extended not only to conventional adhesive application but also to structural glass adhesives, as suitable options are yet to be developed for effectively bonding large surfaces of glass arranged in stacked configurations. In the case of the Glass Angel and the Glass Sphinx, adhesive tapes were employed, while for the laminate house, sets of 10 were pre-adhered on-site using a 2-component silicon compound. Subsequently, these tapes were installed and secured with a silicone-based sealant (Oikonomopoulou et al., 2023). Despite the effert for alternatives for conventional adhesives, fractures emerged in the sphinx. As a precautionary measure, the sculpture was cordoned off for safety reasons. Subsequently, during a press briefing, it was disclosed that the statue has reached a state of irreparability, and needed to be demolished (L1, n.d.).



Figure 31: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass



figure 32: https://www.jockimo.com/about/jockimo-blog/what-you-need-to-know-about-tempered-glass

4.2.4 Structural considerations

Yet, the recent investigation carried out by Sedak unveils the potential of an 18-layer laminated float glass plane, each layer measuring 15mm in thickness and bonded with a SentryGlass interlayer. This laminated safety glass, approximately 30cm thick and weighing 3.6 tons, exhibits remarkable strength, with each individual glass pane boasting a strength of at least 160 MPa (Sedak, 2019). This giant glass piece, finds its purpose in the construction of a highly pressure-resistant underwater yacht viewing window, featuring a 3.6-meter diameter. While Sedak has undertaken a lamination process involving 19 layers, it is essential to note that this configuration has solely served as a display model and lacks structural functionality. While safety glass is typically standardized with 2 to 5 layers, it is possible to extend the dimensions to include an expansive glass pane measuring 3.6m x 20m (Sedak, 2021).

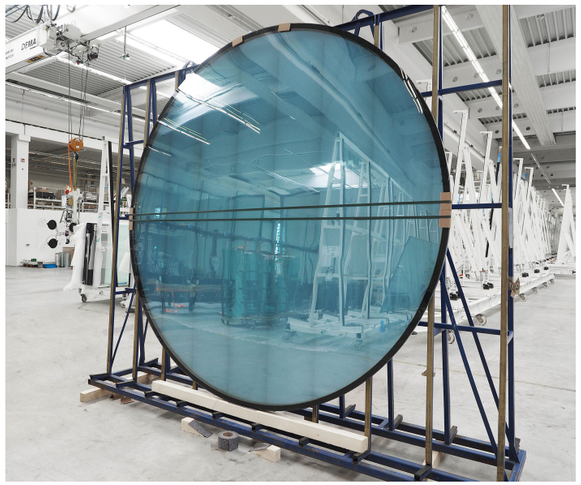


figure 33: Sedak (2022)

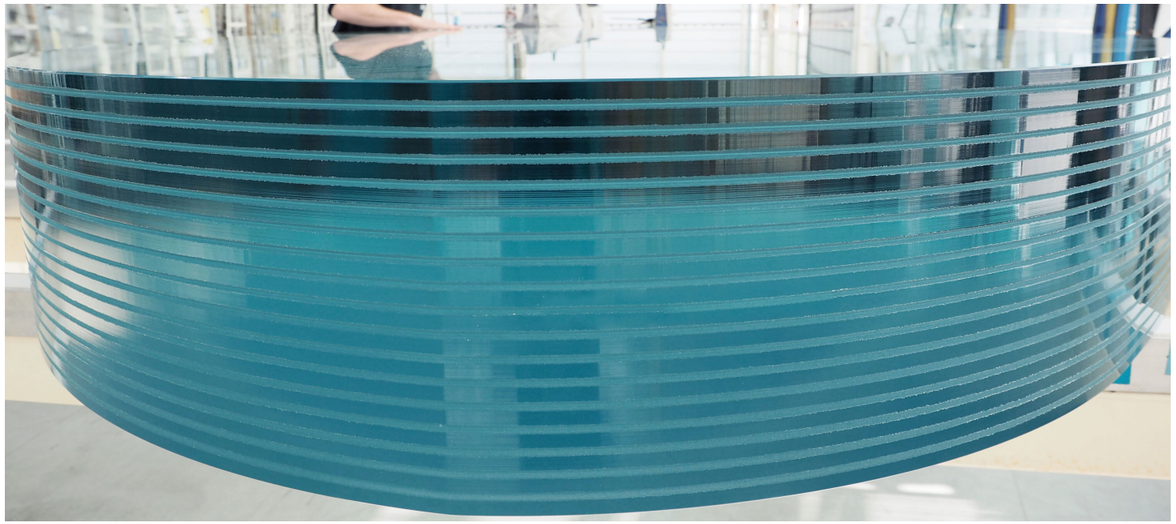


figure 34: Sedak (2022)



Figure 35: https://architizer.com/blog/inspiration/collections/curved-glass-facades/

4.3 Casting

Cast glass, the oldest method of glassmaking, has a rich history dating back to more than 2000 B.C. in Mesopotamia, where beads and small objects made of molten glass were cast in molds (The Corning Museum of Glass, 2001).

The art of casting glass unfolds through two distinct approaches: primary casting, originating from raw materials, and secondary casting, involving the reheating of solid pieces (Oikonomopoulou, 2019). These methods demand different procedures.

Primary casting commences with the creation of a glass melt from raw ingredients in a furnace. The molten glass is then poured into a mold and subjected to the annealing process in a second furnace, making it particularly favourable for large-scale production.
Secondary casting entails creating a glass melt by reheating solid glass pieces like sheets, grains, or powder. Both melting and annealing occur within a single kiln, with glass pieces placed in a holder to allow molten glass to flow into molds when adequately heated. The use of pre-formed solid pieces requires lower temperatures, and the convenience of using only one furnace makes this method the preferred choice for crafting customized components (Oikonomopoulou, 2019).



4.3.1 Moulds

In the realm of kiln-cast glass, another factor to consider is the mold choice. The surface of the mold in contact with the glass can impact roughness and accuracy. Although open and pressed molds, typically made from steel or graphite, achieve high surface quality and accuracy, their cost makes them more suitable for large-scale production and permanent nature with using the hot pouring method (Koopman, 2021; Oikonomopoulou, 2019; Bhatia, et al., 2020). Despite the superior surface quality of these molds compared to disposable molds, dealing with high geometrical complexity can pose challenges.

A comprehensive exploration reveals several advantages in favour of disposable molds over permanent steel molds for crafting customized kiln-cast glass. Steel or graphite molds, due to their expense, require advanced CNC methods for creating custom shapes (Bhatia et al., 2020; Veer, et al., 2018). In contrast, disposable molds, such as those made of silica plaster through the investment casting/lost wax technique and 3D printed sand molds (3DPSM) produced with binder jetting, need post-processing for optical transparency and satisfactory surface quality (Oikonomopoulou, 2019; Bhatia et al., 2020).

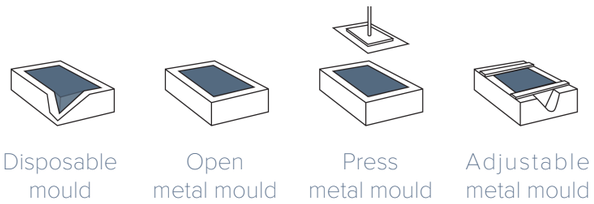


figure 36: Rework off F. Oikonomopoulou (2019)

4.3.1 Sand Molds

Traditional sand mold techniques involve imprinting two positive halves on a sand bed, generating two negatives of the object. These parts are then combined, along with vents and a pouring basin, to create the final mold.
Binder jetting, classified as an additive manufacturing method by ASTM, digitally slices a 3D model into layers. A computer-controlled head selectively applies a liquid binder to layers of fine sand, producing the mold by applying binder only to the negative of the object. Microwave drying, taking 15-45 minutes, efficiently dries the mold from the inside out. Excess sand removal follows through automated and manual processes, and the molds are inspected and stored in a dry environment, with excess sand reused for future molds.
3DPSM offers advantages such as low manufacturing costs, high precision, and size accuracy. The relatively short production time and potential reuse or recycling of recovered sand contribute to a more circular fabrication cycle (Giesecke & Dillenburger, 2022).
However, 3DPSM has limitations. Fully closed cavities are impossible due to the need for material removal after casting. Adequate wall thickness is crucial to withstand hydrostatic pressure, and avoiding sharp corners is essential to prevent air bubbles or unfilled volumes during casting. Although not technically required, splitting the mold aids in releasing the object after casting and ensures a homogeneous application of surface coatings. Surface coatings become necessary due to the rough surface of the mold and the tendency of sand particles to merge with the glass surface after casting (Giesecke & Dillenburger, 2022; Oikonomopoulou, 2019)
Glass casting within 3D-printed sand molds has undergone examination in the MSc theses of Bhatia (2019), Damen (2019) & Ioannidis (2023). Ioannidis explored various coatings to enhance surface quality, but despite encouraging results, considerable manual post- processing, particularly polishing, was required to address surface roughness. Additional tests with different coatings were conducted by Giesecke & Dillenburger (2022). However, in
this case the most promising coatings were tested for foundry casting (hot pouring) and not specifically for kiln casting. The structural performance of the produced elements has yet to undergo testing. As demonstrated in Ioannidis's recent research, inorganic binders and coatings contribute to enhancing the immediate surface quality and transparency of glass. However, achieving a flawless outcome for the glass-cast object still necessitates thorough cleaning and polishing (Ioannidis, 2023).



figure 37: Thesis from Menandros Ioannidis

4.3.2 Lost-Wax technique

In this process, a positive wax model is created and surrounded by clay, forming a mold that, once fired, leaves a negative imprint. Post-processing and cleaning are necessary to enhance the surface quality of the cast element due to the nature of the mold material.
Recent advancements explore the use of 3D printing with wax-based filament, such as FDM (Fused Deposition Modelling), offering advantages in accuracy and computational workflow flexibility. SLA (Stereo-Lithography Apparatus) method, utilizing liquid photopolymer resin cured by a laser system, further enhances precision and surface definition.
However, limitations arise with common investment casting materials like silica plaster, which has low precision and a relatively low maximum firing temperature. This method is recommended specifically for kiln casting due to the brittleness of the mold material. Challenges include the need for open cavities and consideration of edges radii for mold material removal after casting (Oikonomopoulou, Bhatia, et al., 2020).
Applications of this method have been tested, including the fabrication of a complex glass dinosaur shape. While the accuracy of the FDM printer is lower, the method offers insights for printer setup and optimization. The process involves 3D printing the wax positive, creating silica plaster molds, de-waxing, glass placement, and kiln casting, requiring multiple times oven use and extensive post-processing for surface quality improvement (Ioannidis, 2023). In summary, kiln casting with the lost wax technique or 3D printed wax-based filament and SLA presents a historic yet evolving method with advantages in accuracy and flexibility but requires careful consideration of material limitations and involves intricate post- processing steps for optimal results (Ioannidis, 2023 & Giesecke & Dillenburger 2022).

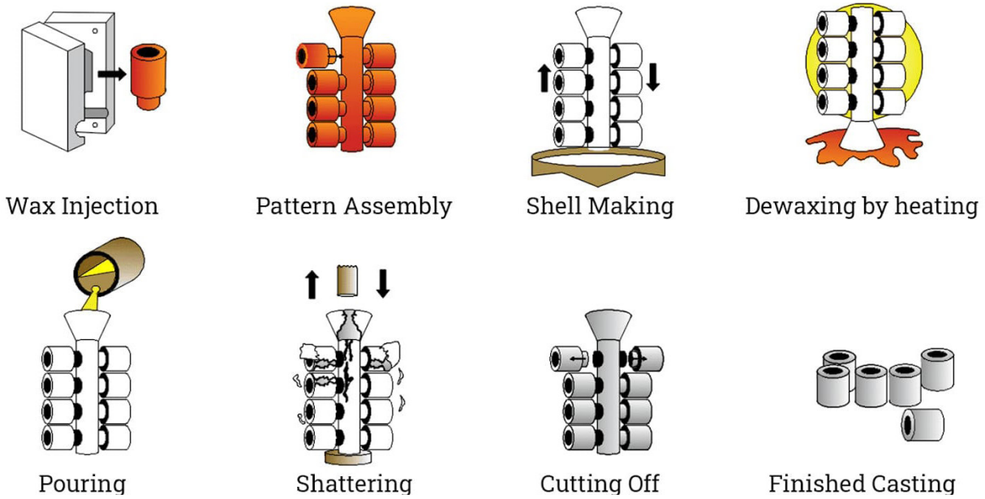


Figure 38: https://www.iqsdirectory.com/articles/investment-casting/lost-wax-casting.html



My experience with the lost-wax technique as a final protoype for this master thesis

4.4 3D printing of glass

Over the past 30 years, 3D printing has rapidly evolved, offering advantages like material savings, high production efficiency, and low cost. Technologies such as Fused Deposition Modelling (FDM), Stereolithography (SLA), Selective Laser Sintering/Melting (SLS/SLM), and Digital Light Processing (DLP) are widely used in diverse fields such as industrial design, sculpture, clothing, automotive, construction, and aerospace (Zhang et al. 2021).
While mainstream 3D printing materials include metals, resins, plastics, and ceramics, the application of glass materials in 3D printing has been demonstrated on a laboratory scale. However, large-scale commercial use of 3D printed glass materials remains elusive in the near future. This is despite glass being widely utilized in our daily lives due to its excellent optical properties, mechanical characteristics, thermal stability, and electrical/thermal insulation (Zhang et al., 2020).
Despite the promising future of Additive Manufacturing (AM), specific design considerations must be addressed during the process planning phase to ensure the desired product quality. Key factors include part orientation, slicing strategy, overhang angle, support generation, and toolpath planning. Neglecting these parameters can result in surface defects, prolonged printing time, material wastage, or even failure of the fabrication process. Various research perspectives, such as adaptive and non-planar slicing, adaptive nozzle speed, support optimization, printers with increased degrees of freedom, mechanical improvement, and orientation optimization, have been explored to optimize fabrication efficiency and quality.

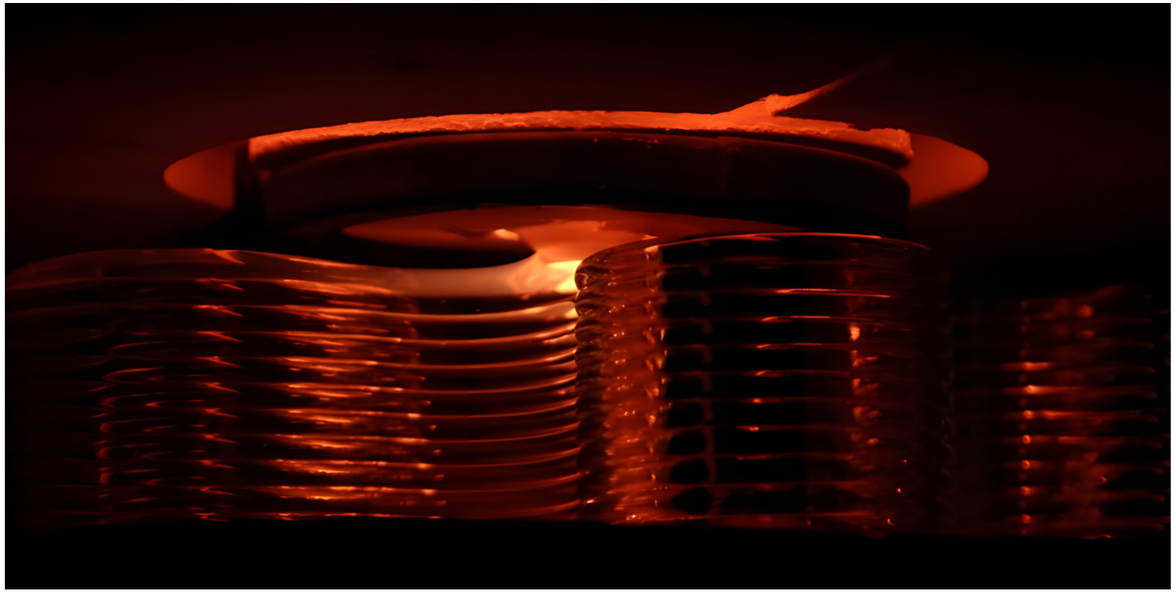
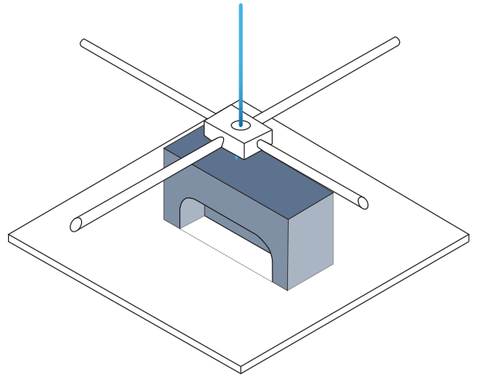


figure 39: G3DP2 oxman.com

4.4.1 Fused Deposition Modelling:

Fused Deposition Modelling (FDM), invented and patented by Stratasys (Zhang et al., 2021), operates by feeding continuous filament through a computer-controlled heated nozzle. The nozzle deposits material layer by layer on the printer's bed within the printable volume, allowing movement of either the nozzle or the plate. This layering process is repeated to create a 3D product. Glass viscosity, dependent on temperature, is critical in FDM. Maintaining the nozzle's raw material temperature during FDM printing is crucial. High temperatures may reduce print accuracy and cause deformation, while low temperatures can lead to nozzle blockage and printing failure (Choi et al., 2016).
Micron3DP and MIT developed a system, G3DP, for 3D printing silica glass, extensively detailed in Klein's research (2015). The G3DP2, an improved iteration, enhanced print accuracy, speed, and printable volume (Inamura, 2017b).



FDM printer

Adapting FDM for glass printing required a controlled annealing process. The system comprises two main elements: the Kiln cartridge and the print annealer. Glass culets in the kiln cartridge melt inside a crucible, flowing through the nozzle based on gravity. The team experimented with nozzle sizes and frequent refilling due to pressure affecting the flow rate. The print annealer controls temperature to prevent uneven cooling, crucial for stability and solidification. The temperature gradient between the last five printed layers, around 350°C, is vital to prevent crack propagation and uneven stress concentration. The Build Chamber supplies 12kW and operates at 480°C during printing. Upon completion, the glass flow stops, and printed objects are moved to a separate annealing chamber (Inamura et al., 2018).
This method exhibits potential for substantial-sized objects, such as glass columns in Project Glass II (Inamura, 2017a). Improvements in printing speed and accuracy have been achieved, and preliminary structural testing provides insights into structural values, safety, and surface quality.

Other FDM tests have also been abducted with the use of other glass type, such as chalcogenide glass, which reduced the temperature significantly (Baudet et al. 2019). Their investigation into the physicochemical properties revealed that the chemical and thermal characteristics closely resembled those of the original glass. The choice of the As40S60 chalcogenide glass was based on its advantageous attributes, including a low glass transition temperature (Tg=188°C) and straightforward synthesis and processing. Utilizing a commercial 3D printer, the glass was extruded at a temperature approximately 140°C higher than its glass transition temperature. Notably, they demonstrated the efficacy of 3D printing in creating intricate optical components from chalcogenide

4.4.1 Fused Deposition Modelling

glass, including multi-material fibre preforms featuring geometries and structures unattainable through conventional methods (Baudet et al. 2019).
The Ningbo Institute of Materials Technique and Engineering team developed a glass melting deposition molding technique, overcoming challenges with high operating temperatures and poor process stability (Engineering Niomta, 2015). They engineered a high-temperature-resistant, high-precision Al2O3 ceramic nozzle and tabletop-scale printing equipment tailored for the Fused Deposition Modelling (FDM) process. This innovative system enabled the successful printing of diverse micron-scale glass components, each with a characteristic size of under 50 mm and a linear printing speed reaching up to 10 mm/s (Engineering Niomta, 2015).

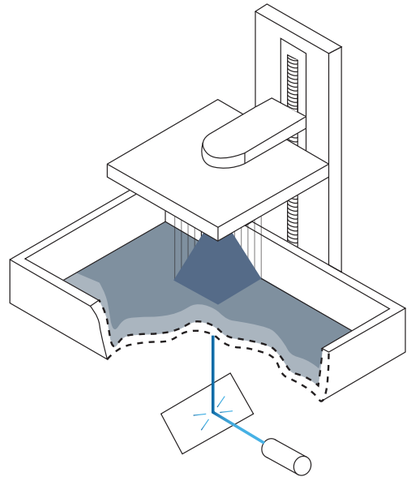
However, their research has limitations; the Maple 2 appears to be the first consumer market glass printer. The procedure involves crafting a glass rod initially, which is subsequently placed into the printer. Within the printer, the rod undergoes heating and is transformed into the desired shape. Typically, these rods have a diameter ranging from 3.5 to 6 mm and are produced with a maximum temperature of 1287°C (Maple Glass Printing Ltd., n.d.).

The maximum nozzle size is 3mm, in contrast to the G3DP2, which has an 11.5mm nozzle. Currently, the extruded glass layer height utilized by the printer is 1mm, with plans to decrease it in the future. But the current Maple 3 is capable to print object of 150mm in height with an overall build volume of roughly 10L. The Maple 3 is currently available for $45,000, excluding the essential $15,000 automated glass filament production unit required for crafting the filament rods used by the printer. The primary focus of the Maple 3 is on the design of the annealer, where the maximum temperature has been reduced, and a 'viewing window' has been incorporated into the front of the annealing chamber (Maple Glass Printing Ltd., n.d.).

Despite FDM's ability to manufacture glass devices with complex shapes and properties, challenges persist. Operating temperature significantly influences the FDM process, impacting discharge form and print precision. Printed samples often exhibit low precision and a "step effect" due to layered deposition, resulting in substandard quality, Mechanical strength, may not meet industrial standards (Zhang et al., 2021).

4.4.2 Stereolithography and digital light processing

Stereolithography (SLA) and digital light processing (DLP) techniques are both based on UV curing of photosensitive resins; however, these techniques differ in the process of printing 3D objects. employs a projector to solidify layers of light-sensitive polymer liquids in succession, projecting the 3D model's shape onto each layer and gradually building up the object. In contrast, SLA uses an ultraviolet laser to solidify photosensitive polymer liquids layer by layer. Unlike DLP, SLA cures the resin point-by-point, scanning the laser to form each layer. Both techniques involve elevating the printing platform after each layer is completed until the entire 3D object is produced.



SLA printing

SLA and DLP techniques can be used to produce glass parts with a small size, high precision, excellent permeability, and superior mechanical properties. Therefore, it is also considered to be the most promising technique for 3D glass precision printing.
The critical factor for utilizing SLA/DLP techniques in printing glass devices lies in the formulation of the slurry. It is essential for the slurry to possess appropriate rheological properties and a high concentration of SiO2 nanoparticles. Presently, only a limited number of resin pastes meet these specific criteria. Moreover, the sole material available for printing is SiO2, thereby constraining the application of SLA/DLP techniques in the 3D printing of glasses. In 2017, Kotz et al. successfully created 3D glass objects using the SLA technique. Initially, they dispersed SiO2 nanoparticles, averaging 40 nm in diameter, within a solution comprising hydroxyethyl methacrylate monomers to produce a glass slurry. Subsequently, a green part was printed using an SLA 3D printing device. The green part underwent a heat treatment process for debinding and densification, resulting in the formation of a transparent glass sample without pores or fissures at a temperature of 1300°C (Kotz et al., 2017).

4.4.2 Stereolithography and digital light processing

However, for the same 3D model file, DLP requires 60 min, while SLA requires 9 h to finish the printing process (Zhang et al., 2021). Recent studies employing DLP technology and phase-changing resins show promise in advancing the 3D printing of glass components with multi-oxide compositions. To achieve transparency, these components undergo pyrolysis, followed by fabrication methods such as sintering, which significantly reduces the object's volume by 60% (Moore et al., 2020).

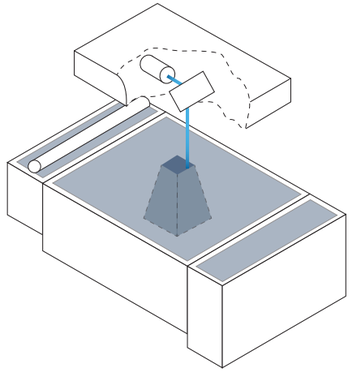
Despite the evident benefits of the DLP method, it currently falls short of surpassing the SLA method due to limitations in maximum printing size. The pixelated projection in the DLP process restricts the print size based on the capabilities of the projector or liquid crystal display screen. In contrast, SLA 3D printers operate differently, enabling the production of objects as large as those created by FDM 3D printing (Zhang et al., 2021).

Computed Axial Lithography (CAL) stands out as an innovative 3D printing method that leverages tomography and Continuous Liquid Interface Production (CLIP) principles. Jointly developed by the University of California, Berkeley, and Albert Ludwig University of Freiburg in Germany, this technique facilitates the fabrication of highly detailed microstructures boasting superior optical quality, design adaptability, and strength. In CAL, the procedure unfolds by crafting 3D objects layer by layer through the targeted solidification of a liquid resin. Tomography, a process that captures 2D X-ray images from various angles to construct a 3D representation, is seamlessly integrated into CAL. The 2D images are projected utilizing a DLP projector. The size of the resultant examples remains presently constrained (Murphy et al., 2022).

4.4.3 Selective laser sintering and Selective laser Melting

The SLM (Selective Laser Melting) method employs a high-power laser beam directed at powdered material, selectively consolidating and melting it. Glass powder is applied to the substrate and selectively melted in successive layers to form the desired object. In contrast, SLS (Selective Laser Sintering) doesn't melt the material but sinters it, resulting in a distinct molecular structure and porosity. Post-printing, the as-printed parts undergo heat treatment for finalization, causing noticeable shrinkage (Zhang et al. 2021).

Klocke et al. utilized Selective Laser Sintering (SLS) to process borosilicate glass powder, studying parameters' effects on sample density and surface roughness. Challenges in 3D printing glass arise from its high melting points and low thermal expansion coefficients, causing cracks and pores (Klocke et al., 2004). Luo et al. applied SLM to print soda-lime silicate glass, optimizing parameters for transparent glass objects (Luo et al., 2016). Despite challenges, SLM for 3D printing of glass presents a promising method for creating objects with diverse shapes, suitable for applications such as glass artworks. Precision improvement is essential for producing accurate optical devices (Zhang et al. 2021).



SLS printing

4.4.4 Direct Ink Writing

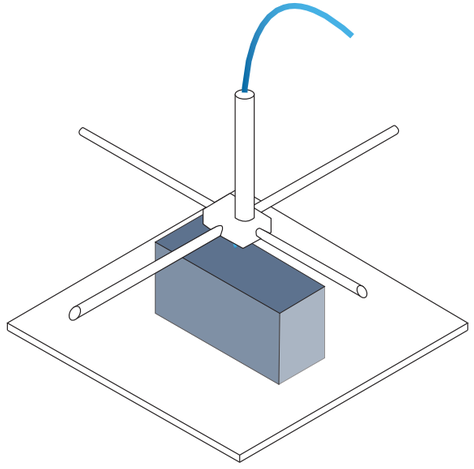
Direct Ink Writing (DIW) is an emerging 3D printing method that involves extruding a semi-solid ink material with shear-thinning properties from a print nozzle. The ink is then rapidly solidified through gelation, evaporation, or temperature-induced phase change. The green body produced goes through heat treatment and sintering processes to create glass components. The critical step in printing glass devices using DIW is the synthesis of ink with dispersed glass powders, necessitating good rheological and shaping properties, as well as pore-free gelation during drying. Unlike Fused Deposition Modelling (FDM), DIW doesn't require high temperatures for glass printing.

Nguyen et al. used silica nanoparticles, tetraglyme, and hydroxyl-terminated dimethyl siloxane to create an organic paste, demonstrating DIW's capability to print glass devices with diverse microstructures (Nguyen et al., 2017).

Sasan et al. showcased the 3D printing of Germania-silica glasses with excellent optical quality. They described a method involving colloidal germanium dioxide raw material suitable for mixing with various silica colloids to change the refractive index. The DIW system operates at moderate temperatures, offering transparency, zero porosity, and sub-millimetre size, without requiring heating, laser, or ultraviolet radiation. However, limitations include the nozzle diameter affecting accuracy, the nozzle path restricting pattern deposition, and the extended drying time for printed embryos, hindering the widespread application of the DIW 3D printing technique (Sasan et al., 2020).

Vat photopolymerization and direct ink writing have advanced high-resolution sintered glass printing. However, drawbacks include a lack of transparency, mechanical limitations, and a slow printing rate of only 0.02 kg/h. Significant shrinkage and distortion during the removal of the binder in the sintering process pose challenges for scaling this indirect technology (Inamura et al., 2018).

To conclude, this method is subject to the limitations of extrusion-based technologies. The accuracy is confined by the nozzle size, the shape freedom is limited by the path of the nozzle and drying time is long (Moore et al., 2020; Zhang et al., 2021).



DIW printing

4.4.5 Comparison 3D printing of glass

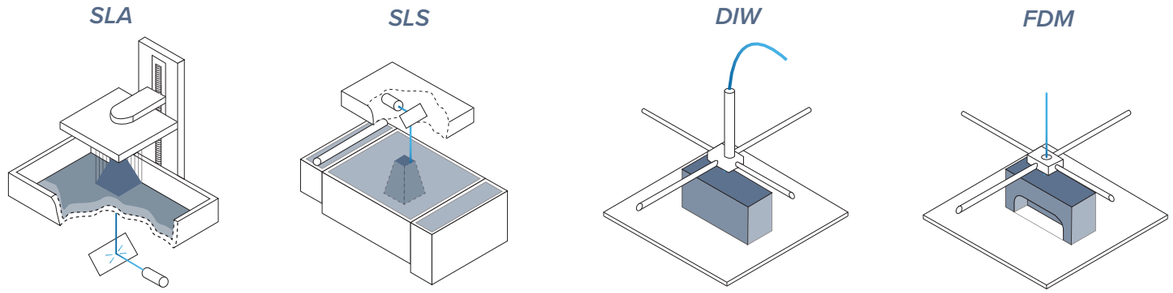
Various methods are available for 3D printing glass, each with unique advantages and drawbacks. Fused Deposition Modeling (FDM) involves laying continuous filament through a heated nozzle to maintain glass viscosity. G3DP(2), developed by MIT, improves print accuracy and volume but requires controlled annealing.

Selective Laser Melting (SLM) and Selective Laser Sintering (SLS) offer precision but struggle with glass's high melting points and shrinkage during finalization.

Direct Ink Writing (DIW) doesn't need high temperatures but necessitates ink synthesis with dispersed glass powders, facing limitations such as nozzle diameter and extended drying times despite achieving diverse microstructures.

Stereolithography (SLA) and Digital Light Processing (DLP) boast high precision and mechanical properties but are restricted by print size and material availability.

Therefore to use 3D printed glass on an architectural scale, size and speed of the printing process for large-scale glass products are the most important criteria. Consequently, FDM remains the feasible option for architectural purposes, as it enables precise layering of glass on a scale bigger than a few mm. The translusity lost by the layering of the glass is therefore considerd as an item which can also bring quality to the final outcome of the object. Further research has to be undertaken to show real potential in the other fabrication methods for 3D printing of glass in architecture.



3D printing techniques

Table 1: comparison of 3D printing of glass for architectural structures

	speed	direct	Transparency	temperature (lower is better)	schrinking	Fabrication cost	Size Limitations	Shape Limitations	total
FDM	++	+	-	-	++	+	+	-	++++
SLA / DLP	-	--	++	-	--	-	-	+	----
SLS / SLM	-	-	++	-	--	+	-	+	--
DIW	-	--	+	++	+	-	-	-	--

4.5 Comparison of laminated, casted & printed glass:

Among the discussed methods, AWJ/Laser - cutting and bonding of float glass emerge as the optimal choice for structural performance, leveraging well documented properties of float glass and redundancy in a multi-layered solution. Further enhancements, such as tempering, can improve tensile strength, resulting in a lightweight solution. However, this method faces challenges in optical performance due to layer directionality and has notable limitations in size and shape, with sustainability concerns related to high waste generation and difficult recycling. The performance implications of combining free-form shapes in laminated float glass represent a relatively unexplored domain at present.

Casting glass offers design freedom, with size limitations addressed through optimized annealing time and the use of glass with a low thermal expansion coefficient. Although post-processing is necessary for achieving a smooth surface, the strength values may be slightly lower than those of float glass. This, in combination with the monolithic nature of cast glass, requires the implementation of increased safety factors, resulting in a somewhat heavier structure. Despite these challenges, cast glass demonstrates promise in terms of sustainability, recoverability, and recyclability. However, the extensive preparation and post-processing times for fabricated elements remain significant drawbacks. Additionally, when combined with topology optimization, a suitable manufacturing mould that would allow for high surface quality without post-processing, high dimensional accuracy and low cost is currently not fully developed. 3D printed sand moulds are the most promising direction although a combination of coatings that enable for a smooth, glossy finish quality of the glass as coming out of the mould is still to be developed.

Additive Manufacturing (AM) of glass, while in an experimental phase, holds potential as an environmentally friendly option with minimal waste and reduced post-processing requirements. It is also the best solution for customized components. Although some challenges arise from the manufacturing method that can limit the shape freedom of the generated shape. Challenges such as directional optical performance arising from visible deposited glass layers, continue. Klein et al. noted a limitation associated with the use of small filament size, where light scatters as it traverses the printed object's walls, resulting in increased translucency rather than transparency (Inamura Chikara et al., 2018). It appears that, for topology-optimized glass structures, this latest FDM printer is the most suitable. The printer excels in form freedom, requires minimal post-processing, and once the printing process begins, labour intensity is nearly non-existent. Hence, further exploration of this printer is being undertaken.

Table 2: comparison of addative manufacturing techniques to create unique architectural elements

	Structural perfromance	Structurally homogenious	Transparency	Finishing quality	Waste generation	Recycleability	Fabrication cost	Post processing	Size Limitations	Shape Limitations	total
Laminated	++	-	+	++	--	+	-	+	+	--	+
Casted	+	+	++	--	-	++	+	--	-	+	++
3D printed	-	--	-	+	++	++	+	+	-	+	+++

4.6 FDM-printer G3DP2:

The G3DP2 project has, for the first time, showcased the architectural-scale capabilities of 3D-printed glass, emphasizing its geometric complexity, accuracy, strength, and transparency. This marks a crucial advancement in unlocking the full structural potential of the material (Inamura et al., 2018c). The project which started as a FDM style glass printer, the G3DP, was the first glass printer developed by the Mediated Matter group (Klein et al., 2015). The G3DP2, a second-generation molten glass printer. Its development was centred on creating a system capable of producing glass at an architectural scale. This necessitated enhancements in speed, scale, repeatability, and reliability. These criteria were employed to assess the system's efficacy in comparison to its predecessor, G3DP, and other glass additive manufacturing systems, particularly those discussed by Kotz et al and the commercial work by Micron 3DP (Kotz et al.,2017; Inamura et al., 2018). The G3DP2 has significant speed and scale improvements, crucial for achieving the desired architectural output. Notably, G3DP2's deposition rate is 5.2 kg/hr, surpassing the Kotz system by 260 times and the Micron 3DP system by 104 times. This deposition rate stands out as one of the fastest in additive manufacturing across all materials (Inamura, 2017b).

The G3DP2 platform comprises a two-part modular assembly featuring a stationary three-zone thermal control module and a four-axis motion control module. These systems are driven by a software interface that transforms digital designs into tangible glass objects. The software workflow for the printer includes several phases. Initially, during the design phase, a three-dimensional object is computationally modelled. Subsequently, custom C# modules translate the geometric path into an executable G-Code file. This file is then loaded into the machine which sends commands to individual motors on the X, Y, Z, and A-axis. The motion control system employed by the G3DP2 platform is a meticulously crafted four-axis CNC system. Nonetheless, during the testing of the fourth axis (A) for extrusion, issues surfaced because of jerk-limited accelerations, manifesting in elevated wall thickness at inflection points. Therefore, only 3axis prints were successfully produced (Inamura, 2017b).

The thermal control module encompasses three independently controlled thermal regions working together to process molten glass stock into a stress-free product. The total power consumption of the thermal module is 19 kW at its peak and 15 kW during the normal printing cycle (Inamura et al., 2018).
The material reservoir, storing molten glass at 1090C, allows the glass to flow down through a temperature-controlled nozzle, conditioning it at 915C to achieve a desirable viscosity for printing. Situated beneath the Material Reservoir, the build chamber comprises a custom-machined alumina nozzle and a two-part thermal control system. Serving as a vital link between the printer's thermal and mechanical systems, the alumina nozzle interfaces with the highly viscous molten glass. An issue known as wetting, where molten glass adheres unevenly to the nozzle's exterior face, posed challenges during prior research phases. To minimize wetting impact, the geometry of the nozzle tip angle was increased from 30° to 45°, reducing the contact surface area with molten glass. The nozzle body temperature is regulated by a resistive wire heating system wrapped around a silicon carbide tubular mandrel, the nozzle kiln operates at 800°C, supplying 600 W during printing (Inamura et al., 2018).

Control over the nozzle tip temperature is achieved through propane combustion or compressed air applied through the nozzle gas manifold. The Build Chamber subsystem is designed with an electric kiln featuring a D1100 × W1100 × H500 mm interior volume, full-height double doors in the front, and a W400 × D400 mm opening at the bottom. Operating at 480°C with 12kW power. Enabling the glass to cool and stabilize during printing. The build chamber maintains a consistent temperature, halting the flow of glass while permitting the relaxation of thermal stress. This guarantees appropriate layer adhesion and yields a glass product devoid of stress. The temperature difference between the last 5 layers is 350C. The magnitude of the temperature decrease ensures the existence of a minimum distance before the 'path' returns to its initial location; this distance measures 475mm (Evenline, 2023). The front doors, equipped with double-pane Neo-Ceram windows, enable process monitoring while maintaining the thermal profile (Inamura et al., 2018c).
Upon completion of the G-Code file, the printed object is subjected to a thermal soak at 525°C for 5min, establishing a temperature buffer as it is transferred to an external annealing oven. Although the Build Chamber within the G3DP2 platform has the capability to anneal each print, an external oven is employed to streamline the production cycle.
Ultimately, the halt of glass flow is achieved by lowering the set point of the Nozzle Control subsystem to the idle temperature of 800°C and initiating the release of compressed air through the ring manifold (Inamura et al., 2018c).
The final extruded glass possesses a standardized dimension, measuring 10mm in width with a layer height of 5mm. But for the offset form the nozzle a standard range of 3mm - 6mm is used. Through our processes, we have observed the creation of isotropic glass within the fully fused sections of the layers (Evenline, 2023). Annealing takes place, effectively mitigating internal stress and inducing minor changes in the refractive index.

In terms of standard range specifications, overhangs fall within the 0°- 30° spectrum. The angle is determined between two vertical points along the object path. Measurement occurs from a vertical plane perpendicular to the print path, where 0° signifies a straight-walled object, and 90° represents a horizontal ceiling. For single-line objects in vase mode, the curvature lies within a predefined range based on input values. It is defined as the inverse of the local radius at a point along the tool path in the XY plane of the object. The standard range for curvature (C) is C=1/15 (R=15mm) to C=1/150 (R=150mm). The minimum tool path length for a single layer, before encountering instability in the printing process, is 475mm. This is primarily determined by the heat dissipation capacity before the subsequent layer is applied. By opting for thinner layers, specifically at 3mm, this threshold can be significantly reduced to approximately 100mm (Evenline, 2023).

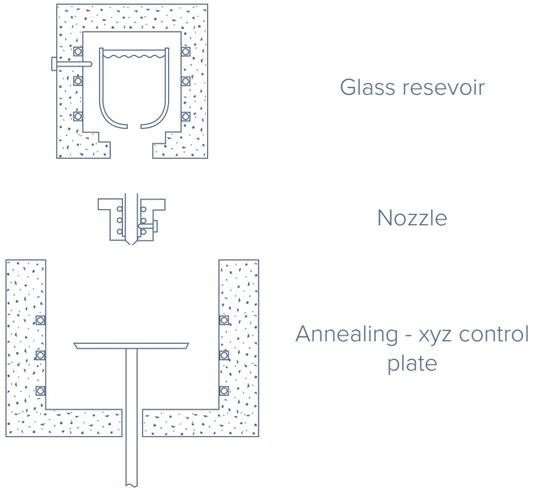


Figure 40: G3DP2 rework by author, copyrights to MIT media lab

4.6 FDM-printer G3DP2:

The G3DP2 demonstrates the most promise for producing architectural-scale objects. Extensive testing has been conducted on the current form and production of printed glass with the G3DP(2). Various shapes and consistencies of the glass inlet have been reconsidered (Inamura, 2017b). Eventually, a method was selected where the glass is poured from another oven where it already has a certain level of viscosity. After allowing the glass to sit in the reservoir for some time to release air bubbles, the printing process can start. Printing is achieved through gravitational pull, although other approaches were considered, the limitations of refilling the glass reservoir led to their dismissal. The glass reservoir can be refilled multiple times during printing; however, due to the small print area (print chamber), it is only necessary to refill the reservoir with a second batch of molten glass at most. The nozzle has been slightly enlarged from the first to the second generation primarily to reduce drag within the nozzle and thereby increase the speed of the printed glass (Inamura, 2017b).

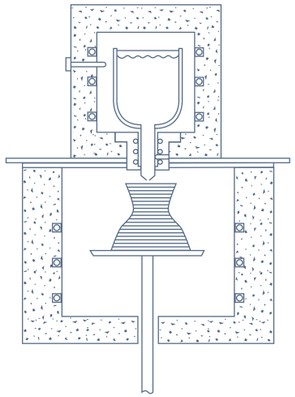


Figure 41: G3DP2 rework by author, copyrights to MIT media lab

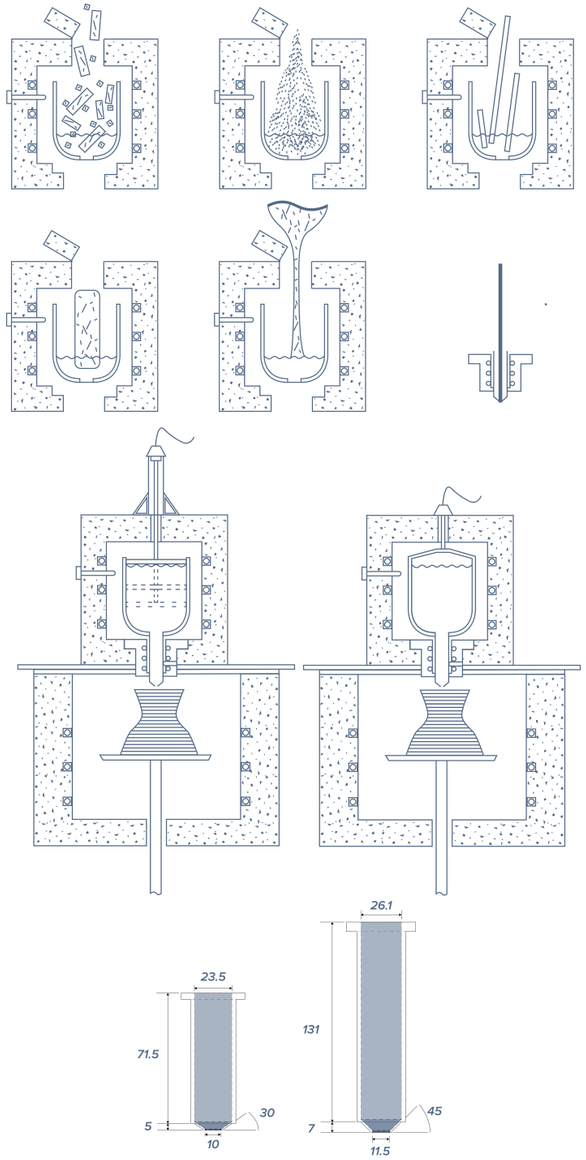


Figure 42: G3DP2 rework by author, copyrights to MIT media lab



Figure 43: G3DP2 oxman.com

4.7 Limitations of glass FDM printing incorperation of the algorithm

Here, the potential and limitations of 3D-printed glass are being considered for integration into the topology optimization algorithm. Given that the FDM method is selected for algorithm generation, two categories of limitations arise: material-specific limitations and printer limitations. In discussions with Kaitlyn Becker (Assistant Professor at MIT) and Daniel Massimino (Researcher at MIT), it was noted that certain printer limitations can be readily addressed with sufficient motivation to invest in their improvement within the context of 3D printing glass via FDM.

The primary limitations, which encompass both material and printer constraints, include the overhang angle of 30 degrees (measured from the vertical axis), print path continuity, path direction, and path length.
Addressing the path length limitations primarily involves printer adjustments. Kaitlyn suggested that these limitations can be mitigated by passing warm air around the freshly printed object, allowing it to release its heat (>600 degrees) more rapidly. The current reduce of temperature difference between the last five layers is approximately 300 degrees Celsius. This can be directly linked to the possibility of radiated heat from the printed layers. While acknowledging that there will always be some temperature differential, Kaitlyn emphasized the potential to shorten the minimum path length at least 475mm.
Concerning path direction limitations, which cannot exceed 150 degrees, expanding the print bed size can easily overcome this challenge. Regarding minimum curvature, which is impacted by nozzle blockage, reinforcing the nozzle can address this issue. The drag of the glass plies a pull on the glass still in the reservoir, thus pressurizing the nozzle. Therefore, these limitations can be easily adjusted accordingly.

The 30-degree overhang angle and path continuity pose more significant material-related challenges. Insufficient adhesion between layers, primarily due to viscosity issues, complicates these aspects. Altering the nozzle angle could potentially address this concern, but it would necessitate a complete redesign of the glass printer.
Moreover, research indicates that interruptions in printing, such as stopping and resuming, negative affect print quality, a phenomenon observed in various materials, including concrete, plastics etc. Therefore, it is (when possible) avoided.

The bead size of the print layers can be readily increased (using a larger nozzle, glass reservoir, and print chamber) once the configuration has been adjusted to accommodate glass printing requirements. As a result, it would be feasible to enhance the glass printing output per hour, although this may come at the expense of accuracy and finer detailing.

In conclusion, the key aspects requiring integration into the topology optimization process are adjustments to the overhang angle and ensuring continuous print paths, which would mitigate interruptions and contribute to improved print quality.

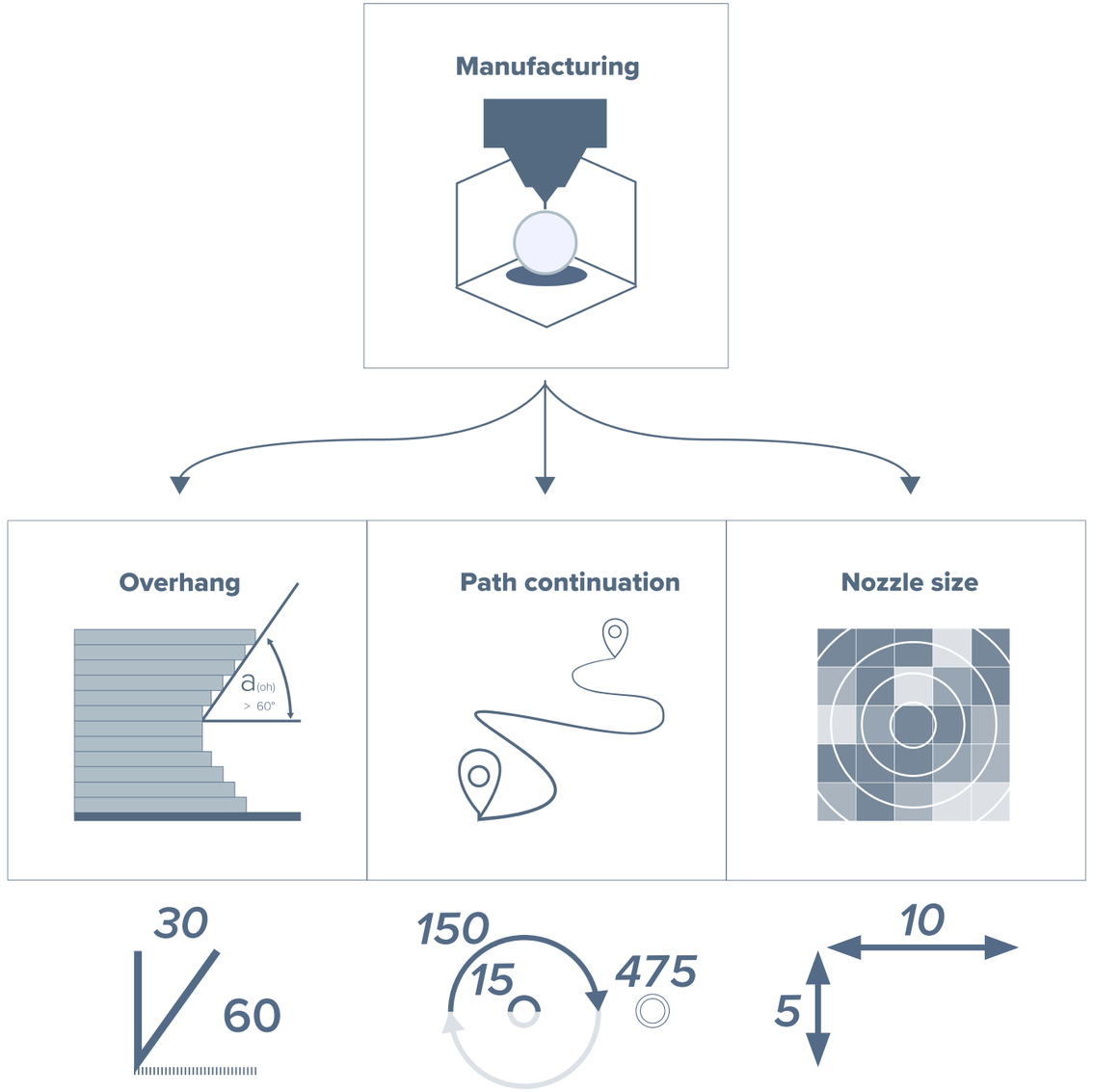


figure: limitations of 3D printed fdm glass algorithm

05

Topology optimization

Algorithms and essentials for topology optimization

5.1 Topology optimization

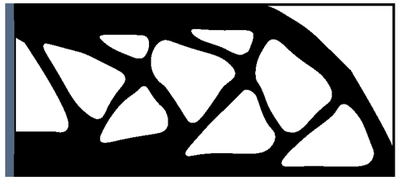
Structural Design Optimization is a discipline within engineering that leverages advanced computational methods to achieve the most efficient material distribution in a structure, given specific design objectives and constraints. The primary goal is to enhance the structural performance while minimizing material usage and overall costs. This field has become increasingly crucial in the construction industry, offering a sophisticated approach to designing structures that are not only structurally robust but also resource efficient.

The optimization process involves complex algorithms and numerical techniques to explore numerous design possibilities. Engineers aim to strike a balance between competing factors, such as strength, stability, and cost-effectiveness, to create structures that meet or exceed performance requirements.

In recent years, the advancements in digital manufacturing technologies have played a pivotal role in the evolution of structural design optimization. These technologies provide the tools and capabilities to translate optimized designs into real-world structures, making it possible to push the boundaries of what can be achieved in terms of efficiency and performance. Key advantages of Additive manufacturing (AM) compared to traditional manufacturing techniques are the enormous geometrical freedom it offers to designers, combined with the fact that costs are nearly insensitive to geometrical complexity (Langelaar, 2016).

As industries increasingly prioritize sustainability and resource efficiency, structural design optimization emerges as a key solution to address these challenges. By harnessing the power of computational tools and digital technologies, engineers can revolutionize the way structures are conceived, designed, and constructed, prelude in a new era of smart and optimized structures. In this context, topology optimization has been universally recognized as a crucial design technology (Langelaar, 2016).





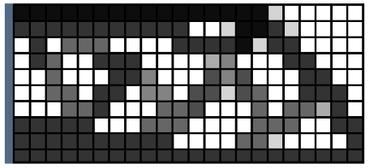
5.2 Structural optimization problems

Structural optimization problems can be classified into three distinct types: size optimization, shape optimization, and topology optimization.
Size optimization involves situations where the problem domain is known and fixed, but the size of the components within that domain is unknown. For instance, in a truss structure, the layout is known, and the optimization focuses on determining the optimal cross-sections of the elements. A commonly utilized algorithm for size optimization is the Genetic Algorithm, which involves the iterative evolution of a population of potential solutions representing various sizes of structural components across multiple generations.
Shape optimization, or morphological optimization, on the other hand, aims to identify the optimal form or contour of a given domain. An example is a plate with a predetermined number of holes, where the boundary and shape of these holes are subject to optimization.
Topology optimization represents the broadest category, encompassing scenarios where the number, location, and shape of holes, along with the connectivity of the domain, are to be determined. For instance, a 2-dimensional domain may undergo topological changes by allowing certain areas to have zero thickness, providing flexibility to determine the number and shape of the holes (Bendsøe & Sigmund, 2004).

5.3 Topology optimization methods

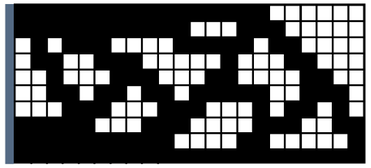
It is evident that Topology Optimization (TO) presents a greater potential for design exploration. The existing methods can be categorized into two groups: Gradient-based methods and the Direct-search methods.

Gradient - Based Methods



- Homogenization Method
- SIMP (Solid Isotropic Material Penalization)
- Level Set Method (LSM)

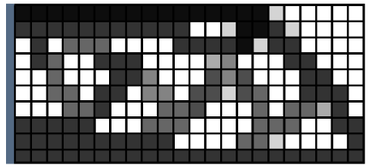
Direct-Search methods



- Evolutionary Algarithms
- Moving Morphable Components (MMC)
- Bi - Directional Evolutionary Structural Optimization ((B)ESO)

5.3.1 Gradient based methods

The adjustment of the variable set is directed by sensitivity analysis, which measures the impact of changes in the variable set on the objective function. The initial advancements in Topology Optimization (TO) involved the utilization of composite materials as an interpolation between void and full material. An example of such a method is the homogenization approach introduced by Bendsøe and Sigmund in 2004. These early developments laid the groundwork for establishing a framework for binary structures, as seen in methods like the Solid Isotropic Material with Penalization (SIMP) method and level set methods.



5.3.1.1 Homogenization

The Homogenization Method in Topology Optimization, initially proposed by Bendsøe and Kikuchi in 1988, stands as one of the earliest optimization solutions (Bendsøe & Kikuchi, 1988). This versatile mathematical technique serves as a vital link between microscopic intricacies and macroscopic behaviours in materials science and engineering design. The homogenization method of topology optimization consists of solving a class of shape optimization problem where the topology is made from an infinite number of microscale voids which produces a porous structure. The optimization problem then consists of finding the optimum values for the geometry parameters of the micro voids, which become the design variables. Transforming the optimization challenge into a sizing problem, seeking optimal dimensions for micro voids (Querin et al., 2017). A general formulation of the homogenization method is given as:

$$
\begin{aligned}
&Maximize: && II(u)\\
&Subject\ to: && \sum_{e=1}^{N}(1-a_e b_e)v_e - \overline{V_s} \le 0\\
&and && a_e - 1 \le 0\\
& && -a_e \le 0\\
& && b_e - 1 \le 0\\
& && -b_e \le 0\\
& && a_e, b_e, \theta_e : e = 1,2,\dots,N
\end{aligned}
$$

5.3.1.2 SIMP - Solid Isotropic Material with Penalization

A direct consequence of the homogenization method was the development of the SIMP method (Querin et al., 2017). It has become the most published and implemented software method of topology optimization. The basis is to use only one design variable per FE. Instead of incorporating finite elements as a combination of solid and void, each element is assigned an artificial element density (ρ_e) within the range 0 < ρ_min ≤ ρ_e ≤ 1, as introduced by Bendsøe and Sigmund in 2004. Here, ρ_min is set to a very small value to prevent numerical issues (Bendsøe & Sigmund, 2004). A penalty p is placed on the density when it is multiplied to the elastic modules of the element. As the penalization is increased from one to higher values, the optimal solution produces a discrete valued (0-1) problem. in SIMP the optimization problem is calculated for p = 1, then this is increased, and the optimization repeated until a (0-1) design is generated (Querin et al., 2017). A general formulation of the SIMP method is given by Bendsøe and Sigmund (2004) as:

$$
\begin{aligned}
&Maximize: && c(\rho_e) = \{F\}^T(u)\\
&Subject\ to: && \left[\sum_{e=1}^{N}\rho_e^p K_e\right]\{u\} = \{F\}\\
& && \sum_{e=1}^{N} v_e\rho_e \le \overline{V_s}\\
& && 0 < \rho_{min} \le \rho_e \le 1 : e = 1,2,\dots,N\\
& && p = 1,2,\cdots, p_{max} : p_{max} > 3
\end{aligned}
$$

The implementation of artificial density poses specific challenges, with one of them involving the interpretation of intermediate values when optimization hasn't fully converted to a 0-1 solution. The characteristics of these elements in the model become evident only when utilizing a composite material. Another issue arises concerning the dependency of solutions on mesh discretization. The optimization process tends to prefer integrating a greater number of inner cavities, optimizing material usage. The incorporation of these inner cavities results in a distinct geometry when the domain is divided into smaller elements, contrasting with the geometry achieved through larger element discretization (Bendsøe & Sigmund, 2004).

5.3.1.3 LSM - Level Set Method

The level set method, introduced by Osher and Sethian in 1988, is designed for the computation and analysis of interface motion in two or three dimensions. In the context of topology optimization, the design boundary, distinguishing between solid and void, is determined by the contour of the level set function (φ(x)) at zero (Sethian, 1999). The level set function φ exhibits certain properties (Osher & Fedkiw, 2001):

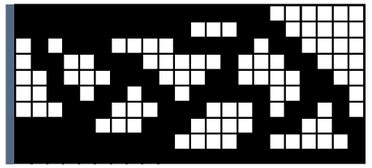
$$\varphi(x,t) > 0 \; for \; x \in \Omega$$
$$\varphi(x,t) < 0 \; for \; x \notin \Omega$$
$$\varphi(x,t) = 0 \; for \; x \in \partial\Omega = \Gamma(t)$$

The topology optimized design exclusively considers the material domain Ω and the interface Γ, later evaluating structural performance. Notably, the domain ðΩ is not treated as entirely empty but assumed to be filled with an extremely compliant material (ε << 1) to prevent singularities in calculations (de Ruiter & van Keulen, 2004). Level-set approaches categorize into two types: those allowing boundary changes while keeping hole count constant and those enabling new hole creation (Sigmund & Maute, 2013). For the first type, the Hamilton-Jacobi equation updates the boundary using the intersection between the cut-off level and the Level Set Function (LSF) graph, altering the domain by moving the cut-off level vertically. The speed of this translation is determined by the velocity function (Osher & Fedkiw, 2001). Conforming discretization analyses only the material domain Ω, leading to mesh changes with overall shape alterations. While providing a good outer boundary approximation, this method requires additional computational time for mesh adjustments, potentially introducing noise. Immersed Boundary Techniques (IBTs) avoid mesh restructuring, keeping the mesh unchanged for structural analysis and modifying only the material-void interface elements for an accurate outer boundary approximation. Techniques like the extended Finite Element Method (X-FEM) allow local variations along the interface line (domain Γ) defined by the iso-contour of the LSF in each iteration.

The last approach, like density-based methodologies, maintains consistent mesh discretization throughout optimization, changing only the material domain Ω with design alterations. The material is described by the pseudo-density function ρ(x), affecting material properties. For boundary interface elements, material proportion calculations scale material properties, but these values are not directly comparable to 'grey' values in the SIMP method.
Shape sensitivities, combined with Level-Set Methods, update the LSF during optimization, emphasizing overall shape changes over topological alterations.
Both approaches eliminate uncertainties related to material existence compared to density-based methodologies. However, caution is necessary to prevent geometric model confusions in stress analysis due to small intersections or non-smooth boundaries, requiring filtering schemes (Sethian, 1999). Various approaches aim to enhance convergence and avoid local minima, incorporating regularization techniques for defining basis functions and managing the slope and smoothness of the LSF. Care is needed to prevent numerical inconsistencies when applying these techniques (van Dijk et al., 2013; de Ruiter & van Keulen, 2004).

5.3.2 Gradient-free | direct search method

The Gradient-free approach was initially introduced by Xie & Steven in 1992. What sets them apart from the previously mentioned methodologies is the absence of a mathematical function serving as the foundation for design creation (Xie & Steven, 1992). Instead, the algorithm commences with an initial material domain and systematically eliminates material or in the more recent algorithms incorporates it into the volume based on stress analysis outcomes (Xia et al., 2016). These methods cannot always guarantee optimality as their outcome is often influenced by an initial guess. Still, they can provide viable efficient solutions (Querin et al., 2017).



5.3.2.1 ESO – Evolutionary Structural Optimization

Evolutionary Structural Optimization (ESO) was pioneering in the field of structural optimization as the first evolutionary algorithm, as introduced by Xie and Steven in 1993. The fundamental idea behind ESO is to enhance structural efficiency by systematically eliminating inefficient material from a given domain, allowing the structure to evolve towards an optimal configuration (Sigmund & Maute, 2013). The removal of material is governed by a rejection criterion, typically based on local stresses, although alternative criteria such as displacement, buckling, or thermal considerations can also be employed (Querin et al., 2017).

$$\sigma \leq RR_s * \sigma_{max}$$

In the context of a stress rejection criterion, elements with low stresses are considered inefficient and are consequently removed. The evaluation involves comparing the stresses of individual elements (σ_e) to the maximum stress (σ_max) across the entire structure (Querin et al., 2017).

5.3.2.2 BESO – Bidirectional Evolutionary Structural Optimization

One problem with the ESO method is that it is one-directional. Once material has been removed, it cannot be reintroduced into the domain (Querin et al., 2017). In contrast to the ESO algorithm, BESO allows for the addition of material in regions experiencing high stresses. Both ESO and BESO offer the advantage of yielding a discrete division of the domain, eliminating any 'grey' areas in the optimization process (Sigmund & Maute, 2013). However, the BESO method has been noted to produce mesh-dependent solutions. This implies that, for a consistent design domain under identical loads, selecting different mesh sizes can result in divergent solutions. Notably, a finer finite element mesh tends to generate a greater number of voids (Huang & Xie, 2007).
Typically, the BESO consist with the structure to satisfy the ESO inequality after which the material can be added through the second inequality (Querin et al., 2017):

$$\sigma \leq RR_s * \sigma_{max}$$
$$\sigma_e \geq IR_s * \sigma_{max}$$

To address this mesh dependency, the introduction of perimeter control to the BESO method can yield mesh-independent outcomes. Nonetheless, determining an appropriate value for the perimeter length constraint proves challenging (Huang & Xie, 2007). Another concern with the BESO method is its potential for non-convergent solutions. This means that allowing the algorithm to run indefinitely may result in a solution that deteriorates concerning the objective function (Sigmund & Maute, 2013).

5.3.2.3 SKO – Soft Kill Option

The soft kill option method of topology optimization simulates the process of adaptive bone mineralization by varying the elastic modulus of a loaded structure according to its stress distribution. The process starts with the entire structure give a target maximum elastic module E_max. Then the stress distribution is calculated. Then the elastic module of every element is distributed with:

$$E_{i+1}^e = E_i^e + k\left(\sigma_i^e - \sigma_{ref}\right)$$

The equation describes how the modulus of elasticity (E) at the next time step (I+1) evolves from its current state E_i^e. This evolution is influenced by a term proportional to the difference between the current stress σ_i^e and a reference stress σref, and the proportionality is governed by a constant (k).

In practical terms, this could be part of a simulation or analysis involving materials that exhibit dynamic changes in their mechanical properties, such as in the field of fluid-structure interaction or material response to varying conditions. The equation allows for the adjustment of the material's elastic behaviour based on the stress it is experiencing, with the reference stress serving as a benchmark (Querin et al., 2017).

5.3.2.4 Other direct search methods

Numerous evolutionary topology optimization methods exist, each demonstrating varying degrees of promise. However, due to limited research in the context of architectural applications utilizing these methods, they are not explicitly enumerated in this study. Despite the potential offered by these alternative methods, the current exploration focuses on those methodologies that have received more attention and validation within the architectural domain. As research expands and delves deeper into the application of evolutionary topology optimization in architecture, additional methods may emerge and warrant consideration for future investigations.



Figure 44: AI generated, prompt: 'glass bridging element'

5.4.1 Gradient vs Direct search

Extensive research has been conducted to rectify specific inaccuracies and assess the performance of various methods through their application in characteristic case studies. Concerning density-based methods, the Solid Isotropic Material with Penalization (SIMP) approach stands out as widely popular. SIMP optimization problems generally exhibit a simple structure, resulting in fast and robust simulations with good convergence. However, challenges arise from geometric defects, result dependency on the penalization value, mesh refinement considerations, and the non-direct definition of boundaries due to 'grey' areas associated with elements having intermediate densities. Notably, the method can lead to jagged boundaries, causing slightly higher compliance outcomes compared to Level Set Methods (LSM) and requiring additional post-processing for the final shape. Although SIMP is not constrained by a specific starting shape and can reach a global minimum, the design appears highly dependent on the formulation of objectives and constraints in the problem. In Level-Set Methods, a notable advantage lies in the clear definition of boundaries. However, attention must be given to filter cut FEs to remove small parts causing false stiffness values. Despite independence from mesh refinement, the result is influenced by the initial guess, making it crucial to attain global minimal (Allaire, 2005). Computational demands are higher, suggesting the use of lower-level languages like C++ or Fortran (Andreasen et al., 2020).

The discussion of evolutionary algorithms forms the last category. While capable of providing rapid solutions, these algorithms face criticism for often converging to local minima. Establishing an appropriate stopping criterion is challenging, as it must effectively monitor convergence, considering factors like structural degeneration, differences in outcomes between steps, or the achievement of set objectives, yet none can guarantee the attainment of a global minimum (Sigmund & Maute, 2013).

5.4.2 Combination

Allaire has proposed a combination of topology optimization, suggesting a fusion of Homogenization with Level-Set Method (LSM). Despite significant differences, LSM can function as a secondary step, utilizing the result of the density-based topology optimization operation as an initial shape (Allaire, 2005).
Shao, on the other hand, has proposed a combination of Solid Isotropic Material with Penalization and Bi-directional Evolutionary Structural Optimization. In this approach, the minimal compliance area is initially defined using SIMP. Subsequently, the boundaries of that outcome are offset, allowing BESO to iterate and determine the final topology-optimized result (Shao, 2019).
While both proposed ideas hold promise, there are currently no developed and proven optimizations. Therefore, this option is not considered for this master's thesis.

5.4.3 Conclusions

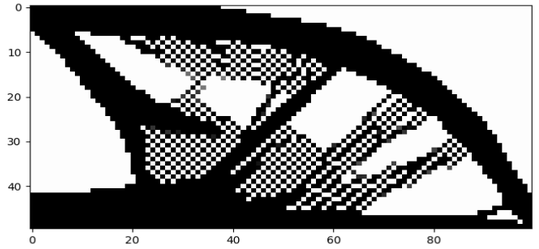
BESO, an improvement over ESO, allows both addition and subtraction of material in the design. However, the gradually imposed volume constraint in the BESO method tends to yield topologies comprising a multitude of thin bars, introducing complexity in shape and manufacturing (Yago et al., 2022). Meanwhile, Level Set Methods offer the advantage of a smooth surface, reducing the need for post-processing. Yet, they still heavily rely on the initial guess (Sigmund & Maute, 2013) and demand more computational time, making it advisable to implement them in lower-level languages such as C++ (Andreasen et al., 2020).
In the context of this project, the SIMP method stands out as the most fitting choice. Despite showcasing some geometric defects, it provides rapid and resilient solutions with commendable convergence. Its autonomy from any initial guess enhances the likelihood of reaching a global minimum, with the outcome predominantly influenced by the objective function and constraints. Additionally, the widespread popularity of the SIMP method ensures a wealth of documented knowledge and practical applications, including the formulation of diverse overhang constraints and real-life case studies. This makes SIMP a pragmatic choice for the optimization tasks undertaken in this thesis.

5.5 Checkerboards, point flexures and islanding

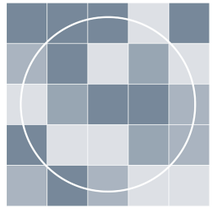
Geometric singularities are common issues that arise through using rectangular cells for the TO process. They should be avoided, since they result in false estimations regarding the performance of the element designed. The most important being checkerboards, point flexures and islanding. Checkerboarding is the occurrence of a checkerboard like pattern in the optimized geometry. Due to the nodal connectivity, this pattern falsely gives the impression of structural stiffness (Bendsøe & Sigmund, 2004). Point flexures are similar to checkerboarding, again solid elements are connected to each other in a single point. Resulting in a geometry that is not feasible to produce and that has local peak stresses. Lastly, islanding is the occurrence of solid parts in the structure that are not connected to any other part of the structure (Kumar, 2017).

Another common issue, which results in inaccuracies regarding the estimation of the structural performance of the element, is related to the boundaries of the element obtained. These are often blurred and not completely discrete, with angles that prevent from the creation of one continuous boundary line. This has an effect not only in the precision of the shape but may also lead to larger compliance during the structural analysis (Bendsøe & Sigmund, 2004). As the SIMP method has been further researched multiple solutions have been formulated for each of the previously explained issues. Filtering methods are employed to mitigate geometric defects and enhance the smoothness of the optimized structures. These techniques involve applying spatial filters to the design variables, influencing the distribution of material and minimizing abrupt changes in the topology. This helps in achieving more physically plausible and manufacturable designs.

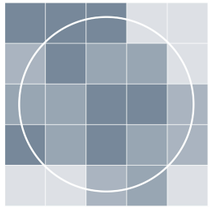
Additionally, the incorporation of size constraints serves to regulate the thickness or dimensions of structural elements in the optimization process. By imposing limits on the sizes of individual components, the resulting topologies can be controlled to meet specific design criteria, promoting structural feasibility and ease of fabrication (Bendsøe & Sigmund, 2004).



checkerboard issue



Unfiltered densities



Weighted average densities neighbourhood

5.6 Finite elements

Within topology optimization algorithms, finite element analysis (FEA) serves as a crucial tool, providing insights into the structural response of optimized geometries and forming the foundation for defining objective and constraint functions. Typically, a finite element analysis commences by discretizing the problem domain into geometric elements. These elements, representing discrete sections of the structure, facilitate the approximation of its behavior.

5.6.1 1D-elements

The displacements of individual elements are assumed in simplified forms, allowing for the assembly of a global stiffness matrix [K] by combining the equations of neighboring elements. Incorporating boundary conditions enables the determination of individual nodal displacements. Subsequently, utilizing these displacements, the strains and stresses within the individual elements are computed (G. R. Liu & Quek, 2014).

The formulation of element equations is grounded in the principle of minimum potential energy. This principle posits that a body subjected to a force will deform to a stationary point where the total energy in the body is minimized. More specifically, the principle asserts that the total potential energy Π of an elastic body equals the sum of the total strain energy (U) and the work potential (wp). The work potential encompasses external forces acting on the body, which can include traction forces (T), body forces (f), and/or loads acting on specific points (Pi). Consequently, the potential energy for a general elastic body is expressed as:

$$= \frac{1}{2}\int_V^\infty \sigma^t \varepsilon \, dV - \int_V^\infty u^T f dV - \int_s^\infty u^T T dS - \sum_i u_i^T P_i$$

The stiffness matrix can be derived from the total strain (U). As the relation between stress (σ) and strain (ε) is given by σ = Eε, U can be rewritten as:

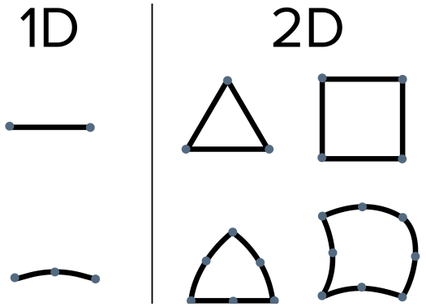
$$\int_V^\infty \varepsilon^T E \varepsilon dV$$

5.6.1 1D-elements

Introducting a matrix that translates strain to displacement (ε = Bq) and for simplicity only considering nodal forces f, the stationary point is given by:

$$\frac{\partial \Pi}{\partial\{q\}} = \int_V^\infty [B]^T[E][B]dV\{q\} - \{f\} = 0$$
$$or\ as: \quad \{f\} = [K]\{q\}$$
$$\text{where: } [K] = \int_V^\infty [B]^T[E][B]dV$$

In a problem domain with known boundary conditions {f}, the stiffness matrix [K] is assembled from element equations, and solving for the stationary point yields nodal displacements {q}. Each finite element type has distinct equations dependent on the element's shape functions [N] (Chandrupatla & Belegundu, 2002).
Finite element analysis is applicable to linear and nonlinear problems. In linear scenarios, stresses are assumed to stay within the material's elasticity limit, resulting in a constant stiffness matrix. Nonlinear problems involve significant deformations or materials stretched beyond their yield point, necessitating adaptations due to changes in the stiffness matrix (Chandrupatla & Belegundu, 2002).



1D / 2D finite elements

5.6.2 2D-elements

Domain division can be accomplished with various elements. Two-dimensional elements, commonly used for problems governed by plane stress / strain, are surface elements with triangular or quadrilateral base shapes containing 3 or 4 nodes, respectively. Isoparametric versions enhance accuracy when required. In planar elements, the displacement vector is denoted as u = [u, v]T, representing x and y directions (Chandrupatla & Belegundu, 2002).

For two-dimensional problems, stresses (σ) and strains (ε) are expressed as vectors:

$$\sigma = [\sigma x, \sigma y, \tau xy]^T$$
$$\varepsilon = [\varepsilon x, \varepsilon y, \gamma xy]^T$$

Triangular elements, also known as constant-strain triangles, feature linear shape functions, resulting in constant strains and stresses throughout the element. Principal stresses can be derived using Mohr's circle. In contrast, quadrilateral elements do not have constant stresses. Stress analysis typically yields four sets of stress values at Gauss points, and for increased efficiency, evaluating stress at one point may suffice, reducing calculation time (Chandrupatla & Belegundu, 2002).

These 2D elements are advantageous for analyzing structures like plates or shells under plane stress / strain conditions. They provide insights into stress concentration locations, aiding in structural assessments (Wai et al., 2013).

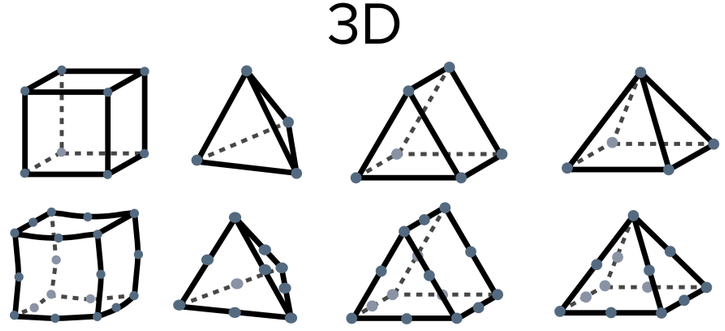
5.6.3 3D-elements

In three-dimensional analyses, which accurately represent engineering problems in their full spatial context, four-node tetrahedra or eight-node hexahedrons are commonly used for domain discretization. Tetrahedral elements, like their 2D counterparts, have linear shape functions, resulting in constant strains and stresses. However, there exist also other types of finite elements which can populate a 3D mesh. To simplify the division of a 3D domain, a master cube can be employed, subdivided into five or six tetrahedra. The assembly of hexahedron element matrices can be computed from the assembly of tetrahedron matrices (G. R. Liu & Quek, 2014).

The displacement vector for three-dimensional problems is denoted as u = [u,v,w]^T, allowing nodes to move freely in the x, y, and z directions (Chandrupatla & Belegundu, 2002). Isoparametric versions of these elements can be generated for higher accuracy if needed. As with 2D elements, the choice of 3D elements depends on the balance between accuracy and computational efficiency. Each type of element is chosen based on the geometry and characteristics of the structure being analysed.

$$\sigma = [\sigma x, \sigma y, \sigma x, \tau yz, \tau xz, \tau xy]^T$$
$$\varepsilon = [\varepsilon x, \varepsilon y, \varepsilon z, \gamma yz, \gamma xz, \gamma xy]^T$$

Principal stresses are obtained by analyzing the (3 x 3) stress tensor using stress invariants. While three-dimensional finite element analysis demands more computational time, there's a preference to simplify structures to 2D or 1D representations. Nevertheless, for complex geometries requiring in-depth analysis, 3D elements provide accurate deformations and reveal the presence of stress concentrations (Wai et al., 2013).



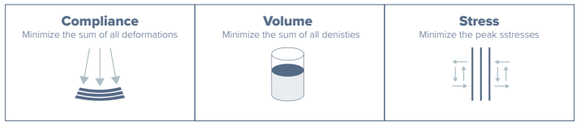
3D finite elements

5.7 Formulation of the optimization problem

According to Boyd and Vandenberghe (2004), a generalized optimization problem can be expressed in the following format:

$$minimize: \ f_O(x)$$
$$g_i(x) \leq 0, for\ i = 1, 2, \dots, m$$
$$h_j(x) = 0, for\ j = 1, 2, \dots, p$$

In the realm of optimization, the objective function (denoted as fo(x) serves as the focal point, embodying the goal to be either minimized or maximized. Complementing this objective, inequality constraints expressed as gi play a crucial role in shaping the optimization landscape. These constraints encapsulate the boundaries or limitations that the solution must adhere to. Concurrently, equality constraints hj provide a set of conditions that must be satisfied for the solution. At the heart of these considerations lies the vector of decision variables (x), representing the parameters or choices that the optimization process seeks to optimize. This fluid interplay between the objective function, inequality constraints, equality constraints, and decision variables encapsulates the essence of mathematical optimization, where the quest is to navigate this multidimensional space in search of optimal solutions.



5.7.1 Compliance-based minimization

A design with minimum compliance seeks to maximize global stiffness. Introducing a volume constraint in optimization aims to achieve an optimal material distribution for enhancing structural stiffness (Bendsøe & Sigmund, 2004). The general form of a minimum compliance objective with a volume constraint is expressed as follows:

$$min_x \quad : c(x) = U^T K U$$
$$subject\ to \ : \frac{V(x)}{V_0} = f$$
$$KU = F$$
$$0 < xmin \leq x \leq 1$$

In this context, the variable (x) signifies the design variable where it denotes the element densities. The permissible range for the variable (x) is constrained between (xmin) and 1, with (xmin) being a minute fraction introduced to prevent singularities in the computation. Throughout the calculation, the structure must adhere to static equilibrium conditions defined by the global stiffness matrix (K), global displacement (U), and the global force vector (F). Additionally, (V(x)) represents the current volume fraction, where (V) is the total domain volume, and (f) is the prescribed volume fraction.

Despite its advantages, compliance-based optimization has certain drawbacks. Notably, the prescribed volume fraction significantly influences the structure's outcome, potentially leading to a suboptimal solution with a stiffer structure having less mass. Another limitation lies in the fact that the general form of compliance calculation does not consider peak stresses, which could contribute to structural failure ((Collet et al., 2017)).

5.7.2 Volume-based minimization

In volume-based optimization, the goal is to minimize the volume while adhering to a constraint related to structural performance. Typically, in a volume-based objective, assessments are made regarding compliance and stress evaluation. The compliance constraint ensures the achievement of a stiff structure, while the stress constraint guarantees the feasibility of the final outcome (Bruggi & Duysinx, 2012). A common formulation for expressing volume-based optimization is:

$$min_x \quad : V(x) = \sum_N x_e v_e$$
$$Subject\ to : KU = F$$
$$: \frac{c}{c_L} \leq 1$$
$$: x_e^{(p-q)} \frac{\sigma_e}{\sigma_{Lt}} \leq 1, \qquad e = 1, \dots, N$$
$$: 0 < x_{min} \leq x \leq 1$$

Where v_e is the volume of each element; c represents the compliance of the structure, and c_(L)is the minimum compliance. Stress evaluation is once again conducted in each finite element, resulting in a robust yet computationally expensive calculation. It is plausible to convert this constraint into a global constraint. However, as mentioned earlier, this doesn't guarantee the avoidance of all peak stresses.

An alternative to mitigate calculation time is to employ a material failure criterion instead of assessing absolute stress values in each element. The Von Mises criterion is suitable for materials with equivalent strength in tension and compression. Alternatively, for materials with distinct tensile and compressive strengths, the Drucker-Prager failure criterion provides an option beyond the Von Mises criterion (Bruggi & Duysinx, 2012). Another notable failure criterion is the Tsai-Wu criterion, which is specifically designed for composite materials.

5.7.3 Stress-based minimization

An alternative formulation for the optimization problem is through a stress-based objective. In contrast to minimum compliance, this approach focuses on minimizing stresses within the structure while adhering to a volume constraint. The standard formula for this stressed-based objective is provided by Yang and Chen (1996):

$$min \quad : G_{(x)}$$
$$subject\ to : \int_\Omega^\infty \mathrm{x d}\Omega \leq M_0$$

Where:
G((x)):Global stress function for von Mises Stresses
M0: Volume that is to be maintained

Yang and Chen (1996) emphasize that optimizing for minimum stress yields notably different outcomes than a compliance-based objective. The findings suggest that while it is possible to minimize peak stresses, it may come at the expense of a reduction in structural stiffness. To address this trade-off, combining both stress and compliance into a unified objective function proves advantageous, resulting in enhanced outcomes and faster convergence.

However, stress-based optimization introduces additional challenges beyond the potential decline in structural stiffness. Firstly, stress is inherently a local quantity, necessitating individual checks for every element to effectively control peak stresses, significantly increasing computational time. One approach to mitigate this challenge involves introducing a uniform stress constraint approximating the maximum stress value. Nevertheless, this uniform constraint may fall short in identifying local peak stresses. A second issue arises when elements with very low density incorrectly exhibit non-zero stress values, leading to the 'singularity' problem (Bruggi & Duysinx, 2012).

5.8 Topology optimization algorithms

Several algorithms offer valuable perspectives on topology optimization methods and finite element analysis. Subsequent sections will delve into the specific algorithms pertinent to the objectives of this thesis.

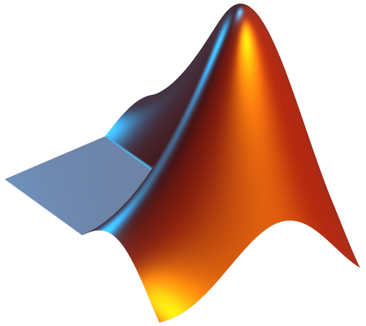




Figure 45: https://pemrogramanmatlab.com/2023/09/09/panduan-pemrograman-matlab-untuk-pemula/ & https://junilearning.com/blog/guide/what-is-python-101-for-students/

5.8.1 99-line for 2D topology optimization

Sigmund (2001) crafted a 99-line topology optimization code in Matlab, presenting a two-dimensional algorithm designed for compliance minimization in statically loaded structures. This code employs the Solid Isotropic Material with Penalization (SIMP) methodology and integrates the Optimality Criteria (OC) method for solving the optimization problem. To prevent checkerboarding, a mesh-independency filter is incorporated. Notably, the code introduces a symmetric boundary condition, effectively calculating only half of the structure to enhance computational efficiency. The implementation involves nested loops for assembling the stiffness matrix and preparing the filtering process (Sigmund, 2001).

5.8.2 88-line for 2D topology optimization

Andreassen et al. (2011) developed an 88-line topology optimization code, building upon Sigmund's 99-line code (2001). This program focuses on compliance minimization for statically loaded two-dimensional structures. Significant enhancements in calculation time are achieved through more efficient utilization of Matlab. Key optimizations include loop vectorization and memory preallocation. Vectorization optimizes array operations by executing them on entire arrays simultaneously, avoiding the need for costly loops. Memory preallocation involves reserving memory for a data structure before populating it, preventing performance issues associated with dynamic resizing. The 88-line code is structurally refined compared to the 99-line version, with certain calculations moved outside the optimization loop to be computed only once. These improvements result in a substantial increase in efficiency (Andreassen et al., 2011).

5.8.3 200-line 2D topology optimization algorithm

This Python code, initially crafted by Niels Aage and Villads Egede Johansen in January 2013, represents a topology optimization implementation spanning 200 lines. Niels Aage subsequently updated the code in 2016. While discernible similarities exist between this 200-line implementation and an 88-line counterpart, our focus lies predominantly on elucidating their distinctions.
The Python-based 200-line code leverages Matplotlib for dynamic visualization, providing real-time insights into the design evolution during the optimization process. In contrast, the 88-line code, developed in MATLAB, relies on the native `imagesc` function for rendering the optimized design. But the what sets the 200-line Python code apart is its elevated level of complexity, integrating additional features like external modules to incorperate the overhang constraints. For detailed specifications regarding the overhang constraint, readers are directed to chapter: (5.8.2).

5.8.4 2D-topology optimization for cast glass structures (Koniari)

Koniari (2022) developed a two-dimensional topology optimization algorithm for cast glass structures, utilizing the SIMP methodology. The algorithm optimizes for compliance and volume while addressing specific constraints for cast glass design. Implemented in Matlab, it employs the Fmincon function for constrained nonlinear multi-variable function minimization, choosing the standard interior-point algorithm. Glass-specific constraints, such as principal stress and maximum member size, are integrated to ensure annealing time compliance. The code structure aligns with the logic of pre-assembling data to reduce redundant recalculations, similar to the 88-line algorithm. Regrettably, despite extensive research conducted on her thesis, the code couldn't be assembled to function properly.

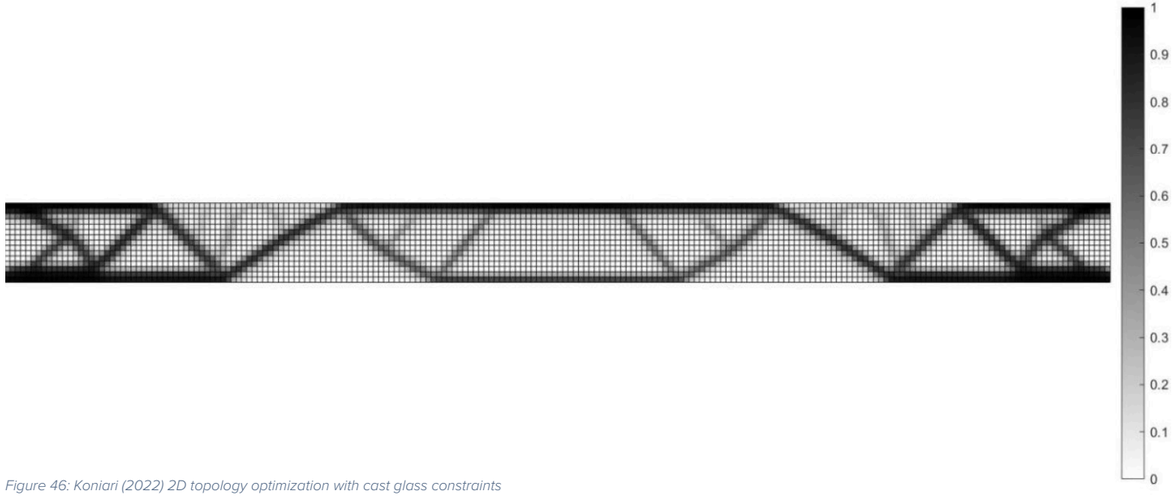


Figure 46: Koniari (2022) 2D topology optimization with cast glass constraints

5.8.5 169-line 3D topology optimization algorithm (top3d)

K. Liu and Tovar (2014) crafted a comprehensive 169-line three-dimensional topology optimization algorithm, encompassing finite element analysis, sensitivity analysis, density filtering, and a plotting function for result visualization. This algorithm is adept at addressing minimum compliance challenges within rectangular domains, employing a density-based approach with a modified Solid Isotropic Material with Penalization (SIMP) interpolation. The default optimization algorithm integrates the gradient-based optimality criterion method. The formulation incorporates partial derivatives of the compliance objective and volume constraints. Moreover, the algorithm provides flexibility by allowing the implementation of nonlinear programming strategies like Sequential Quadratic Programming (SQP) and Method of Moving Asymptotes (MMA). The finite element discretization utilizes uniformly sized eight-node hexahedral elements.
Despite featuring a density filter, the code offers alternatives such as sensitivity and grey scale filters. Building on the efficiency principles from the 88-line code, this algorithm ensures optimized problem calculations. It introduces an iterative solver tailored for handling large models.

5.8.6 3D topology optimization for cast glass structures (Schoenmaker)

Schoenmaker (2023) introduced a three-dimensional topology optimization algorithm tailored for cast glass structures. This algorithm adopts the SIMP methodology, with its foundation primarily derived from the 169-line code discussed in Chapter 5.7.5. Nonetheless, it distinguishes itself through the incorporation of alternative filters and constraints, with a primary emphasis on ensuring the feasibility of casting glass.

The introduced constraints encompass the enforcement of maximum member size, minimum gap size, and the prevention of internal voids. The latter is addressed by employing a finite element method (FEM) to simulate heat transmittance. Additionally, a Drucker-Prager stress constraint has been implemented, accounting for the material properties of glass.

Although the code was successfully generated, challenges persisted in addressing the islanding phenomenon observed in several elements of the topology optimization. Consequently, a considerable post-processing effort was required to mitigate these issues.

5.9 Constraints

The manufacturing process of an object imposes limitations on the size and arrangement of its elements. A pivotal criterion set by the manufacturing process includes constraints on the dimensions of individual members and the overhang between different elements.

5.9.1 Filters

As discussed in earlier sections, filters are employed to mitigate numerical instabilities such as checkerboard patterns, point flexures, and islanding. Additionally, filters play a role in diminishing solutions dependent on the mesh. Filters can be applied either to sensitivities, densities, or both. For a more comprehensive understanding of a diverse array of filters, refer to Sigmund's detailed description in the year 2007.

5.9.1.1 Sensitivity filter

The sensitivity filter functions by computing sensitivities in the conventional manner and then adjusting them as weighted averages of sensitivities within a specified neighborhood. This adjustment is performed using the equation:

$$\frac{\partial c(x)}{\partial x_i} = \frac{1}{\max(\gamma, x_i)\sum_{j\in N_i} H_{ij}} \sum_{j\in N_i} H_{ij} x_j \frac{\partial c(x)}{\partial x_j}$$

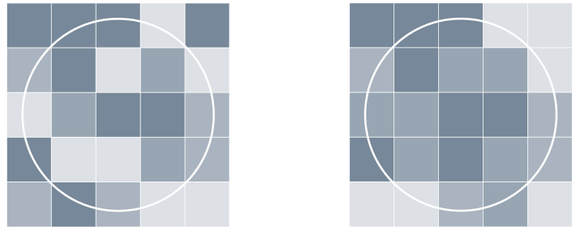
In this context, γ(= 10−3) is introduced to prevent division by zero, and H_ij represents a weight factor defined as Hij = rmin − dist(i, j). The distance is measured between the center of element i and the center of element j and the weight factor becomes zero outside the region determined by rmin . The implementation of the filter involves utilizing the filtered sensitivities for the Optimality Criteria update (Sigmund, 2001).

5.9.1.2 Density filter

With a density filter, each element's density is redefined as a weighted average of the densities in a given neighbourhood Ne of the element x_i. The filtered density is given by:

$$\bar{x_i} = \frac{\sum_{j \in N_e} w_{ij} x_j}{\sum_{j \in N_e} w_{ij}}$$

In this equation, w(xe) is a linear weight factor defined by w(xe) = R dist(e, j), where dist(e, j) denotes the distance between the center of element e and the center of element j, and R is the radius size of the neighborhood. This weight factor linearly decreases over the distance from the centroid to rmin The original densities x are then replaced by the filtered densities x, also known as the physical densities (Sigmund, 2007).



Denisty filtering

5.9.1.3 Density filtering with a Heaviside step function

The density filter can undergo modification by incorporating a Heaviside function to enhance convergence towards a 0-1 design. The integration of a Heaviside function ensures that elements with a filtered density greater than 0 will be assigned a value of 1, while only those with a filtered density of 0 will result in a physical density of 0.

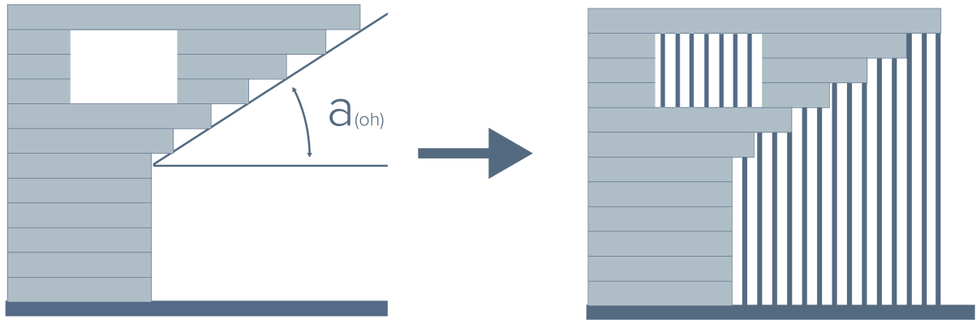
$$\bar{x_i} = \frac{1}{1 + e^{-\beta(\rho_i - 0.5)}}$$

In this modified approach, when β is set to zero, the filter remains identical to the original density filter. However, as β approaches infinity, the density value of the central pixel is adjusted to one if any of the pixels within the neighborhood have a density larger than zero. To facilitate stable convergence, it is recommended to gradually increase the value of β from 1 to 500 (Sigmund, 2007).
This enhancement introduces a dynamic element to the density filter, facilitating a seamless transition from intermediate density values to a binary 0-1 design. The integration of the Heaviside function enhances the filter's adaptability, promoting convergence toward distinct and well-defined structures in the topology optimization process. Consequently, this refinement reduces the potential for misinterpretations of the optimized structure by yielding a more pronounced output.

5.9.2 Overhang filters

To create topology-optimized structures, it is crucial to establish constraints that ensure the feasibility of the manufacturing technique for production. However, as additive manufacturing processes come with their own specific limitations (see chapter 4). Failure to consider these limitations during design optimization often necessitates modifications in post-processing. These post-processing adjustments not only incur additional costs but also diminish the optimized performance of the structure (Langelaar, 2015). Given the selection of FDM 3D printing as a potential solution for fabricating topology-optimized structures. It is essential to acknowledge the infeasibility of creating overhangs that exceed the allowable limitations inherent in the specific type of printing technique employed, in this case a maximum of 30 degrees is allowed.

As depicted in the figure below, the necessity to incorporate the overhang angle in topology optimization is crucial. Without it, substantial support would be required to construct the structure effectively. Failure to implement support or integrate the overhang angle would result in structural failure, as subsequent prints would lack a stable foundation layer. This master's thesis specifically revisits the idea of planar printing and utilizes only the bed as the support area. Consequently, optimization is constrained to utilizing the bed as a support area, determining the print direction, and employing the desired overhang angle as the angle of growth from the base plate.



The incorporation of the overhang angle is essential to avoid the need for support material.

5.9.2.1 Overhang filters overview

Several overhang filters have been explored over the years. The following section presents an exploration into various types, unveiling distinct characteristics and attributes associated with each.

Brackett et al. introduced a method to identify overhanging sections' angles in 2D designs generated by Topology Optimization (TO). They suggested incorporating this information into the optimization process as an additional constraint. However, no subsequent work has demonstrated the realization of this idea.
As an alternative, Leary et al. proposed augmenting the originally optimized design with additional structures, ensuring the final geometry is entirely self-supporting. An iterative procedure was developed to minimize the amount of added material. In this approach, the additional structures are not treated as sacrificial parts that need removal after printing; instead, they form an integral part of the final structure. While successful in generating self-supporting 2D designs, this method tends to deviate from the original optimized geometry.

Gaynor and Guest recently proposed a wedge-shaped spatial filter to control the boundary orientation during TO. However, the filter definition doesn't account for the layer-wise support conditions crucial in Additive Manufacturing (AM) processes. This allows the optimization process to exploit configurations that are not inherently self-supporting. The filter considered the part above a threshold density average in the wedge as properly supported. Although the 2D results showed a degree of self-supporting designs, introducing support requirements only in an average sense allows the optimizer to use intermediate density material to support fully dense structures, which is undesirable.

Van de Ven et al. addressed the overhang challenge by introducing a speed function that originated from the base plate of the topology optimization (TO) mesh. This function determined the delay of overhanging parts, and the implementation of this approach proved successful in both 2D and 3D scenarios.

Langelaar developed an additive manufacturing (AM) filter designed for topology optimization, which has demonstrated success in various research studies (references). In 2D, the filter checks if the direct neighbors underneath the node are present. If not, the code reallocates a certain amount of material to the area to ensure structural integrity.
The investigation in this research extends to the filters proposed by Van de Ven et al. and Langelaar.

5.9.2.2 Front propagation based overhang filter

Van de Ven et al. (2018) introduced an overhang filter utilizing front propagation to identify overhanging regions. In this approach, a front is initialized at the boundary and propagated throughout the domain, resulting in an arrival time field T(x) indicating the time at which the front reaches each location. The filter is designed to detect overhangs in each geometry, considering the height from the base plate.

$$T^{layer}(x) = b * \frac{x}{f_0}$$

$$\tau\,(x) = \;T\,(x) - \;Tmin(x)$$

$$T(x) = T(x') + \frac{\parallel x - x' \parallel}{f(a)}$$

$$f\,(a, \alpha_{oh}\,) = \frac{f_0}{\max\,(tan(\alpha_{oh}\,) \parallel Pa \parallel, \parallel b \cdot a \parallel)}$$

The arrival time fields are crucial for detecting overhangs. The first field T^layer is proportional to the height from the base plate, representing the printing sequence. The second, T(x), results from front propagation in the design domain Ω, with speeds adjusted based on the minimum overhang angle αoh.

A delay field τ(x) between these arrival time fields is then calculated. In regions where τ(x)>0, indicating overhanging structures, the printable density field is determined from the delay field using a delay-density relation. The delay-density relation introduces a parameter related to the element size. Higher values result in a shorter transition from full density to void, accommodating fine meshes. The speed function in front propagation is scaled with densities, ensuring that all densities are printable at the base plate.

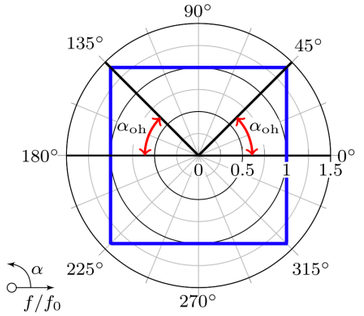


Figure 47: Rework by author of working principle Van de Ven et al (2018)

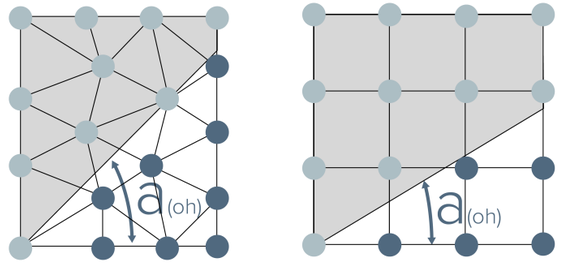


Figure 48: Rework by author of working principle Van de Ven et al (2018)

5.9.2.2 front propagation based overhang filter

The anisotropic speed function is decomposed into a density-dependent part and a direction-dependent part, where the latter relates the propagation direction to speed. A rectangle-shaped function with an aspect ratio defining the minimum overhang angle α_oh is used in the direction-dependent part. The speed-density relation is linearly scaled with density, introducing geometry dependence.

The continuous overhang filter underwent enhancements through the incorporation of a refined continuous filter, particularly emphasizing improved dependency within the initial 10 iterations. This modification allows the structure to develop without the overhang constraint, leading to more favourable outcomes. This adjustment is crucial as the typical growth pattern of structures often originates solely from the baseplate (Van de Ven et al., 2020). This continuous overhang filter, developed by Van de Ven et al., has been successfully adapted for 3D applications. The extension to 3D involved the use of tetrahedra in mesh generation, as conversion to hexahedrons was restricted by the front propagation's underlying principles. However, despite the successful implementation, Van de Ven (2024) noted challenges in comprehending the 3D code's complexities. In discussing the outcomes, Van de Ven mentioned, "the 3D code I use in that paper was part of a parallel topopt framework for unstructured meshes. This is quite an extensive piece of code, based on PETSc and where the overhang filter is also programmed completely in parallel with MPI. In itself useful, but I must honestly say that because it was my first C++ project, in combination with the almost complete absence of comments, it is actually impossible to get through that code." As a result, further implementations were not pursued due to the complexities involved, aligning with the time constraints of a master's graduation project.

Hi Pim,

That sounds like a fun project! Are you planning to actually make a glass bridge?

Regarding the overhang constraint, the 3D code I used in that paper was part of a parallel topopt framework for unstructured meshes. This is quite an extensive piece of code, **based on PETSc** and where the overhang filter is also programmed **completely in parallel with MPI.** Useful in itself, but I must honestly say that because it was **my first C++ project,** in combination with **almost completely missing comments,** **it is actually impossible to get around the code.**

Especially with the **limited time of a graduation project,** I would recommend using a simple implementation for overhang constraints.

Good luck!

Greeting,
Emiel

Figure 49: Mail contact with Emiel

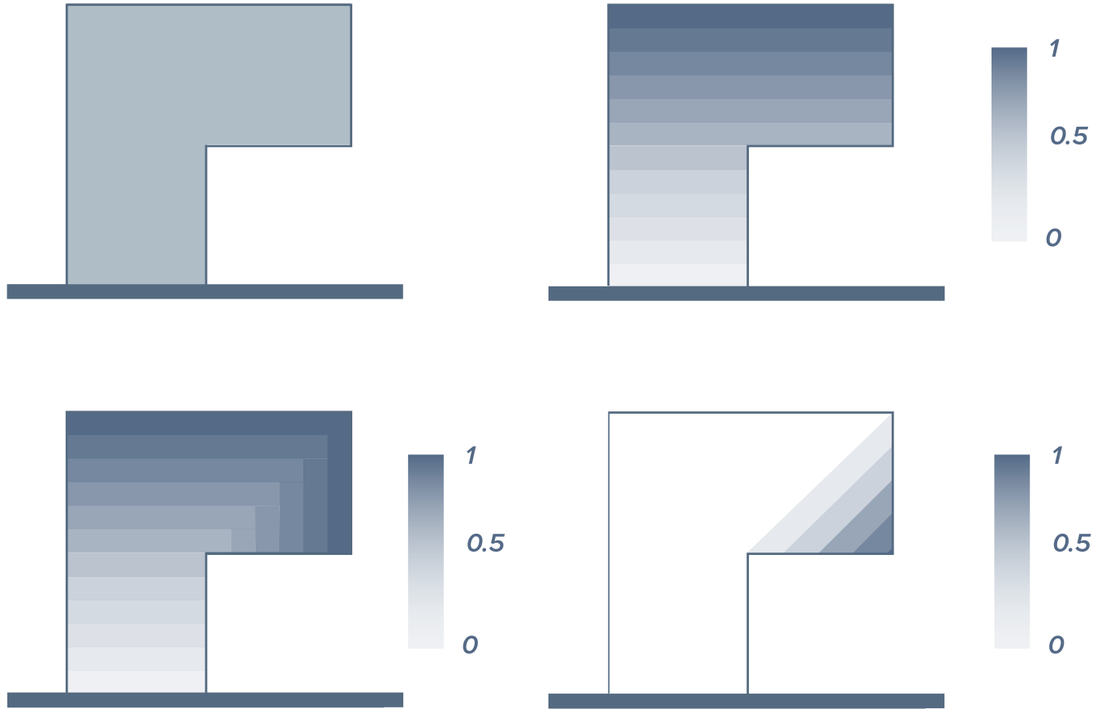


Figure 50: rework by author of working principle Van de Ven et al (2018)

5.9.2.3 Layer – by – Layer overhang filter

The layer-by-layer overhang filter outlined in Langelaar (2017) is designed for a structured rectangular grid. By integrating a simplified AM process model into the problem formulation, the imposed overhang angle restriction for the specific AM processes is systematically enforced throughout each stage of the optimization process. This is accomplished through the application of a layerwise spatial filter, mimicking the layer-by-layer printing of the component. The optimization process is driven by the evaluation of the as-printed geometry's performance, resulting in the generation of self-supporting designs that are fully compatible with the printing constraints (Langelaar, 2017).

In a 2D context, each element in a structured mesh can be uniquely identified by its row and column indices (i, j). The filter's operation can be succinctly described in two principles: for an element at position (i, j), its supporting elements consist of the three elements directly below it— (i-1, j-1), (i-1, j), and (i-1, j+1). Additionally, the printable density Ξi,j of the element must not surpass the maximum printable density Ξ of the supporting elements.

$$\Xi_{i,j} = \max(\xi i-1, j-1, \xi i-1, j\,, \xi i-1, j+1)$$
$$\xi_{i,j} = \min\left(\Xi_{i,j}, \rho_{i,j}\right)$$

Assuming the printing direction is from the bottom to the top row, the filter is propagated through a domain by initializing the bottom row of elements with Ξ1,j = ρ1,j. Or,(ρ)e is the printed density, ρe^S is the maximum printed density in the supporting region Se, (ρ_e)˜ and is a blueprint density variable, which is the projected density. Subsequently, the evaluation is carried out layer-by-layer in the printing direction, with sensitivity evaluation performed in the reverse order. To ensure continuous differentiability for gradient-based optimization, non-smooth min and max operators are replaced by differentiable counterparts. When using a suitable structured finite element mesh, this filter can be directly applied to the elemental densities. In cases where a mapping between the FE mesh and a separate overhang grid is needed (Langelaar, 2017).

In the 3D context, the filtering principle remains largely unchanged. When examining a specific element at position (i, j, k), the filter ensures that the 5 or 9 elements directly below it in S(i,j,k) do not exceed the maximum printable density of the supporting elements. In this simplified AM process model, Langelaar (2018) defines that each printed density p(i,j,k) cannot be higher than the maximum printed density in its supporting region, pˆS:

$$p_{(i,j,k)} = \min(b_{(i,j,k)}, \hat{p}_S)\; with\; \hat{p}_S = \max\left(p \in S_{(i,j,k)}\right)$$

Or

$$\hat{\tilde{\rho}}_e = \min\left(\tilde{\rho_e}, \hat{\tilde{\rho}}_e^S\right), \hat{\tilde{\rho}}_e^S = max_{i \in S_e} \hat{\tilde{\rho}}_i$$

Various approaches to accommodating different overha angles include adjusting the aspect ratio of elements proposed by Yang et al., 2023) or employing interpolati techniques (as demonstrated by Delissen et al., 202 These methods exhibit promise for extension to thr dimensions (3D). The interpolation approach involv detecting a minimum of 5 nodes, with a preference 9 nodes, to ensure adherence to overhang limitations the 3D context.

The layer-by-layer function closely mirrors the / process, as it propagates only in directions printable fr its current position. This makes it a more intuitive choice, and it was initially explored for the continuous filter as well (Van de Ven et al., 2020). For more info on the use and exploration of this filter the reader is refered to chapter 6.

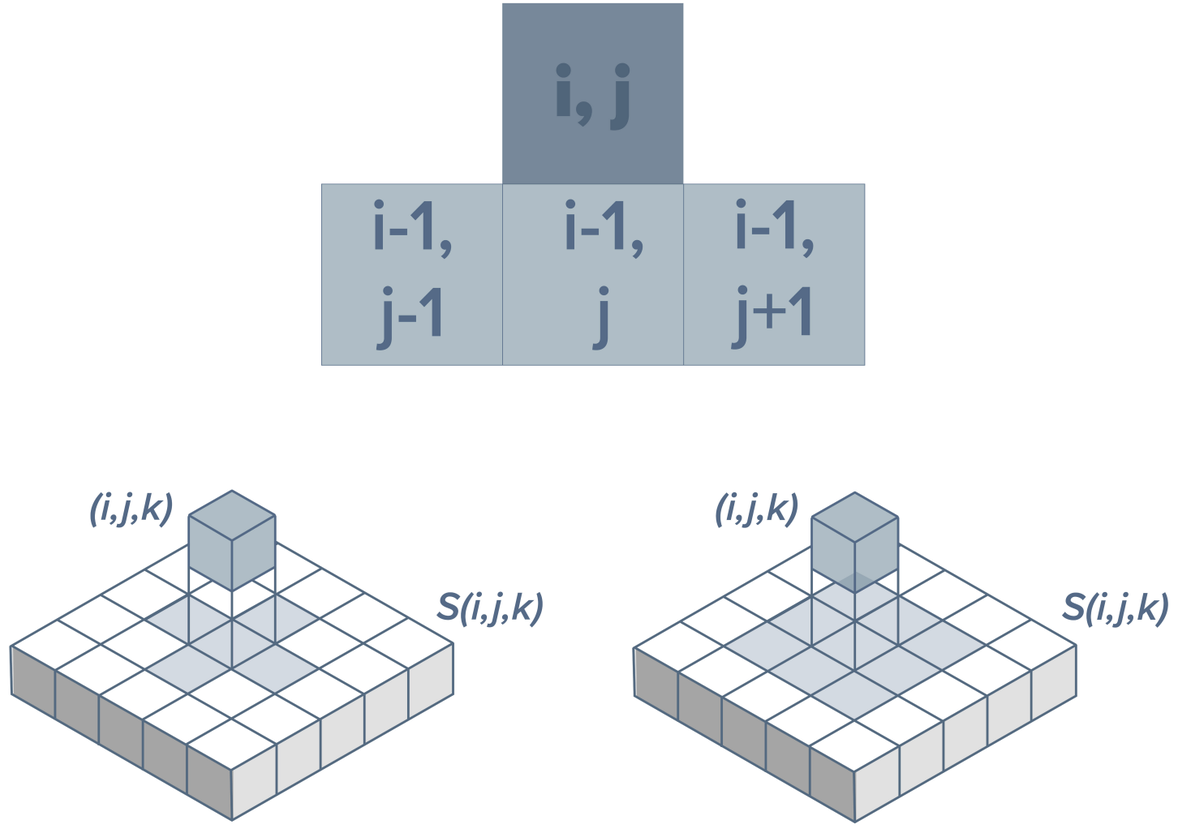


Figure 51: rework by author of working principle of layer by layer filter (Langelaar 2018)

5.9.3 Structural constraints

Implementing a stress constraint can be done by assessing the principal stresses for each individual element, yet this approach demands significant computational resources. The von Mises yield criterion is frequently employed to compute and mitigate peak stresses, especially suitable for materials exhibiting similar behaviour in tension and compression (Bruggi & Duysinx, 2012).

Dealing with brittle materials introduces distinctive challenges in optimization processes. In tackling these issues, a practical strategy involves employing the Drucker-Prager failure criterion. This methodology, as outlined by Bruggi and Duysinx (2012), Koniari (2022), and Schoenmaker (2023), offers a tailored solution to the distinctive properties and failure mechanisms inherent in brittle materials like glass. By incorporating the Drucker-Prager criterion into the optimization framework, a more nuanced and effective strategy emerges for handling the intricacies associated with these materials.
Composite materials can be anticipated by usin
g the Tsai-Wu criterion. The Tsai-Wu criterion serves as a predictive tool for anticipating the failure of composite materials, accounting for various failure modes such as tension and compression. This criterion employs a mathematical expression that incorporates material strength parameters and stresses, ultimately yielding a failure index. Notably, the Tsai-Wu criterion is designed to accommodate the anisotropic characteristics of composite materials, making it particularly suitable for the analysis of laminated composites. This comprehensive approach enables a more accurate assessment of potential failure scenarios in composite structures (Yang et al., 2023).
The principal stresses of a deformed three-dimensional hexahedral element can be derived from the stress-strain relationship. The principal stresses in 3D elements can be derived from the stress tensor in combination with the stress invariants. These invariants are given by (Chandrupatla & Belegundu, 2002):

$$I_1 = \sigma_x\sigma_y\sigma_z$$

$$I_2 = \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_x\sigma_z + r_{yz}^2 + r_{xz}^2 + r_{xz}^2$$

$$I_3 = \sigma_x\sigma_y\sigma_z + 2\tau_{yz}\tau_{xz}\tau_{xy} + \sigma_x\tau_{yz}^2 + \sigma_y\tau_{xz}^2 + \sigma_z\tau_{xy}^2$$

The Drucker-Prager failure criterion defines the feasible domain for a material based on an equivalent stress measure, denoted as σ^eq. This criterion is utilized to assess the material's ability to withstand loading conditions. The expression for the equivalent stress measure, σ^eq, encapsulates both tensile and compressive stresses, providing a comprehensive representation of the material's performance under various loading scenarios. The Drucker-Prager criterion establishes boundaries within this stress space, guiding the evaluation of material failure and ensuring that the optimization process adheres to realistic mechanical constraints. The feasible domain for the material is given by the equivalent stress measure σeq can be written as:

$$\sigma^{eq} = \frac{s+1}{2S}\sqrt{3J_{2D}} + \frac{S-1}{2S}J_1 \leq \sigma_{Lt}$$

In this context, s represents the ratio between compressive strength and tensile strength of the material The first stress invariant is denoted as J1, which is equivalent to I1. J_2D refers to the second deviatoric stress invariant, as defined by Liolios (2020):

$$J_2 = \frac{1}{3}I_1^2 - I_2$$
$$= \frac{1}{6}[(\sigma_x-\sigma_y)^2 + (\sigma_y-\sigma_z)^2 + (\sigma_z-\sigma_x)^2] + \tau_{yz}^2 + \tau_{xz}^2 + \tau_{xy}^2$$

5.10 Solver

Gradient-based methods typically involve the calculation of gradients or sensitivities with respect to the design variables. These gradients guide the optimization algorithm to iteratively update the design in the direction that reduces the objective function. These methods are part of the broader class of mathematical optimization techniques and include approaches like the Method of Moving Asymptotes (MMA), the Optimality Criteria (OC) method, and the Sequential Linear Programming (SLP) method.

Schoenmaker used the Fmincon solver from the optimization toolbox with is a solver build into MATLAB. This is a solver for finding the minimum of a constrained nonlinear multivariable function. It offers multiple different algorithms for both large-scale and small-scale problems (Schoenmaker, 2023).

5.11 Path filling

Analogous to a maze problem, the principle of toolpath generation is to fill a closed 2D region obtained from slicing using the shortest possible route. The performance of the constructed toolpath can be assessed from four aspects: underfilling, overfilling, number of sharp corners, and total path length (Yang et al., 2023). It is crucial to effectively combine the digital design of topology optimization and the AD technique to realize the maximum performance potential of manufactured structures.

Pastore et al. proposed a Bezier based biased random key genetic algorithm that addressed only the nozzle size and path-continuity constraints.

Bi et al. addressed various manufacturing constraints of 3DCP (3D concrete printing). However, there exist many short zigzag paths causing to accumulate concrete materials because the nozzle size constraint is not considered (Yang et al., 2023). Anisotropy has not been fully utilized, and path planning is based on optimized geometric shapes and cannot achieve collaborative optimization in the design process.
Bi et al. discretize four categories of path filling techniques for a continuous toolpath. In brief, these techniques will undergo examination and description, with the preferred method ultimately being selected.

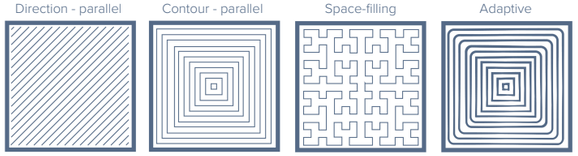


Figure 51: rework of author of different path planning

5.11.1 Direction-parallel paths

Direction-parallel paths, celebrated for their simplicity and computational efficiency, play a prominent role in Additive Manufacturing (AM) applications. Research endeavours in this domain concentrate on fine-tuning parameters such as path orientation and width. A common strategy to mitigate the staircase effect near boundaries involves the placement of contours near borders, with zigzag paths intricately embedded within them (Bi et al., 2022).

While direction-parallel path algorithms are esteemed for their straightforwardness and effectiveness, challenges such as dealing with sharp corners and addressing the persistent staircase effect near transitions pose ongoing concerns. Continuous advancements are imperative to enhance and refine the planning of direction-parallel paths in the context of additive manufacturing processes (Bi et al., 2022).

5.11.2 Contour-parallel paths

In contrast to direction-parallel paths, contour-parallel paths are distinguished by their ability to generate smooth curves that closely adhere to the boundary profile. The fundamental principle underlying the creation of contour-parallel paths involves the offset operation applied to boundaries. Various methods have been proposed in the literature and can generally be categorized into four groups: Voronoi diagram, level-set method, edge offset, and vertex offset (Bi et al., 2022).

A primary drawback of contour-parallel paths arises in scenarios involving underfilling and overfilling areas within the interior space. This occurs, firstly, when the dimensions of the 2D region are not multiples of the nozzle size, leading to innermost contours being too close or too far apart. Secondly, the offset operation introduces sharp boundaries, contributing to underfilling and overfilling areas due to the bottleneck effect. To address issues arising from sharp corners, Lin et al. introduced dual offset and clear-up toolpath procedures, resulting in a noticeable reduction in sharpness. However, the sharp corners were not eliminated (Lin et al., 2015).

Within complex 2D geometries, contour-parallel paths excel at volume-filling compared to direction-parallel paths due to their smoothness and compatibility with intricate boundary shapes. However, current continuous path algorithms in the literature still exhibit sharp corners and underfilling areas that require further improvement (Bi et al., 2022).

5.11.3 Space filling paths

Various types of space-filling paths have been proposed in the past. Space-filling paths, also known as space-filling curves, are mathematical constructs that traverse a 1-dimensional space in a way that efficiently covers or fills the entire space. These curves exhibit a unique property where they manage to fill a 2-dimensional or even higher-dimensional space through their continuous and self-replicating patterns. Some well-known examples of space-filling curves include the Hilbert curve, Peano curve, and Z-order curve. Spiral paths, a specific type of space-filling curve, are often utilized in numerically controlled machining due to their smoothness.
The benefits of using space-filling curves include their ability to reduce toolpath length, enhance print quality, and improve the efficiency of certain processes.
Although space-filling curves offer improvements in print quality and efficiency, their intricate shapes may impose constraints on computational efficiency and suitability in specific situations. Bedel et al. have introduced a closed space-filling curve with controllable printing orientation, thereby augmenting the filling capabilities and optimizing the path for Additive Manufacturing (AM) (Bedel et al., 2022).

However, it's essential to note that the complexity of space-filling curves in terms of shape may pose challenges in computational efficiency and limit their applicability in certain contexts (Bi et al., 2022).

5.11.4 Adaptive paths

Toolpath methods discussed in the preceding sections rely on uniform filament width and layer height. Another approach involves adaptive toolpaths, where feed rate control and non-uniform layer height are employed to optimize toolpath design.
Adaptive paths prove effective in reducing underfilling and overfilling by adjusting extrusion rates near problematic regions. However, they introduce additional complexity to toolpath design and may not be compatible with certain printer systems. Nozzle movement speed constraints limit the range of motion control, potentially destabilizing the printer beyond the maximum allowable printing speed. Optimizing toolpath geometry becomes crucial for addressing underfilling and overfilling issues that adaptive paths may not resolve (Bi et al., 2022).

5.12 Nozzle offset generation in topology optimization

It is crucial to effectively combine the digital design of topology optimization and AM technique to realize the maximum performance potential of manufactured structures. The integrated design considering the manufacturing constraints of AM enhances the manufacturability of the optimized structures without post-processing (Carstensen, 2020; Fernandeze et al., 2021; Yang et al., 2023). Although Carstensen and Fernandez et al. successfully considered the nozzle size constraint of FDM-AM, the anisotropy, path-continuity constraint, and overhang angle constraint, which are highly relevant for material extrusion-type additive manufacturing, have not been considered. Breakpoints are unfavourable and multiple start-ups and shutdowns will lead to unstable extruded materials or worse printing quality. Moreover, the removal of supporting materials is very difficult, which can easily cause structural damage (Yang et al., 2022). To generate a path for a TO structure Yang et al use the adaptive control of nozzle size for structural features is achieved through the medial axis transform (MAT), a technique applied to precisely manage the minimum and maximum length scales. The integration of adaptive nozzle size control into topology optimization involves various steps, including boundary extraction, medial axis extraction, width selection, path planning, and modification of design variables. Yang et al use the corresponding steps to integrate the nozzle size in the topology optimization;

1. To ensure smooth structural boundaries for medial axis extraction, an initial contour plotting algorithm is employed, followed by curve fitting using a B-spline curve of order 3 to achieve smoothness.
2. Following structural boundary extraction, the main force transmission path is obtained through medial axis extraction. Constrained Delaunay triangulation on a sample of points along the geometry boundary leads to the extraction of the medial axis. Undesired minor branches are removed, and any unconnected sections are extended until they precisely fit the structural form.
3. After obtaining the final medial axis, the structural form is modified by offsetting the adaptive width. The width of each segmented central axis is calculated using inner tangent circles at intervals along the axis. The K-medoids clustering analysis method is employed for accurate width selection, mitigating inaccuracies introduced by large widths at the axis ends. Silhouette values are examined to assess cluster quality, with values exceeding 0.8 indicating substantial separation. The offset of the segmented medial axis is then performed based on chosen width settings to derive the new structural form. Nozzle size control for each branch is expressed by the equation:

$$d_{new,i} = d_{p,i}, i = 1, 2, \dots, n$$
$$\left|d_{p,i} - d_{k,i}\right| = min_{j=1,2,\dots,m}\left|d_j - d_{k,i}\right|$$

n is the total number of branches, d_j represents discrete width values, and d_(k,i) is the clustered value for the ith branch's width obtained through the MATLAB function for K-medoids clustering. The final width (d_new) of the segmented central axis can be selected. Following adaptive width selection, offset path planning aligns optimized results with the actual printing path, crucial when design variables for printing angles are element-based. The last step involves modifying design variables, incorporating results into the iterative process, adjusting angles to fit the actual printing process, corresponding to the toolpath's slope. To prevent oscillation in optimization iterations due to abrupt changes in element density (0–1), the initial element density is averaged with the density obtained through medial axis offset, serving as the initial value for subsequent iterations. This process successfully integrates nozzle size control into the optimization iteration, yielding results after repeated iterations (Yang et al., 2023).

5.13 Path continuity

It cannot be emphasized enough for certain materials like glass that discontinuation of the print process can result in imperfect prints or even outright failures (Yang et al., 2023). Therefore, it is crucial to incorporate strict continuation rules within the algorithm. Despite extensive literature review, no sources have been found reporting on print path continuation with glass. Similarly, concrete exhibits similar limitations in this regard. Bi et al. (2022) and Yang et al. (2023) have tried to integrate path continuation into their topology optimization methodologies.

The literature on structure optimization utilizing concrete often relies on the vase model principle, wherein the outline of the structure is initially printed in a single extrusion, and later the mold is filled with concrete, sometimes with reinforcement. However, employing vase mode, followed by filling, introduces significant stresses in glass, rendering it unsuitable for glass printing.

Bi and Yang have proposed incorporating a path reconnection mechanism within MATLAB to address this issue. They assess, for each layer in the topology optimization, Bi et all.,(2022) uses '1' elements due to the use of BESO. The label matrix function in MATLAB is utilized to label the different boundaries in each layer. Subsequently, the MATLAB code connects the isolated regions based on their proximity and position relative to other regions. Once a region meets a certain threshold, it is filled with material ('0' are changed to '1'). This process continues until all boundaries are interconnected.

The labelling function in MATLAB offers the option to base connections on either the four corners of each element or the eight surrounding elements, including diagonals. However, for the purpose of printing, the latter possibility is excluded.

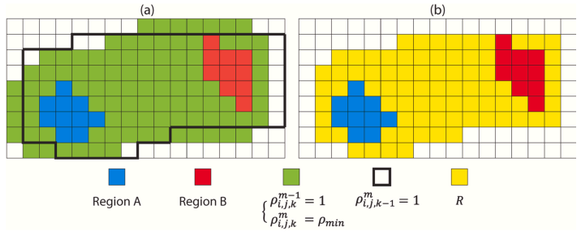


figure: MATLAB labelmatrix where the different boundaries are labelled (Bi et al.,(2022)

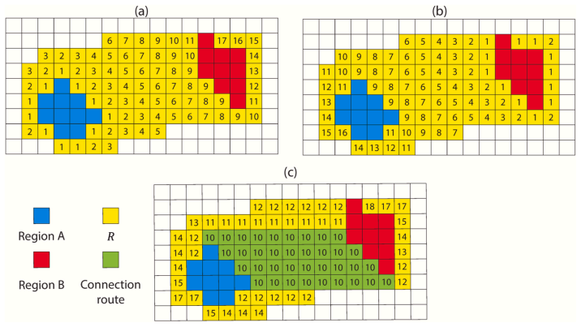


Figure 52: MATLAB counting the distance from each boundary and finding the lowest numbers, assigning value to the green 'connection route' (Bi et al.,(2022)

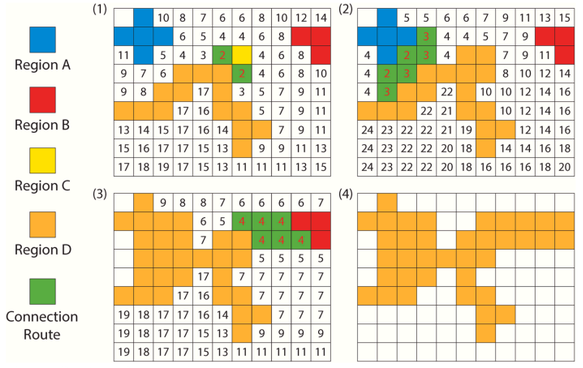


Figure 53: MATLAB configuring different boundaries and reconnecting them (Bi et al.,(2022)

06

Development of the algorithm

Progress and developments of the topology optimization algorithm for 3D printed glass

6.1 Introduction

This chapter explores how algorithms for topology optimization have evolved over time. It starts by looking at the early stages, when different parts of the algorithm were developed separately to solve specific problems, like dealing with overhangs in 2D and 3D designs.

One big challenge in topology optimization is handling overhangs, especially in 3D designs where you have a lot of freedom in shape. At first, algorithms were made to deal with overhangs separately in 2D and 3D. Then, a breakthrough came with adaptive overhang angles in 3D optimization. This allowed the algorithm to adjust overhang angles automatically,.

Another improvement was in managing the size of the printing nozzle. It was found that by adjusting the boundary of the design based on nozzle size, they could make the transition from design to actual object smoother, avoiding problems like printing errors and weak spots in the structure.

As the chapter goes on, it shows how these different parts of the algorithm were brought together into one unified system. The quest for smooth, continuous shapes in topology optimization also became important. This led to algorithms that could create shapes without any breaks, making structures stronger and reducing waste material.

Overall, this chapter highlights how the process of developing algorithms for topology optimization is done. Each improvement, from dealing with overhangs to optimizing nozzle boundaries and creating continuous shapes, adds to the overall progress of a topology optimized glass design.

As previously outlined, in the context of topology optimization, there are three function options which can be minimized: Compliance, Volume, and Stress.

In this particular master's thesis, the decision was made to focus on minimizing compliance (maximizing the global stiffness).

This choice was justified by the experimental nature of topology optimization for glass and the inherent brittleness of the material. Consequently, volume optimization was deemed unsuitable as it could not guarantee the structural integrity of the entire structure. Additionally, there wasn't an approved function available for stress minimization specifically tailored to glass structures.

Hence, minimizing compliance was deemed the most appropriate approach for this thesis.

$$
\begin{aligned}
min_x \quad & : c(x) = U^T K U \\
subject\ to \quad & : \frac{V(x)}{V_0} = f \\
& \quad KU = F \\
& \quad 0 < xmin \leq x \leq 1
\end{aligned}
$$



6.2 Introduction to design domain specifics

For the algorithm development, I opted to use the classic MBB (Moving Morphable Barrier) beam as the basis. This structure is well-documented and commonly used in topology optimization examples. I implemented three variants of this beam to demonstrate the capabilities and outputs of the algorithm.

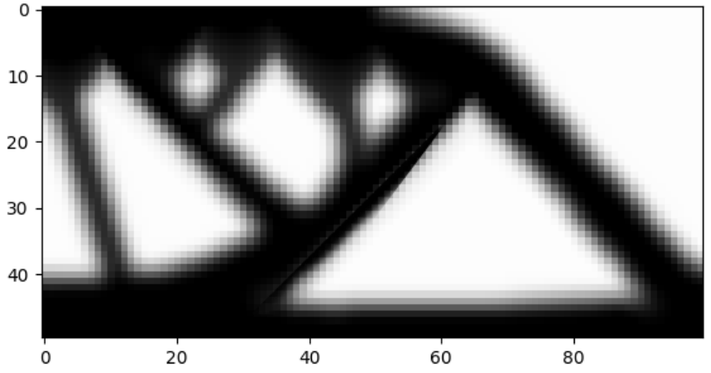
These design domain choices were made to efficiently illustrate various aspects of the algorithm's performance and to ensure that the computations could be handled effectively on my laptop. Increasing the domain sizes for final testing allows for a more comprehensive evaluation of the implementations.

1. Optimization Domain 1: 30 x 30 x 10. This domain was selected primarily for the increased height (10 layers), which effectively showcases the differences in element dimensions.

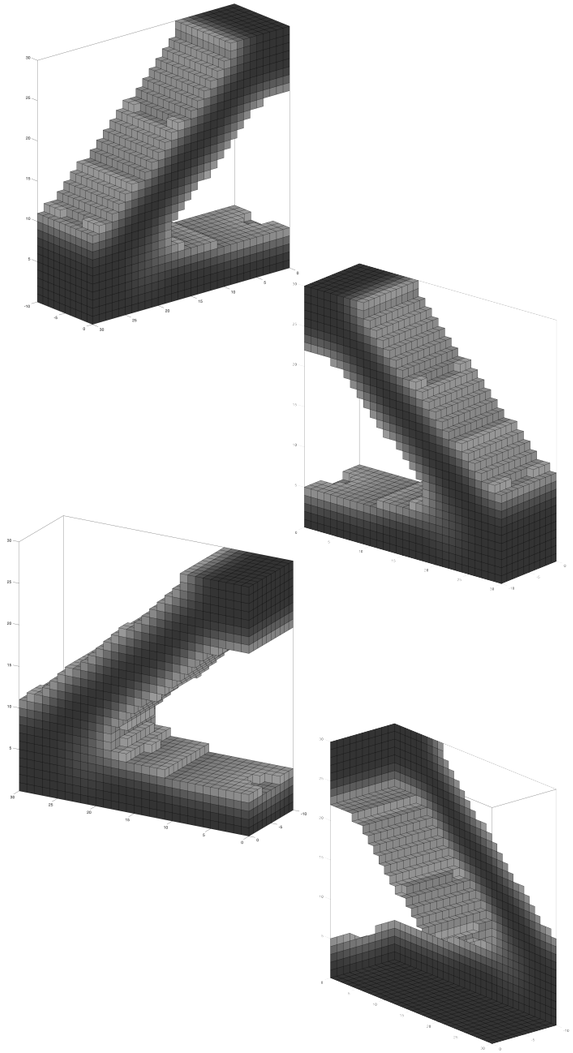
2. Optimization Domain 2: 30 x 10 x 10. This domain has a height of 10 layers, making it manageable, and it demonstrates how the shape is prone to the islanding effect in the y-direction when the overhang filter is not included.

3. Optimization Domain 3: 50 x 100. This is a 2D domain used for visualizing the path inclusion, which aids in making the steps more readable and understandable. The variant in line count and contouring helps clarify the process.

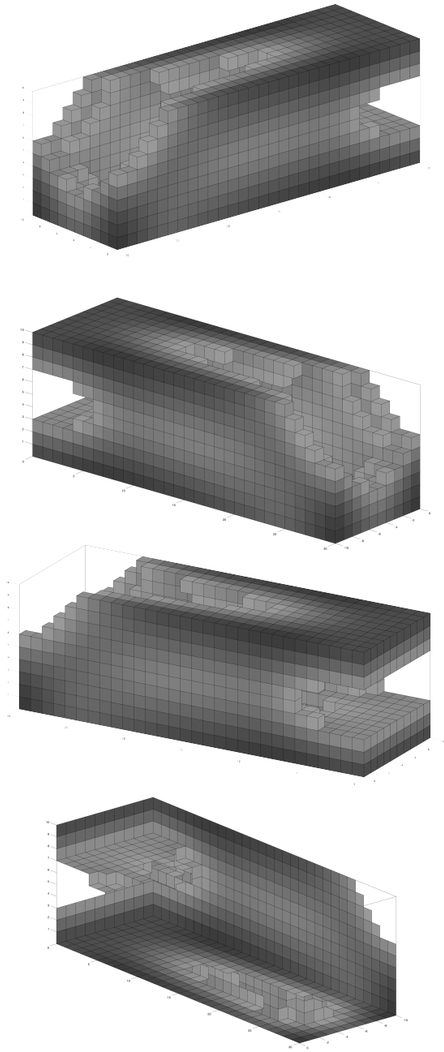
Optimization domain 3



Optimization domain 1



Optimization domain 2



6.3 Introduction to setup complete step by step algorithm

Initialization

Clearance

Clear all existing commands
clear all existing windows
Get screen information

Set visualization and outcome settings

Boundary

Define parameters for the design domain, loads, material properties
and loop parameters

Loop / material conditions

Define loop parameters and material properties like Young's modulus and Poisson's ratio

Preprocessing

Overhang setup

Calculate the overhang angle based on user-defined input

Adjust the mesh grid size based on the overhang angle

Initialize setup

Initialize FEA parameters like degrees of freedom, stiffness matrix
and element properties

Prepare the filter for sensitivity analysis.

Optimization

Optimization loop

Iterate through the optimization process until convergence or maximum iterations

Perform FEA, calculate objective function (compliance), and update design variables based on the Optimality Criteria method

Overhang loop

Optionally itterate through overhang filter and rearange xPhys

Islanding loop

Optionally calculate continuous path using Dijkstra's algorithm

Post processing

Export

Export objective graph, connection png, stl converions and 3D Matlab figure

Mirror the element,
(optional Dijkstra algorithm),
converge to STL (cube and ISO)

Optimization values

Measure and print the performance metrics such as total time, time for specific tasks, and iteration count

Offset nozzle path

Slice element, offset boundaries to nozzle size, export to new element with nozzle boundaries

Fabrication

Slicing

Export new STL file, create g-code

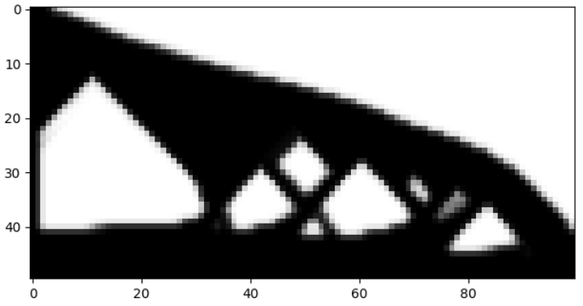
Part creation

3D-print

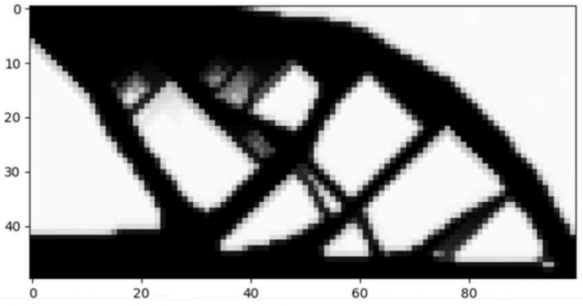
6.4.1 2D overhang constraint

As previously mentioned, two distinct approaches were identified for adjusting the overhang angle in a 2D topology optimization. Initially, setting up both algorithms required some effort, but after straightforward implementations, they proved highly effective, particularly when utilizing the MBB beam and the standard 45-degree overhang angle.

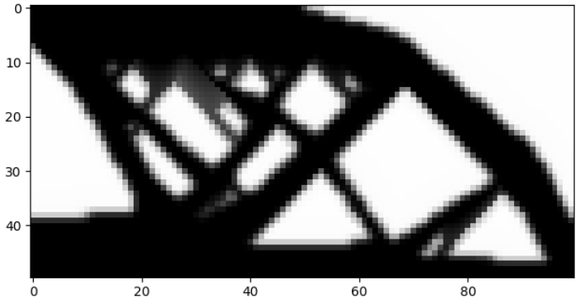
While the methods operate with slight variations, readers interested in their underlying principles are directed to sections 5.9.2.2 and 5.9.2.3. Ultimately, the Layer-by-Layer overhang angle was also incorporated for use in 3D topology optimization.



Failed overhang filter



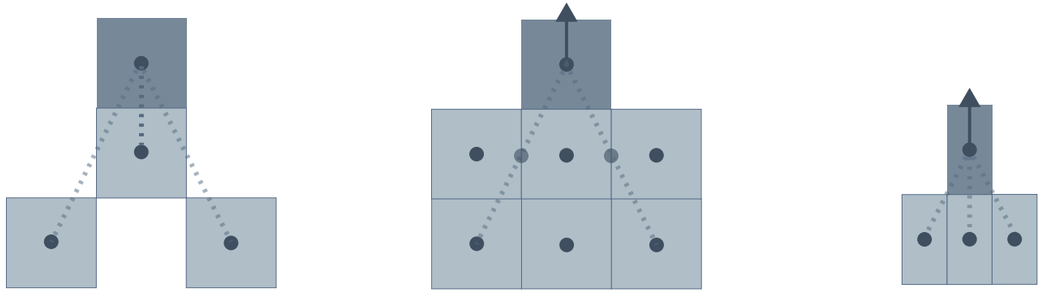
Front propagation 2D 45degree overhang filter



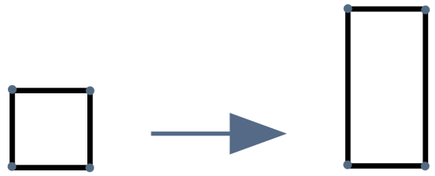
Layer by layer 2D 45degree overhang filter

6.4.2 2D overhang constraint adaptation 60°

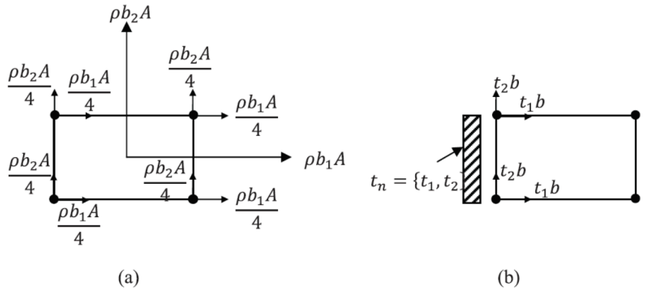
For altering the overhang angle in 2D topology optimization, three different approaches can be explored (refer to figure:). Numerous hours and attempts were devoted to this task, but progress remained elusive. It wasn't until a meeting was arranged with Matthijs Langelaar, the developer of the 2D layer-by-layer filter, that a breakthrough occurred. Langelaar suggested modifying the linear quadrilateral elements, a method initially dismissed based on recommendations from Charalampos. However, as the project had transitioned to 3D topology optimization and considering time constraints, adjusting the linear quadrilateral elements in the 2D space was excluded from the objectives of the master's thesis.



Three approaches to change the overhang angle for 2D topology optimization



Linear quadrilateral elements



Linear quadrilateral elements

6.3 3D overhang constraint

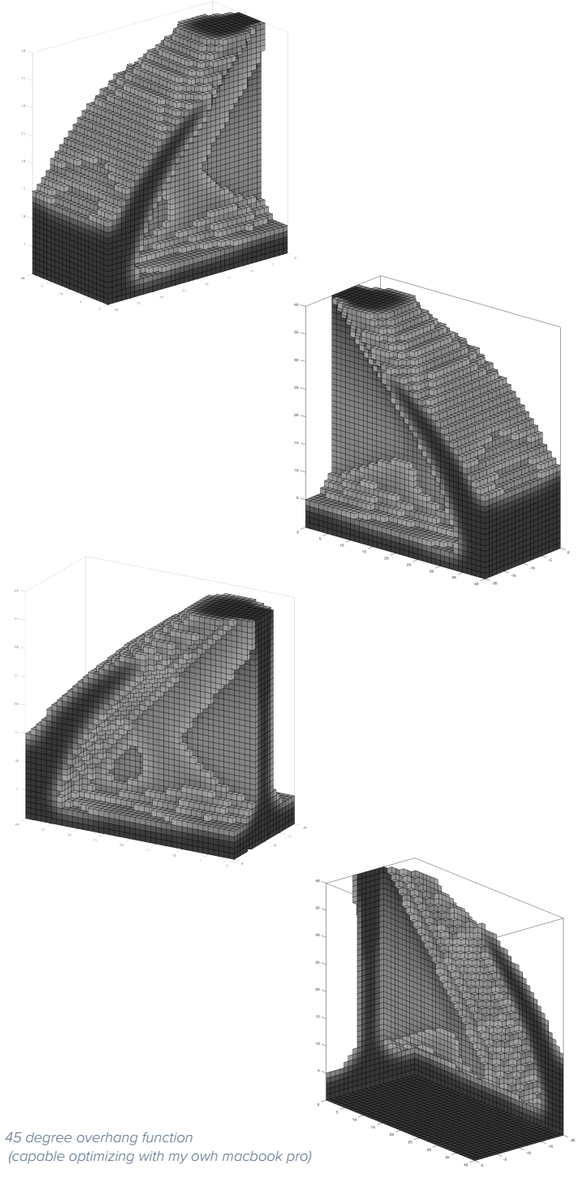
Implementing the 3D functionality proved to be quite challenging. A thorough comprehension of the algorithm's steps is essential. Eventually, the filter is integrated into the algorithm.

For 2D considerations, a ramp is introduced alongside the filter, allowing the design domain to establish its initial shape independently of the filter's influence on element values. Consequently, the objective chart often shows a steep decrease until iteration 10. During this phase, the filter prioritizes elements less crucial for overall strength, enabling the design domain to solidify its shape with minimal compliance.

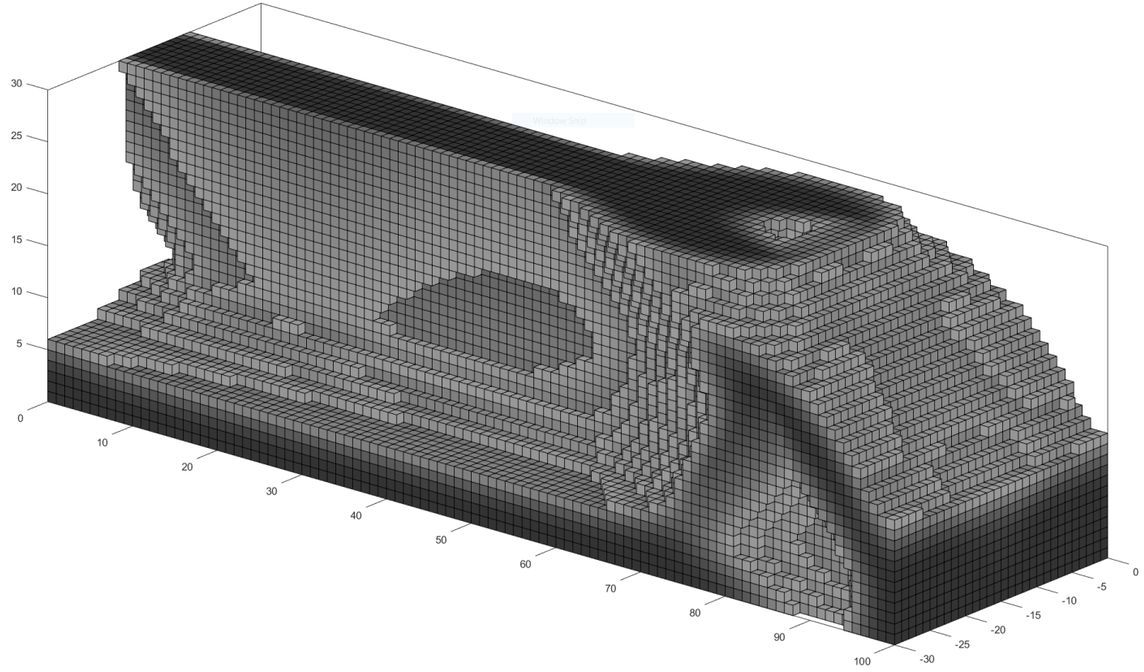
While the introduction of this filter marginally slowed down the algorithm, typically wihtin 1-15%, the main concern lies in the volume allocation within the design domain. Inadequate volume boundaries may lead to suboptimal outcomes in topology optimization. An experienced individual is necessary to ensure the compatibility of these outcomes with established boundary conditions and load/support settings.

In many cases, the overhang filter eliminates the need for the path continuity tool since the shape naturally maintains continuity across layers, thereby reducing the time spent on algorithmic reconnections within the Dijkstra algorithm function.

In the following pages, the adjustment of the overhang angle is observed to be contingent upon the size of the elements. In instances where extreme design boundary conditions are selected, the algorithm occasionally produces undesired outputs. This occurs because the surrounding elements, although necessary for structural performance, are included primarily due to the filter.



45 degree overhang function
(capable optimizing with my owh macbook pro)



45 degree overhang function

6.4 3D overhang constraint adaptation 60°

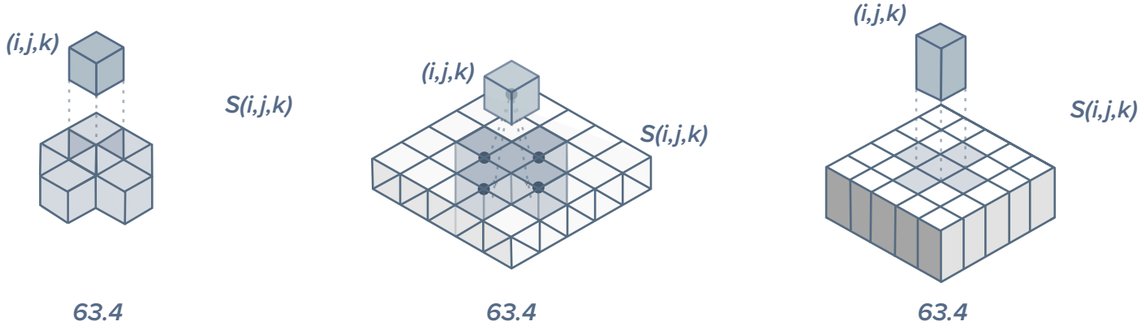
To alter the 3D overhang angle, the linear quadrilateral elements are adjusted. Similarly to the 2D case, three different approaches were identified in the literature, all showing promise. Langelaar suggested altering the size of the elements, which was then implemented. In the provided figure, this is exemplified by the implementation on the far right, showcasing a 63.4-degree angle.

The value of 63.4 arises from the distance between the center points of each element. When the height of the element is doubled, the center point scales accordingly. Consequently, the angle between the nodes to the left and right of the desired node is adjusted to 63.4 degrees.

This method was eventually embraced by formulating a formula to adjust the element sizes, enabling the attainment of various angles through this approach. The height of the four top Gaussian points was determined by 1/tan α. When the angle α is introduced, the elements resize accordingly, resulting in different overhang angles. It was found that the G3DP2 printer could achieve a maximum overhang angle of 60 degrees as viewed from the base plate, which became the target angle. Since 63.435 degrees or 26.565 degrees respectively double or halve the height of the original element, they facilitated a clearer overview of the grid, revealing the total design domain boundaries post-optimization.

```
%Setup of the overhang angle of the Finite Elements
A_deg = 45;      %Angle in degrees 63.435 = 2, 45 = 1, 26.565 = 0.5
A_rad = deg2rad(A_deg);
ws = tan(A_rad);
```

Simple lines of code to change the distance the points move up or down

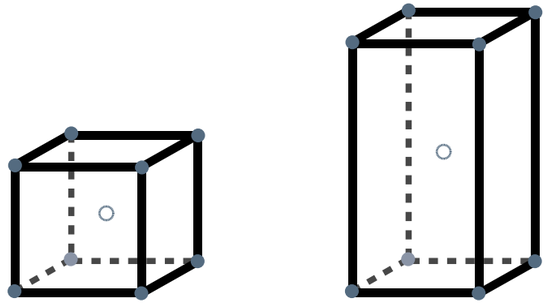


The 3 possible solutions of changing the overhang angle to bigger than 45 degree

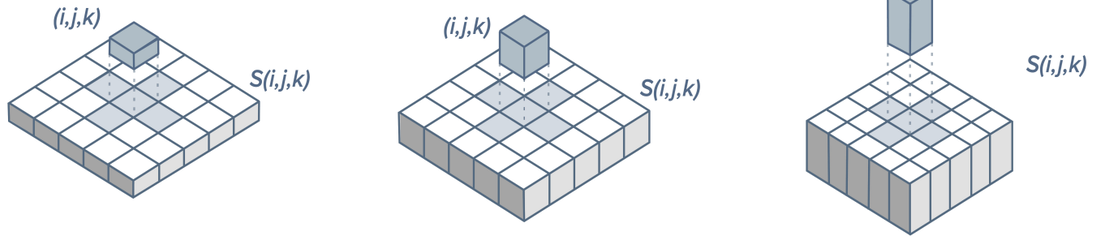
6.4 3D overhang constraint adaptation 60°

Thus, primarily 63.435 degrees is utilized, though other variables are also viable (as discussed later in this chapter).
However, since this method alters the element sizes, the total number of elements within the design domain changes unless the domain itself is adjusted accordingly.These adjustments are made in the code. The adjustment for the sizing are also controlled in the visualization outputs and the STL formats (for printing conversion).

I've selected this particular design domain to showcase overhang differences due to its clear element height and simplicity, the amount of ellements ensured smooth operation / calculation time on my MacBook Pro.
(Intel i9, 2.3GHz 8-core, 16GB 2667MHz DDR4).

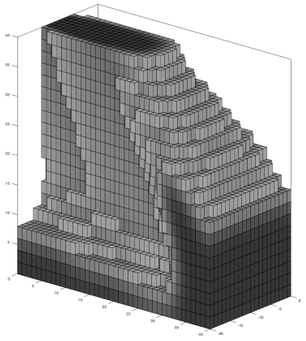


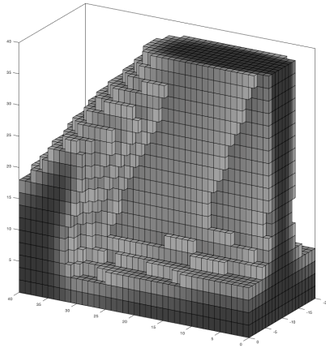
This diagram illustrates the difference in centerpoint angles (63.4 degrees) when the height is doubled compared to the length or width.



The addaptation that is used for the overhang problem

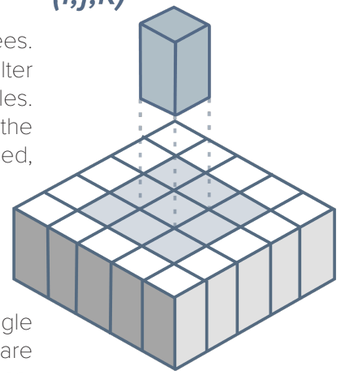
6.4 3D overhang constraint adaptation 63.435°

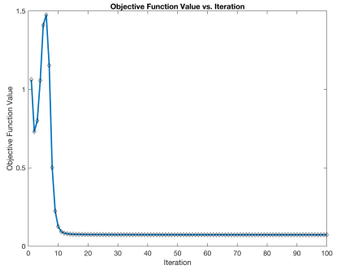




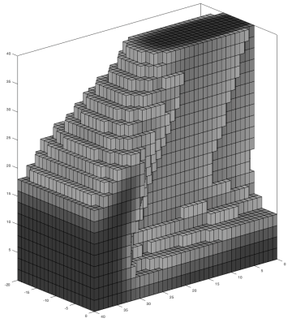
The overhang angle of 64.435 degrees. It's evident that computations for the filter are slower compared to smaller angles. However, within the design domain, the number of elements is also reduced, resulting in a faster total computation time for the same design domain. The detailing of the total shape is lower than with lower overhang angles. To adhere to the recommended 60-degree angle, using this 64.435 angle is advised, as it ensures the elements are twice as high as they are wide or long. This enhances visual appeal and clarity.

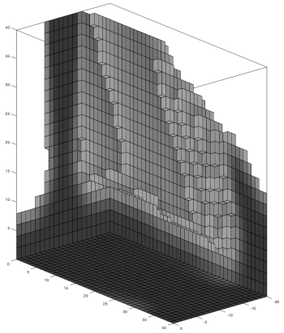
(i,j,k)



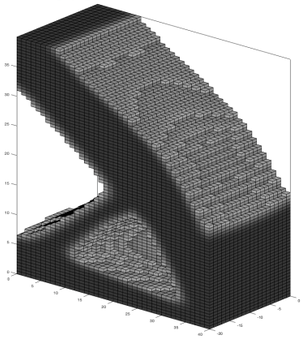


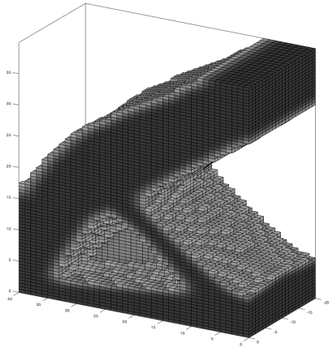
63.4





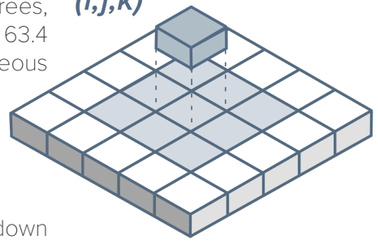
6.4 3D overhang constraint adaptation 26.6°

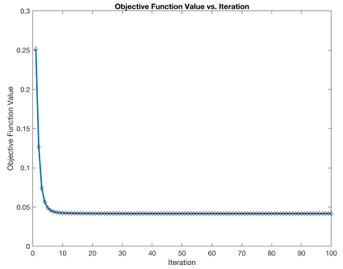




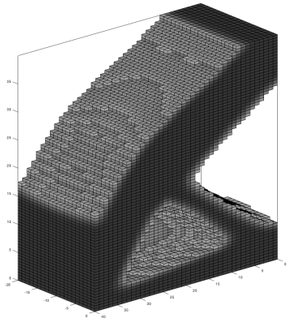
The overhang angle of 26.565 degrees is notable. It's apparent that computations are marginally swifter compared to 45 degrees, and progressively quicker than at 63.4 degrees. This renders it an advantageous overhang angle, provided the material and printing method permit. The algorithm leans towards employing a slightly lower element height. While this may increase the total number of elements, it doesn't necessarily slow down the algorithm. In fact, it could lead to a more sophisticated outcome across the entire design space.

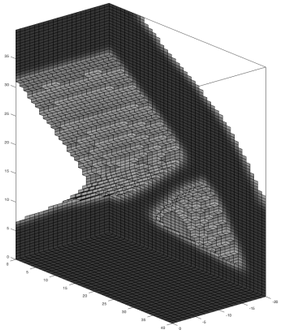
(i,j,k)



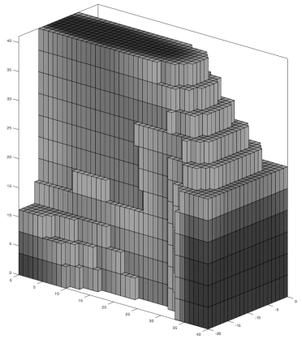


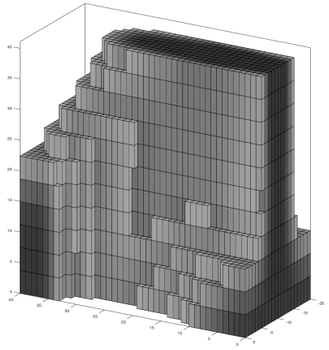
26.6



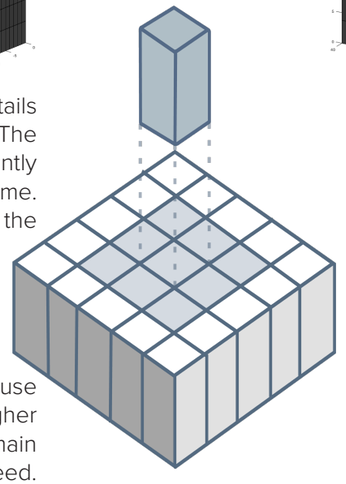


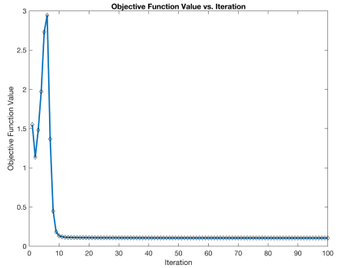
6.4 3D overhang constraint adaptation 75°

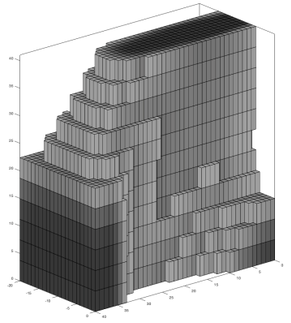


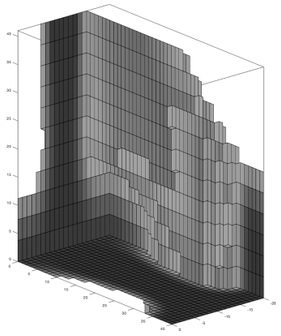


With a height of 3.7 per length, the details of the overall shape are limited. The number of elements is also significantly reduced, which reduces calculation time. The objective graph really shows the implementation importance of the overhang angle. Because the crucial number of elements to be kept for each element is also lower. For the use of the overhang filter, the recommendations would be not to use this implementation for many cases higher than 65°, except for the design domain where there is a relatively high speed.

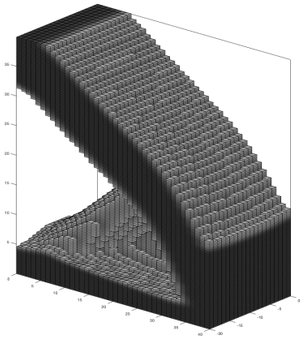


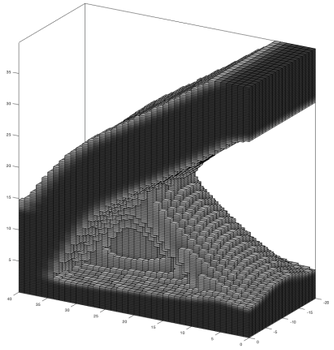




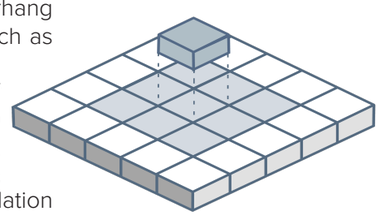


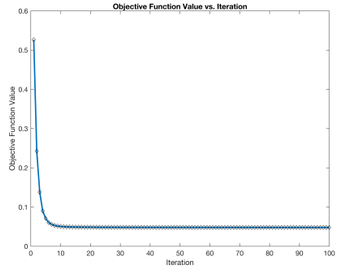
6.4 3D overhang constraint adaptation 15°

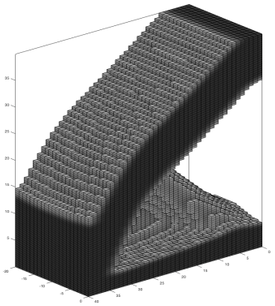




With only a degree of 15° the algorithm has a lot of freedom of creating the desired shape it wants therefore the overhang filter isn't restricting the shape as much as higher overhang angles. The design domain is populated with a lot of elements as the height only is 0.2679 while the lenght and depth are still 1. This demonstrates nearly 4 elements for each normal 1 x 1 x 1 cube. Calculation time was increased significantly due to the increase in element numbers.







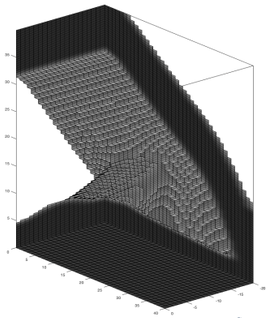
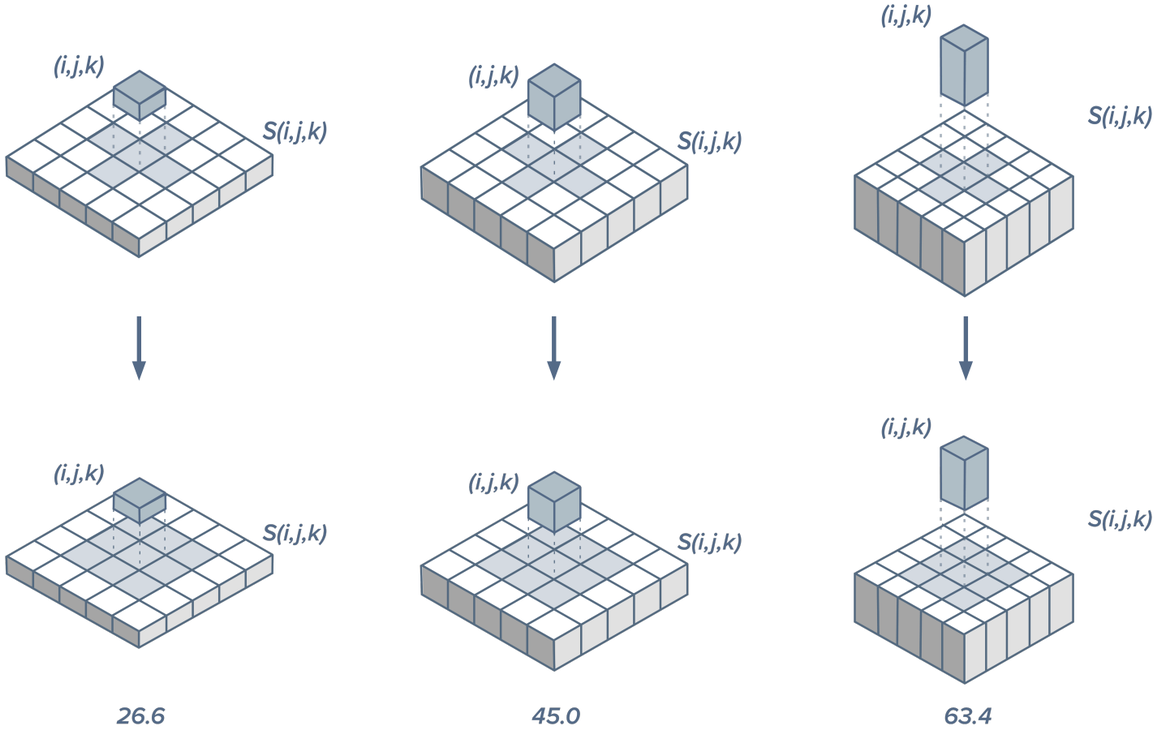


figure: futuristic 3D printer (author)

6.4 3D overhang addition

To enhance the precision of the AM overhang filter, the code could be modified to examine not just the 5 elements below the current element, but rather 9 elements. This adjustment would marginally increase the computational time required for calculations, as each element would need to consider 4 additional elements. However, it would significantly enhance the freedom in shaping the final structure, as there are parallel pathways for growth associated with every element.

While the use of 5 elements sufficed for this master's thesis, expanding to 9 elements would offer better optimization outcomes Thus, in 'real life' applications, this implementation would be advisable for achieving more accurate outcomes. And to ensure that the min objective is achieved, As for now the freedom of the elements can eliminate this minimun.



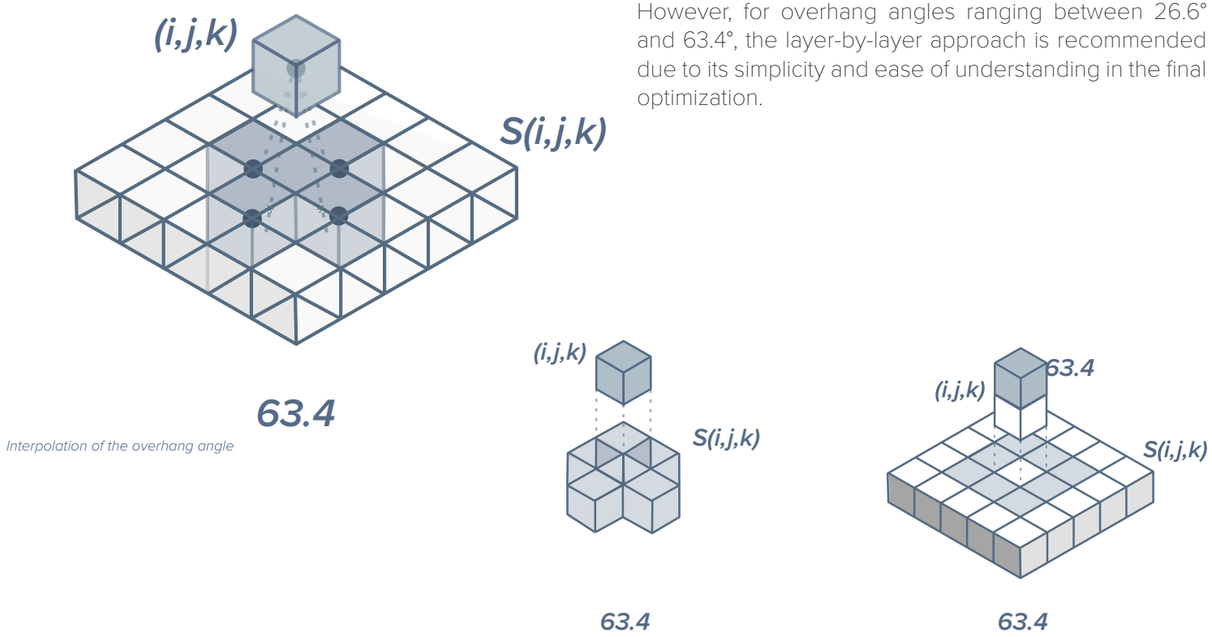
Checking for 9 neigbours instead of 5

6.4 3D overhang conclusion

The utilization of the overhang angle is considered crucial for the feasibility of the 3D printing process, particularly achieving overhangs greater than 60°. In the literature, there are various methods proposed for adjusting the concluding angle of the overhang within topology optimization. Langelaar noted the surprising diversity of these methods amidst other approaches. One example discussed was checking for elements higher than the original one, but it was deemed impractical as it could lead to unsupported elements, potentially compromising the overall optimization.

During meetings with Charalampos and Matthijs, several approaches to modifying the overhang were explored. In the subsequent chapter, the performance and outcomes of the layer-by-layer overhang filter will be detailed. However, it's worth noting that this approach may not be applicable in all cases, especially when the overhang angle deviates significantly from 45°, as it can greatly increase computational time or reduce the precision of the final outcome.

Therefore, when dealing with extreme variations in overhang angles, it's advisable to consider alternative methods such as the propagation filter or interpolation. However, for overhang angles ranging between 26.6° and 63.4°, the layer-by-layer approach is recommended due to its simplicity and ease of understanding in the final optimization.



Interpolation of the overhang angle

Langelaar himself discourages the approach of checking for element +1 due to its complexity later in the algorithm.

6.5 Path continuity

In the realm of 3D printing, meticulous consideration must be given to the print path. Section 5.13 delves into the intricate rationale behind this necessity. For materials like glass, halting the print mid-process can lead to catastrophic failure, primarily due to the discontinuity between the stop and start points of the printing operation. The procedure involves lowering the temperature with blown air, followed by reheating the nozzle to facilitate the flow of glass. Such interruptions can compromise the strength, surface finish, and overall completion of the final printed object. Thus, ensuring connectivity is imperative when engaging in 3D printing with glass materials.

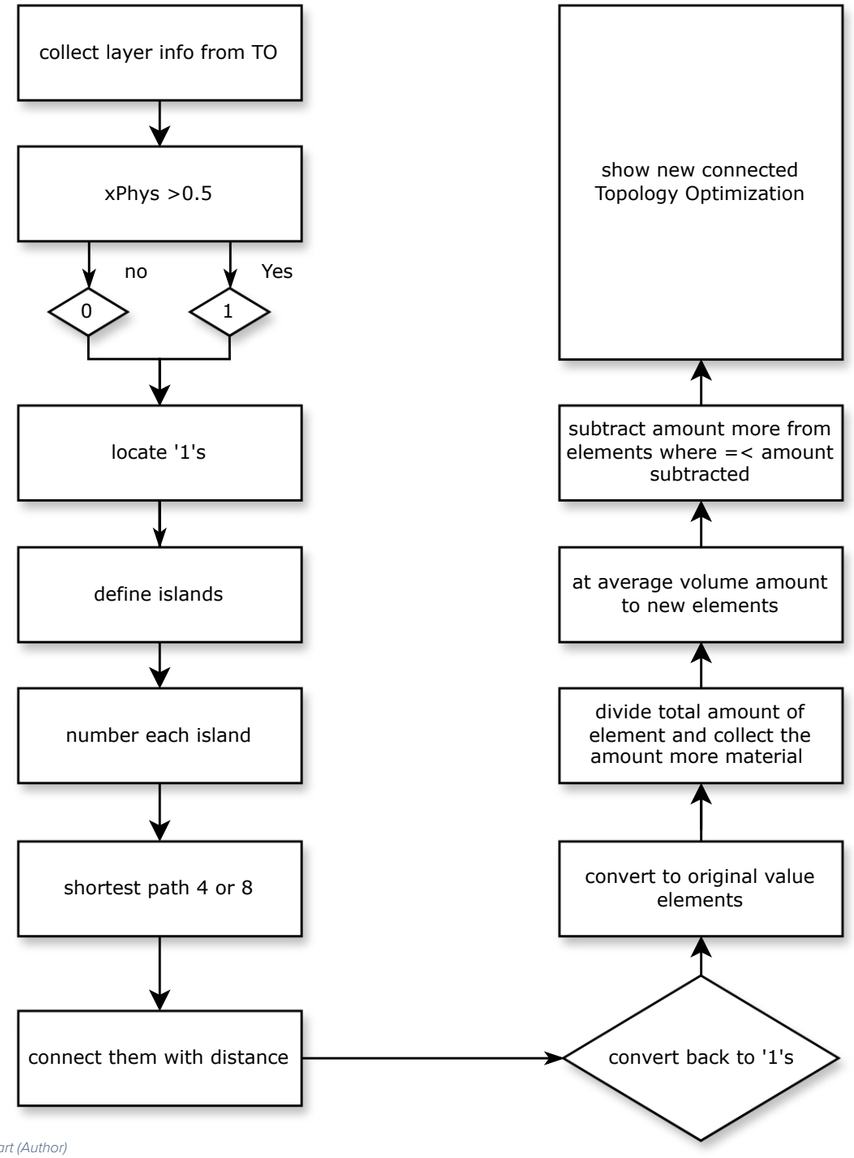
Initially, the implementation strategies of Bi et al. and Yang et al. were explored. Despite slight disparities in their descriptions, the approach involving the utilization of MATLAB remained consistent, owing to the common application of techniques such as SIMP and BESO. The schematic on the right page illustrates a structured sequence devised and interpreted by the author for both implementations.

The MATLAB plugin was employed to assign numerical labels to various islands, subsequently reconnecting them using methods like 'Euclidean, cityblock, chessboard, or quasi-Euclidean.' The script was crafted based on these principles, and after initial testing and minor adjustments, it operated smoothly and efficiently. The processing speed for small iterations (<100 blocks) was approximately equivalent to that of a regular iteration, while for iterations involving >100 blocks, it could be up to 1/6 of the time taken for a standard iteration (particularly when combined with the overhang filter, accelerating the connectivity algorithm even further).

Initially, the script designated connection elements with an average value of elements per layer greater than 0.5, marking them as 'grey' elements. However, these elements were sometimes discarded from the design domain after just one iteration. This aspect of the algorithm, where the average per layer is established and integrated into the connection nodes, was eliminated, and the nodes were assigned a value of 1. Despite this adjustment, the script typically removed the connections between 3 to 5 iterations, as these connection nodes were not deemed structurally critical.

However, a certain issue emerged over time. In cases where the design domain was relatively sparse, and the islands were equidistant from each other, the algorithm tended to fill the entire space with elements of uniform spacing. This inadvertently resulted in a design domain with significantly increased volume. Consequently, an alternative algorithmic approach was pursued, as detailed in the subsequent chapter.





Flowchart: path continuity chart (Author)

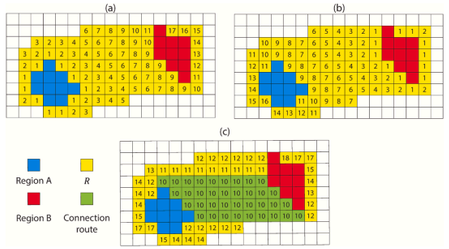


Fig. 3. Demonstration of how the shortest possible route is found within the potential area for reconnection: (a) the distance map of region A; (b) the distance map of region B; (c) the sum of the two distance maps; the shortest connection route is located by adding elements with the lowest summed distance value.

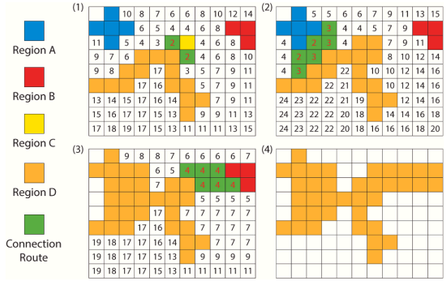


Figure 54: Connection by MATLAB function Bi et al., 2023

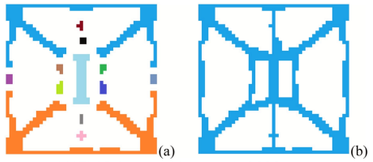


Figure 55: Connection by MATLAB function Yang et al., 2023

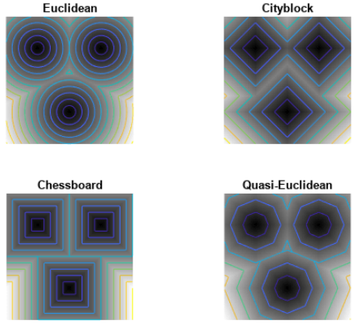


Figure 56: Connection by MATLAB, MATLAB connection

Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma

Topology optimization for 3D concrete printing with various manufacturing constraints

Minghao Bi [a], Phuong Tran [a], Lingwei Xia [b], Guowei Ma [b], Yi Min Xie [a,*]

[a] Centre for Innovative Structures and Materials, School of Engineering, RMIT University, Melbourne, Victoria 3001, Australia
[b] Tianjin Key Laboratory of Prefabricated Building and Intelligent Construction, School of Civil and Transportation Engineering, Hebei University of Technology, Tianjin 300401, China

Contents lists available at ScienceDirect

Engineering Structures

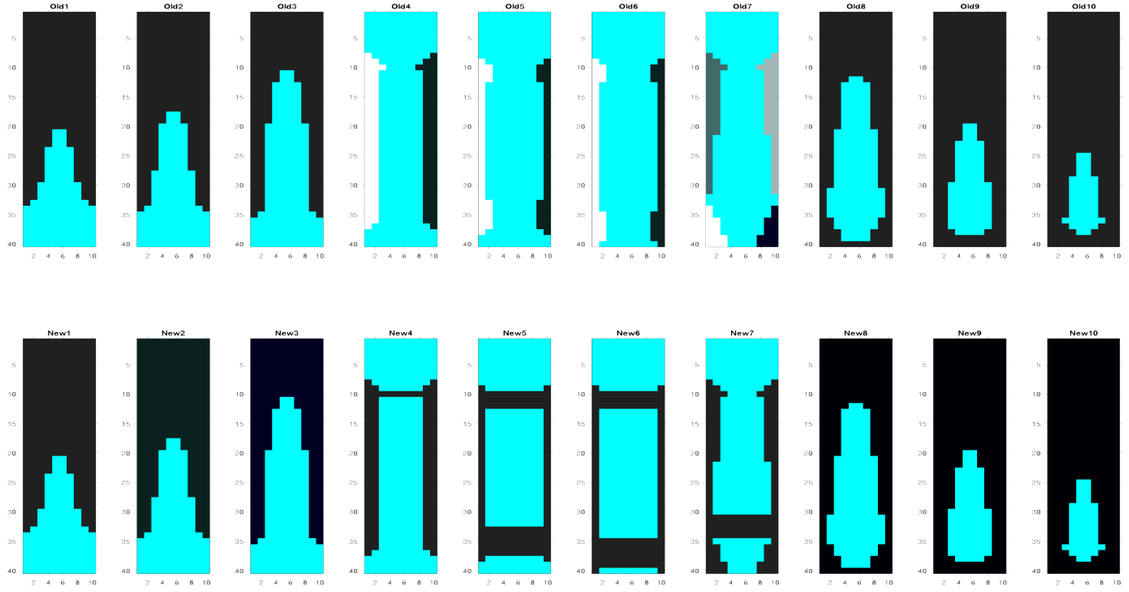
journal homepage: www.elsevier.com/locate/engstruct

An integrated method of topological optimization and path design for 3D concrete printing

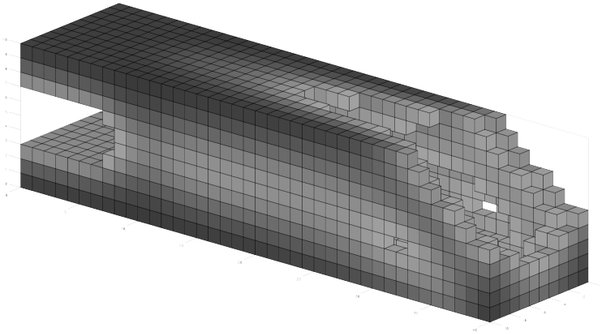
Wenwei Yang [a], Li Wang [a,b,*], Guowei Ma [a,b,*], Peng Feng [c]

[a] School of Civil and Transportation Engineering, Hebei University of Technology, 5340 Xiping Road, Beichen District, Tianjin 300401, China
[b] Tianjin Key Laboratory of Prefabricated Building and Intelligent Construction, Hebei University of Technology, 5340 Xiping Road, Beichen District, Tianjin 300401, China

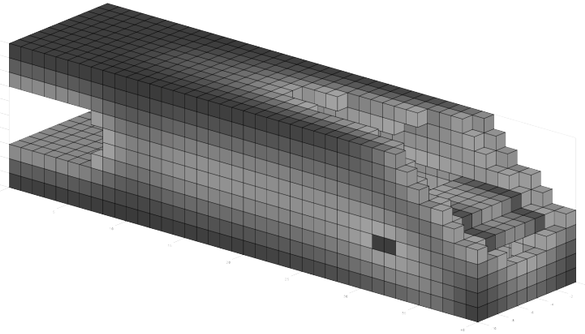
overview of connection by layers Topology Optimization (MBB beam)



Not connected Topology Optimization (MBB beam) volume 0.5



connected Topology Optimization (MBB beam) volume 0.517



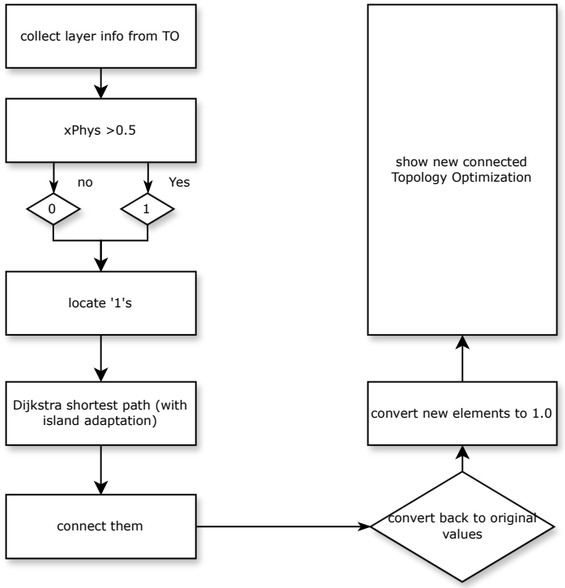
6.6 Path continuity Dijkstra

When establishing connections between topology optimization islands, the earlier method utilizing the MATLAB labeling algorithm and reconnection functions exhibited limitations, particularly when dealing with lower volume design domain boundaries. This issue arose from the algorithm's tendency to prioritize connectivity based solely on the shortest path, resulting in excessive material introduction—an outcome often unnecessary for enhancing the strength of topology optimization designs.

To address this challenge, a different approach involving shortest path algorithms was explored. These algorithms, such as Dijkstra's algorithm, A* search, and Johnson's algorithm, were considered during the master's thesis research. Ultimately, Dijkstra's algorithm was selected due to its well-documented examples and straightforward working principle.

The script was enhanced to organize the islands differently. Instead of searching from one node to another node within the design domain, the algorithm was modified to start from the endpoints of the islands and then identify the shortest routes to connect them with other islands. This iterative process continued until all islands were successfully reconnected.

This revised implementation significantly improved upon the previous method, consistently delivering fast and accurate results by establishing the shortest routes between different islands. Notably, this approach proved to be more efficient than the MATLAB functions, mainly because it focused solely on connecting relevant elements ('1's) while disregarding non-relevant areas ('0's).

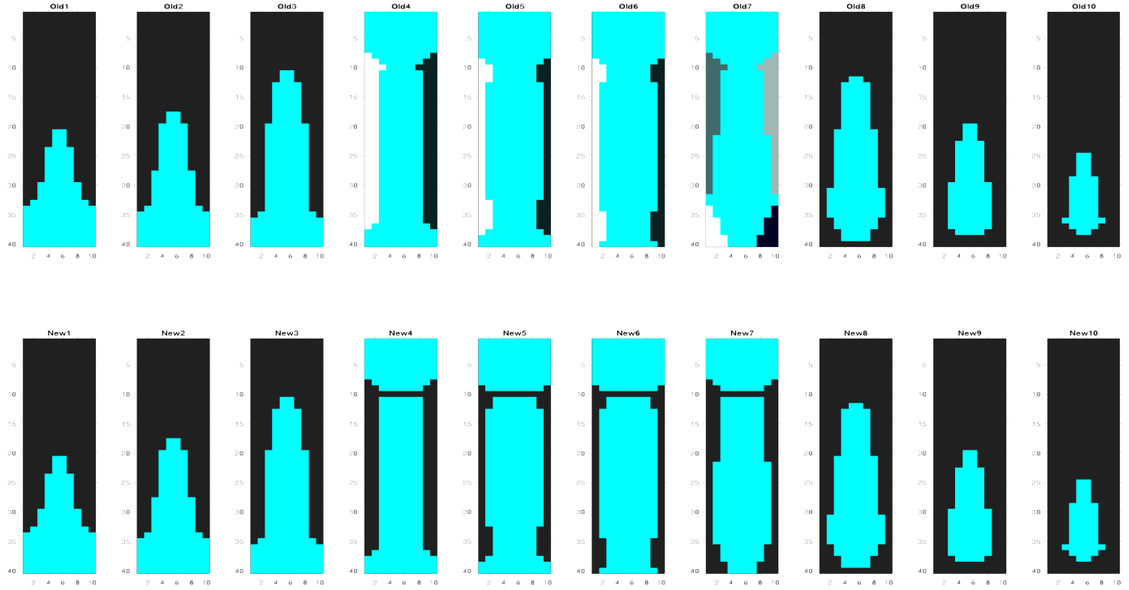


Flowchart ..: path continuity Dijkstra chart (Author)

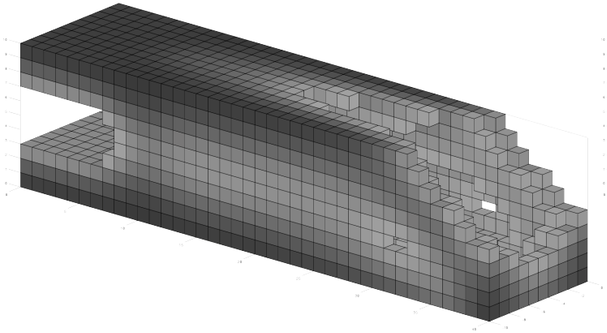
```
function endpoints = find_endpoints(map)
    [N, M] = size(map);
    endpoints = [];
    for i = 1:N
        for j = 1:M
            if map(i, j) == 1
                waters = 0;
                if i-1 < 1 || map(i-1, j) == 0
                    waters = waters + 1;
                end
                if i+1 > N || map(i+1, j) == 0
                    waters = waters + 1;
                end
                if j-1 < 1 || map(i, j-1) == 0
                    waters = waters + 1;
                end
                if j+1 > M || map(i, j+1) == 0
                    waters = waters + 1;
                end
                if waters >= 2 || 3
                    endpoints = [endpoints; [i, j]];
                end
            end
        end
    end
end
```

MATLAB code for an overview of the adjustment to the Dijkstra algorithm

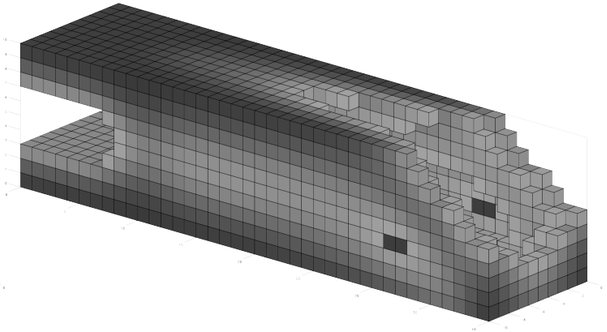
overview of connection Dijkstra by layers Topology Optimization (MBB beam)



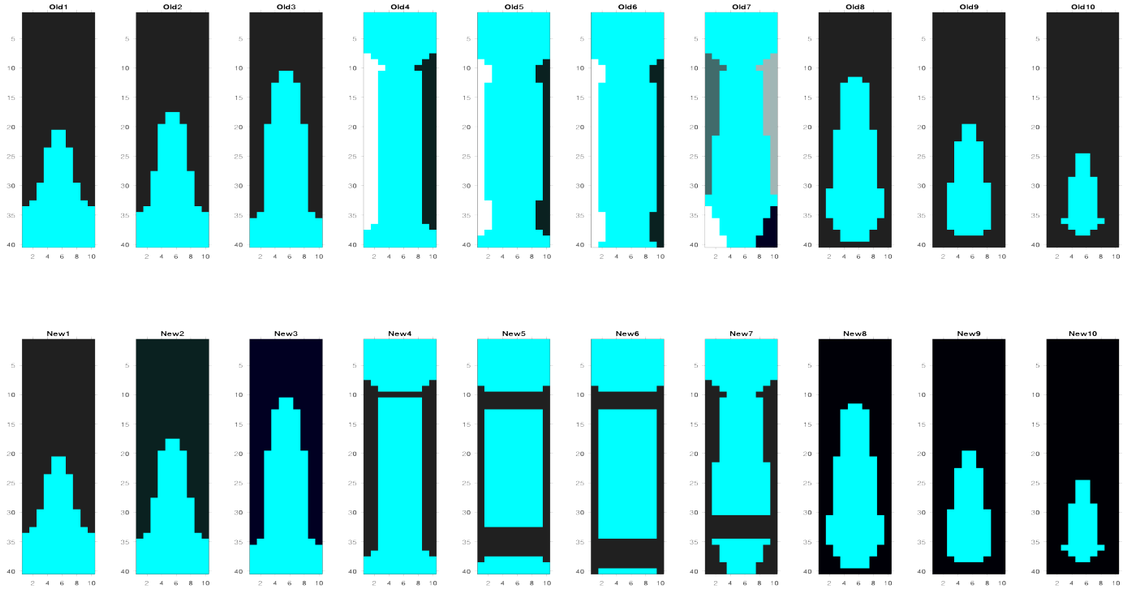
Not connected Topology Optimization (MBB beam) volume 0.5

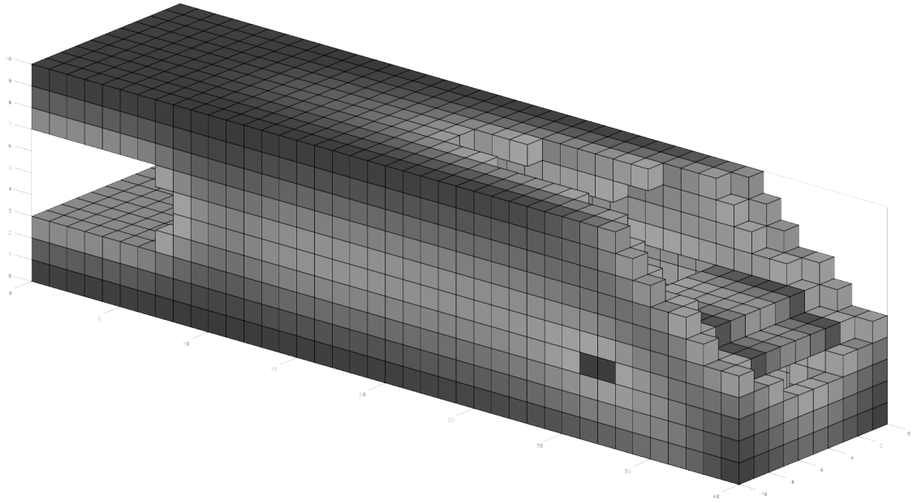


connected Topology Optimization Dijkstra (MBB beam) volume 0.503



6.6 Path continuity Dijkstra

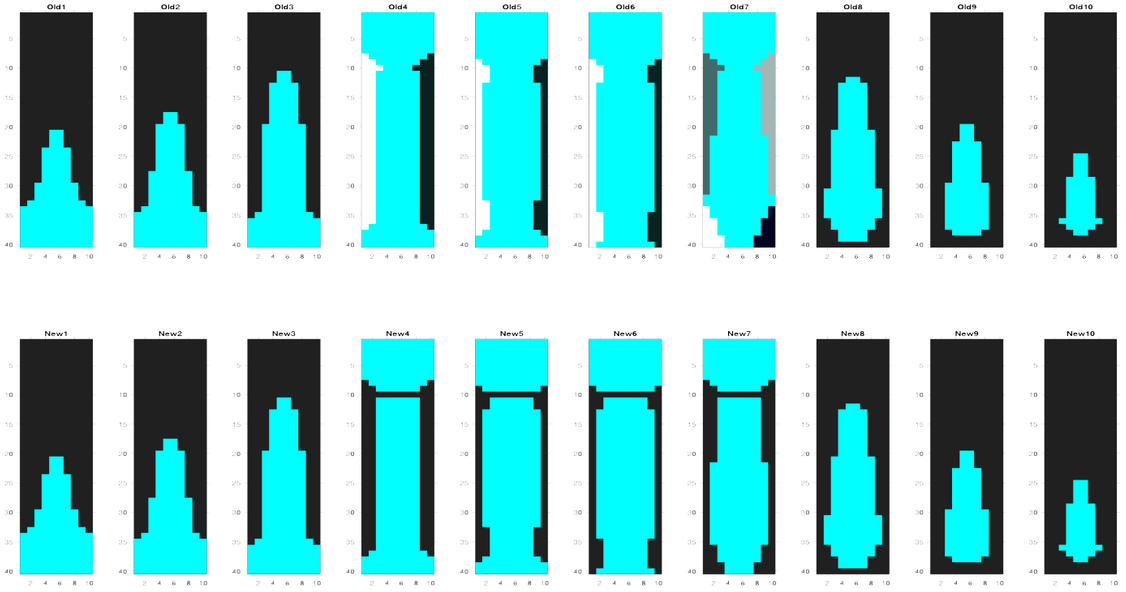


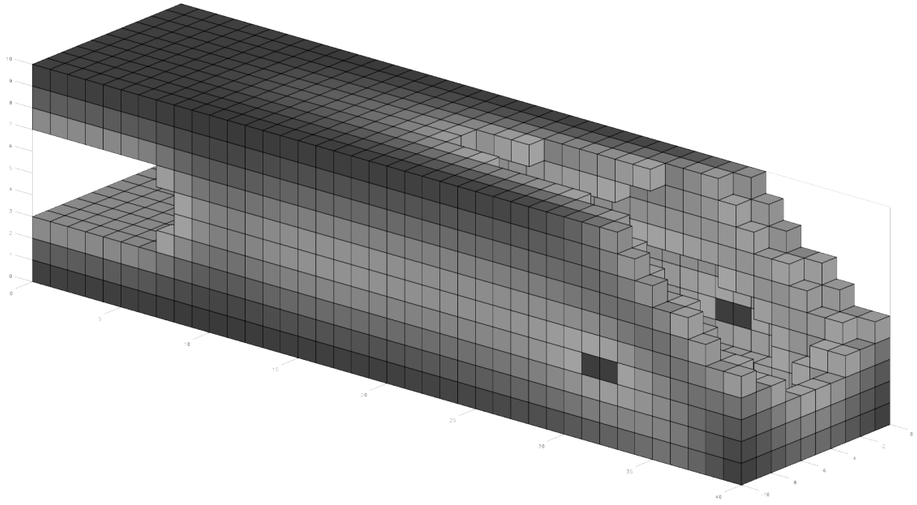


Comparison of MATLAB function used by Bi et al., & Yang et al., Topology Optimization (MBB beam)

volume needed for reconnection 0.017

6.6 Path continuity Dijkstra





Comparison of adjusted Dijkstra function in Topology Optimization (MBB beam)

volume needed for reconnection 0.003

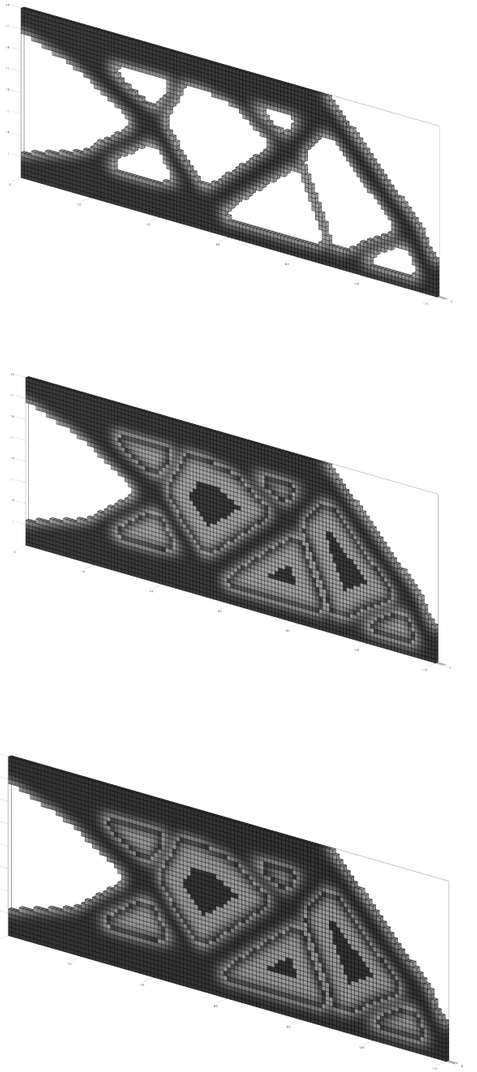
6.6 Path continuity Conclusion

The comparison between the path continuity setups revealed that the Dijkstra algorithm generally outperformed the MATLAB islanding function, although there were instances where the latter showed better performance. This discrepancy underscores the significance of volume-based versus symmetric properties.

On the following page, the superiority of the Dijkstra algorithm becomes evident as it typically requires only one string to connect different “islands,” whereas the islanding function connects items with equal distance values, resulting in a different pattern.

In some cases, the simplicity of the islanding function led to better outcomes, particularly in generating symmetric results, which can be advantageous in certain scenarios. A more detailed explanation is provided on the subsequent page.

In the figure accompanying this text, the final outcome of connecting different topology optimization results exhibits a striking similarity. Notably, the infill reacts to the less dominant areas by further reinforcing them, causing them to appear almost “black” in the figures. If these less dominant areas receive insufficient additional material, they are likely to be eliminated within one or two iterations, rendering the connectivity in the topology optimization ineffective.



Difference between matlab function (middle) and Dijkstra (top) on 2D+ optimization

6.7 path continuity discussion

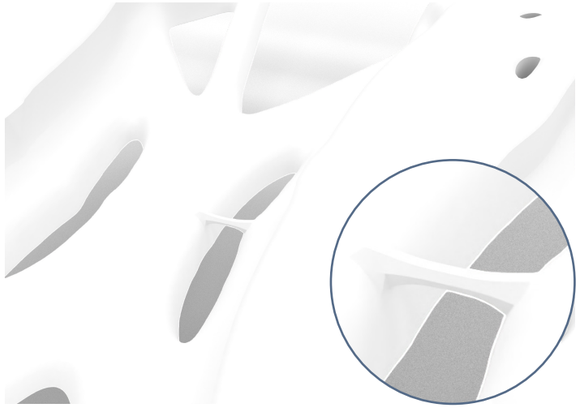
After experimenting with these two tools and evaluating their performances and outcomes, it can be concluded that both methods are effective in producing printable results without interruptions in each layer, thus eliminating potential flaws in the printing process such as stopping and retracting.

However, there are certain drawbacks, particularly when using no overhang angle or when employing a different orientation for the overhang angle, as this can result in “problematic” outcomes. In both cases, the printing path remains connected, but it may create bridged lines in the 3D structure. Consequently, when printing continuously, the printed part may fall on the layer at least -2 or more below, diminishing the aesthetic quality of the print unless the excess material is removed. The images on this page visually illustrate the impact of these connections, showcasing various scenarios, including larger variants.

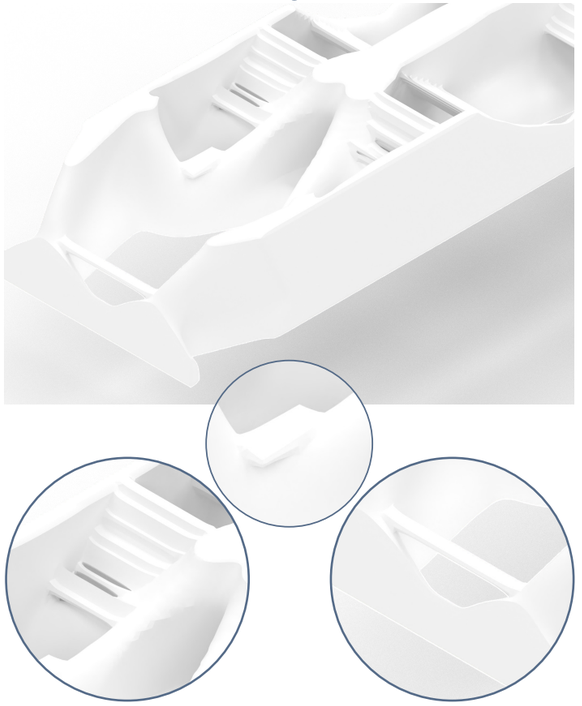
The continuity of the printing path is significantly enhanced by the overhang filter, primarily because the outcome in several cases is already improved when the importance of connected nodes within the overhang filter is considered. Therefore, the assistance provided by the overhang filter addresses cases where minimal reconnection is required. however the islanding tool doesn’t incorperate the overhang filter. So sometimes reconnecting is done using steeper or not achieveable overhang situation.

In the following section, several comparisons between the two approaches are presented, and their performance is evaluated.

Dijkstra algorihm



Matlab islanding reconnection



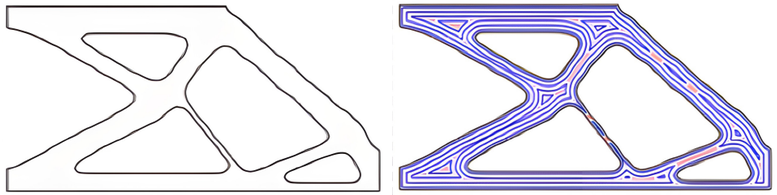
6.8 Nozzle offset

The path filling method was employed to ensure that the nozzle could adequately cover the entire area without resorting to zigzag patterns within the domain. This approach aimed to produce 3D printed structures with continuous lines while minimizing both overfill and underfill. Numerous studies, such as those by Yang et al., Bedel et al., and Bi et al., shed light on the importance of integrating path width considerations into 2D printing paths. Figure .. 'toolpath generation and boundary offsetting' visually shows the necessity of path filling.

While previous researchers discuss the potential modifications to topology optimization, they often fail to provide a comprehensive overview of the steps involved or insights into the coding aspect of topology optimization. Despite several attempts to contact these researchers, no replies or insights into these aspects were obtained. Even though i obtain a code, the correct .npz file was not configured properly, making it impossible to discern its construction process.
To kickstart the coding, a structured plan was formulated, detailed on the subsequent page. This blueprint delineates the coding tasks requisite for generating an offset boundary. In this thesis, offsetting is perceived as a post-boundary procedure, given the negligible disparity between the nozzle width and the eventual element width. Hence, optimization was delegated to a post-step adjustment via a Python script, facilitating integration of nozzle sizing into the final product.

Consequently, no instances of zigzagging, underfilling, or overfilling occurred. The next pages present a guided explanation, visually illustrating the offsetting of the shape. This demonstration employs a 2D optimization with a 60-degree overhang angle, providing insight into the process.

Toolpath generation and planning post-topology optimization



Toolpath generation integrated post-topology optimization

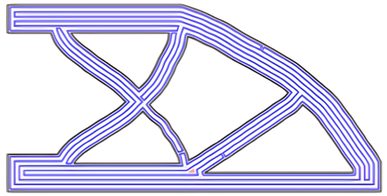
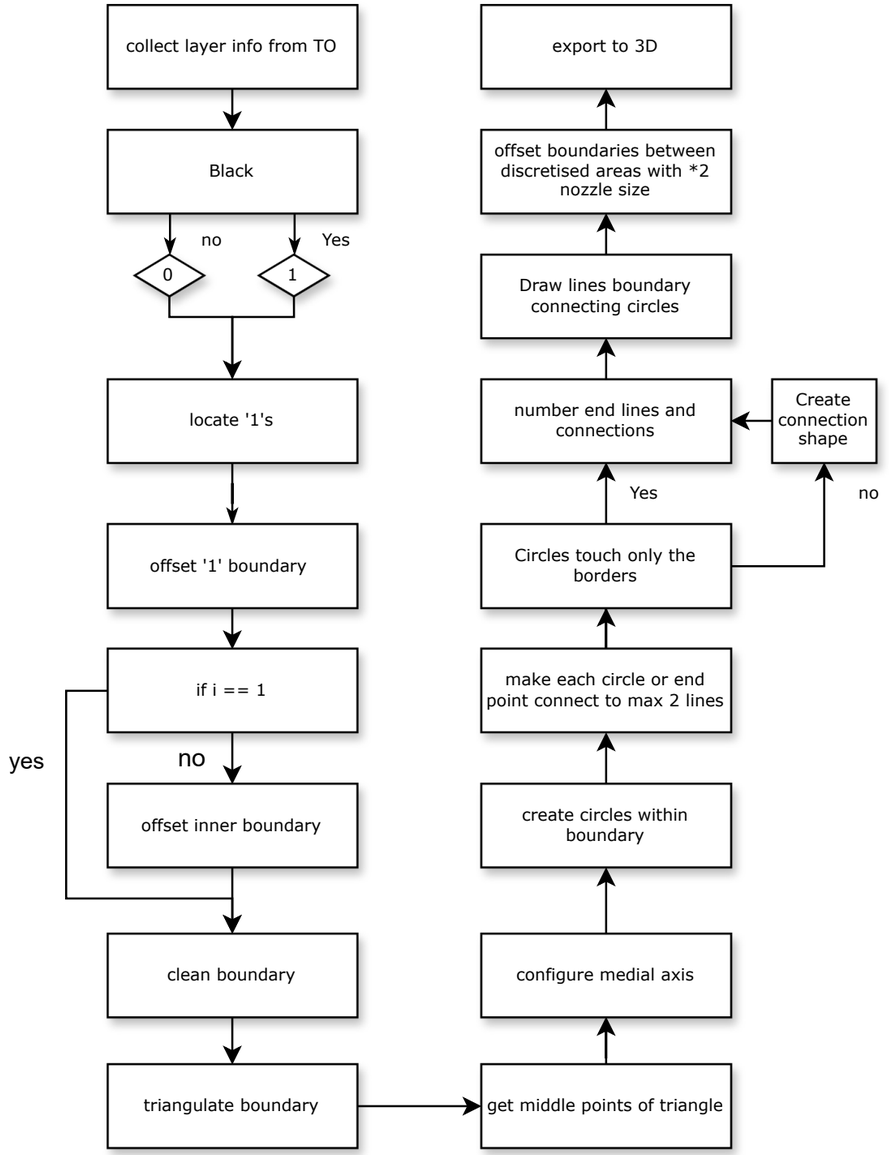


Figure 57: Rework of author of toolpath generation and boundary offsetting (Bi et al., 2022)



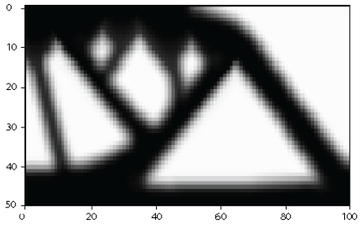
6.8 Nozzle offset

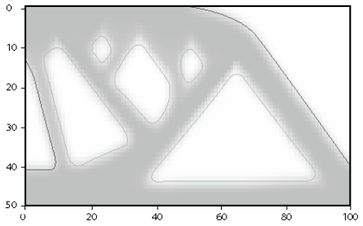
The initial step may seem straightforward on paper: establishing the boundaries of the topology optimization to define the overall shape. However, implementing these boundaries becomes more complex due to the nature of SIMP optimization, which operates within gradients ranging from 0 to 1. Thus, accommodating even slightly gray areas within the boundaries becomes necessary to prevent offsetting the total boundary.

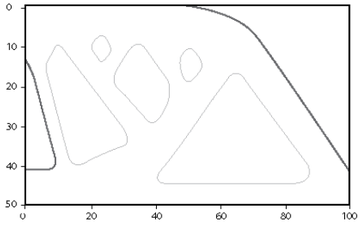
Initially, two primary boundaries are outlined to delineate the main shape within the design domain. Subsequently, additional inner boundaries are introduced, each requiring a specific area allocation that can be adjusted by the user. Following this, only the boundaries are considered, leaving the space within to be filled, completing the topology optimization area. This ensures a clean starting point for the subsequent stages of the algorithm.

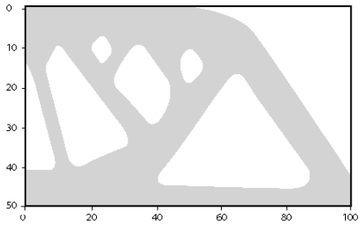
In the final thesis, boundary creation is integrated into the Matlab algorithm, with layers extracted accordingly. The boundary can be set from 0 - 1. But in this cases regardly chosen as > 0.5.
These layers are differentiated, typically represented in blue and black, enhancing clarity in boundary setup.

However, to mitigate the risk of sharp angles in the final print caused by the cubic nature of optimization, a polyline is employed, smoothing the boundary configuration. Once this step is completed, the optimization process can proceed to the next stage, presenting a refined boundary and the total optimized area.









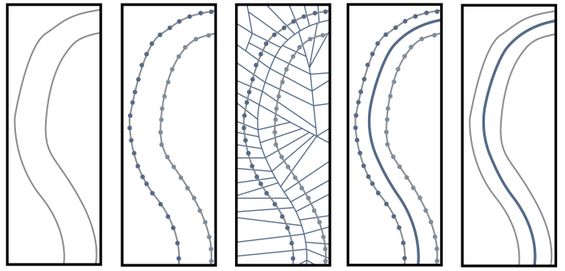
6.8 Nozzle offset

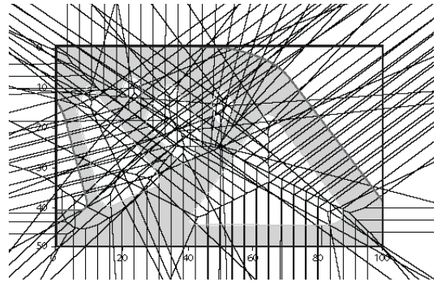
As the topology optimization boundary is fully defined, the next step involves identifying the medial axis. This is achieved by densely populating the boundaries with points and triangulating the optimization area. The following diagram illustrates the process of generating the medial axis, which is then superimposed over the existing optimization area.

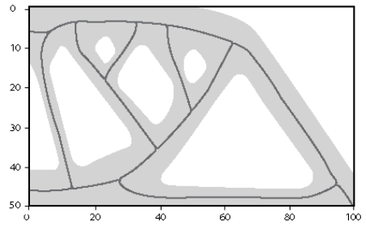
Subsequently, the points connecting two or more lines are enumerated and surrounded by circles whose sizes are determined by the intersecting boundary lines. These circles are also labeled, establishing the foundation for offsetting the optimization boundaries.

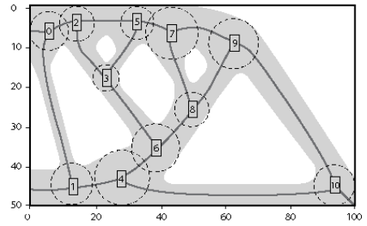
In the subsequent stage, intersecting circles are merged and smoothed out using polyline offsetting techniques applied to both circles. This operation results in distinct boundaries for each line, as overlapping portions where circles intersect are removed. This step is crucial as it prepares the boundaries for further configuration.

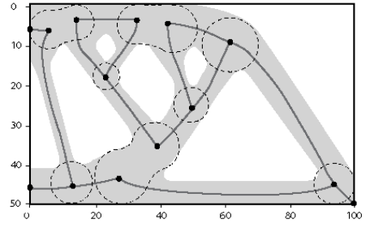
To facilitate the forthcoming calculations and provide clearer insights into the points' interactions, they are once again enumerated. This numbering scheme simplifies subsequent steps and enhances understanding of the points' behavior throughout the process.











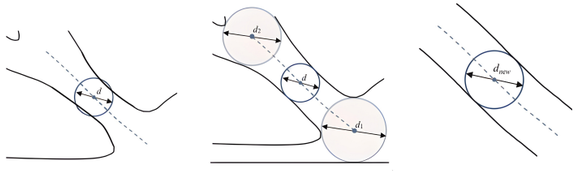
6.8 Nozzle offset

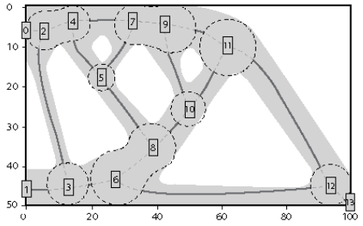
In the concluding steps of the process, the circles connecting points with boundaries redefine the boundaries, slicing through the previously determined medial axis from one boundary to another. This configuration, as depicted in the accompanying image, reveals 12 medial axes intersecting eight boundary boxes and seven optimization boundary lines.

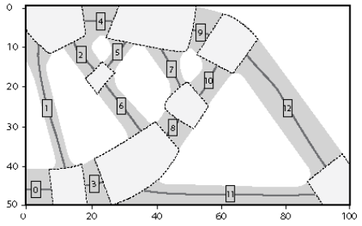
The offsetting procedure utilizes the illustration below: medial axis lines serve as guides. When the space between boundary lines cannot accommodate twice the size of the nozzle, these lines are shifted. By positioning multiple circles along these medial axes, the boundary lines expand.

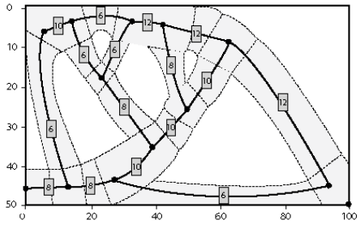
The resultant shape, now adjusted with offsets, undergoes the entire process anew. Upon determining various offsetting circles, the outcome is illustrated adjacent, showcasing blocks representing twice the nozzle size along each line. The utilization of medial axes consistently facilitates this transformation, ensuring effective optimization each time.

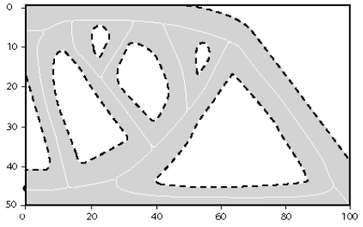
The final shape optimization seamlessly transitions into a 3D optimization through the incorporation of ISO cubic elements. This progression is facilitated by STL conversion, detailed on the subsequent page.



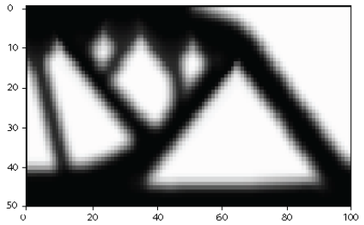


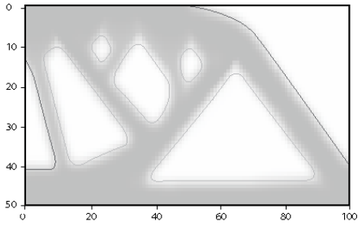


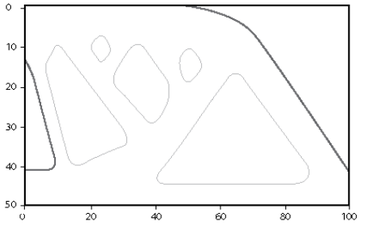


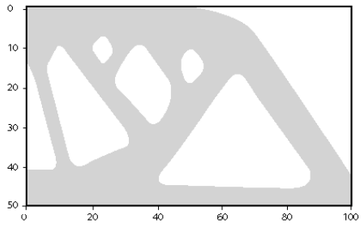


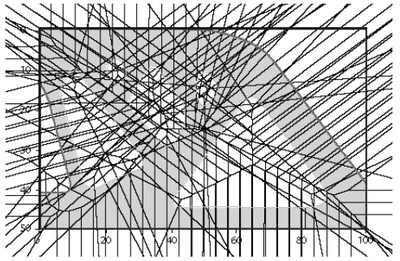
6.8 Nozzle offset

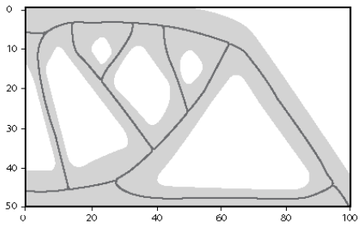


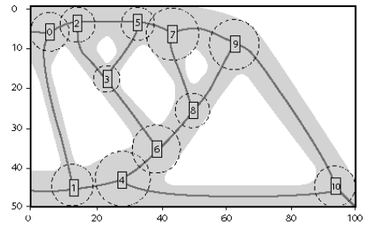


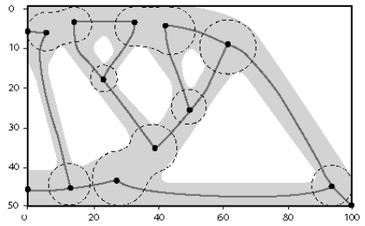


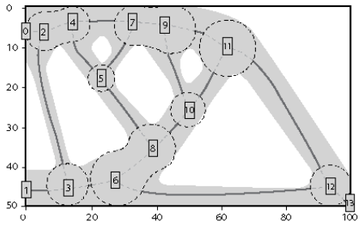


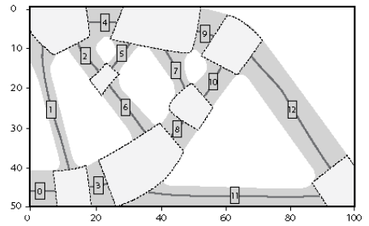


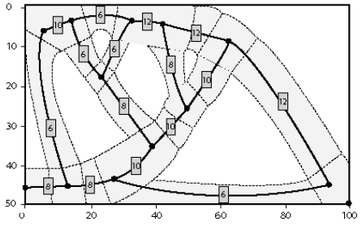


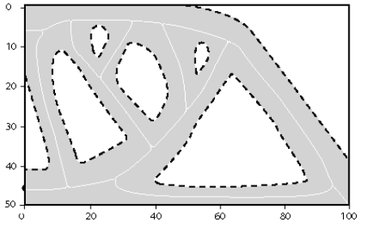








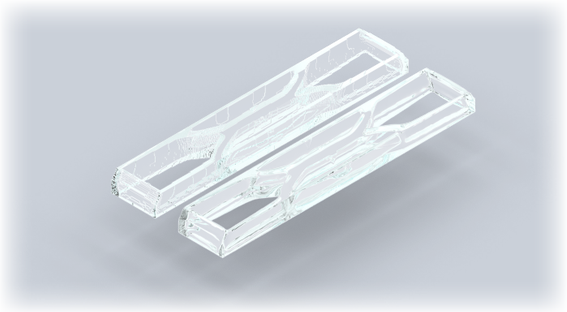


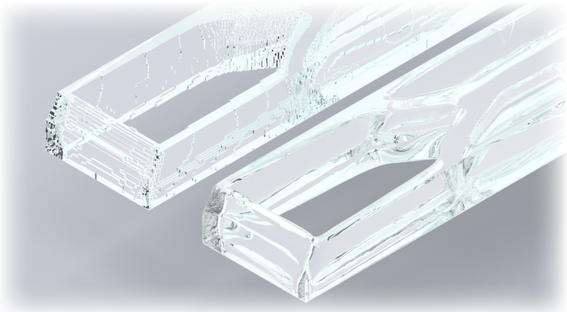


6.9 STL conversion

To convert the file into STL, a well know format for extracting objects into a G-code slicer, the process from MATLAB is swift and seamless. The triangular mesh shapes are relatively lightweight compared to the 3D figures generated by MATLAB.
This code was sourced from top3d.app and developed by Liu (Apr 2015). While the conversion in chapter 7 wasn't the fastest, it was consistently successful, preserving the design domain with specified boundary conditions. Changes were made to minimize the volume required for elements to be included in the final output.

The outputs were modified to first generate the standard output (cubic) and subsequently an ISO version of the elements. Within the ISO version, elements are divided into triangles and pre-connected in a smoothed triangular configuration. This approach eliminates the need for post-smoothing in Grasshopper, streamlining post-processing. The ISO file is then loaded into slicing software such as RoboDK (slic3r) or Cura, where the complete shape is sliced.





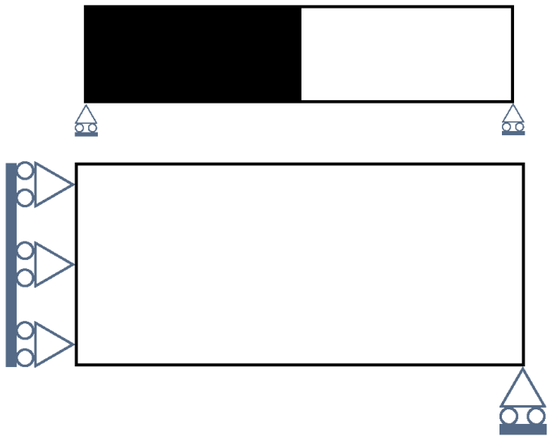
function Top3dSTL_v3(fout, varargin)

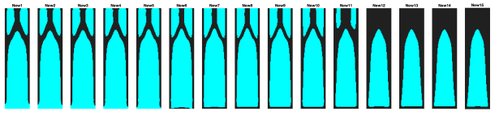
```
if strcmp(save, 'on')
tic;
if strcmp(mirror, 'on')
outputFileName = sprintf('1.STL_%dx%dx%d_%s_%.f_%d.it_%.2fvol%s.ISO.stl', elementsCount(2)*2, elementsCount(1), elementsCount(3), AMfilterLetter, A_deg, maxloop, volfrac, '.mirror');
saving_time = saving_time + toc;
else
outputFileName = sprintf('1.STL_%dx%dx%d_%s_%.f_%d.it_%.2fvol%s.ISO.stl', elementsCount(2), elementsCount(1), elementsCount(3), AMfilterLetter, A_deg, maxloop, volfrac, '');
saving_time = saving_time + toc;
end

STL(outputFileName, xPhys, 'Cutoff', 0.5, 'Plot', false, 'Mode', 'iso');

if strcmp(mirror, 'on')
outputFileName = sprintf('1.STL_%dx%dx%d_%s_%.f_%d.it_%.2fvol%s.stl', elementsCount(2)*2, elementsCount(1), elementsCount(3), AMfilterLetter, A_deg, maxloop, volfrac, '.mirror');
saving_time = saving_time + toc;
else
outputFileName = sprintf('1.STL_%dx%dx%d_%s_%.f_%d.it_%.2fvol%s.stl', elementsCount(2), elementsCount(1), elementsCount(3), AMfilterLetter, A_deg, maxloop, volfrac, '');
saving_time = saving_time + toc;
end
STL(outputFileName, xPhys, 'Cutoff', 0.5, 'Plot', false);
end
```


6.9 mirror of the element

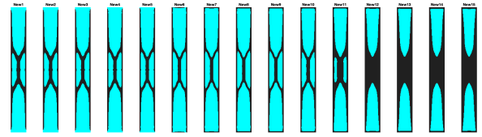
After completing the optimization process, the mirror functionality is introduced to ensure symmetry and integrity in the final shape. This mirror operation aligns with the predetermined direction specified in the algorithm. However, a decision is made to bisect the 'bridge' as detailed in Chapter 7.2, akin to the mirroring technique employed for MBB beams.
Executing the islanding script for the final time after mirroring safeguards against potential issues such as excessive volume addition or asymmetrical outcomes, which are undesirable. Hence, it's crucial to prioritize the mirroring step following the islanding script to maintain the desired symmetry.

To further enhance the process, the Dijkstra grid can be called either before or/and after the mirroring operation as needed, providing flexibility and control over the final shape's continuation.





mirror and dijsktra connection





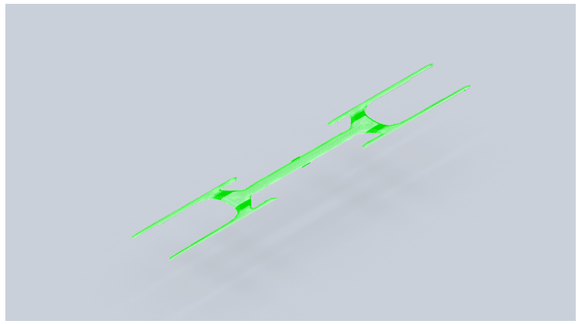
6.9 Grasshopper and visualisation steps

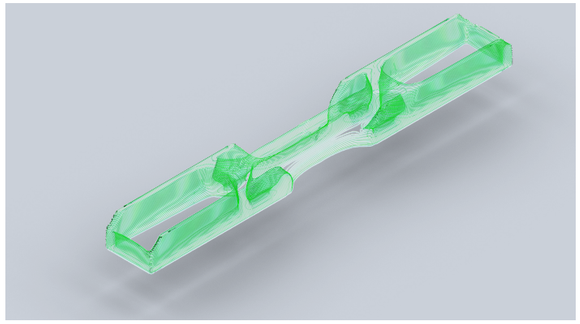
Now, with the final output achieved through STL conversion (ISO), the algorithm's smoothing process had already occurred, following the nozzle path offsetting. A sleek geometry emerged from the algorithm, typically ready for direct implementation into a slicer to generate the g-code, which the printer would then use to produce the element where mainly by it's working principle the familiar "lines" characteristic of 3D printing would appear.

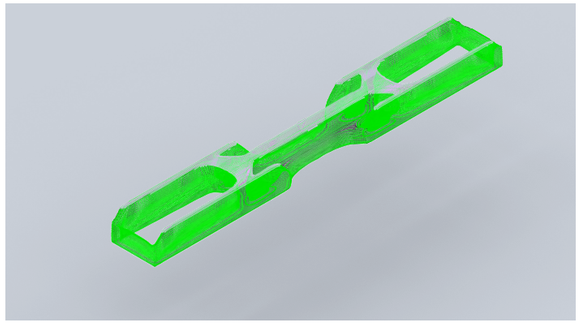
However, in this master's thesis, there was also a need for numerous visual representations, as printing the full-size element wasn't feasible. To ensure these visual representations mirrored the characteristics of the final printed version, a brief study was conducted within Grasshopper to alter the appearance of the final shape.

Several attempts were made, with most files eventually becoming too large for the computer to handle. The Grasshopper script at the bottom of this page presents a simple yet effective method. Outer contours are gathered, divided, and circles are created along the borders to mimic the appearance of a printed object.

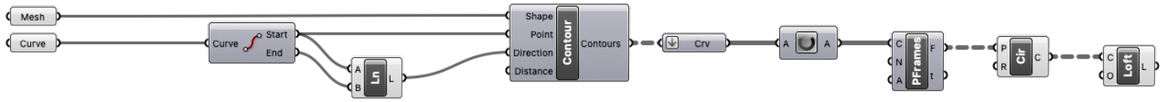
This visualization concept significantly enhanced the visual impact of the final result. A digitally crafted element with the appearance of a 3D print was successfully achieved.









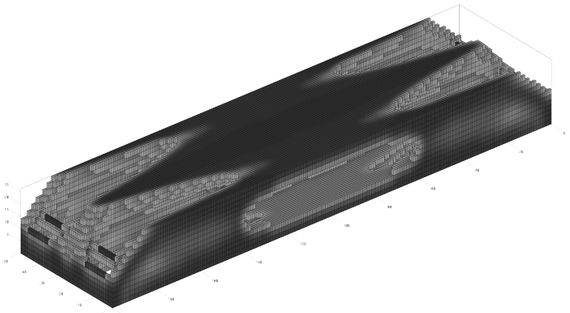


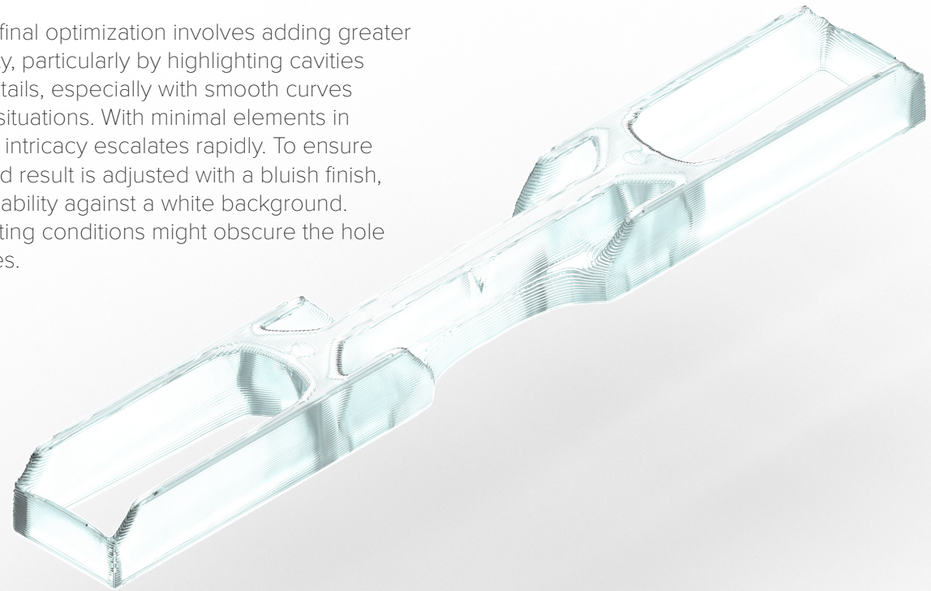
6.10 Conclusions & Other implemenations

Regrettably, due to the increased complexity of elements in the design domain aimed at achieving a more intricate and refined result, the utilization of the 3D figure in Matlab is no longer viable. Opening images larger than 15MB causes Matlab to freeze, subsequently leading to system-wide freezing.

Consequently, it has been decided that when the design space is densely populated with elements, then the example provided alongside this text, the image will be displayed in STL format, resembling glass. While this approach eliminates the ability to visualize the graytones of the design domain, it significantly improves workability and provides a more comprehensive view of the final outcome.

Enhancing the final optimization involves adding greater detail and clarity, particularly by highlighting cavities and refining details, especially with smooth curves and overhang situations. With minimal elements in the design, the intricacy escalates rapidly. To ensure visibility, the end result is adjusted with a bluish finish, facilitating readability against a white background. Otherwise, lighting conditions might obscure the hole element at times.







07

Algorithm testing

Conducting performance testing on the algorithm developed from the previously selected case studies is crucial. This stage involves rigorous evaluation to ensure the algorithm's efficiency and effectiveness. Through systematic testing procedures, the aim to assess its performance across various parameters and scenarios.

7.1 Introduction

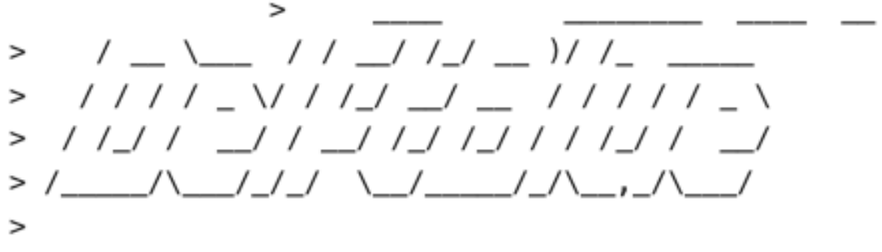
The utilization of the supercomputer significantly accelerated the completion of the chapter by drastically reducing computational time. It enabled the optimization of much larger design domains than what would have been feasible using my personal MacBook or the desktop available from the VR-Lab.

The ability to set up multiple optimization loops simustanisly within Delftblue proved to be a valuable means of generating comparative outputs.

Getting started with the Delftblue server posed some challenges. However, with Prateek's assistance, I managed to set it up through the terminal. In the next meeting, integration with Visual Studio Code was accomplished. This latter improved notably the usability and efficiency of Delftblue's computational power. Initially, the setup for saving the individual files generated by the MATLAB script was performed on a local machine. This decision was made because Delftblue does not support visual outputs. Output specific details were put in the name as that made it possible for several simultanious outputs and configuring and comparing them afterwards.

As shown in the image, the optimization process can consume a significant amount of computational memory. (Standard computers typically have around 16GB, whereas the script can utilize to more than 500GB of memory.) Consequently, my Educational license was upgraded to a Research license, with Charalampos's approval, enabling me to experiment with faster and larger domains. Also it enabled longer/more optimization processes at the same time. The use of the Delftblue server deemed significant for the output of this master thesis.

Using the Delftblue server, a maximum of 3.840.000 elements have been evaluated within the design domain. This represents a substantial leap compared to the 6.300 elements analyzed by Anna-Maria Koniari & 36.000 elements analyzed by Eva Schoenmaker. Although the calculations experienced a slight increase, utilizing Delftblue, an average design domain of 420 x 50 x 100 takes approximately 6 hours, utilizing 24 cores and 50GB of RAM for 200 iterations.



```
Job ID: 3679423
Cluster: delftblue
User/Group: pbrueren/domain users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 24
CPU Utilized: 6-07:51:16
CPU Efficiency: 53.69% of 11-18:49:36 core-walltime
Job Wall-clock time: 11:47:04
Memory Utilized: 94.36 GB
Memory Efficiency: 52.42% of 180.00 GB
```

```
Job ID: 3687858
Cluster: delftblue
User/Group: pbrueren/domain users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 36
CPU Utilized: 27-18:40:53
CPU Efficiency: 77.15% of 36-00:05:24 core-walltime
Job Wall-clock time: 1-00:00:09
Memory Utilized: 450.85 GB
Memory Efficiency: 60.11% of 750.00 GB
```

```
Job ID: 3686057
Cluster: delftblue
User/Group: pbrueren/domain users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 36
CPU Utilized: 29-00:02:46
CPU Efficiency: 57.37% of 50-13:19:12 core-walltime
Job Wall-clock time: 1-09:42:12
Memory Utilized: 162.19 GB
Memory Efficiency: 87.67% of 185.00 GB
```

```
Job ID: 3687856
Cluster: delftblue
User/Group: pbrueren/domain users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 36
CPU Utilized: 28-02:32:12
CPU Efficiency: 78.06% of 36-00:05:24 core-walltime
Job Wall-clock time: 1-00:00:09
Memory Utilized: 474.52 GB
Memory Efficiency: 63.27% of 750.00 GB
```

```
[pbrueren@login03 ~]$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           3675365 compute-p matlab_t pbrueren  R   11:37:30      1 cmp125
           3675369 compute-p matlab_t pbrueren  R   11:38:03      1 cmp135
           3677240 compute-p matlab_t pbrueren  R    2:17:26      1 cmp035
           3677241 compute-p matlab_t pbrueren  R    2:15:17      1 cmp115
           3677239 compute-p matlab_t pbrueren  R    2:27:04      1 cmp159
           3677238 compute-p matlab_t pbrueren  R    2:30:19      1 cmp086
           3677236 compute-p matlab_t pbrueren  R    2:36:49      1 cmp202
           3677233 compute-p matlab_t pbrueren  R    2:43:14      1 cmp043
           3677235 compute-p matlab_t pbrueren  R    2:42:12      1 cmp053
           3677232 compute-p matlab_t pbrueren  R    2:47:32      1 cmp179
           3677228 compute-p matlab_t pbrueren  R    2:52:49      1 cmp168
           3677229 compute-p matlab_t pbrueren  R    2:50:39      1 cmp036
           3677246    memory matlab_t pbrueren PD       0:00      1 (Resources)
           3671919    memory matlab_t pbrueren  R 1-04:32:10      1 mem009
           3677244    memory matlab_t pbrueren  R      15:48      1 mem007
           3677243    memory matlab_t pbrueren  R      47:40      1 mem006
           3677242    memory matlab_t pbrueren  R    2:50:19      1 mem007
```

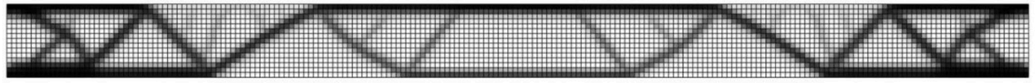
7.2 Algorithm performance 2D

To compare this new developed algorithm with the predeccors, the code was altered to the design load and supports which these 2 algorihms use. This design domain is different then used in this thesis. However as can be seen on the next page, the code was set up by giving a depth of 1 node. so nelz = 1. Thereby creating a similar 2D structure as Ann-Maria and Eva. Where the bridge is ‘clamped‘ in from the bottom and the top creating the start on the left side where the two sides are fully present due to the excess of the clamping.

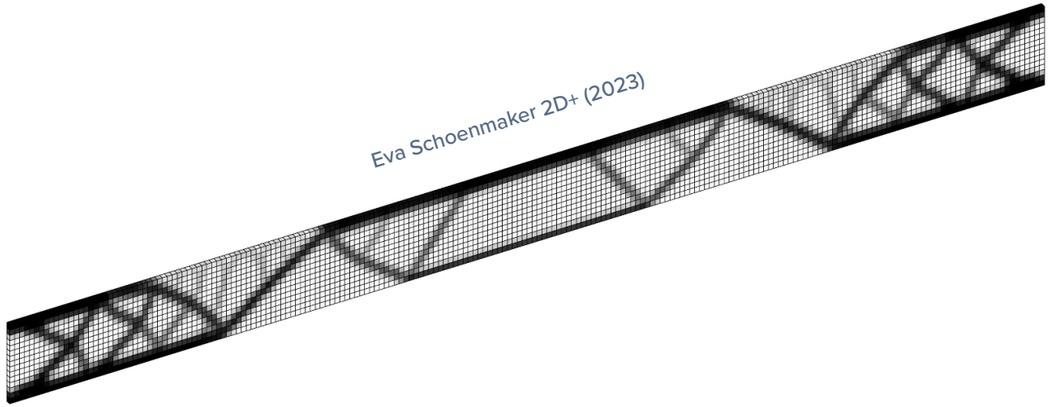
The algorithm was tested with DelftBlue and given more computing power than it could consume to give it all the possible bandwidth. The same number of elements were used as in the girls’ domain. creating the elements with a size of 20x20x20. Because Eva’s code most closely resembled the new algorithm, it was chosen as the ‘base’

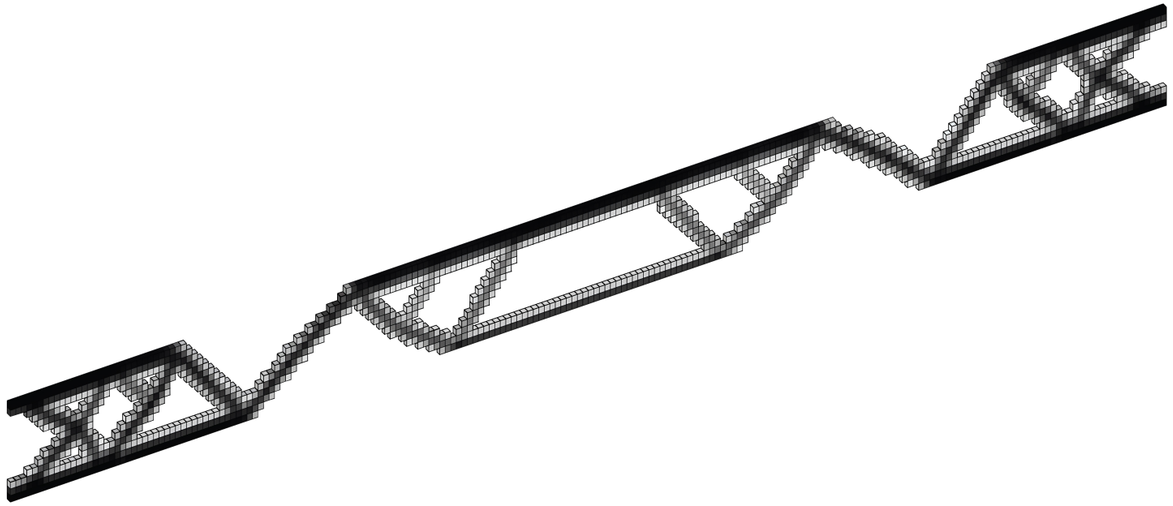
There is a significant time reduction comparing with the other codes. (Only the time required for the optimization loop was taken into account). this code is rougly 1118% faster than the original code by Anna-maria. This improvement would become useful as the design domain would consist of significantly more elements in the optimization. One of the most important key factors for the time reduction is the prelocation of the sparse matrix. Due to this fact, several elements are quickly no longer considered due to the low volume of the SIMP method. After this test, the real boundary conditions can be drawn up and tested. Because this is slightly different from this comparison domain.

Anna-Maria Koniari 2D (2022)



Eva Schoenmaker 2D+ (2023)





	Volume %	Iterations	Percentage	Function evaluation	Time (hours:min:sec)	percentage
2D (Koniari)	29	231	64%	365727	46:32:00	0,17%
3D (Schoenmaker)	28.9	149	100%	274	00:04:51	100%
3D (New)	30	152	98%	268	00:02:28	196%

7.3 Design - Case study (boundary conditions)

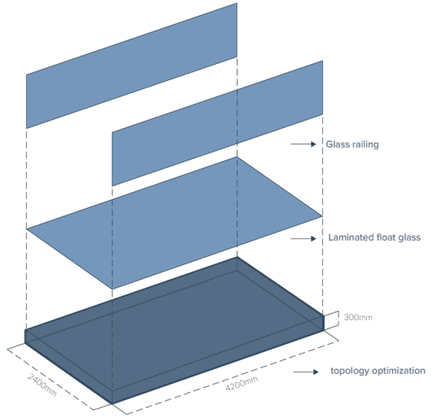
The selected case study must undergo “algorithmization,” which involves establishing the load and support boundaries.
Differing from its predecessors, the load and support conditions have been slightly adjusted. The aim is to construct a single supported bridge where the support area can be ‘concealed’ within the wall. This approach involves establishing a support area at the bottom end of the design domain. In this configuration, the support area only features fixed nodes on the bottom side, as clamping would not be advantageous for the compressive forces acting on the bridge, rendering it unnecessary.

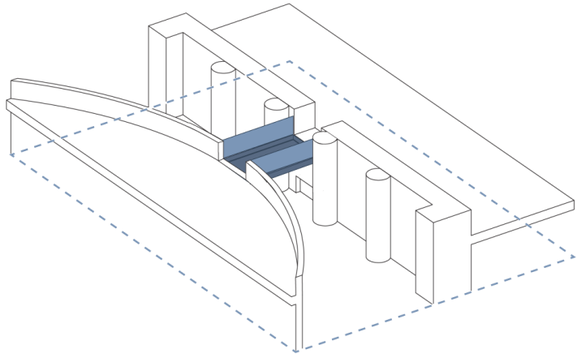
Regarding the load conditions, the opposite approach was taken. The edges were not loaded to create space for the bridge to ‘flow‘ inside of the cavity made for the supporting area of the bridge.
The entire top layer was designated to accommodate three distinct load types.
Firstly, the weight of the two float glass planes is factored in. The glass planes assure the bridge’s impact resistance. Secondly, the load imposed by people traversing the bridge. And thirdly, an additional maintenance loading, is considered. Furthermore, the self-weight of the bridge itself is duly taken into consideration.

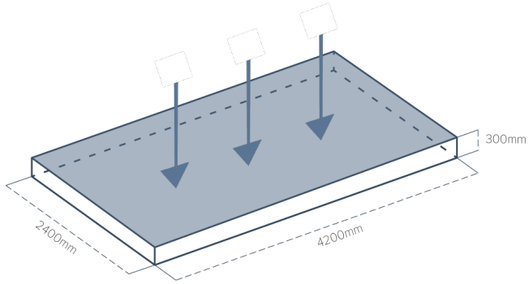
The loading and support conditions were algorithmically formulated. To simplify the algorithm, the design domain was scaled down to 6 different design domain variant. which differ from the hole width of the bridge till 1/6 of the bridge. This was done to illustrate the impact and differences resulting from the subdivision of the design domain. By doing so, the optimal outcome for this project and algorithm could be determined.

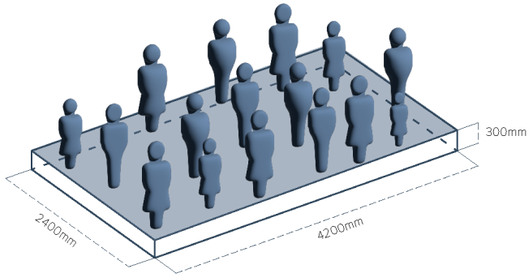
After executing the algorithm, the bridge came into existence.

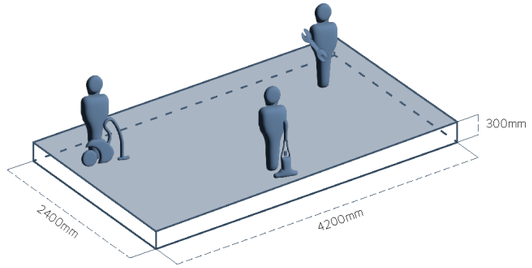
	Symbol	Units	Value
Permanent load **Self weight (slab)** Uniform applied all nodes on top	f_{sw}	kN/m²	12.8
Permanent load **float glass** Uniform applied all nodes on top	f_{fl}	kN/m	1.2
Short term load **people** Uniform applied all nodes on top	f_{sLp}	kN/m²	5
Short term load **Maintenance** Uniform applied all nodes on top	f_{sLo}	kN/m²	0.4
Safety factor **permanent loads**	S_p	-	1.2
Safety factor **short-term loads**	S_s	-	1.5
Total load	f_{tot}	kN/m²	$1.2*(f_{sw}+f_fl) + 1.5(f_{sLp}+f_{sLo})$

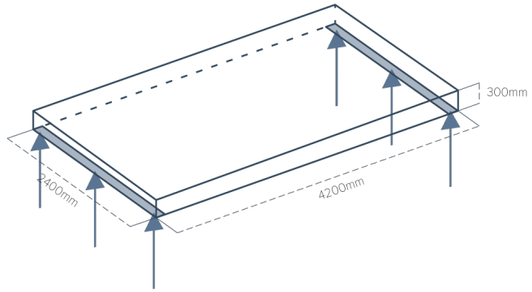


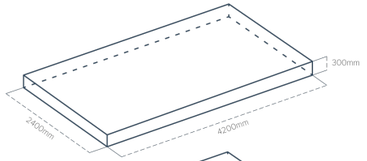


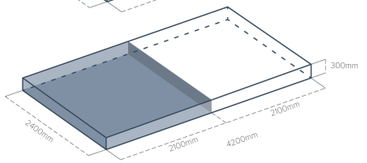


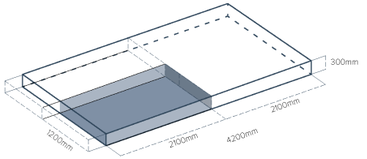


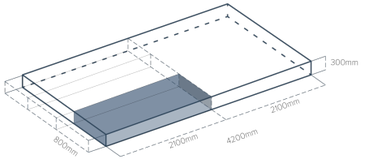


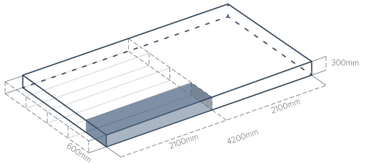


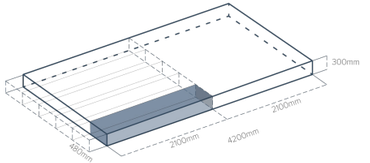


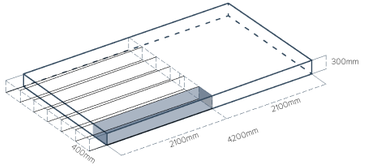




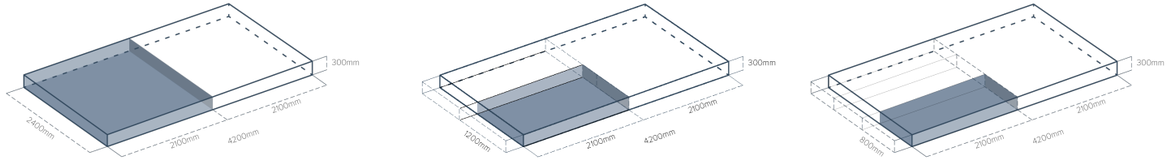




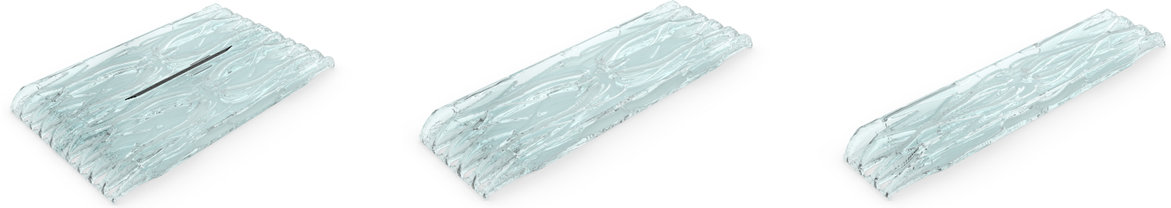




7.4 Design domain isotropic vs brittle (glass)



No material specific (Isotropic)



Glass material specific algorihm (better in compression than tension)

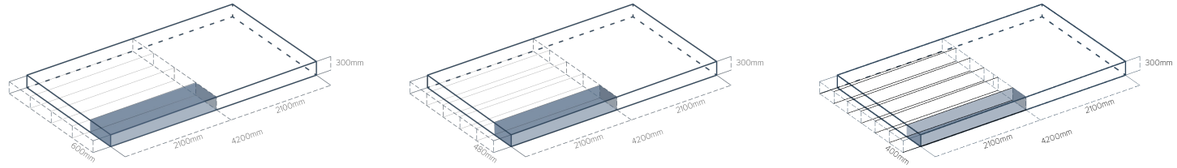


The complete bridge design requires substantial RAM capacities, sometimes exceeding 450GB. The impact of incorporating the specific characteristics of glass is immense. Primarily, the middle-bottom section experiences significant growth to accommodate the additional material needed to withstand the tensile forces at the bridge's base.

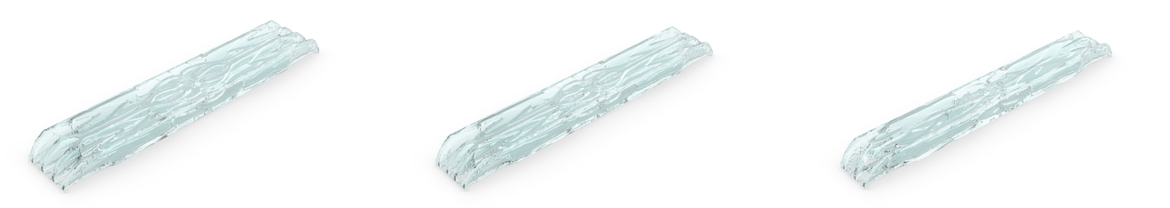
The half of the design domain truly highlights the potential for creating two symmetrical parts.
When mirrored, the resemblance to the entire bridge is shown, concluding that this objective is relatively similar and possible more favorable considering the calculation time required for the algorithm.

The algorithm gradually begins to form more "beam-like" structures, which includes the importance of the tensile forces placed upon the middle bottom part of the bridge.

Similarities persist with the larger version of the domain. When using a volume restriction of 50%.



No material specific (Isotropic)



Glass material specific algorihm (better in compression than tension)

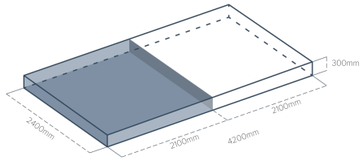


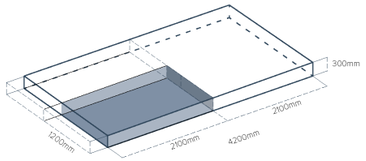
The first width were beam-like structures span from one side to the other, bridging the supported area and effectively counteracting the tensile forces in the middle part of the bridge. The majority of the material is concentrated there, with the beam structures specifically positioned to handle these tensile forces.

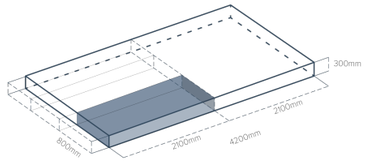
The appearance closely resembles that of the wider shape, where the algorithm must accommodate the lower tensile forces in the middle-bottom part of the bridge. It's evident that the best outcomes with the utilization of 0.5 volume for narrower domains exhibit similar shaping.

Also this appearance is similar, however with this width. In cases of smaller volume, only the boundary beams are formed, and additional material fills the middle-bottom part to meet the glass's tensile force requirements. This distribution differs from the isotropic version, where material is evenly spread.

7.5.1 Layer by Layer







No overhang filter







With overhang filter (bottom - top)



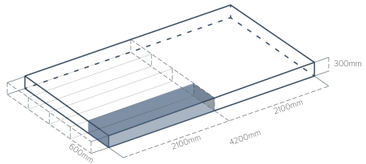


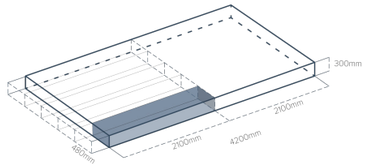


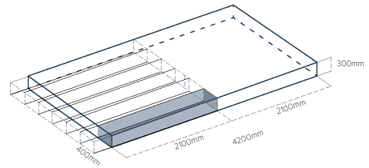
The boundaries noticeably shift outward from the domain's center due to the overhang filter. It's evident that several elements are essential for the object's feasibility, resulting in a more distinct shape. The middle part is predominantly filled at the bottom, prioritizing tensile stresses in that area.

Because of the specific characteristics of glass, more material accumulates in the lower part of the domain. This makes it easier for the overhang angle to shape a boundary similar to that of the non-overhang filter. However, the middle part, where floating elements are present, becomes clustered with the other elements.

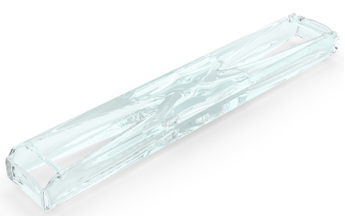
The first smaller domain where the flanges become visibly prominent upon achieving a 50% reduction in volume. The side boundaries are thereby filled with elements which form the base shape togehther with the middle part. With that the base shape becomes more pronounced, resulting in clear and well-defined boundaries.

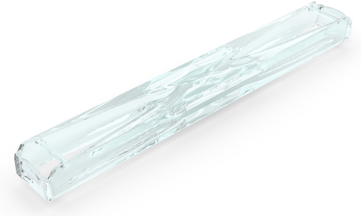


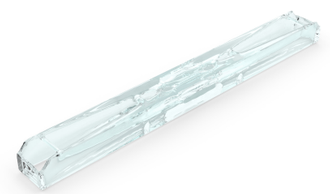




No overhang filter







With overhang filter (bottom - top)







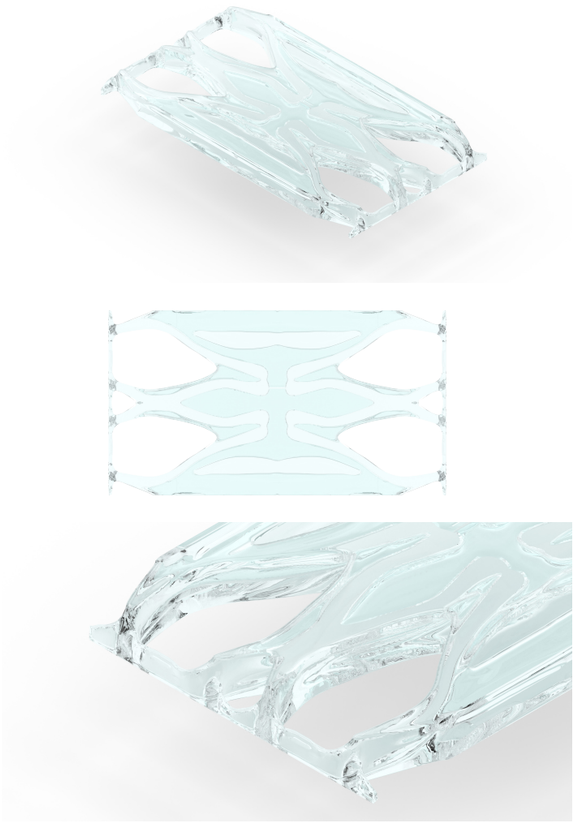
Consistent results are evident across the smaller domains. Clear boundaries are defined by two flanges at the domain's edge, with a filled bottom-middle section and compressive elements in the middle distributed to accommodate the remaining available volume within the design domain.

The appearance mirrors that of the wider shape. The overhang angle effectively aligns with these shape outcomes, facilitated by the clustering of elements at the boundaries of the domain. This clustering benefits from the presence of material beneath, providing a boundary setting for the elements to be placed upon.

The overhang angle preclaims all the elements to the side of the design domain. As there is material to 'grow' the final shape from. Thereby using elements first located in the middle top part to the flanges of the domain. This version would be most likely to be very prone to buckeling.

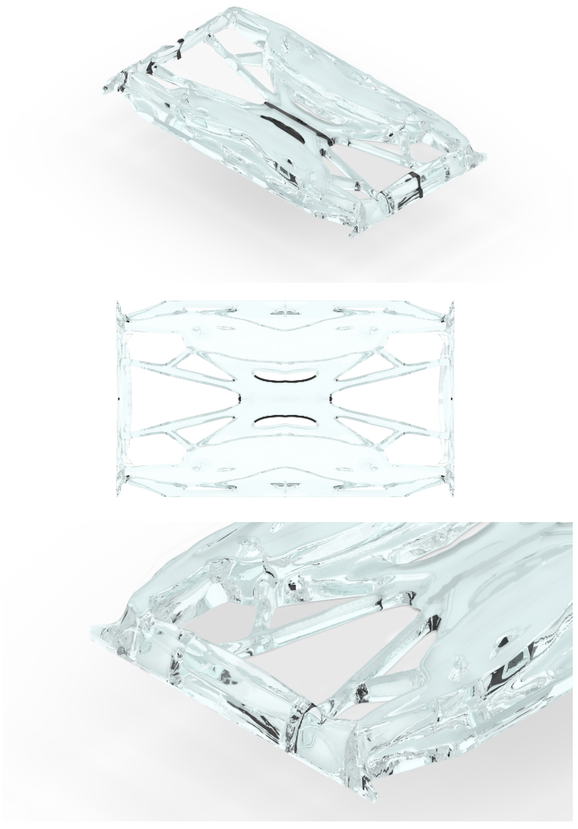
7.5.2 Layer by Layer orientations

No overhang filter volume 40%



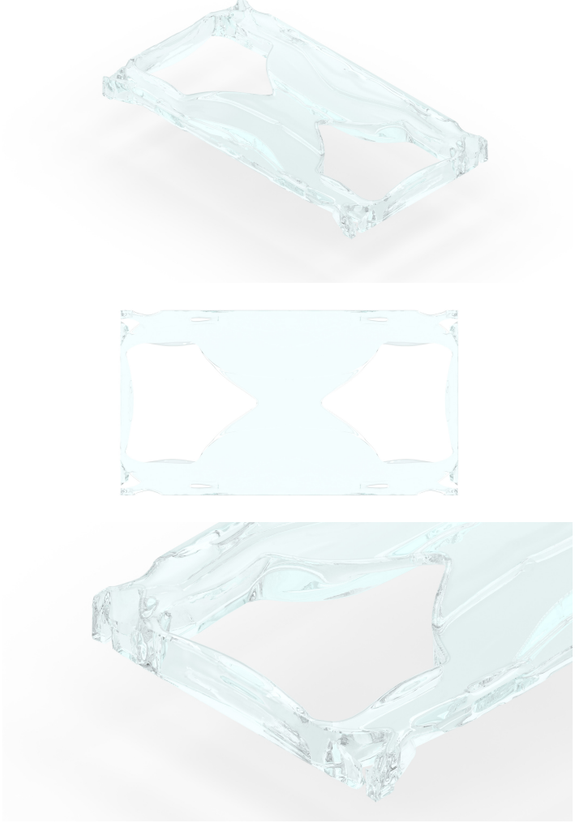
This overview displays the various overhang angles frequently tested for the final structure. Numerous conclusions can be drawn from this domain, which spans the total length and width of case study 1. Given glass's susceptibility to tensile forces over compression forces, the algorithm prioritizes addressing these tensile forces as more important to place more elements.

Bottom to top overhang filter volume 40%



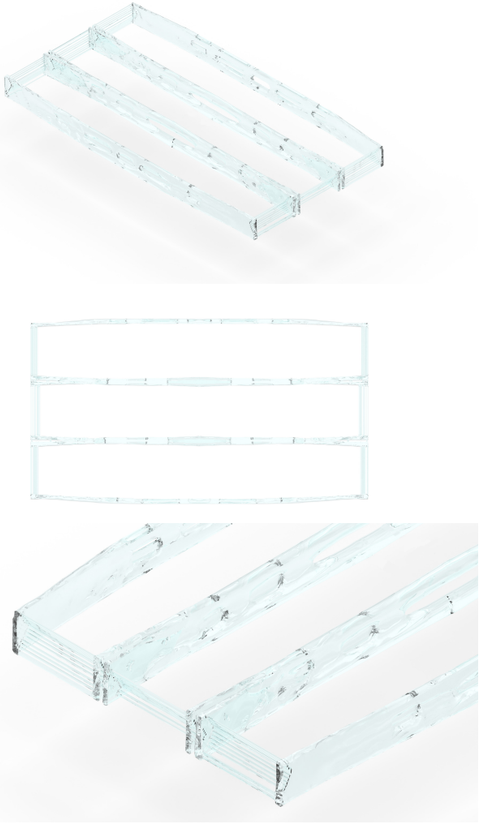
The bottom-to-top overhang filter performs optimally when compared to the other orientations. Given the tensile weakness of glass, it requires more material in the bottom middle part of the structure. Thus, it's preferable to initially fill the base of the element with elements where later additional elements can be placed in the top direction. This strategy is due to the brittleness of glass prioritized.

Top to bottom overhang filter volume 40%



The filter requires numerous elements to support the necessary structural components for achieving the objective. This results in the formation of a top sheet where compression is primarily placed, addressing a different objective compared to when printed from the bottom plate. The growth from the top plate towards the supported areas is crucial, and the shaping of these arms occurs early in the optimization loop.

Left to right overhang filter volume 40%



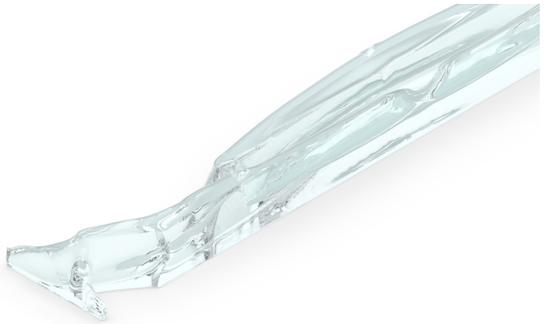
Placing elements in spaces where other elements already occupy sufficient volume poses a significant challenge for the algorithm in forming a coherent boundary. It's evident that "beams" emerge in areas where elements must support those below, underline their critical importance. The stringing on the right and left side is done by the islanding function (Dijkstra), which will be evaluated in the next sub-chapter.

7.5.3 Layer by Layer orientations

Several tests were conducted using the overhang filter, adjusting it for both the front and back perspectives in a 3D view (Instead of left <-> right, or bottom <-> top).

This adjustment eliminated the possibility of utilizing overhang angles greater than 60 degrees, restricting it to 45 degrees for the purposes of this thesis. As the change was made to the 4 top nodes of the hexahedral.
However, given the design loads, it was essential to ensure an adequate number of nodes for the overhang angle, aiming for an extremely low final objective, as evidenced by the visualization on this page.

The overhang filter functions to position “placeable” material on top of elements can be cahnged in the algorithm. The algorithm could be adjusted to accommodate the overhang angle with less density in the elements below, potentially by lowering Xlo to <0.5. As for this master thesis consisently 0.5 was used as the outcomes were in most cases more feaseable and the final optimization would establish within fewer itterations. However, within the scope of this thesis and the constraints of the design domain, these outcomes have not been tested extensively, primarily due to the impracticality of quickly achieving fully printed items.



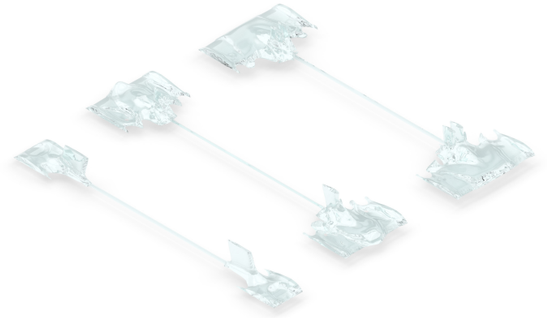




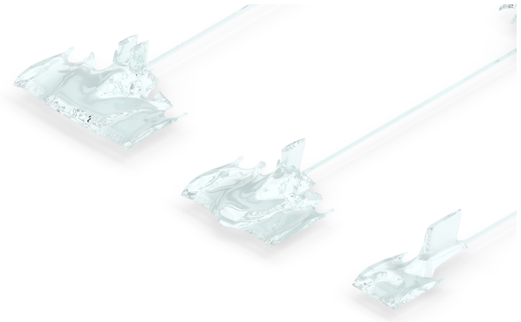
7.5.4 Layer by Layer -> little volume

Upon evaluating the overall algorithm with a minimum volume requirement of less than 20%, the results demonstrate a high priority to overhang angles, as visuallized below. The conversion process will consequently throttle and remain within this shape, as there are insufficient elements available to initiate optimization ‘growth’. The objective is consequently exceptionally high, labeling it as unfeasible.
Consequently, domains with a lower volume bound of less than 20% were not subjected to testing for this master thesis.

When implementing a volume restriction of 20% in combination with the overhang filter, domains consistently show a beam-like structure. Even when widening the domain, elements tended to cluster in the middle portion or, when utilizing wider domains than lengths, two beams would form inside the domain. This characteristic was only observed when employing this specific volume constraint. Otherwise, extended beams would be placed on both sides of the domain, significantly reducing the objective.









7.6.1 Islanding problem speed and comparison

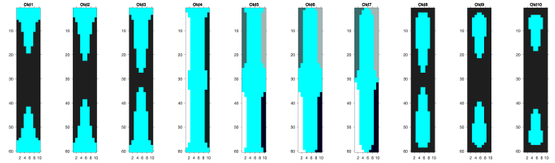
Matlab islanding function

The initial test setup features a design domain of 30 x 10 x 10 with a volume of 0.4. Mirrored after the last itteration. Normally, the function would limit itself to roughly 70 iterations with a time limit of 0.01, but for this example, it's set lower to demonstrate how the objective "dips" throughout the iterations.

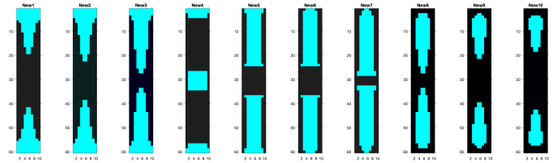
The overall script took roughly 35 seconds to execute. The Matlab islanding function for reconnecting, which occurred 8 times, took 15 seconds, constituting approximately 42% of the normal computation time. The substantial increase in volume for reconnecting the final shape is clearly evident.

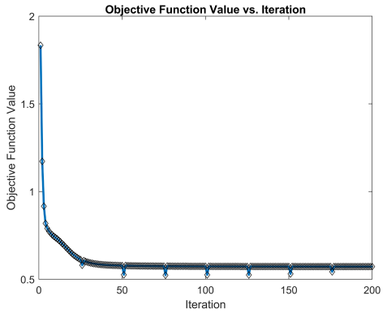
This simple optimization highlights how the function sometimes calls upon an enormous amount of additional volume into the domain, leading to a significant drop in the objective. However, within 2 iterations, this drop is mitigated due to the substantial material additions and volume constraints, which remove the excess volume from the domain.

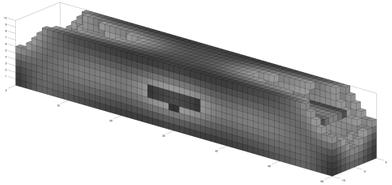
Before algorithm

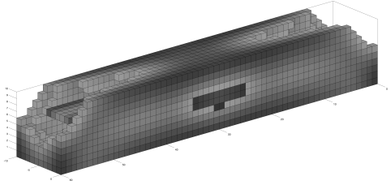


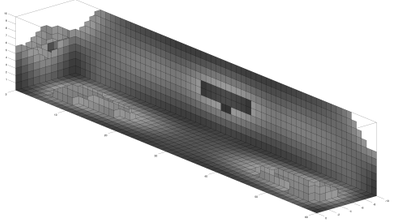
After algorithm











7.6.1 Islanding problem speed and comparison

Dijkstra function

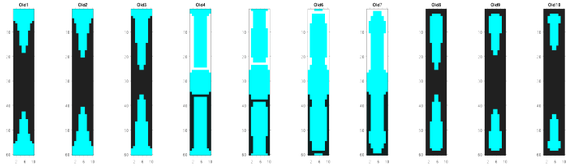
The configuration mirrors the Matlab function presented earlier, yet the absence of "objective dips" really shows the significance of the change in the reconnection algorithm.

Single-element paths are generated, significantly reducing their impact on the total volume within the domain. Consequently, some elements remain within or are not directly expelled from the domain, facilitating reconnection with less volume required for additional reconnections.

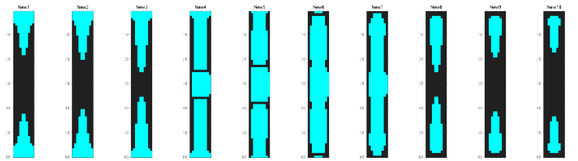
The Dijkstra method exhibits a slight speed enhancement, completing eight iterations within the domain in a total of 12 seconds, marking a 125% increase in efficiency. Nonetheless, the entire optimization loop still necessitates 35 seconds.

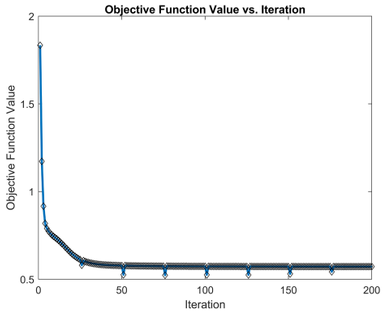
Displayed below is the minimal volume essential for island reconnection in a bottom-to-top reconnection scenario.

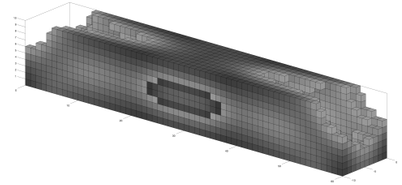
Before algorithm

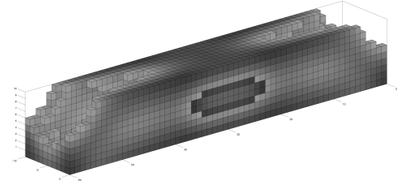


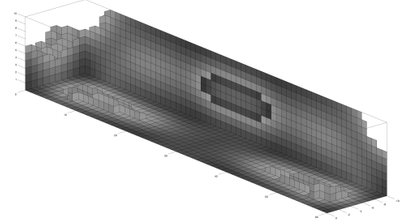
After algorithm











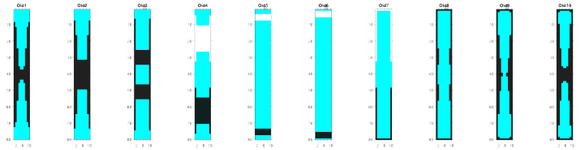
7.6.2 Islanding problem speed and comparison

Matlab islanding function

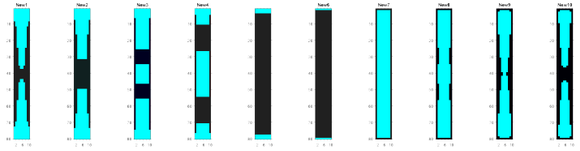
In the second test setup, the design domain spans 30 x 10 x 10 with a volume of 0.2. Modeled closely after the preceding iteration, this test employs the Matlab function in the 200th iteration. Similar to the prior test, the presence of “dips” is discernible but far less pronounced due to fewer parts requiring reconnection.

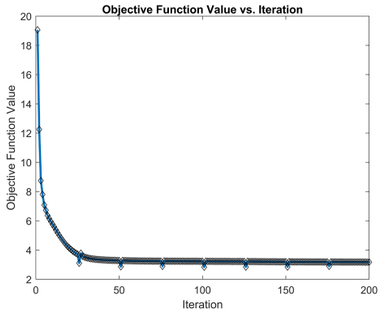
During this optimization, the Matlab function adeptly reconnects layers 4 and 7, forming symmetrical connections given the consistent distance for reconnection.
However, layers 5 and 6 pose a considerable challenge, demanding a substantial number of elements for island reconnection. Since all elements are evenly spaced, volumetric filling across the entire layer occurs, deviating from the intended optimization scope. Something that can be seen is the difference in these layers were there are elements still present due to earlier connections by the Matlab function. As they are not apparant in the Dijkstra function.
Hence, this optimization coupled with the Matlab fucntion is not advisable.

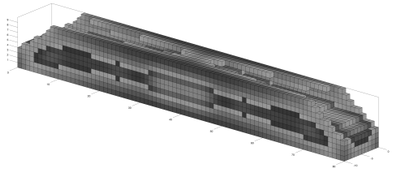
Before algorithm

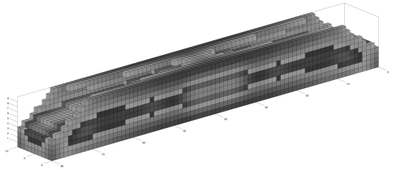


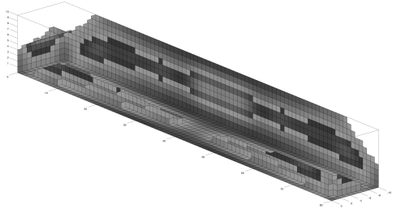
After algorithm











7.6.2 Islanding problem speed and comparison

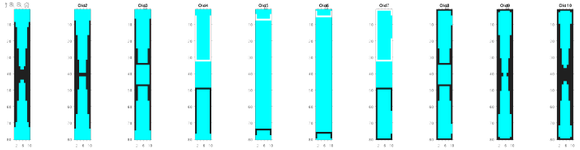
Dijkstra function

Here the same the same design domain spans 30 x 10 x 10 with a volume of 0.2. Modeled closely after the preceding iteration, however this time with the Dijkstra.

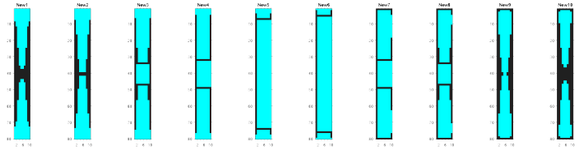
Here the princples are the inverse of the Matlab function as, the reconnection show promise in layer 5 and 6. Due to the addition of a less possible volume addition to the nodes as possible. However for layer 4 and 7. Dissimularity is the answer of the reconnecting. The islands are thereby connected however, the overal optimization would arguable perform better when there is more simularity within the final domain. Also here small reconnection path are visuable by the earlier itterations, only consisting of 1 row of elements.

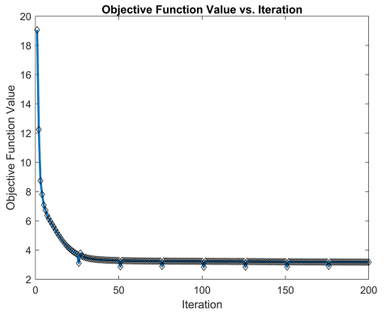
Recommendations would strangly enough for this typical optimization to combining the Matlab function and Dijkstra, by desiding which algorihm to surpress on which layer.

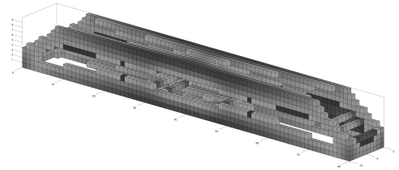
Before algorithm

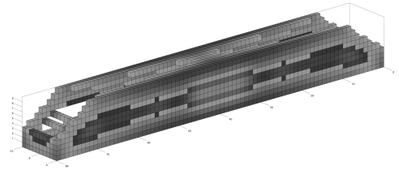


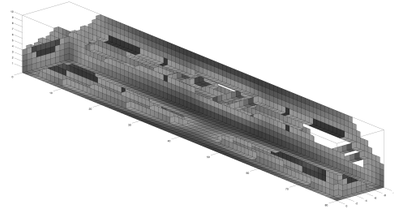
After algorithm











7.6.3 Islanding problem speed and comparison

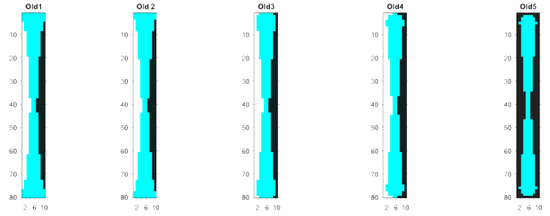
Matlab islanding function

The domain now consists of 30 x 10 x 5 elements, configured to accommodate an overhang angle of 63.4 degrees (2 times higher than the lenght or width), extending from the base plate to the domain's top.

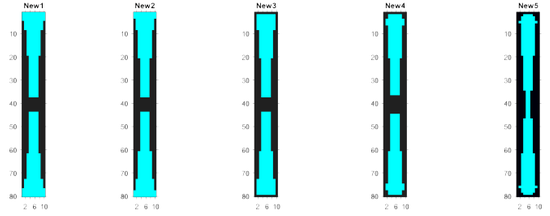
In this setup, the striking resemblance in the final output suggests the potential for symmetrical results on both sides. However, certain layers exhibit elements distributed across the top, middle, and bottom due to their even spacing. Consequently, these elements are constrained to fit within the domain.

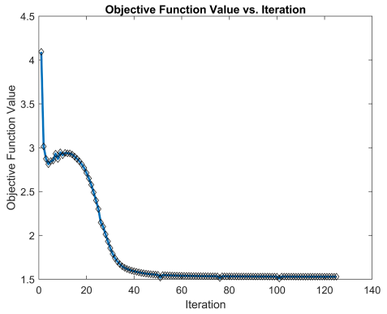
Regarding the objective, employing the overhang angle necessitates some time for the optimization to stabilize. This is attributed to the alignment where certain elements support the higher elements within the domain, impacting the optimization process.

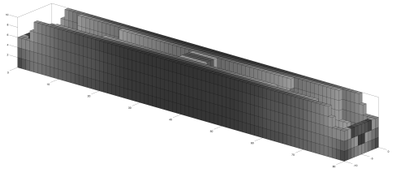
Before algorithm

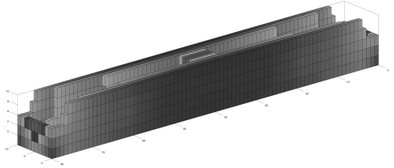


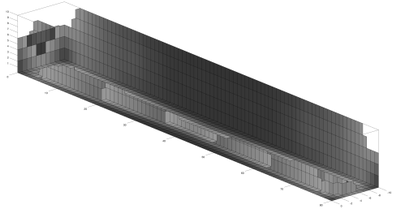
After algorithm











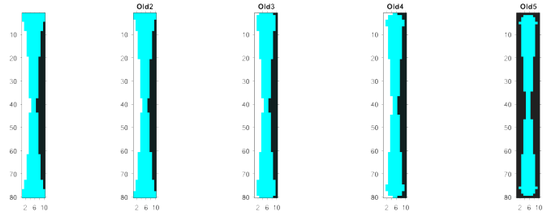
7.6.3 Islanding problem speed and comparison

Dijkstra function

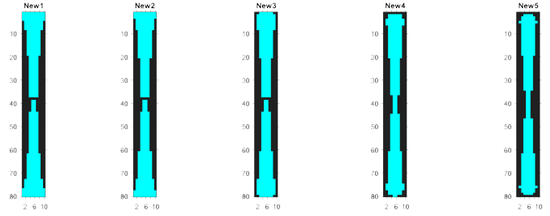
The objective function remains nearly identical, reflecting in similar objectives and final outcomes. In this scenario, Dijkstra's algorithm demonstrates its efficiency by requiring connection to only one item, typically closest to the top-right of the islanding field. Consequently, it demands significantly less volume for island reconnection.

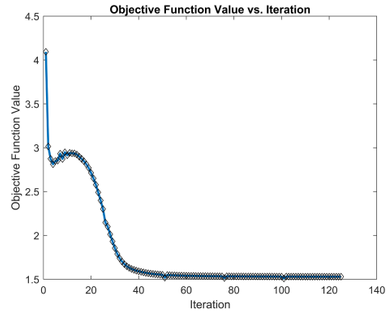
Comparatively, Dijkstra consistently outpaces the Matlab function due to its streamlined approach. While Matlab necessitates assigning different numbers to distinct islands before initiating sequential connections, Dijkstra simply labels all islands as one and identifies if they lack interconnections, irrespective of their quantity. Importantly, there appears to be no limit to the number of islands that can be reconnected for both algorithms, with a 100% assurance of successful reconnection in every case.

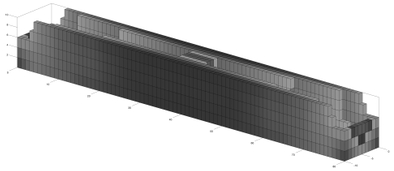
Before algorithm

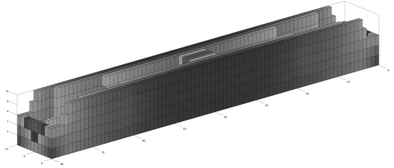


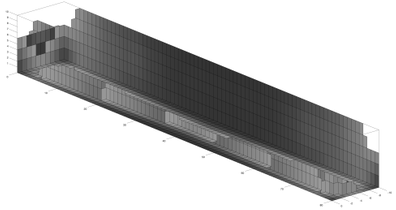
After algorithm











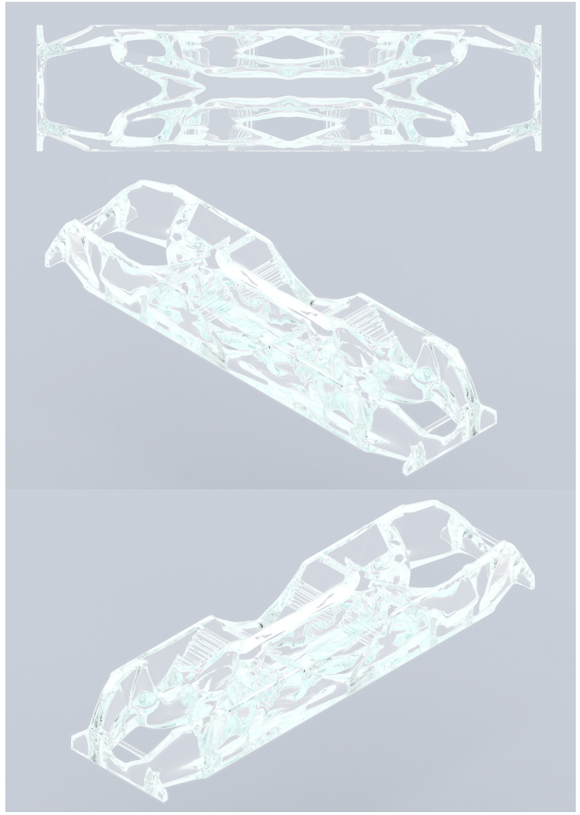
7.6.4 Islanding problem speed and comparison

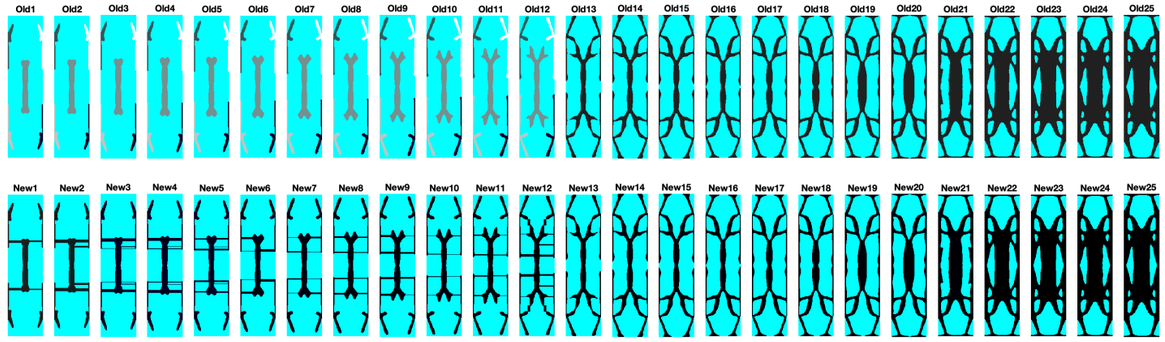
Matlab islanding function

This domain has been specifically chosen to highlight the disparities between the two algorithms. Here, one can clearly observe the distinctiveness of the connection types and their significant impact on the final geometry.

The selected shape measures 400 x 50 x 70, with a volume occupying just 0.25%. This dimension really challenges both algorithms, as numerous islands persist throughout the layers. While the feasibility of the final shape is not fully achieved due to excessive tensile forces, it effectively showcases the island reconnection, highlighting the differences between the two, particularly in this MATLAB function domain. The increasement of material can be really good seen on for example layer 2 till 12. where the algorihm choises both sides of the island to connect with the other islands.

It's evident that neither of the islanding functions adequately addresses the overhang filter, resulting in considerable bridging in the final geometry. Substantial post-processing would be necessary to refine the connections in a manner that ensures minimal bridging.





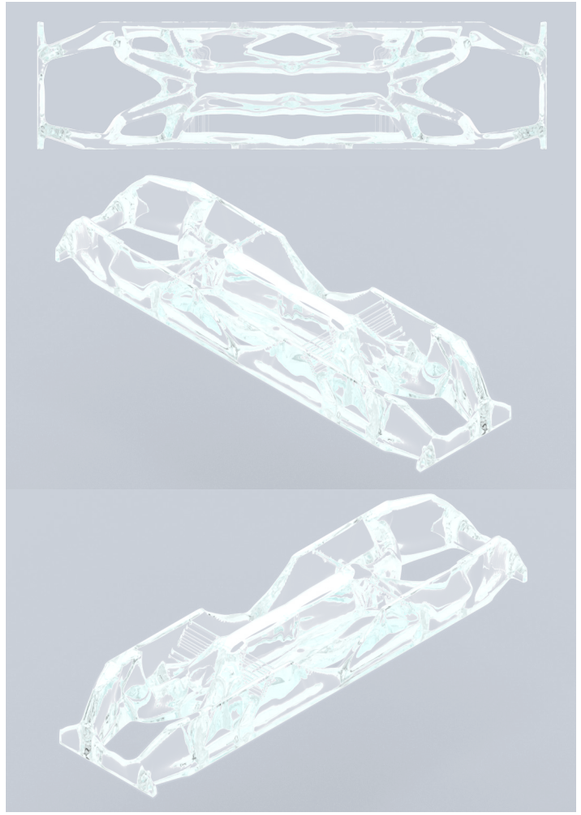
7.6.4 Islanding problem speed and comparison

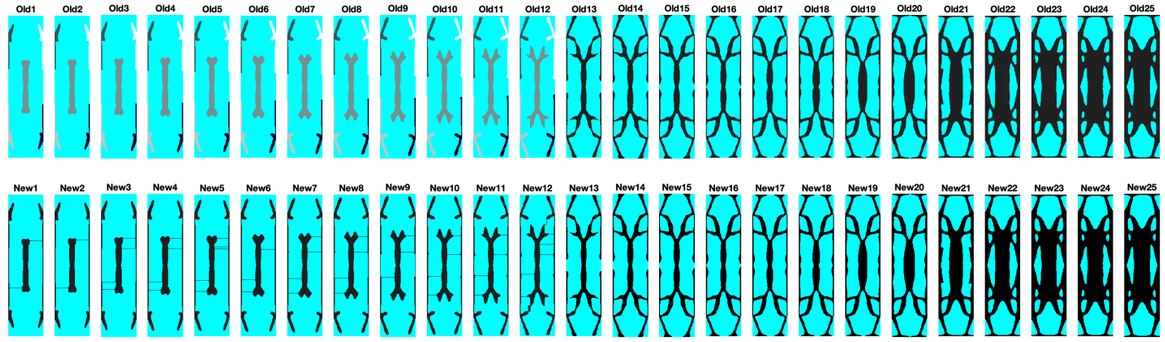
Dijkstra function

On the other hand, the Dijkstra function demonstrates a different approach. Although it still relies on bridging elements for connection, the requirement for fewer additional elements is clearly visuable.

The reason why most islandings are positioned at the top part can be attributed to the tensile forces within the structure. Hence, more material is necessary to withstand these forces compared to compressive forces, which glass handles better. One of the drawbacks is seen in this output, the algorithm fits the reconnection on the shortest route path, while it would be handy to cluster the islanding connecion lines close together so they can be more structurally and more feaseable.

Consequently, in the lower boundaries, nearly all layers are fully connected. With the introduction of more material within the domain, logically, fewer connections between paths need to be configured, as the additional elements will likely resemble the bottom layers more closely.





7.6.5 Islanding problem orientation reconnecting (print-direction)

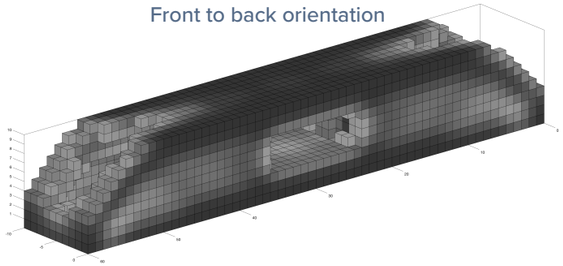
Currently, the filter has only been showcased for the Bottom-Top orientation. However, in certain scenarios, the orientation might vary, such as Front-Back or Left-Right. Hence, these alternatives are illustrated here.

This approach proves effective, allowing users to select the direction for overhangs and printing connections. This flexibility ensures optimal optimization, as the Bottom-Top orientation may not always yield the best results.

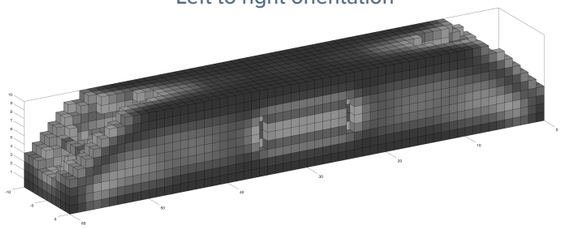
The difference can sometimes be even more pronounced. For instance, the Dijkstragrid takes approximately 20% longer when the longer orientation is displayed (left-right) compared to the more layered optimizations. However, the most time-consuming tasks are the saving and plotting of outcomes, consuming nearly 60% of the total time. Therefore, it's advisable to disable plotting when frequent testing is conducted. This allows the algorithm to focus solely on performing the necessary reconnecting before continuing with the optimization loop.

With this final resutls, all options for this filter are thoroughly evaluated, and conclusions are drawn. These findings are detailed on the following pages.

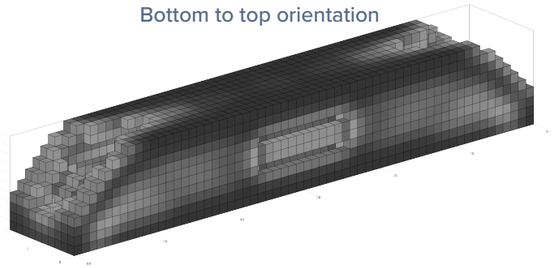
Front to back orientation



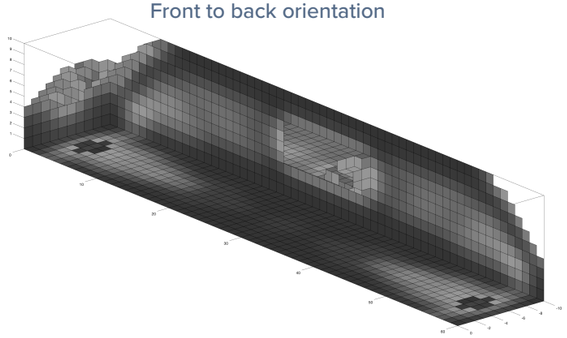
Left to right orientation



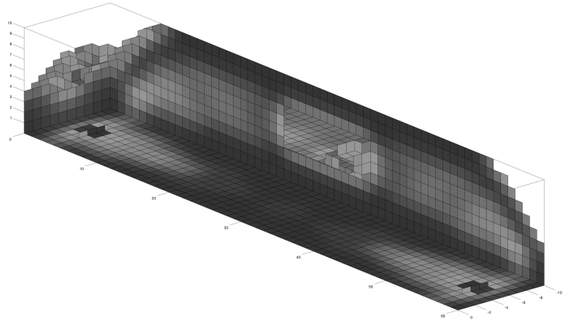
Bottom to top orientation



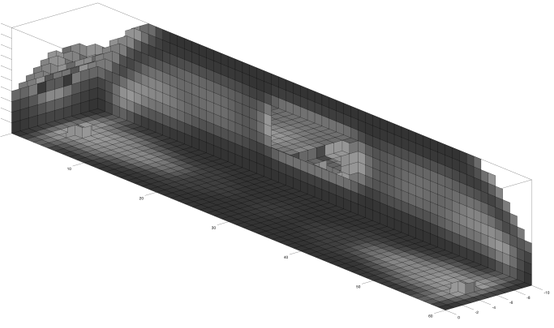
Front to back orientation



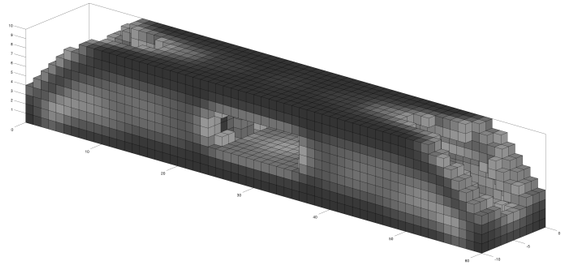
Left to right orientation



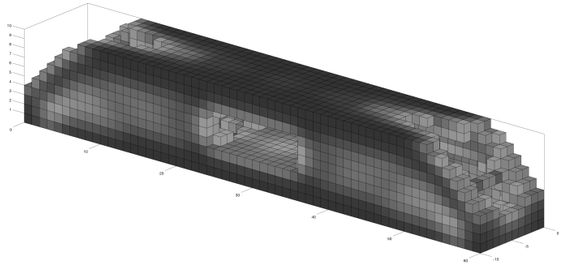
Bottom to top orientation



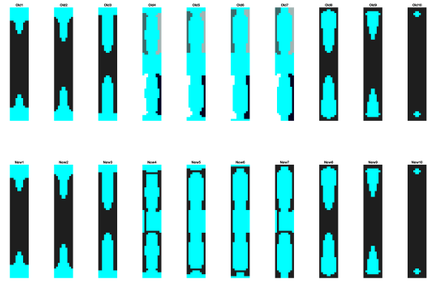
Front to back orientation



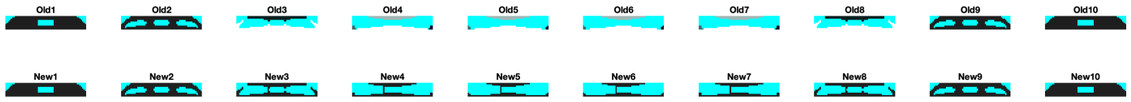
Left to right orientation



Bottom to top orientation



Front to back orientation



Left to right orientation



7.6.6 Islanding problem conclusion & discussion

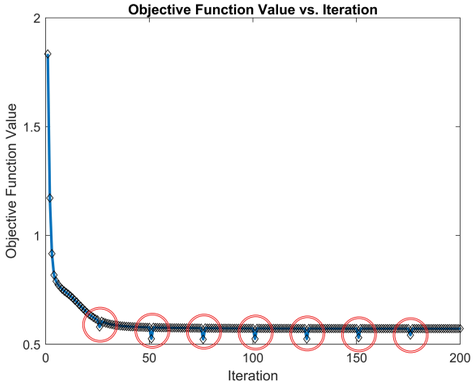
Both connection algorithms are efficient, consuming less than 1% of the total calculation time for larger domains. The transition from 15 seconds to nearly 6 minutes occurs with larger domains where extensive reconnecting is necessary, yet in the context of the overall optimization loop, which spans hours, these increments are negligible.

When properly configured, the algorithm operates flawlessly every time, ensuring continuous paths for printed parts by reconnecting all components. However, one drawback of the Matlab function is its occasional requirement for a significant increase in material for reconnection. Attempts to offset this increase by subtracting from other elements proved unsuccessful in achieving feasible optimization convergence. Therefore, the excess material was instead included within the user-defined boundary. Graphs depicting dips alongside illustrate the resultant objective improvement based on this approach. They also highlight a common issue wherein some elements remain for reconnection, while others, deemed structurally unimportant, are removed from the domain. The remaining elements contribute to a favorable function within the optimization loop, though in certain scenarios, it may resemble a post-processing step. Integrating it immediately after the optimization loop, without user intervention, ensures efficiency and seamless operation.

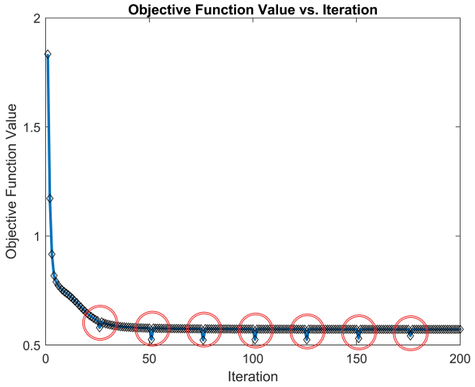
Recommendations

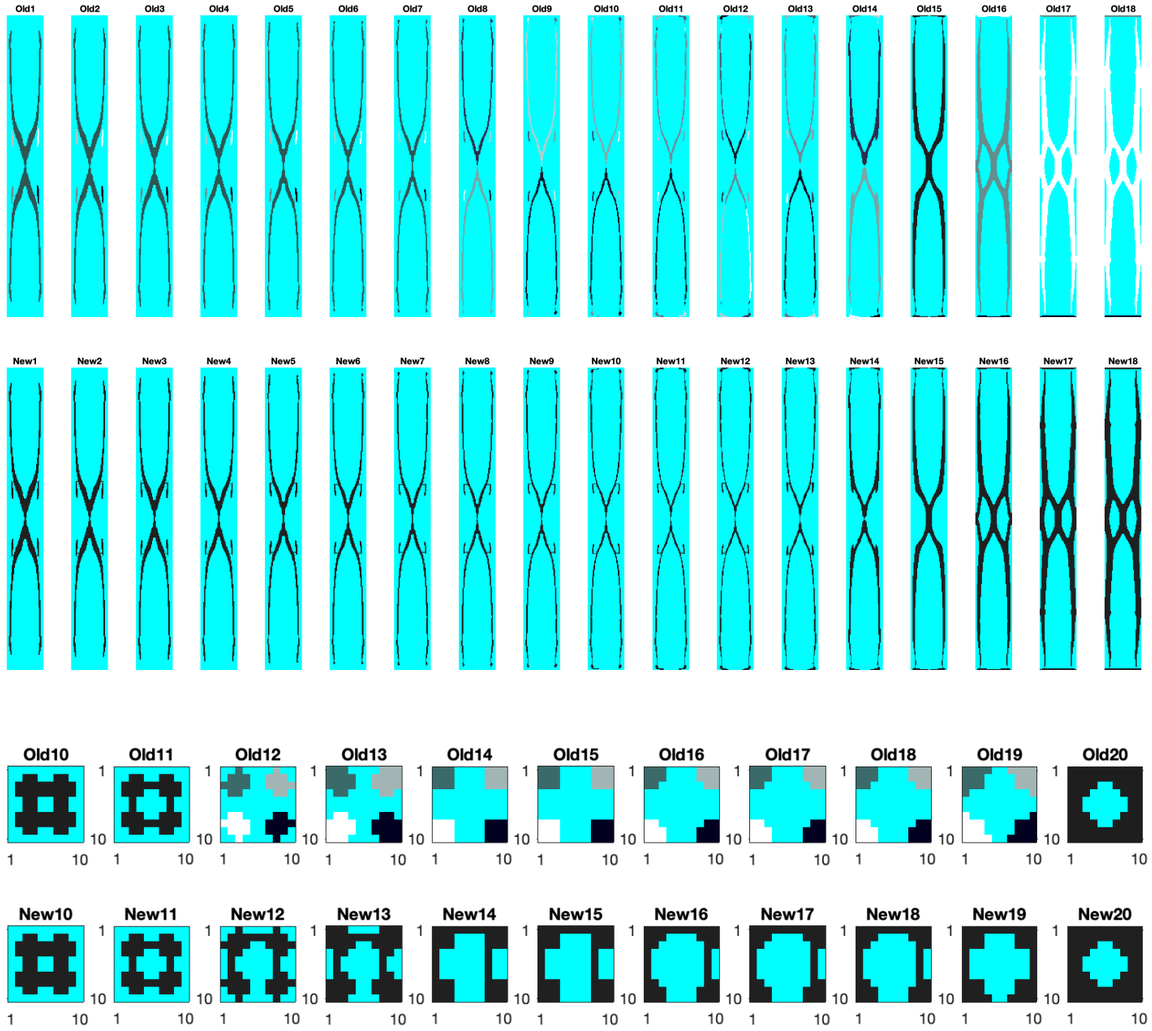
Replacing Dijkstra with an algorithm that prioritizes nodes already possessing values, guided by a blueprint, would enhance its suitability within the overall optimization loop. This approach shifts focus away from always selecting the shortest path as the optimal choice. Moreover, integrating considerations for overhang angles directly into the algorithm would further improve its efficacy. Currently, connections often rely on bridging, offering limited optimization benefits. By incorporating these two enhancements, a more refined reconnection process could be achieved, contributing significantly to the overall optimization objectives.

low volume additions (MATLAB function)



low volume additions (Dijkstra function)





7.7 Boundary offsetting

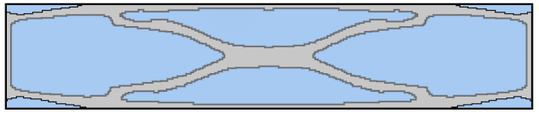
The boundary offsetting process remains consistent between the tested versions and the output of the final algorithm, resulting in a structure where each substructure acts as a path for a nozzle path multiplied by a factor of 2. Unlike the previous version, the color schemes are already predetermined, which eliminated the gradient filter and simplifying the generation of outlines on the inner and outer boundaries of each layer.

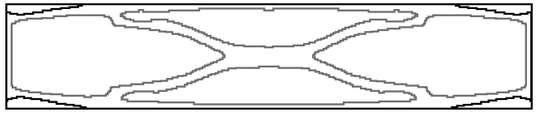
Smoothing the shape becomes a more straightforward task as the algorithm doesn't require adjustment with a gradient filter. However, if the color isn't black, then the boundary is omitted. While the resolution suffices for this master thesis, recommendations suggest utilizing a more detailed starting domain for larger domains or smaller nozzle sizes.

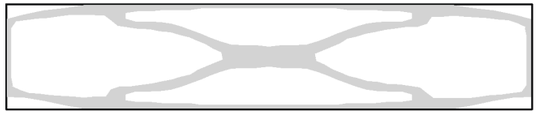
The domains are smoothed out and classified into inner or outer boundaries to comply with the maximum area of optimization. Once this is achieved, a clean and neat boundary extrusion is created. Precision in smoothing isn't critical since the offsetting process will eventually enlarge the overall domain.











7.7 Boundary offsetting

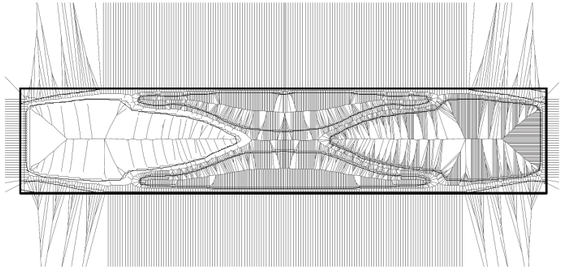
The Voronoi diagram is applied to the boundaries, processing the middle points of the triangles within these boundaries to generate the medial axis. An angular medial axis is then depicted.

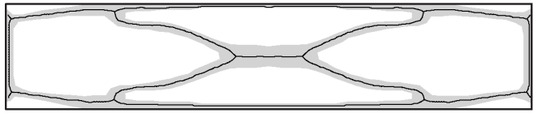
Circles are generated for each line where the medial axis intersects. Initially, very short extensions of the medial axis on the main line are deleted using a minimum distance filter, primarily due to the roughness of the boundary settings. Occasionally, due this roughness the medial axis attempts to connect to corner pieces, necessitating this filtering process. The final medial axis is represented as a continuous line with an additional four short lines, each connected to one end and extending to one of the corners.

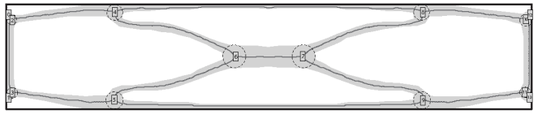
Circles, each accommodating precisely twice the nozzle size, are positioned at the ends or intersections of the medial axis. These circles denote connecting points with the boundaries where the medial axis intersects. They are numbered sequentially from the top-left corner to the bottom-right corner.

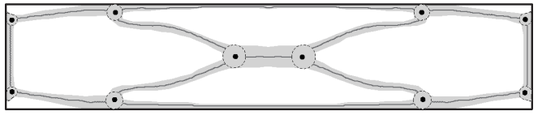
In cases where circles intersect, they merge into an average one, forming a single final boundary area for offsetting. This facilitates creating space for the nozzle to define the boundaries of different parts between these merged boundaries.

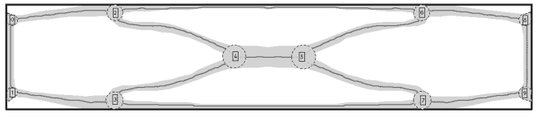
The medial axis within the circles is of less importance, as the subsequent step involves generating an area within all the connection points between the circles and the boundaries they intersect with. This area initiates with only medial axis lines running between these offsetted boundary areas, thereby connecting different boundaries with the appropriate path length between them.





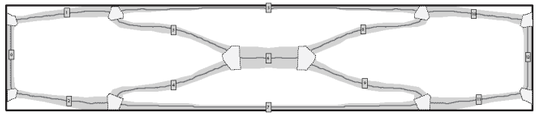




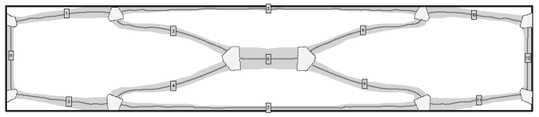


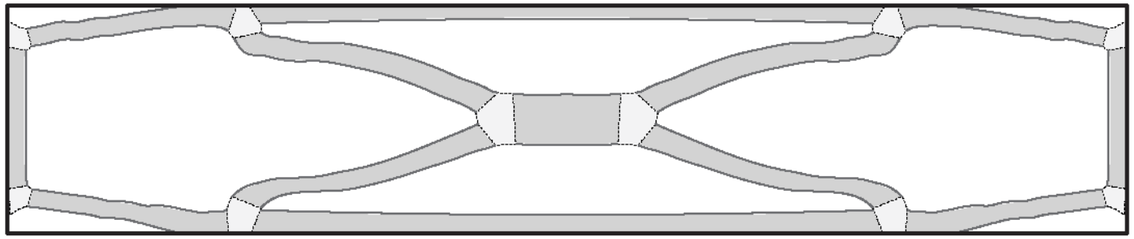
7.7 Boundary offsetting

The medial axis between the connection areas is smoothed out, which has proven to be a better alternative then to directly outputting the algorithm or offsetting the nozzle distance, as it resulted in angular outcomes.
Consequently, the lines are smoothed out first and then offset. The end result is a boundary accommodating a multiple of the nozzle size.
Next, all planar sheets are extruded, connected, and smoothed out using the same principle as the optimization algorithm, leading to an STL file. Which can be imported into various software applications.



Due to time constraints within the duration of the master thesis, no path planning was conducted. Instead, this was eventually carried out using either SLIC3R or CURA, software that can generates G-code for the printer. Further research could potentially yield advancements where under- or over-filling is considered, altering the path accordingly. Or even changing the total path planning to perform better by connecting the path on different order. Non-planar printing is tried for this thesis. However, to conclude for the nozzle offseting, it was sufficient to demonstrate the concept of fitting the nozzle path within the domain.





7.8 STL conversion time and performance

The conversion of STL files for large design domains can be exceedingly time-consuming, ranging from a mere 10 seconds to over 60 minutes. This timeframe fluctuates within a range of 0.1% to 10%.

While this thesis did not specifically target reducing conversion time, there is potential to optimize the overall script. One attempt to save time involved only preserving one of the two STL file variations (ISO or cubic). This saving operation occurred after 100 iterations within the loop, aiming to avoid reaching the time limit of DelftBlue before hitting the maximum loop count or validating the tolerance (tolx).

Notably, mirroring elements during the saving process significantly extended the duration. Nevertheless, this step holds value as it ensures the creation of a perfectly mirrored element, particularly when utilizing the ISO output.



Figure 58: Ai generated cubic art

```
Total time normal script: 07:13:12
Total time for saving: 00:20:08
```

```
Total time normal script: 06:40:25
Total time for saving: 00:26:36
```

```
Total time normal script: 02:31:45
Total time for saving: 00:13:52
```

```
Total time normal script: 02:04:03
Total time for saving: 00:11:18
```

```
Total time normal script: 15:04:53
Total time for saving: 00:44:43
```

```
Total time normal script: 02:36:23
Total time for saving: 00:00:10
```

```
Total time normal script: 02:46:31
Total time for saving: 00:00:16
```

```
Total time normal script: 02:15:24
Total time for saving: 00:00:14
```

```
Total time normal script: 19:43:03
Total time for saving: 00:41:52
```

```
Total time normal script: 05:12:08
Total time for saving: 00:23:52
```

```
Total time normal script: 01:19:35
Total time for saving: 00:09:04
```

```
Total time normal script: 11:19:55
Total time for saving: 00:57:22
```

```
Total time normal script: 04:37:42
Total time for saving: 00:05:59
```

```
Total time normal script: 14:19:10
Total time for saving: 00:47:12
```

7.9 Layer by Layer objectives and maximum stresses

Here, the objectives are compared against the baseline when the overhang filter is positioned at X, indicating it's not in use. This establishes a baseline for average time and objective comparisons across different orientations.

Three different domains are selected, with one version having a low volume of 0.2, equivalent to 20% of the domain filled with volume. Significant differences are observed when using the overhang angle, particularly on the right and left sides. This disparity is mainly due to the domain's inability to accommodate elements that are both objective-wise suitable and capable of supporting other elements.

Consequently, bottom and top orientations are deemed most favorable in this thesis. Interestingly, in these examples, the overall calculation time is quicker when utilizing the overhang filter compared to when it's not used. This is primarily attributed to the sparse matrix, as the overhang filter at times excludes elements from the optimization loop, thereby increasing speed.
However, this may also lead to some issues, although when combined with the MMA solver, it shouldn't significantly limit the possibilities for element placement, as stated by Langelaar.

Furthermore, the Dijkstra grid is demonstrated here, consuming only a small percentage of the total algorithm time. When used with the overhang filter, the process is simplified as the filter clusters elements, reducing the need for the islanding function as no islands are recognized within the first step of the algorithm. The reconnection process takes therefore no more time than this initial step.

200 x 30 x 50 vol.0.4 (63.435)

Orientation	Script (time)	Time (seconds)	Percentage	Dijkstra	Percentage	objective	percentage
X (no-overhang)	01:54:51	6894	100%	01:22	100%	23.14	100%
Bottom	02:01:04	7624	110%	01:26	105%	29.60	128%
Top	01:59:56	7196	104%	01:21	99%	42.25	182%
Right side	02:12:40	7960	115%	00:41	50%	186.16	800%
Left side	02:11:38	7952	115%	00:43	52%	198.54	858%

200 x 30 x 80 vol.0.4 (63.435)

Orientation	Script (time)	Time (seconds)	Percentage	Dijkstra	Percentage	objective	percentage
X (no-overhang)	03:59:39	14379	100%	03:03	100%	39.76	100%
Bottom	04:09:41	14981	104%	02:52	94%	47.12	119%
Top	04:11:09	15069	105%	02:39	87%	52.91	133%
Right side	04:13:16	15196	106%	00:33	18%	245.98	619%
Left side	04:14:04	15244	106%	00:48	26%	285.75	719%

200 x 30 x 80 vol.0.2 (63.435)

Orientation	Script (time)	Time (seconds)	Percentage	Dijkstra	Percentage	objective	percentage
X (no-overhang)	04:13:20	15200	100%	01:10	100%	98.58	100%
Bottom	03:18:18	11898	78%	00:40	57%	152.43	155%
Top	03:26:35	12395	82%	00:28	40%	179.50	182%
Right side	04:06:35	14795	98%	00:29	41%	1503147.65	>..%
Left side	04:22:40	15760	104%	00:30	43%	437364.56	>..%

7.10 Conclusions comparison

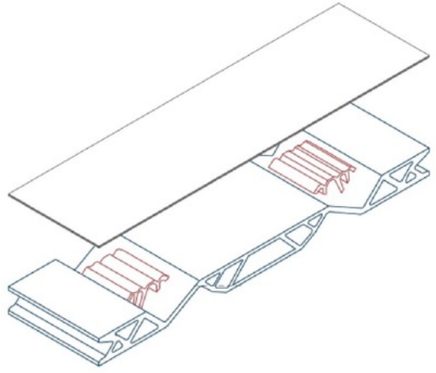
The comparison is conducted utilizing the final output element, with dimensions of 4200x800x300 and occupying 30% of the volume, consistent with the predecessors in both design domain and volume. The only big difference is that the predecessors focused on minimizing volume, this thesis prioritizes compliance minimization instead. (This particular design wasn't chosen for the master thesis due to concerns regarding structural integrity, a topic delved into further in the subsequent chapter.)

Nonetheless, it exhibits the most notable disparities among the various elements and unmistakably demonstrates the feasibility of construction, particularly in the realm of 3D but also applicable to 2D. The number of elements saw a significant increase, with Eva's implementation increasing from 6,300 to 36,000 elements, marking a 570% increase. In this thesis, 1,008,000 elements were employed for this shape, reflecting a staggering 2800% rise. Consequently, this approach offers a much more precise depiction of the shape and intricate detailing of the element.

Moreover, it obviates the need for post-processing, thus eliminating potential interpretations from the researcher regarding the connection between elements, which could sway the final outcome either favorably or unfavorably.
As previously discussed, the incorporation of the overhang filter, coupled with the heightened elements, constitutes the primary addition driving the alteration in the element's shape. This alteration stands out as the most significant visual change for feasibility assessment. It's evident that Eva's optimization would pose considerable production challenges, particularly when considering different manufacturing methods such as casting. This new algorithm clearly addresses these concerns.

Furthermore, it can be demonstrated that even in casting processes, with a 3D printed sand mold, this 3D print-ready optimization would outperform the predecessors. This is because more spatial elements are directly accessible, facilitating easier treatment and cleaning, resulting in the glass achieving the visual appearance it's renowned for.

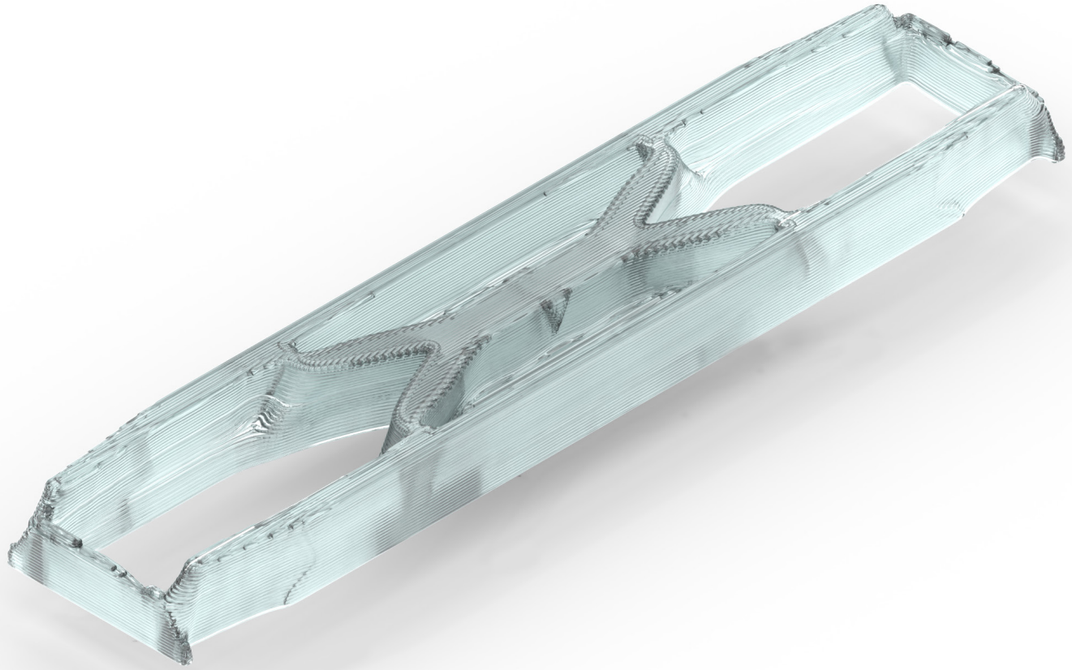
Anna-Maria Koniari 2D extruded optimization with 6.300 elements



Eva Schoenmaker 3D optimization with 36.000 elements



This 3D algorihm with constraints for 3D printing with 1.008.000 elements

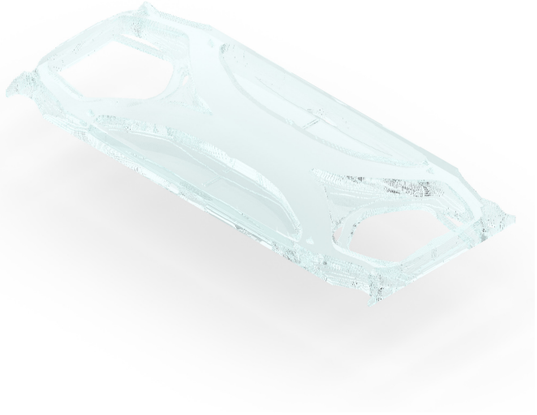


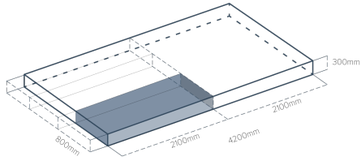
7.11 Conclusions final case study 1

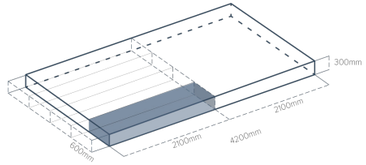
During algorithm testing, exclusions were made regarding the dimensions of the bridge. Among the six different variants initially considered, three were eliminated following algorithmic assessment. The first two were discarded due to the experimental nature of the monolithic glass bridge. It was crucial to ensure that the bridge wouldn't completely collapse if a particular section experienced forces beyond its capacity. Therefore, a safety factor was integrated, ensuring that the glass float panels remain in place even if a segment collapses or cracks. Consequently, both the total width and half-width configurations were ruled out.

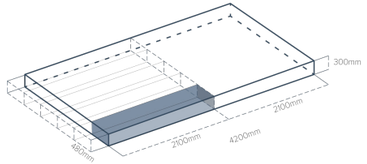
Additionally, the smallest version, with a width of 400, divided the bridge into six parts, was abandoned. This decision comes from the observation that in smaller configurations, optimization tended to densely fill the design domain with two side "beams," suggesting a structure prone to buckling. A visualization of the 400-width bridge on the following page reveals that only the middle bottom section is filled, compensating for the glass's weak tensile performance. The connection between the two islands is centrally located, while the rest of the glass forms two nearly straight beams towards the opposite side, prompting discontinuation of this version.

Further experimentation was done on the versions measuring 800mm, 600mm, and 480mm in width, dividing the bridge into three, four, or five parts respectively. This approach aimed to satisfy the safety factor requirement, ensuring that the failure of one part wouldn't compromise the entire structure.

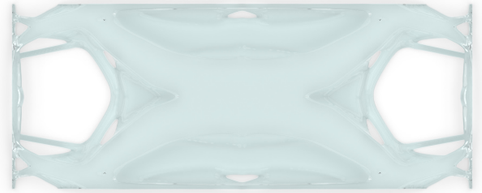




















7.13 Discussion case study 1

One of the primary limitations of developing this algorithm is the expansion of the domain and consequently the increase in the number of elements within it. This growth demands substantial computational power, particularly in terms of RAM. Meeting this requirement necessitates significant computational resources. In cases where there is an insufficient amount of RAM allocated, the algorithm may halt due to an out-of-memory handler error.

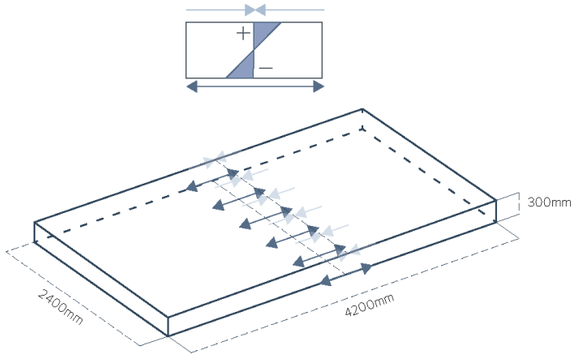
To address this issue, Eva implemented a useful tool in her script, which processes steps of the domain at a time. While this approach aids in managing memory usage, questions arise regarding the feasibility of integrating it into this algorithm seamlessly. Nevertheless, there is significant potential in incorporating such a method into this algorithm to enhance its efficiency and scalability.

The algorithm tends to place material predominantly in the middle bottom part of the design, which can be easily explained by the fact that the central area of the bridge experiences the most tensile forces, as illustrated in the figure on this page. This tendency is also advantageous for the overhang filter, as it allows the structure to ‘grow’ more easily to higher areas within the design domain. Consequently, the overhang filter contributes to a more optimal outcome when using monolithic glass.

However, the current boundary condition presents opportunities for improvement. Currently, nodes at each bottom end of the design domain plate are fixed, which differs from the approach used by Anna-Maria and Eva. To achieve better results, it would be beneficial to explore multiple variants with different support configurations, and to evaluate the various outcomes.

Additionally, the front or back orientation of the overhang filter is not ideal within this design domain, as the limited height restricts node growth.

Further research is necessary to verify whether the results are desirable and perform as expected in this master thesis.




```
slurmstepd: error: Detected 1 oom-kill event(s) in StepId=3678947.0. Some of your processes may have been killed by the cgroup out-of-memory handler.
srun: error: mem009: task 0: Out Of Memory
srun: launch/slurm: _step_signal: Terminating StepId=3678947.0
slurmstepd: error: Detected 1 oom-kill event(s) in StepId=3678947.batch. Some of your processes may have been killed by the cgroup out-of-memory handler.
```


7.12 Conclusions final case study 2

To demonstrate that the algorithm can produce outcomes beyond just bridges, an object with maximum dimensions of 320 x 320 x 350 was generated, focusing on areas where compression is most likely to occur. These dimensions were selected to create a stepping stone, intended to replace the corrugated concrete shown in the figure on this page.

The loading conditions, depicted here, were chosen to accommodate the weight of a single person, ensuring the objective is easily met (overdimensioning).
This approach proved useful when experimenting with different variables, as lower volume designs are highly susceptible to sharp corners and discontinuities throughout the layers. Various printing orientations were evaluated, but ultimately a top-down print order was selected.

Extending the study, future research could explore additional shapes and structures to further validate the algorithm’s versatility. By testing under varying load conditions and material properties, the robustness of the algorithm can be better understood.

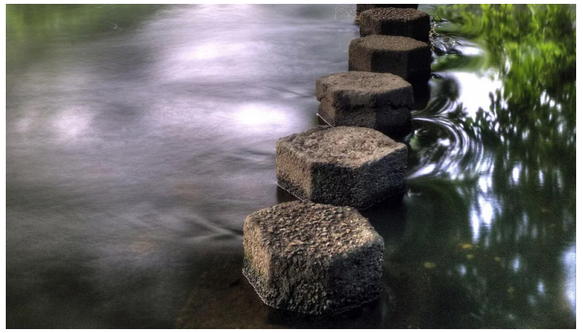
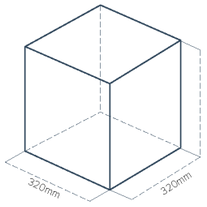


Figure 10: https://www.nationaltrust.org.uk/visit/surrey/box-hill/box-hill-stepping-stones



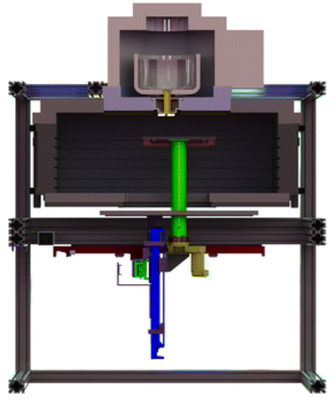
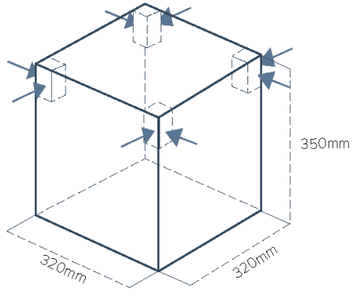
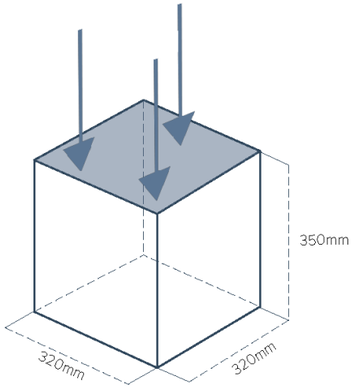
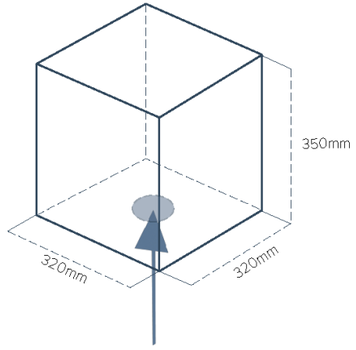


Figure 11: oxman.com

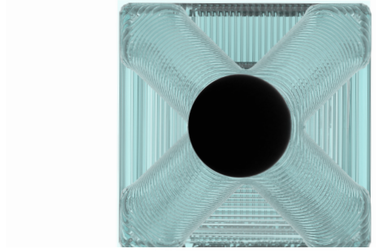


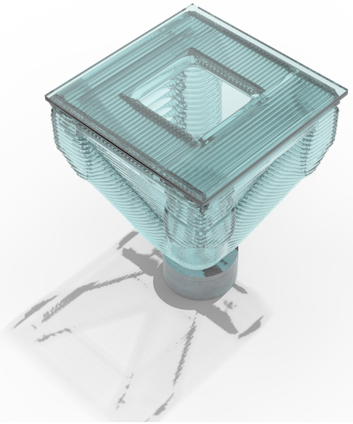


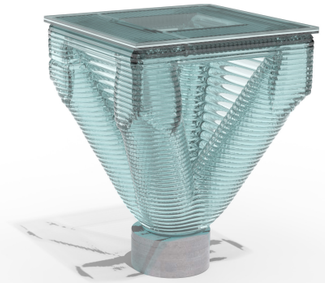


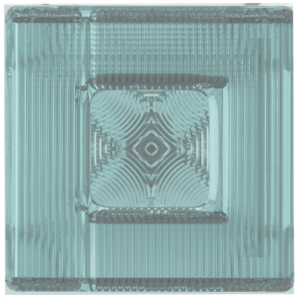
7.12 Conclusions final case study 2

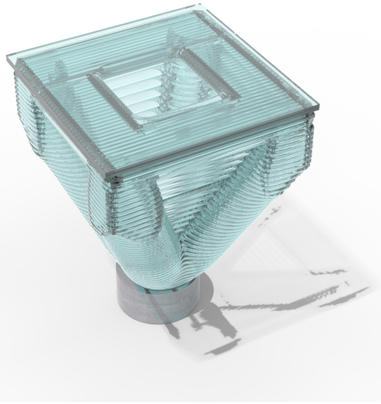
The outcome of this optimization is, in my opinion, very aesthetically pleasing. The symmetrical portions of the stepping stone form a harmonious combination. The rounded columns are beautifully sliced using the STL slicer built into the algorithm. These renders were created with the Cycles render engine in Rhino 3D, while the environmental renders were made using Blender. Despite initial challenges in setting up, I am very pleased with the results. Many programs struggle with glass reflection and transparency, but with the correct settings, Blender proves to be an excellent choice for rendering glass elements. The end renders can be found in the final, chapter 8.







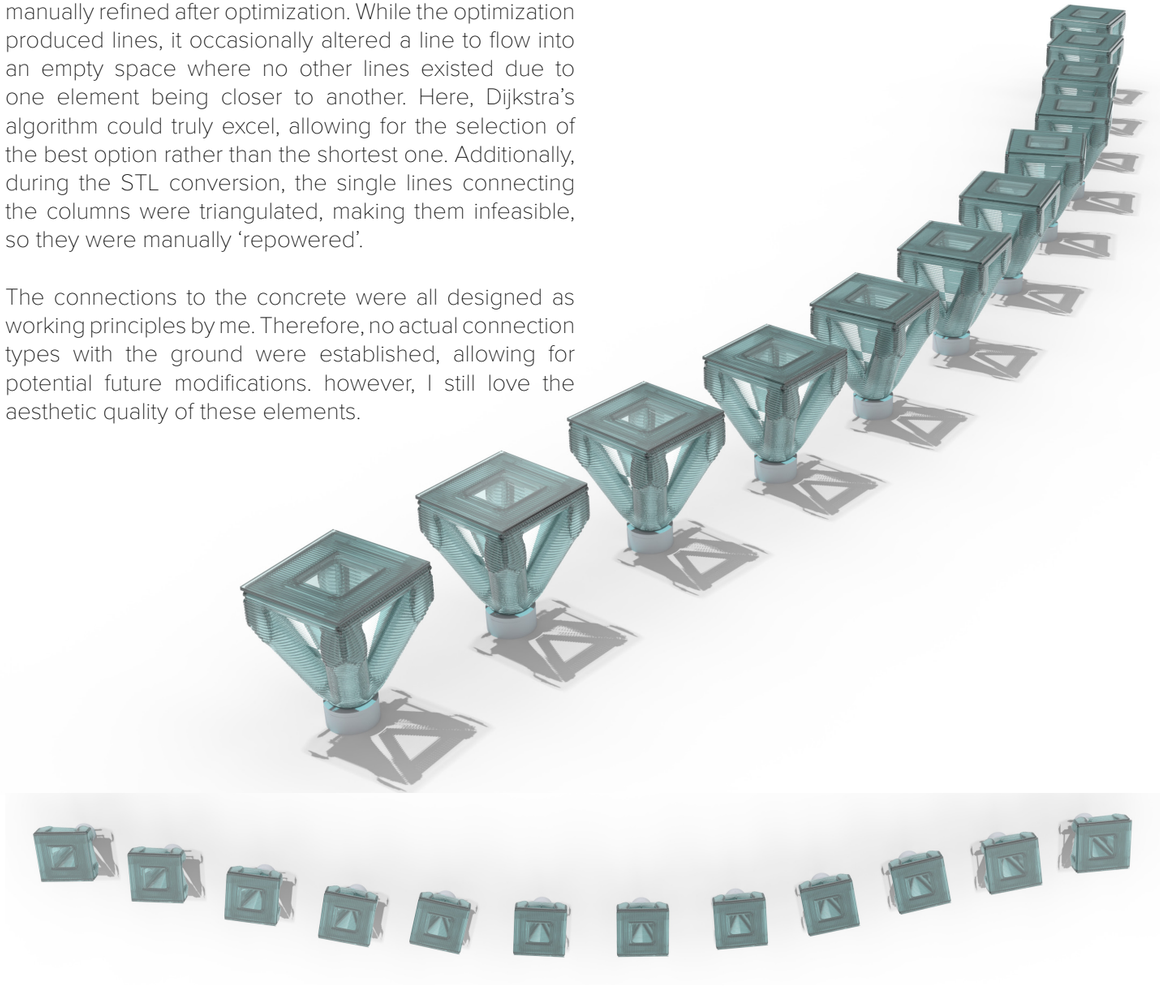




7.13 Discussion case study 2

The result of this algorithm features a plateau with four inward-tapering round columns connected to the concrete using three continuous lines. These lines were manually refined after optimization. While the optimization produced lines, it occasionally altered a line to flow into an empty space where no other lines existed due to one element being closer to another. Here, Dijkstra's algorithm could truly excel, allowing for the selection of the best option rather than the shortest one. Additionally, during the STL conversion, the single lines connecting the columns were triangulated, making them infeasible, so they were manually 'repowered'.

The connections to the concrete were all designed as working principles by me. Therefore, no actual connection types with the ground were established, allowing for potential future modifications. however, I still love the aesthetic quality of these elements.



08

Final

Presented here is the finalized form, meticulously configured to meet specifications and requirements. Additionally, this chapter delves deeper into the intricacies of its design and functionality.

8.1 Final geometry 480

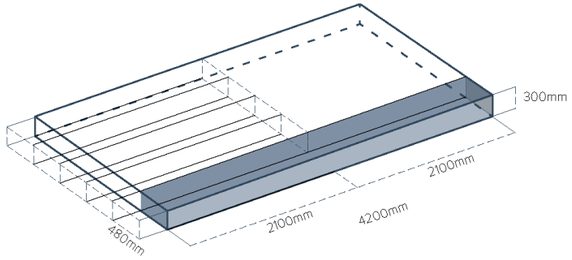
The smallest version of the domain considered for the final geometry has dimensions of 480mm in width and 4200mm in length, with the design domain consisting of half the length mirrored to form the whole bridge element.

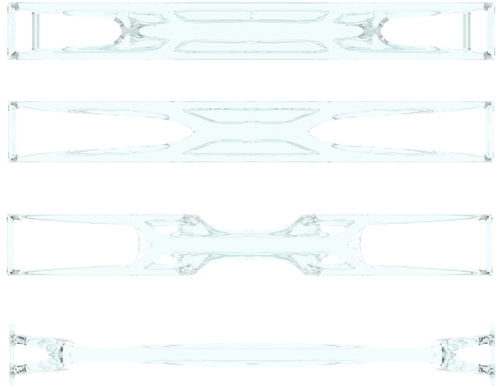
The differences between the various elements are clearly visible. As demonstrated in the following pages, the different versions of this algorithm output vary only in the volume restriction of the total domain. The total volume in the domain ranges from top to bottom: 50%, 40%, 30%, and 20% of the total domain.

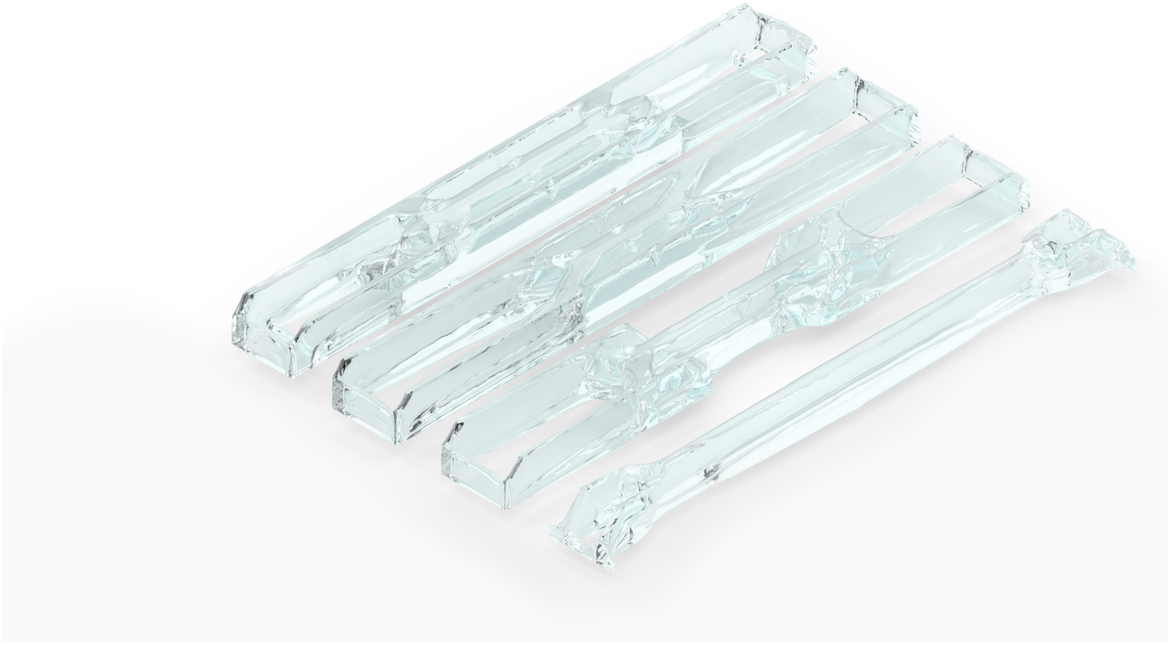
The islanding function is invoked through the algorithm, ensuring feasibility and connectivity between the mirrored parts. It's evident which parts of the final geometry are most critical for the structure. However, the difference between the 30% and 20% elements shows the most variation due to the discontinuation of the flanges on the side of the domain. Unfortunately, these two do not comply with the minimum tensile forces projected on the bridge and thus seem unfit.

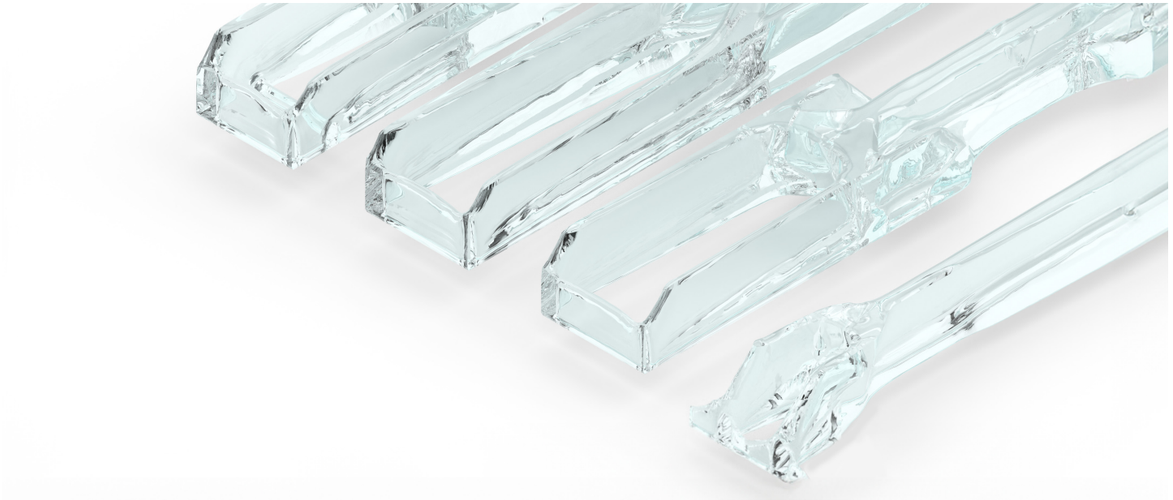
For the domains filled with 40% and 50% volume, feasibility is ensured. Therefore, the optimum lies somewhere between 30% to 40% volume of the domain.
This variant slightly struggles to achieve the right objective compared to the larger domains, therefore this variant is not further tested. However, the smaller variant could still be further investigated to comply with other domains. Personally, I find the 30% volume of this domain interesting due to the discontinuation of the flanges.

Therefore, the reader is referred to the larger domains to establish the final geometry based on the characteristics of the domain.









8.2 Final geometry 600

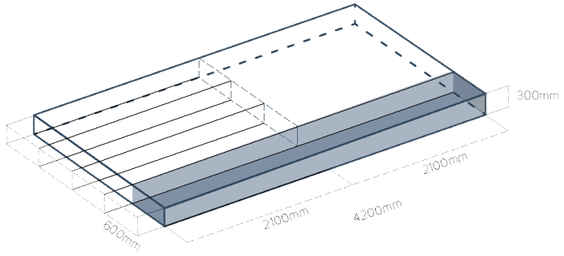
This version of the domain considered for the final geometry measures 600mm in width and 4200mm in length, with half of the length mirrored to form the entire bridge element.

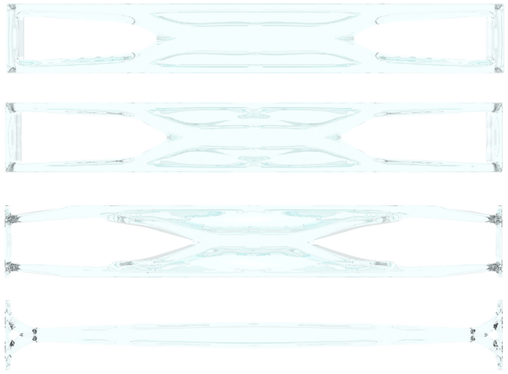
The distinctions between the various elements are evident. The total volume in the domain ranges from top to bottom: 50%, 40%, 30%, and 20% of the total domain. The islanding function is employed through the algorithm, ensuring feasibility and connectivity between the mirrored parts.

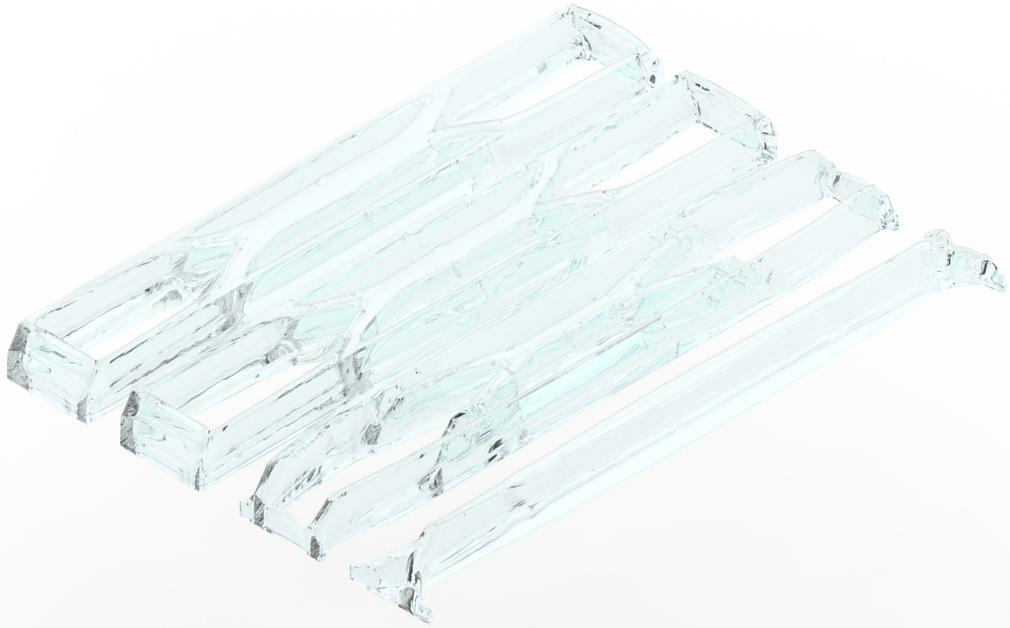
Unlike the smaller domain, the 50%, 40%, and 30% volumes exhibit similarity. While the discontinuity of the flanges was necessary for the smaller domain due to distance and material constraints, here the 30% volume still maintains a consistent beam, forming the primary arm of the bridge.

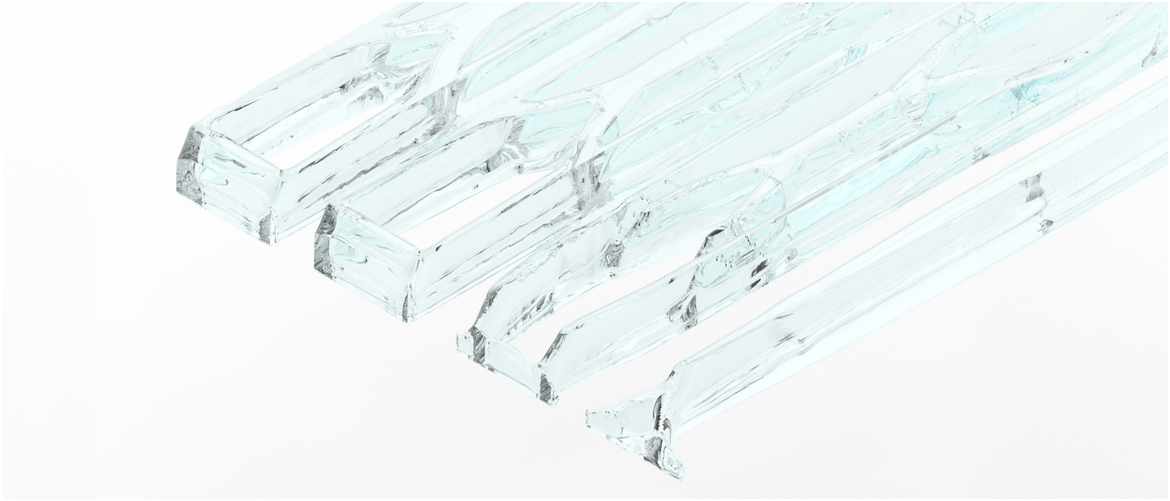
Similarly, in this domain, the 20% and 30% volumes fail to comply with the minimum objectives and/or tensile/ compression forces. Therefore, the optimum volume for the domain lies within the range of greater than 30% to 40%.

This variant, along with the slightly larger variant consisting of three pieces instead of four, is considered for the final geometry of the bridge.









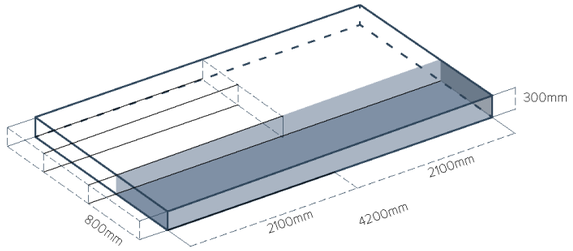
8.3 Final geometry 800

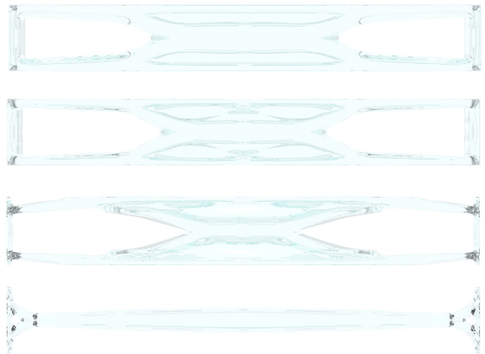
The final variant utilizes one-third of the domain, with approximate measurements of 800 by 4200. For this domain, half the length is utilized and mirrored for algorithmic calculations.

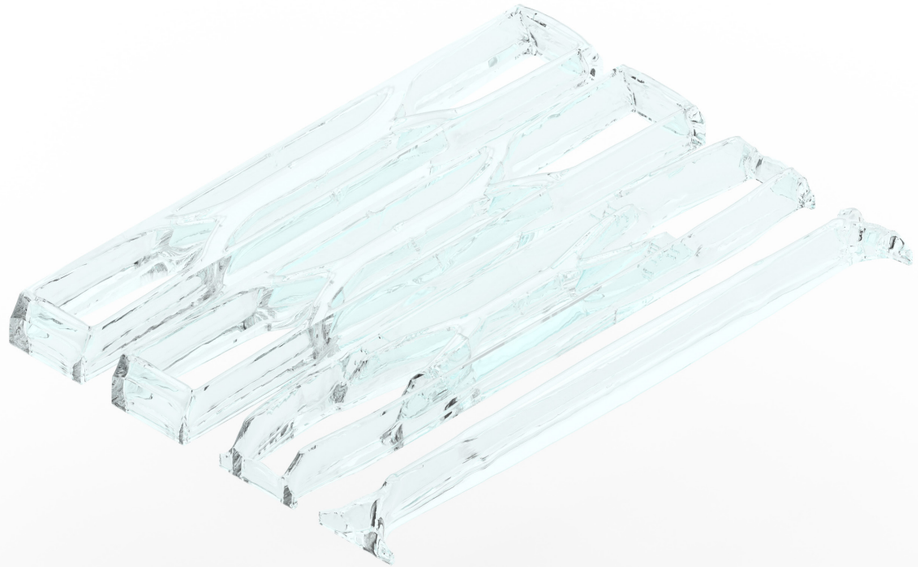
The characteristics of this larger version are also employed by Anna-Maria and Eva, as it presents several opportunities and large panels that are easily divisible with a width-to-length ratio of approximately 5.25.

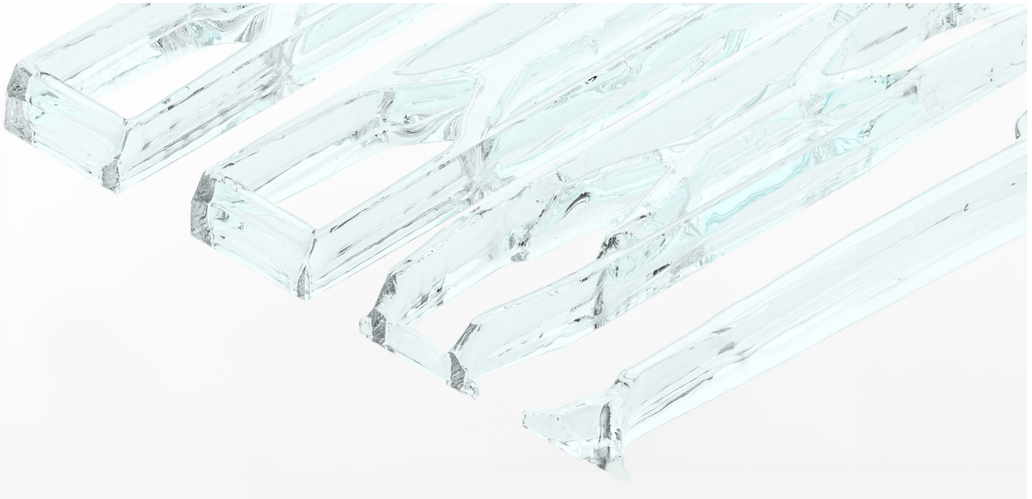
The characteristics of this larger version resemble those of the smaller version measuring 600 in width. It can be assumed that the width difference between these elements is less restrictive to the overall shape than the width ratio of 480mm.
Therefore, it is concluded that the final shape, based on the objective and tensile forces exceeding the maximum amount, will be either the 600 or 800 width version with a higher volume of 30%. Due to the increase in both these amounts due to the use of the overhang filter. The islanding filter is only of slight influence on the objective and thereby less relevant.

Consequently, the overall shape geometry of the bridge will roughly resemble the 30%-40% volume version, either with a width of 600 where four elements will be placed side by side or a width of 800 where three elements will be placed.
The next subchapter will conclude the final geometry and processing.









8.4 Final geometry

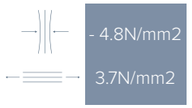
The final shape is determined to be one-quarter of the total width of the domain of case study 1, with a volume reduction of 0.63%. So there is 0.37% of volume left within this domain.

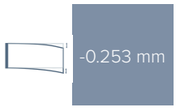
These values are derived from a minimum volume reduction of 30%, utilizing objectives and maximum tensile stress criteria. Multiple checks have been conducted to ensure the minimum volume needed to be added in combination with the overhang and islanding algorithms to comply with both the objective and maximum tensile/ compression forces.

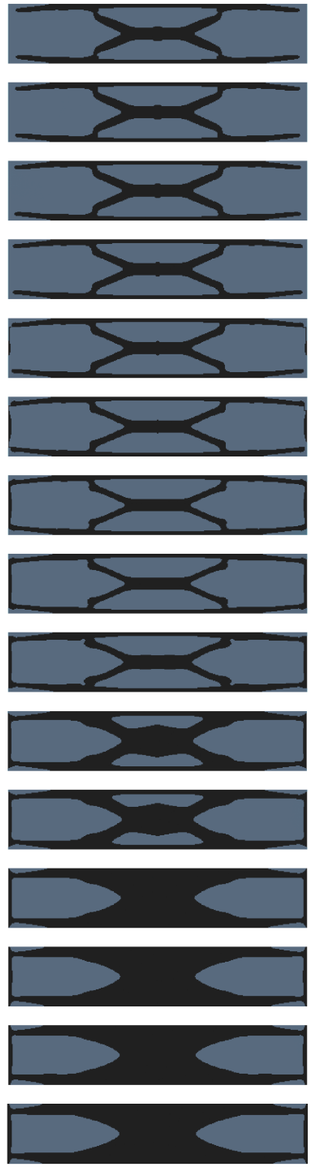
Furthermore, the shape model representing one-third of the total score similarly, as the total number of parts has been multiplied to create an averaged hole for the bridge. This choice was made based on the smaller diameter of glass for each of the elements, contributing to a shorter annealing time. Annealing is considered less important than with casting since, during printing, the glass is already held at a temperature of 600 degrees, gradually decreasing in temperature within the final printing lines.

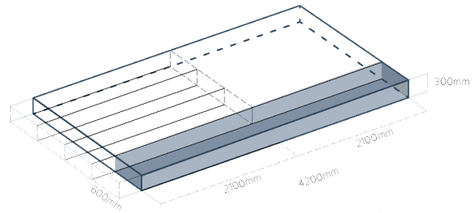
The final weight of the elements...
Dimensions ....
max diameter.....
forces....
calculations...

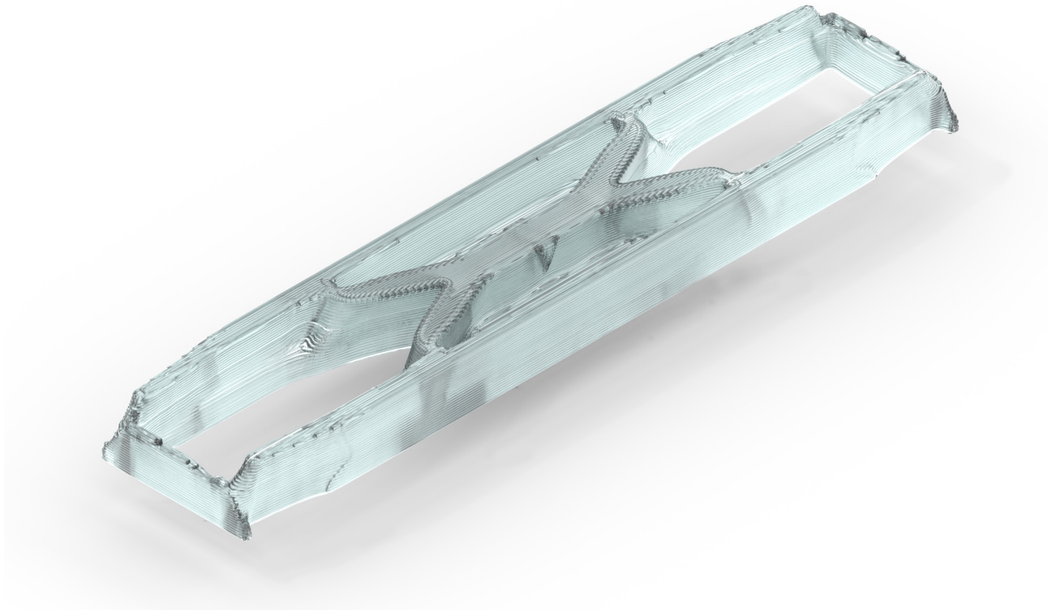




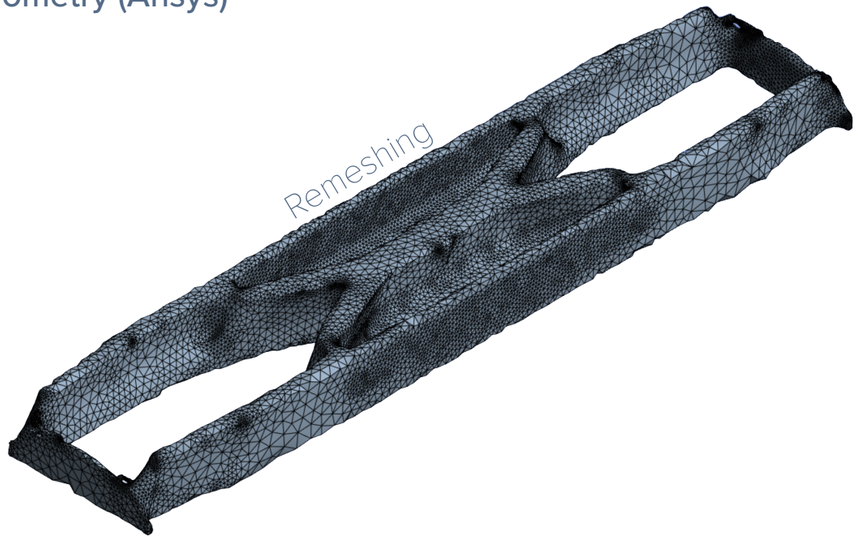




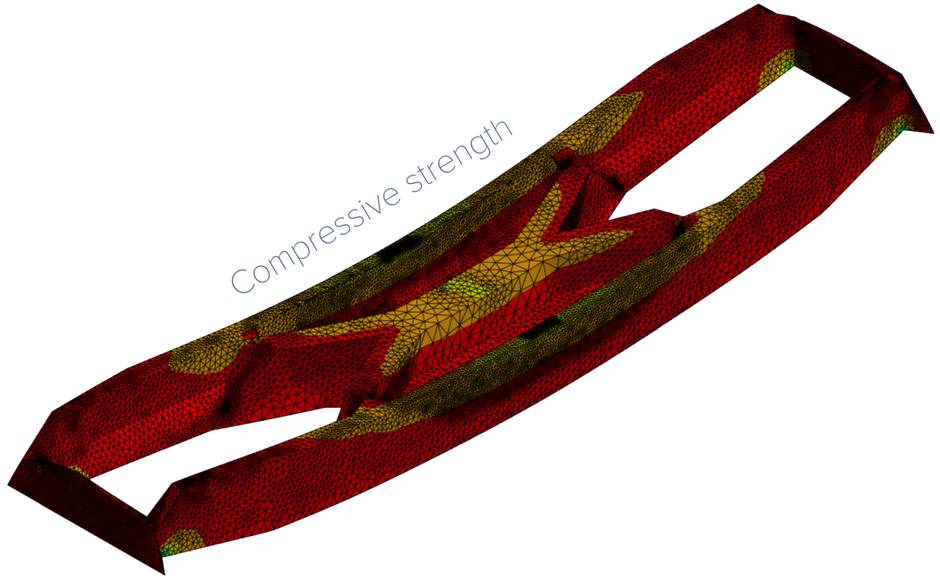




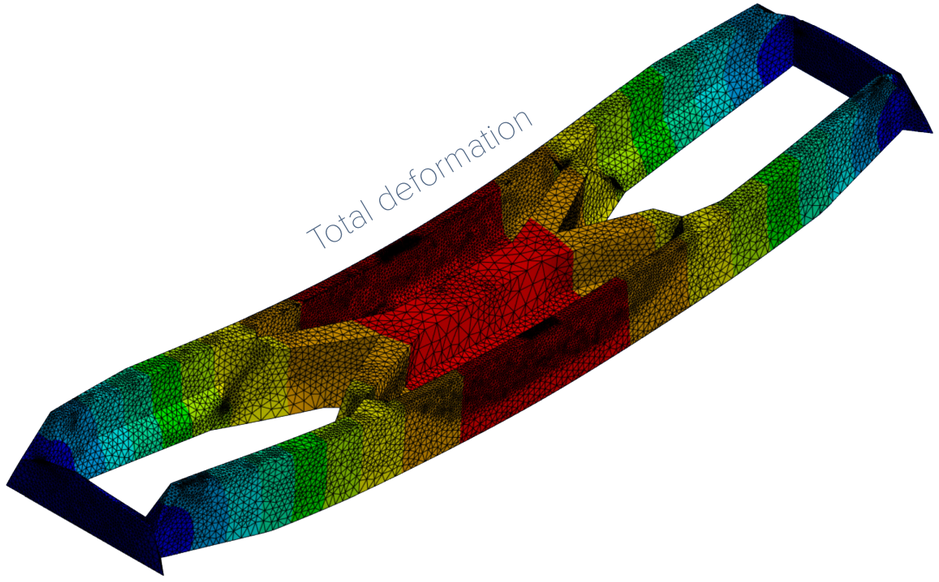
8.4 Final geometry (Ansys)



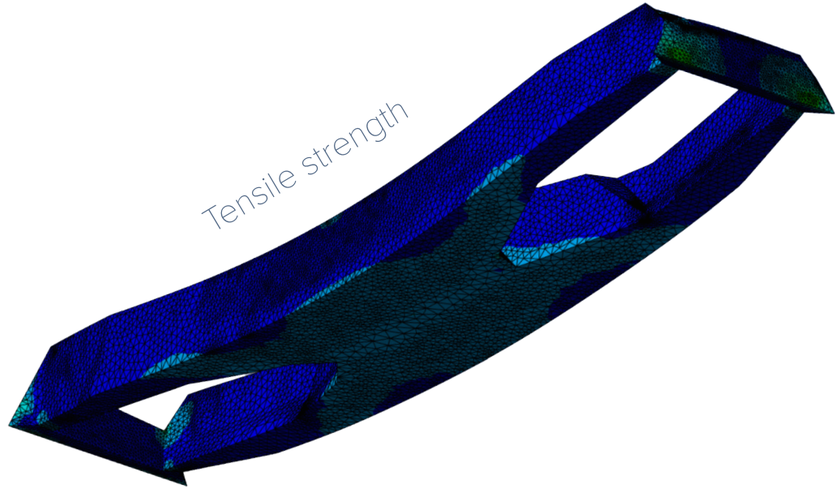
Remeshing



Compressive strength

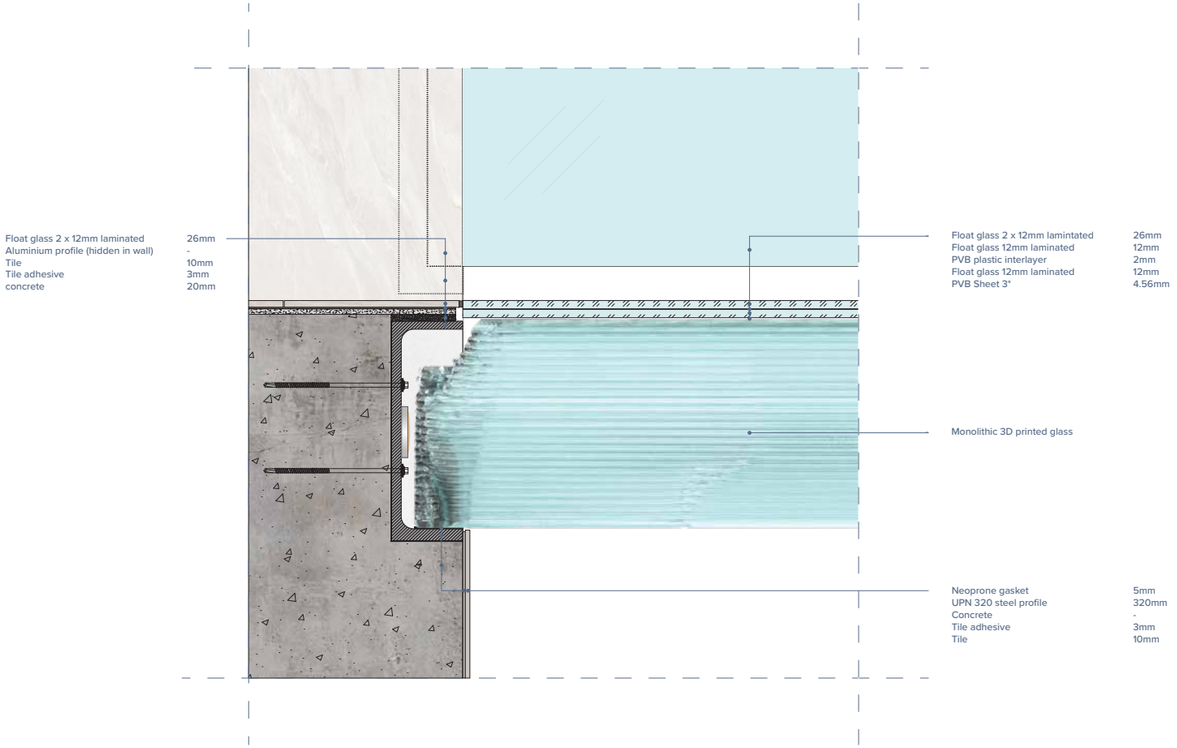


Total deformation

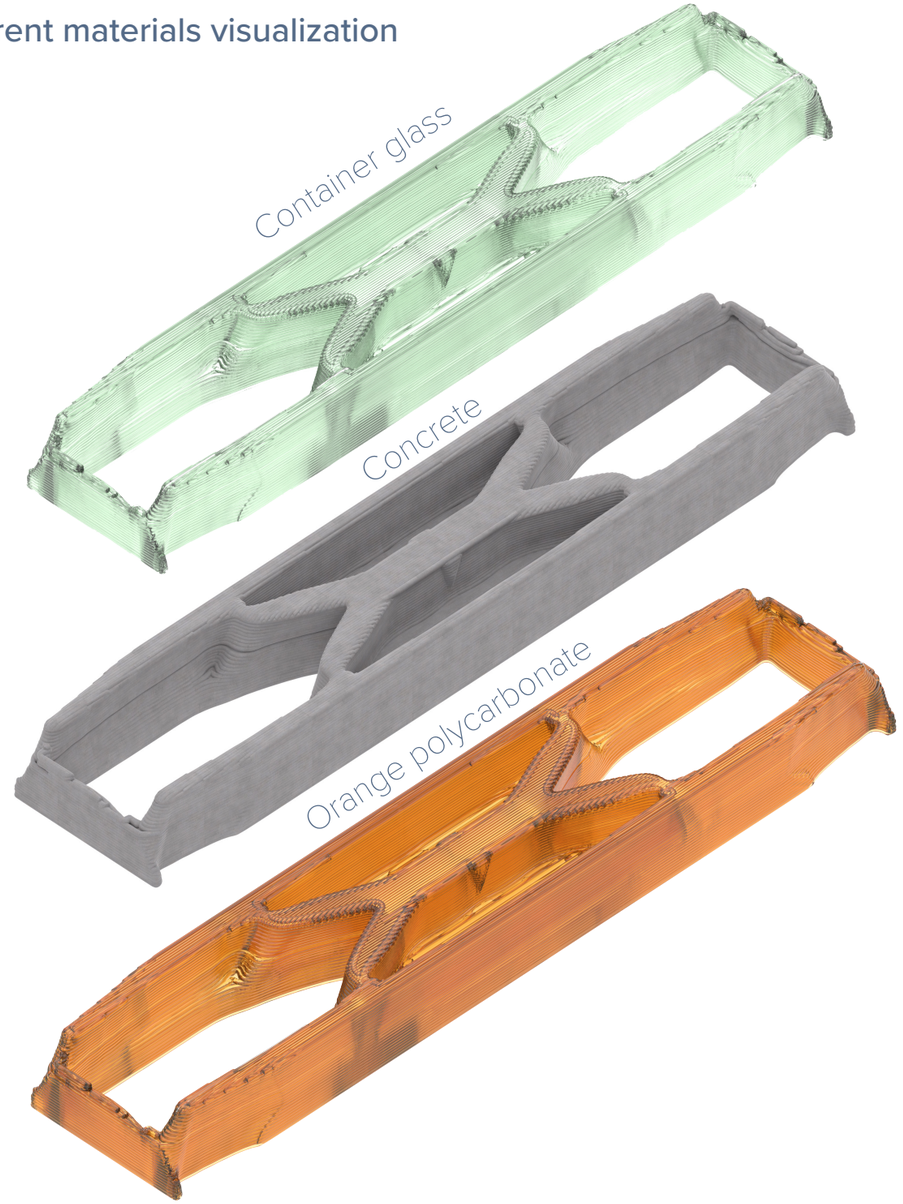


Tensile strength

8.6 Final Detailing



8.7 Different materials visualization

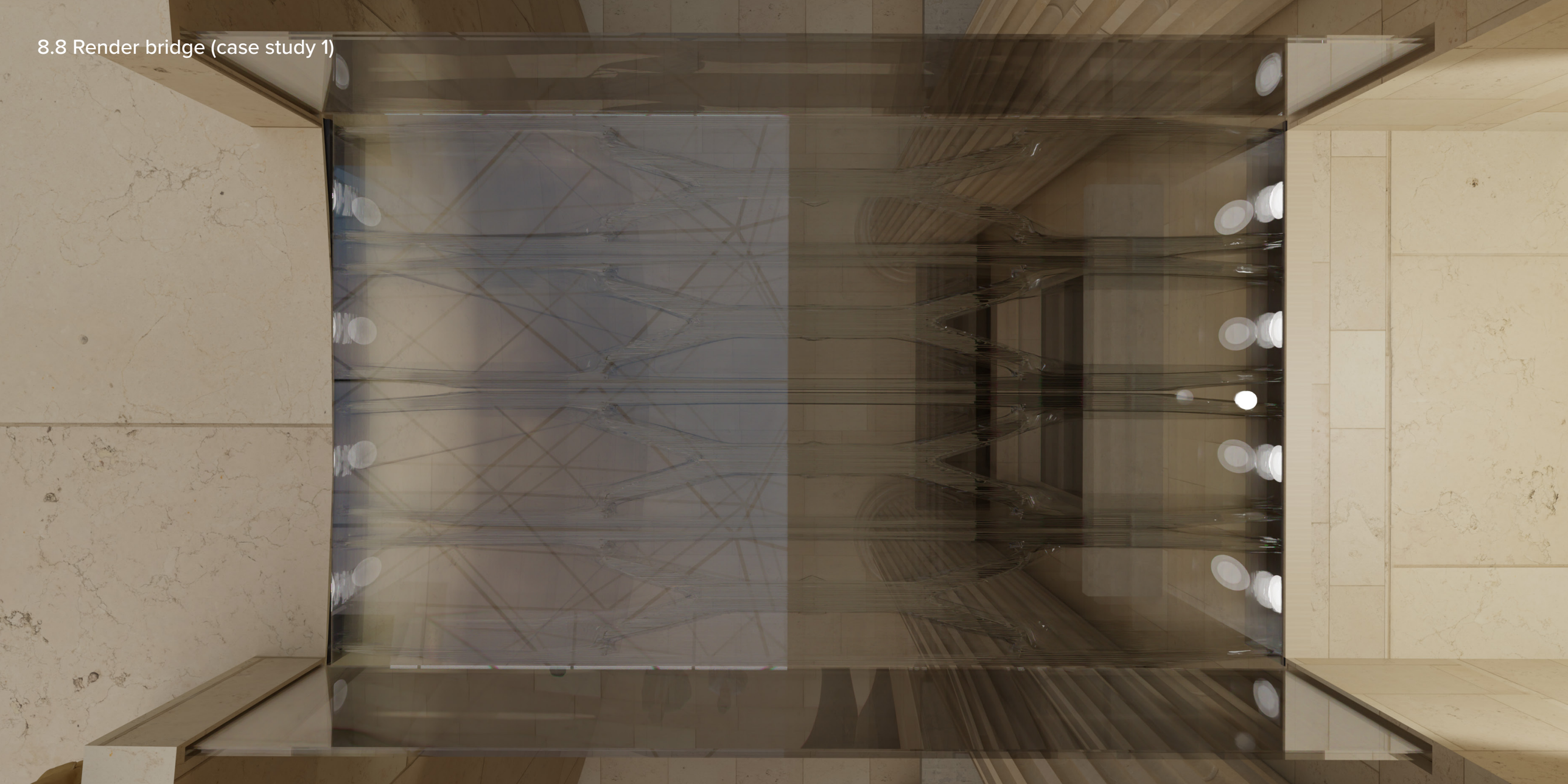


8.8 Render bridge (case study 1)

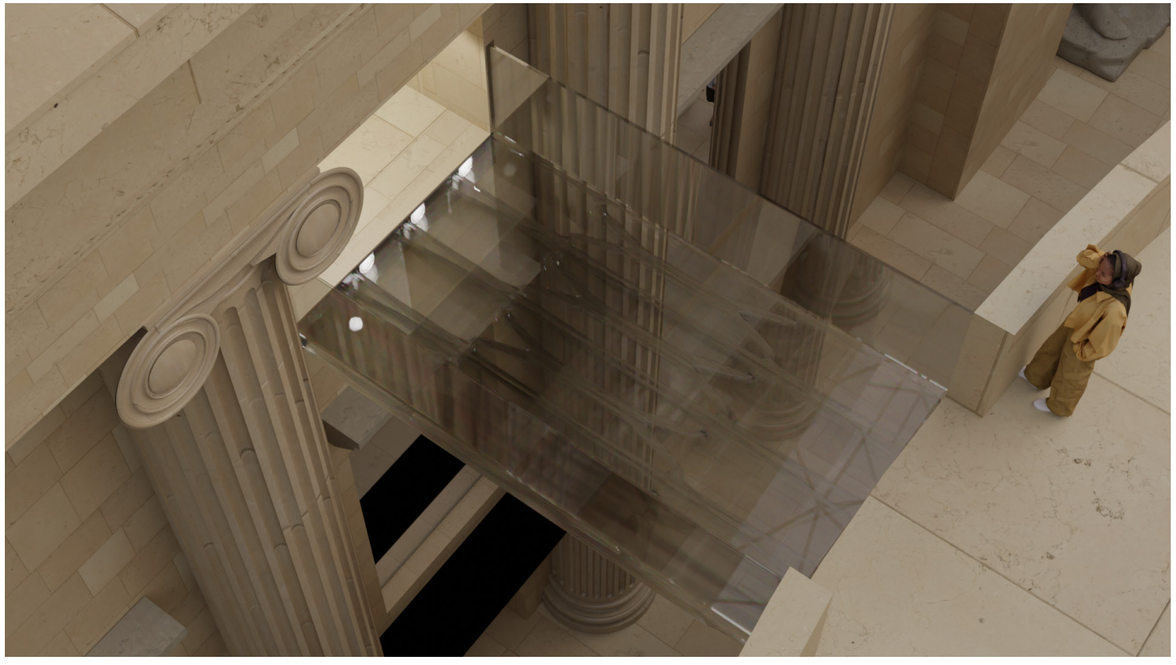


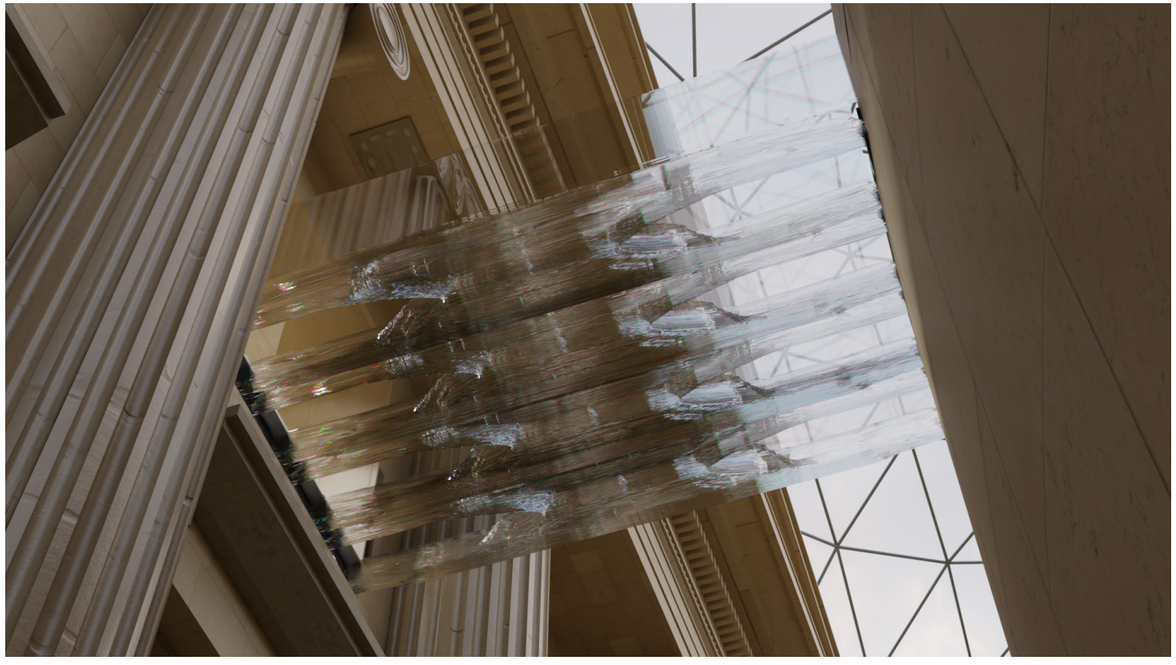
8.8 Render bridge (case study 1)

8.8 Render bridge (case study 1)



8.8 Render bridge (case study 1)

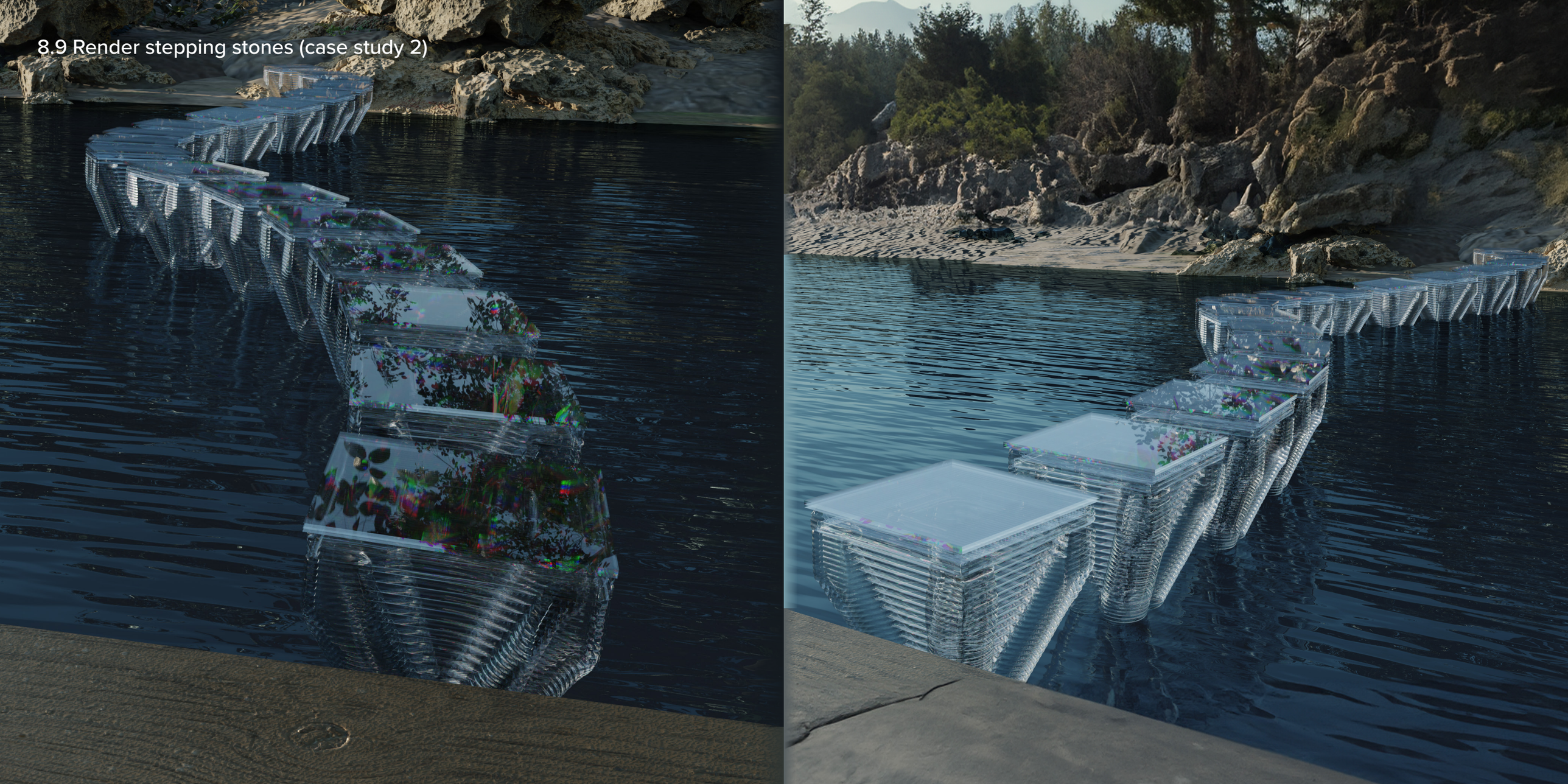




8.8 Render bridge (case study 1)




8.9 Render stepping stones (case study 2)


8.9 Render stepping stones (case study 2)


8.9 Render stepping stones (case study 2)

09

Prototype Production

Progress and developments of the 1:1 prototype and scale model of glass



9.1 Small scale plastic

Before the P3 and for testing new variant small prints out of PET-G were printed. This quick, easy and cheap way of testing ensured that the bigger arm could also potentially created these items in a bigger scale without directly visible problems by the stl file. For P3 this gave already insights in globally what the final outcome could look like. My experience with normal sized FDM printers increased the effiecency and trial of these prints.

This testing continued also after P3, as the use of the comau robot with big plastic printer took way longer than expected. And the working principles for P4 needed to be demonstrated out of the digital enviroment. Unfronately at this time the biggest element could only be printed at one of the anycubic chiron with a maximum bed plate of 400 x 400 mm. Hence this would scale the final shape 10 times.

A final prototype of the bridge was constructed, featuring four adjacent parts made of PET-G and concrete. This showcased a mockup of the final element's appearance. Initially, there was consideration of incorporating LEDs within the scale model to highlight the translucency of the final design and demonstrate the potential impact of lighting.

Unfortunately, due to my experience with standard FDM printers, there are limited pictures available from the process of making prototypes with PLA or PET-G. The final PET-G prototype lacks some transparency and has a slightly greyish tint, which distinctly sets it apart from the glass prototypes. For the final 1:10 scale model, please refer to the end of this chapter.



figure: Final PET-G maquette, light eddited from other picture on this picture, to show the transparent qualities of the final object

9.2 ROBODK

Paul recommended using ROBODK software for creating the print path for the robot. Since I hadn't used this software before, setting it up initially was a bit challenging. The first steps involved implementing both the robot and the extruder in the software environment. The robot was easily downloadable from Comau's website, as was the extruder from Dyze. However, I had to model the connection piece of metal myself.

To integrate all these components into a functional workspace, I needed to establish control points for the robot and the extruder. This ensured that the program understood where the print should begin relative to the end of the nozzle. To achieve this, a relative distance point was created with respect to the robot's end point. In the field of robotics, this point is known as the TCP (Tool Center Point). It's essential for generating a relative path for conversion into G-code for the robot. Once this setup was complete, the prototype could be loaded, and the printing path could be laid out.

The printpath was eventually done using SLIC3R, as the final STL would be perfect for any normal slizing software to be 'slized'. no extra attention was needed to modify own printing path's as these software are specificly designed for this. However as i never used this slicer before, i must say it was not the most 'friendly' slicer i have used. But as this is integrated within ROBODK this was the option used. The other option the plug-in from grasshopper was also tried but eventually abboned by driver issues due to not updated ROBODK software and printpath recognition for mac-os.

The illustration below illustrates that despite precise work, the orientation of the tool center point (TCP) was mistakenly positioned 180 degrees in the wrong direction. As a result, the robot was unable to access all elements within its range. Eventually, these initial hiccups were resolved, and a final g-code was generated, enabling the robot to follow its designated path accurately.

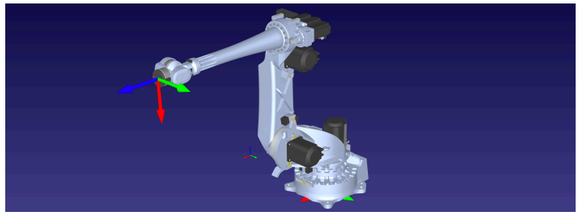


figure: ROBODK first setup robot

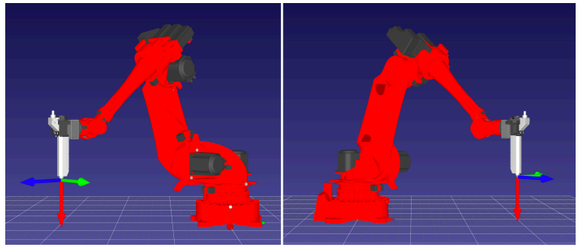


figure: ROBODK final setup Comau robot with correct TCP

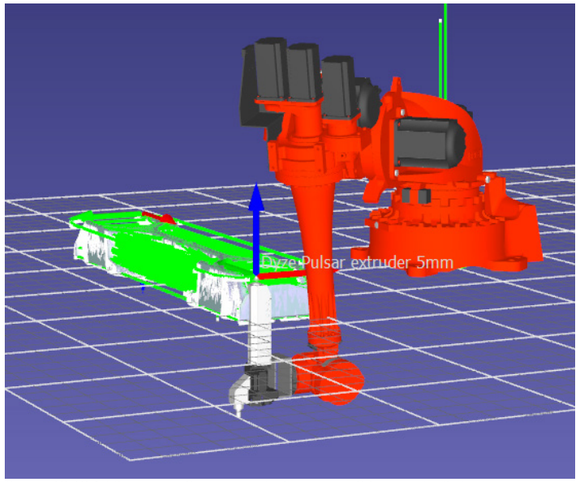
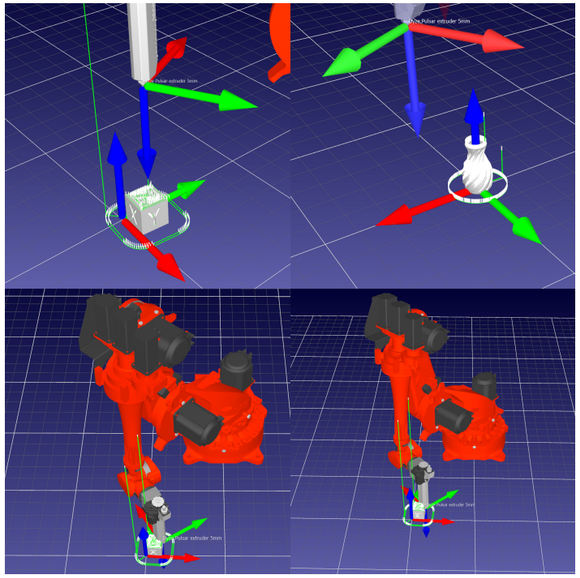


figure: ROBODK final setup Comau robot with wrong TCP and path (first attempt)

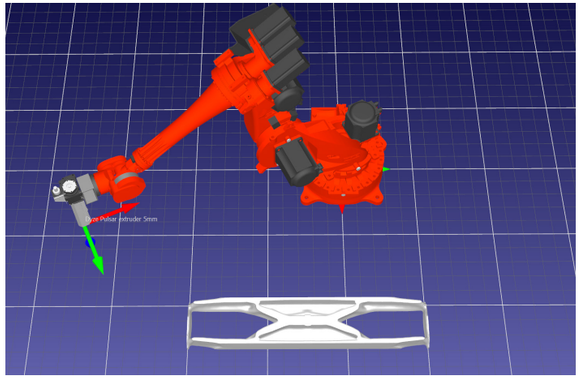
9.2 ROBODK

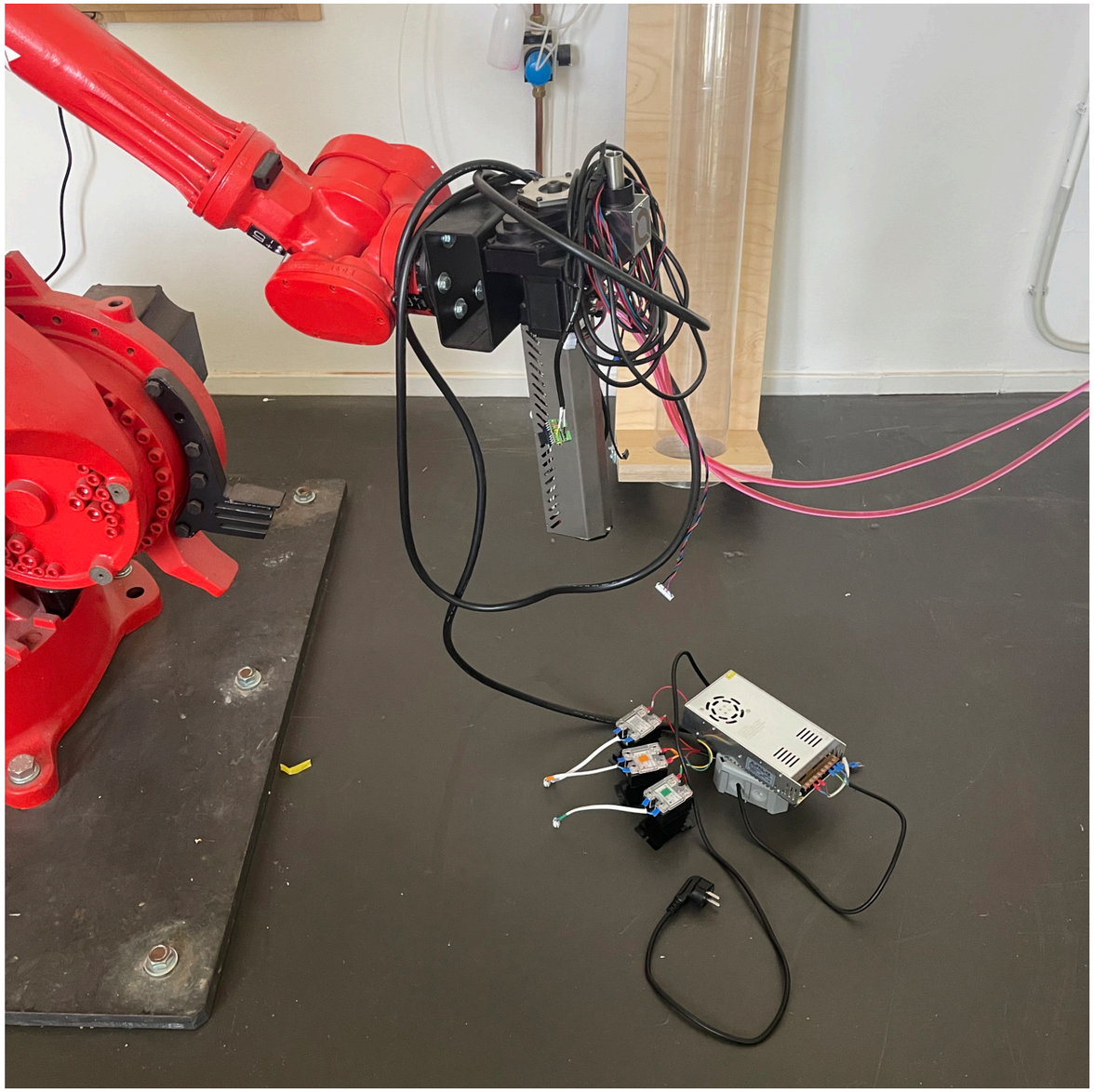
Two test prints were evaluated before proceeding with the final optimization print. Unfortunately, the Dyze extruder wasn't functioning properly, preventing us from working on the robot. This was especially disappointing since everything on my end was set up and ready to go. However, the lab technician hadn't fixed the Dyze extruder. Despite repeatedly mentioning since early January that I wanted to use this robot, it's unfortunate that it won't be used for my master's thesis.

Using the robot would have been an impressive way to demonstrate the algorithm's capabilities and produce a scale model significantly larger than what a standard 3D printer can achieve. However, the control over this aspect was out of my hands. I even attempted to set up the robot without the extruder, but the lab technician didn't allow it, which is unfortunate.





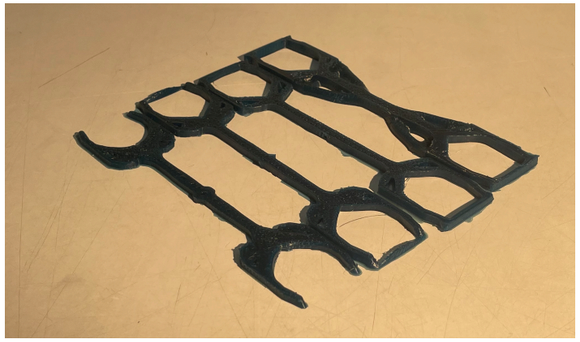




9.3 Glass prototype

Two primary approaches were attempted to develop a scale model for the casting process.

The first method, which involved using wax-based PLA, resulted in several failed prints during initial attempts (detailed in the appendix). The wax component exhibited a high susceptibility to ‘warping,’ a phenomenon where certain materials deform or curl upwards at the edges during the printing process. This deformation arises from uneven cooling of the material as it is extruded and deposited layer by layer.





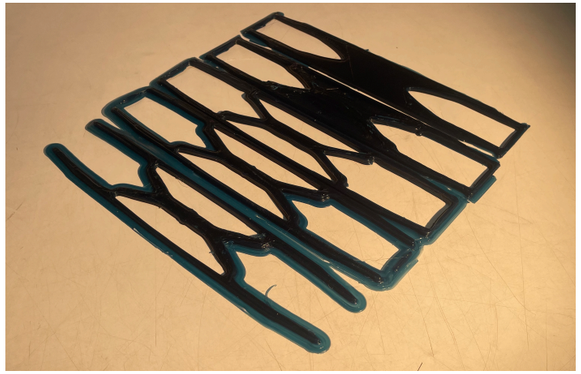
With the bed temperature maintained at 100°C and the nozzle at a stable 175°C, the temperature disparity between the existing and newly deposited layers caused the final shape to curve upwards, leading to unsuccessful prints.

After a few attempts, I experimented with layering and stacking the prints. This method was successful when the individual layers were printed slowly with over-extrusion set at 125%. The final shape was then assembled and ‘glued’ together using candle wax.

I tested this technique with both small and larger versions. However, this method was ultimately not adopted because using a silicone mold with a different wax, which can be steamed out instead of burned, produced better results. Therefore, this remained merely an experiment.

Additionally, the silicone mold method offered greater precision, leading to more consistent and reliable casting outcomes.

For a better overview of this approach, the reader is referred to the appendix.



9.3 Glass prototype

The second approach involved printing an ‘inverse’ mold using PLA and subsequently pouring molten candle wax into the mold.

This wax could later be pressed out. Initially, this step showed promise. However, as the wax cooled within the mold, it contracted, leading to detachment from some edges.

The primary challenge emerged when attempting to release the candle from the original PLA mold. While the detailing of the mold was better than anticipated for larger pieces, the smaller ends of the candle wax proved difficult to remove gently from the PLA mold, resulting in the total piece breaking into several fragments.

An additional test was conducted to evaluate the feasibility of reheating the mold and then releasing the slightly warmed candle wax. However due to getting not the resulting shape and finish that was aimed for. Evantually there was chosen for the wax-based PLA approach.

The final approach, previously executed by other researchers, includes some additional steps detailed on this page. The inclusion of silicon plaster and wax with lower thermal expansion has been deemed beneficial for the project, leading to better outcomes. For the final shape, the reader is referred to the next page, which provides an overview of all steps through pictures.

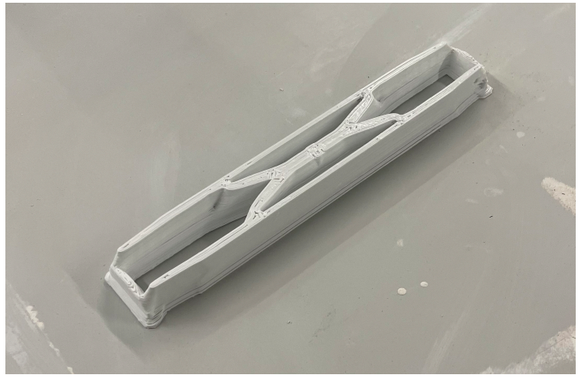
This step was eventually completed between P4 and P5 to gather the final prototype for the graduation presentation.

Fabrication

- 3D printing PLA
- Silicon plaster mold
- Release PLA from silicon plaster
- Wax inside silicon plaster mold
- Release Wax from silicon plaster
- Crystalcast mold around wax
- Steam Wax out of Crystalcast
- Put flowerpots with glass on top of Crystalcast in oven
- Release glass from Crystalcast
- Polish glass

9.3 Glass prototype

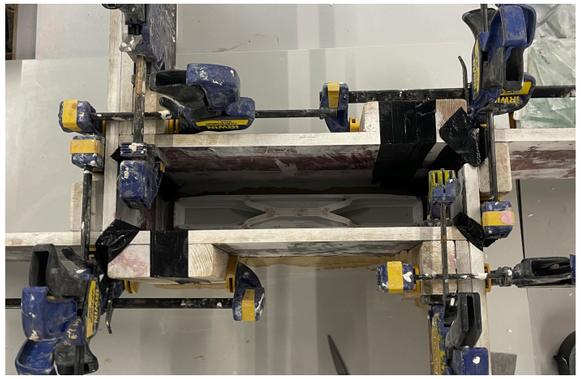
After P4 there was time to made a ‘mock up‘ of the glass bridge in a scale 1:10. This was done by the 10 steps on the previous page. The minimum diameter of the glass needed to be 6mm so to achieve a glass protype of this scale in combination with minimum 6mm, the overall shape was offset, to generate a shape which would fit. Also was the bottom extended with an extrusion facing outwards to provide support material for the molds to follow as otherwise more clay or wax was needed. This would eventually speed up the process. The images show good the process steps that were taken to create a glass prototype.

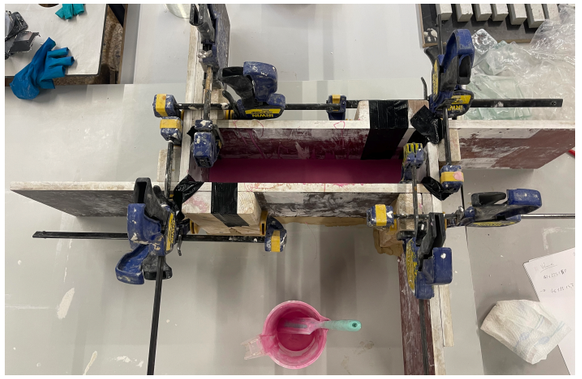


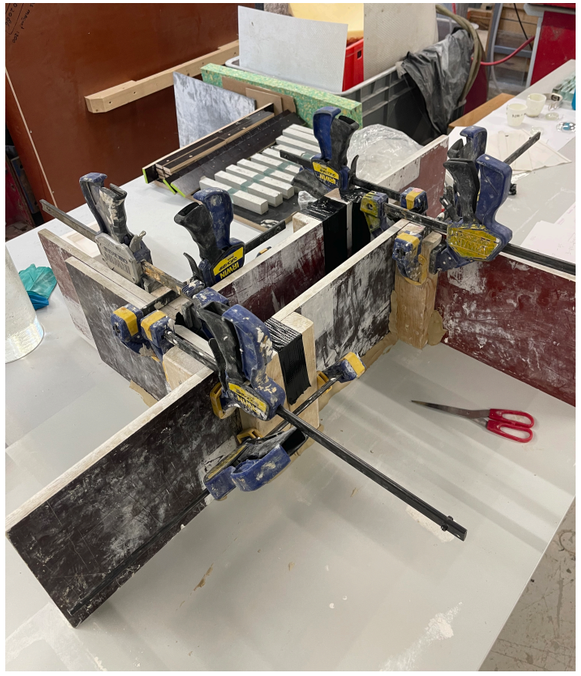


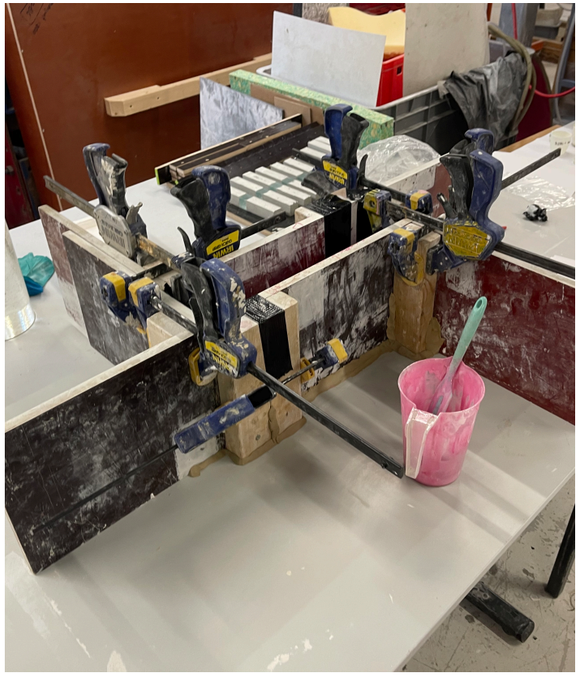




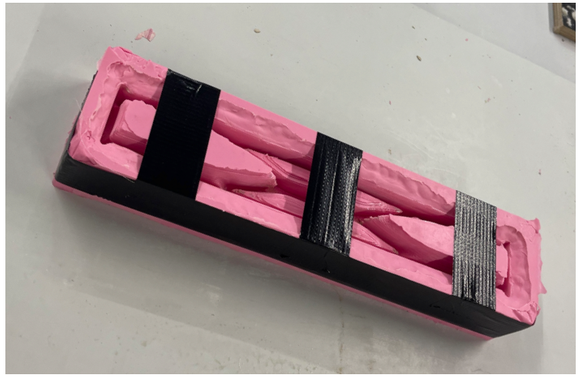




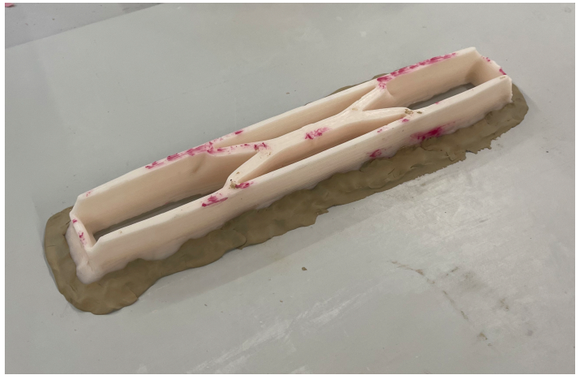
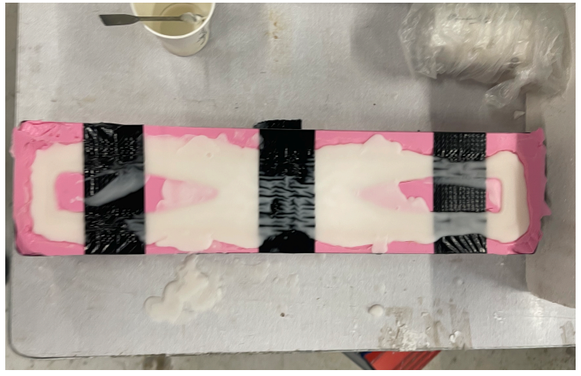


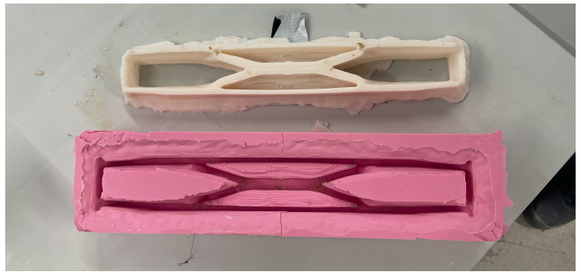


9.3 Glass prototype





9.3 Glass prototype



















9.3 Glass prototype















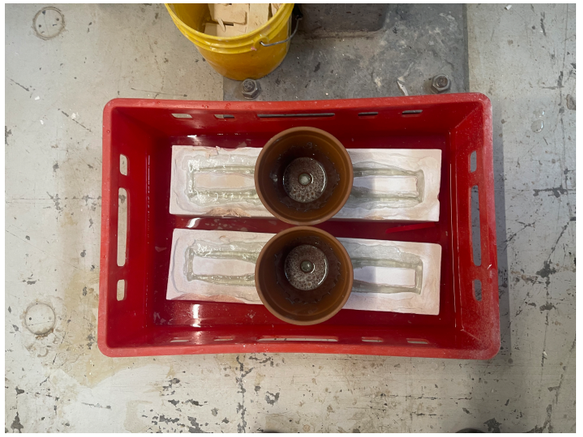




9.3 Glass prototype

As seen on this and the previous pages, the first attempt at creating the glass prototype failed. The bridges supporting the flowerpots collapsed under high temperatures, causing the flowerpots to fuse with the glass as they cooled. This breakage led to the middle part of the glass shattering, and the prototype could only be salvaged at the ends of the bridge where the glass was further from the flowerpot. Even two end pieces couldn't withstand the integration of the flowerpot into the glass.

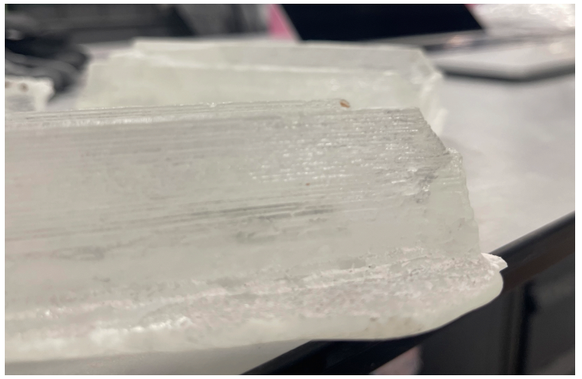
On the following pages, you will see that a new version was attempted. Due to time constraints, as the glass needed to be in the oven for five days, and the oven was also being used by other students, there was an opportunity to share the oven with Veronique, who was making a final glass prototype with contaminated glass. The oven had to be adjusted slightly in temperature. Therefore, toughened glass was chosen instead of the artistic glass used in the first prototype. The final mold was made slightly higher so that all the glass could fit within it without using flowerpots. Once placed, the oven was fired to 1070 degrees Celsius.













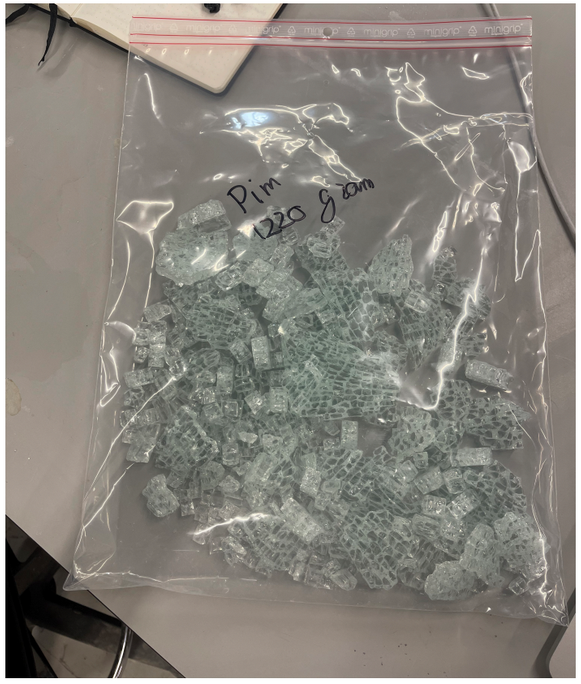


9.3 Glass prototype











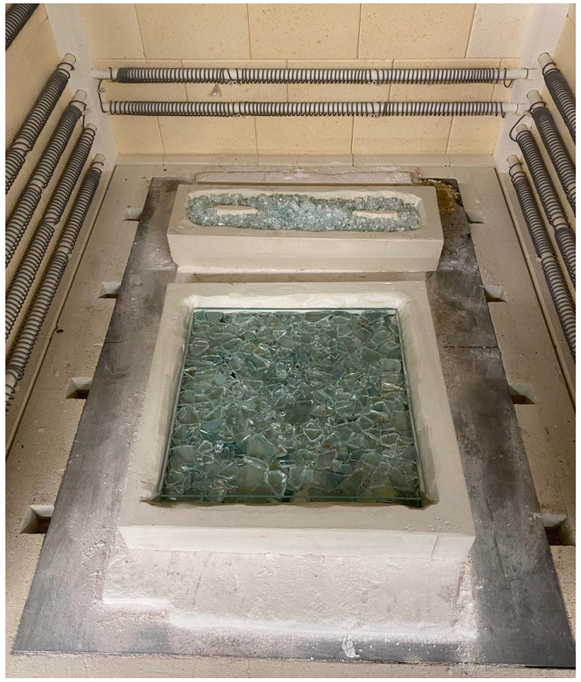
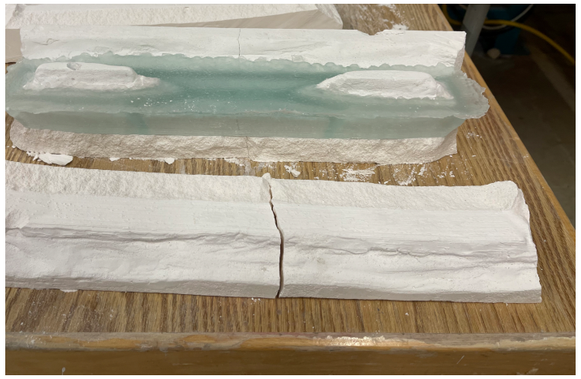
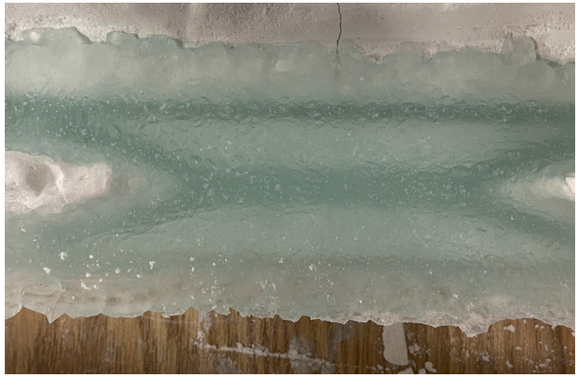






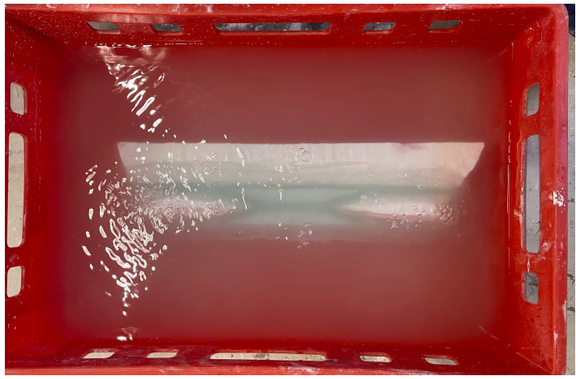
9.3 Glass prototype



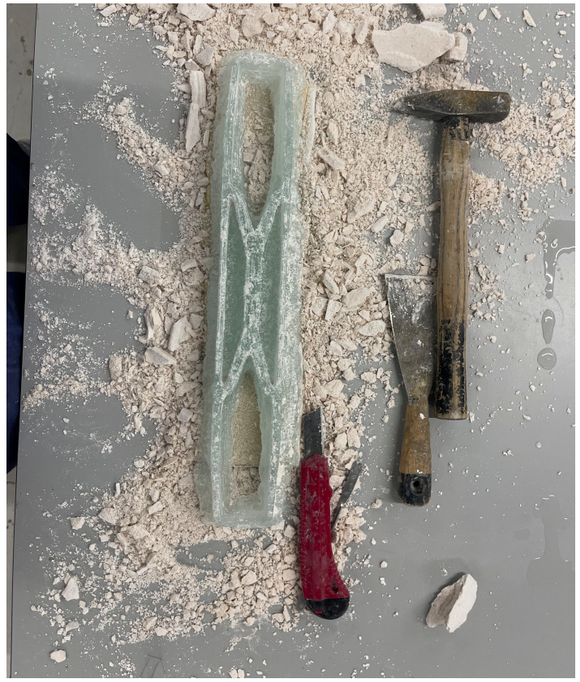





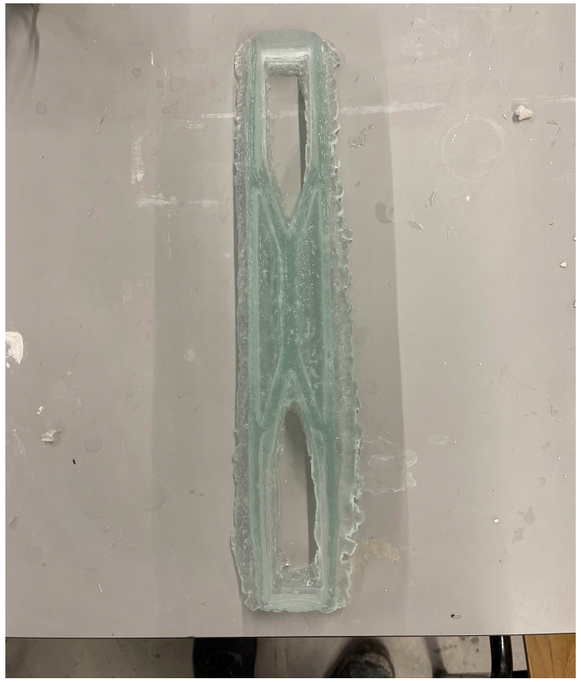


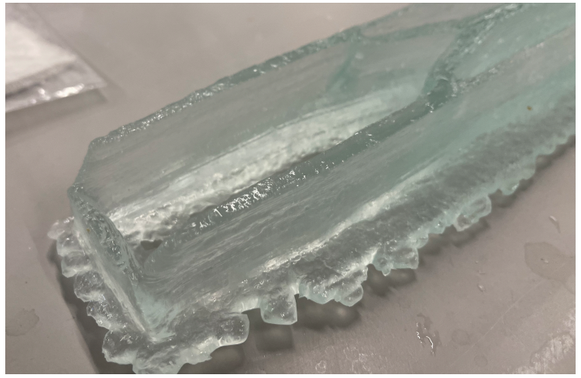
9.3 Glass prototype

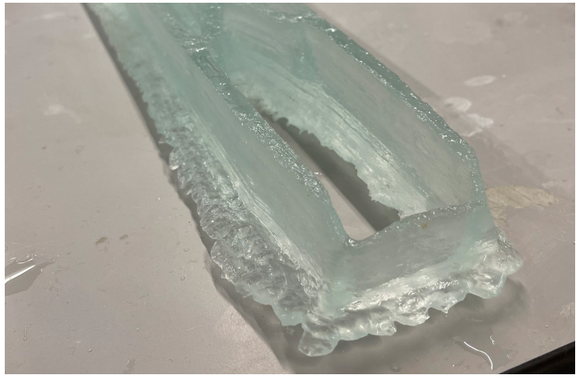


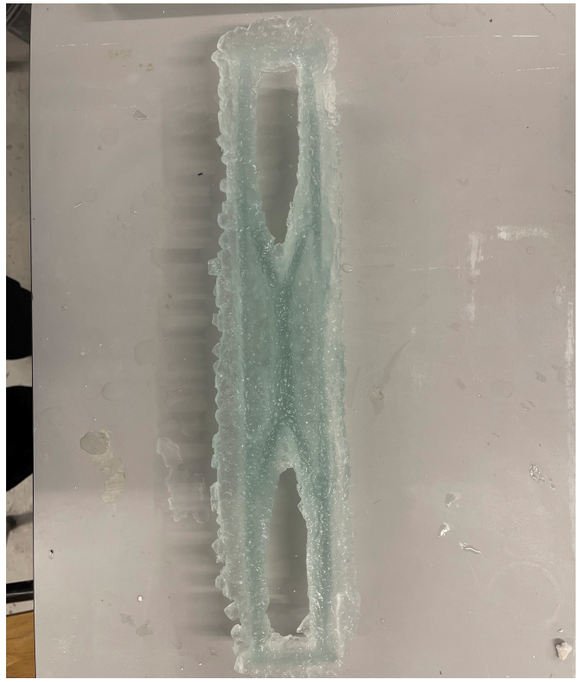


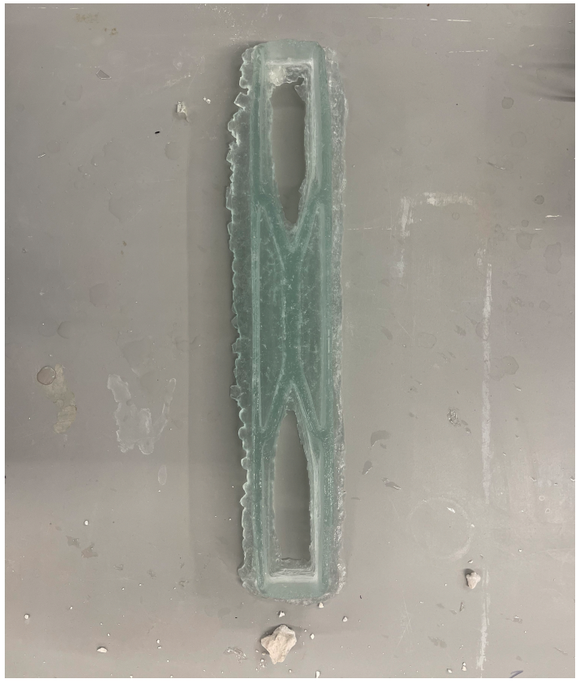




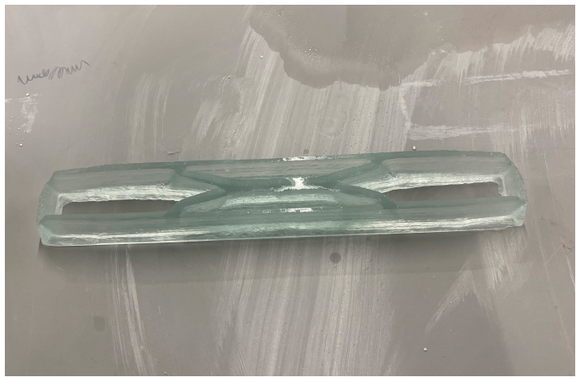


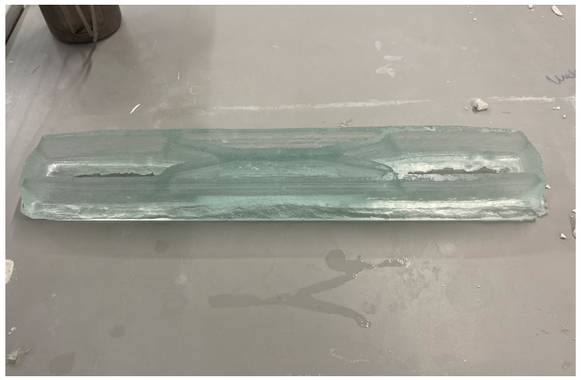


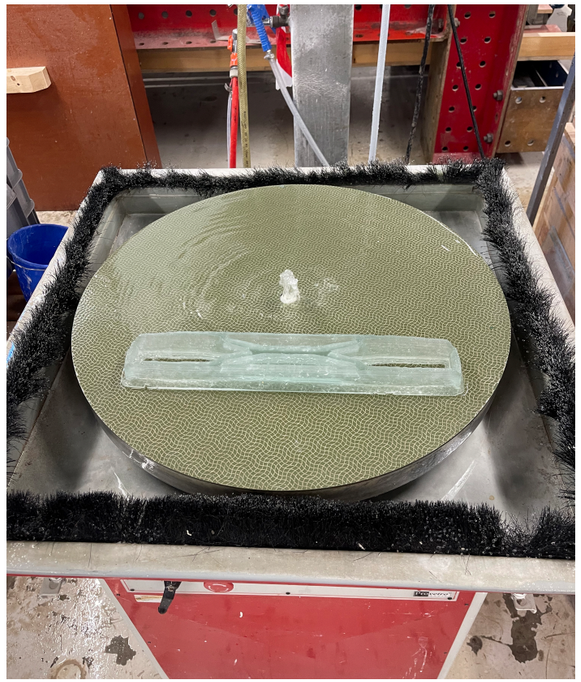


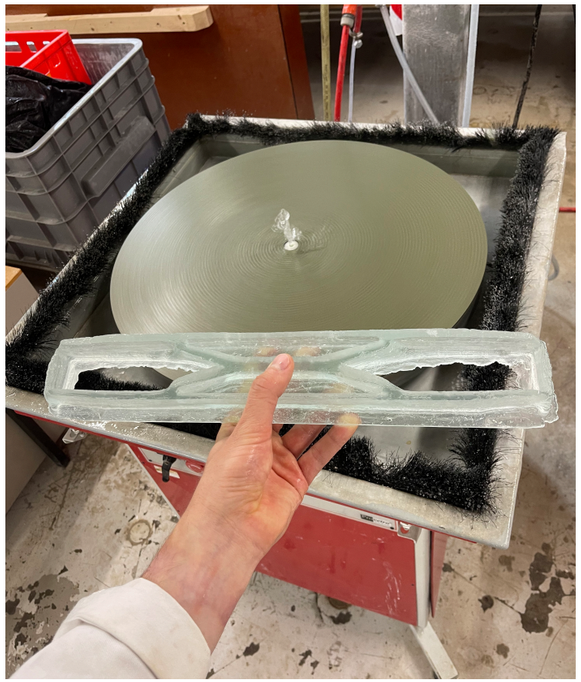


9.3 Glass prototype











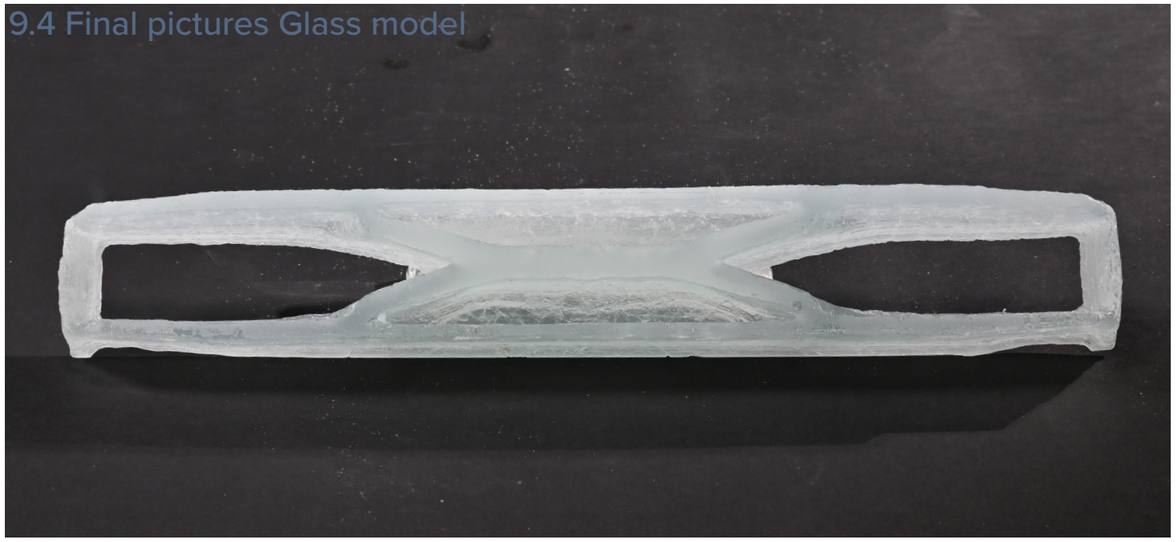
9.4 Final pictures Glass model

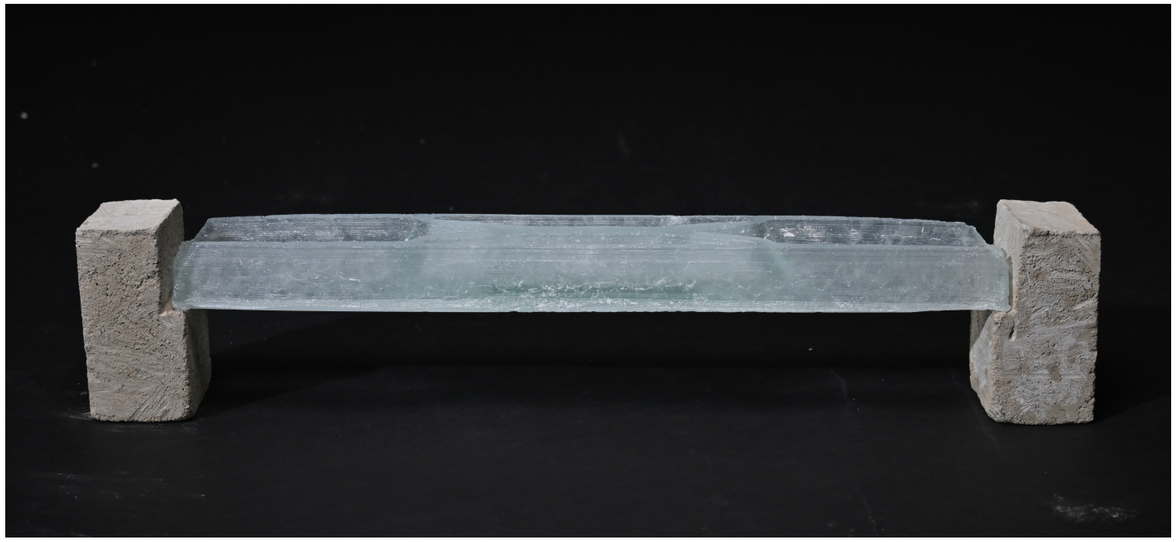




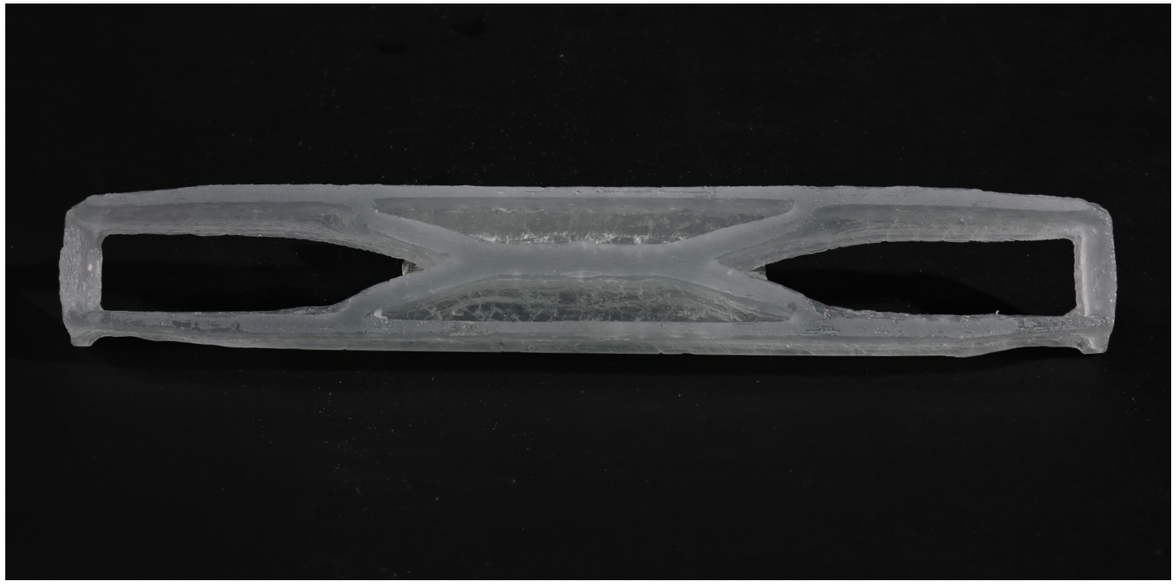


9.4 Final pictures Glass model

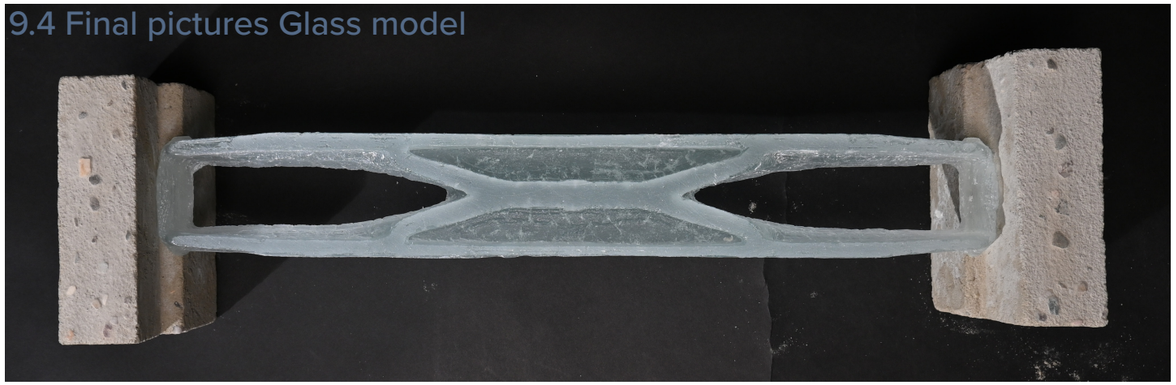


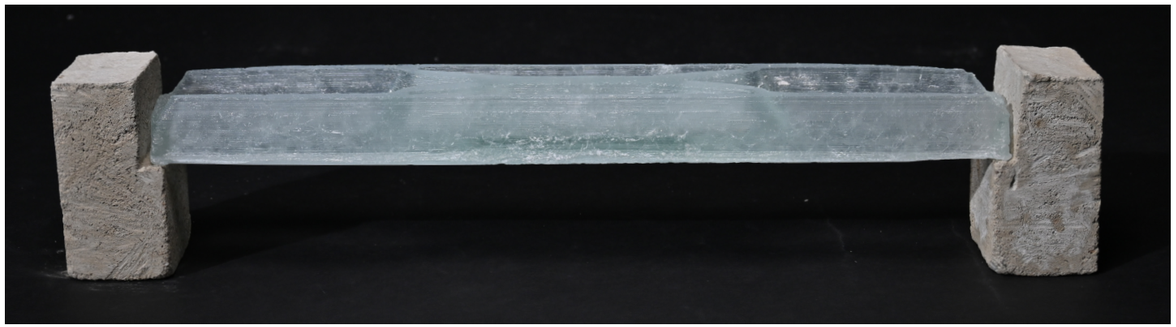


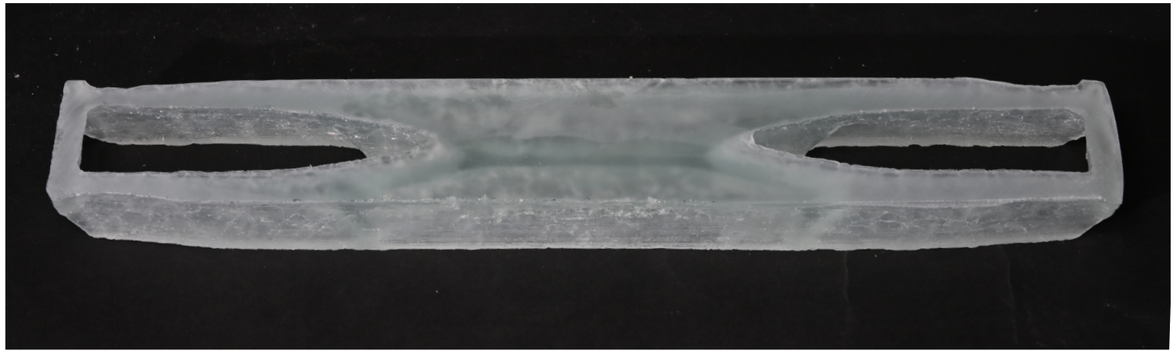




9.4 Final pictures Glass model

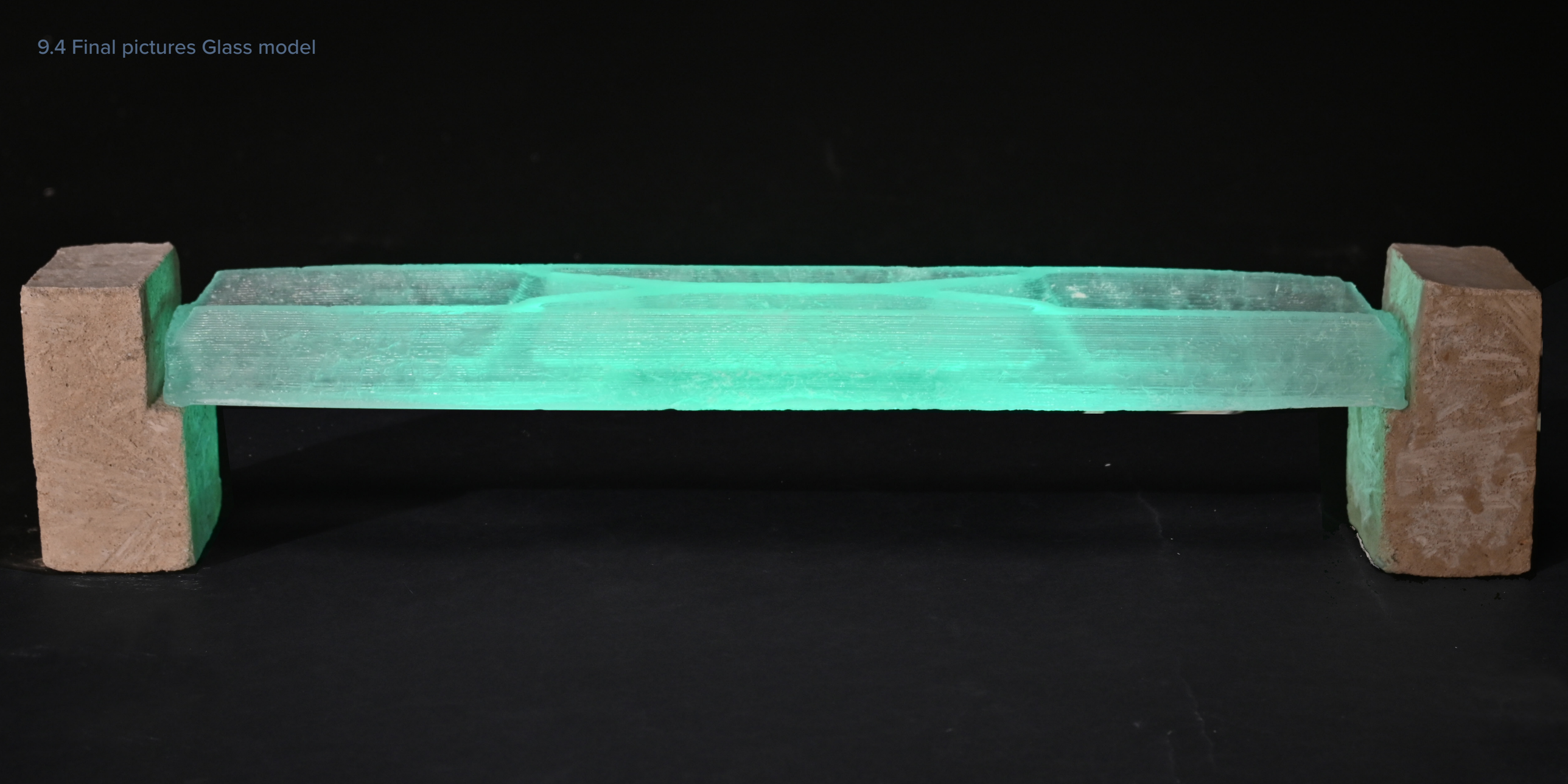




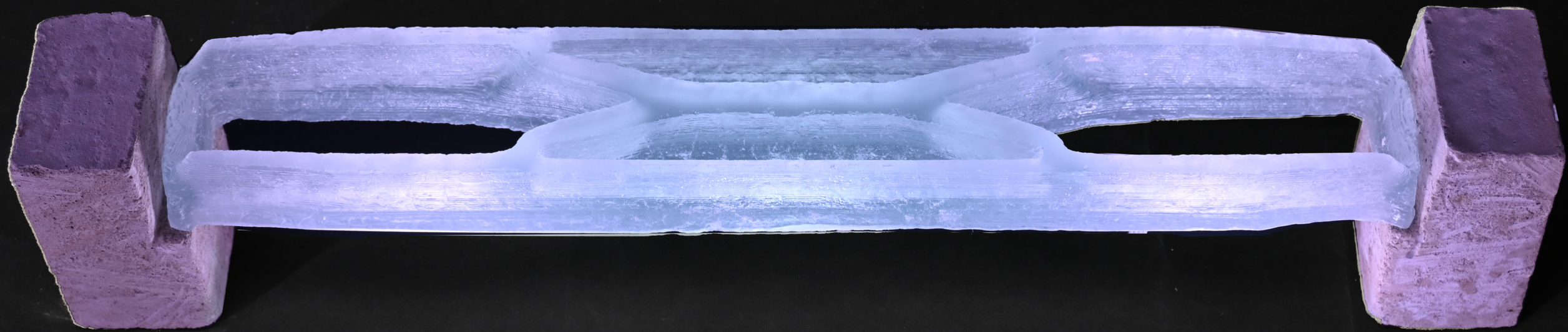




9.4 Final pictures Glass model



9.4 Final pictures Glass model

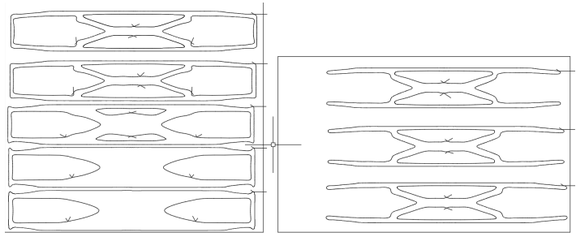


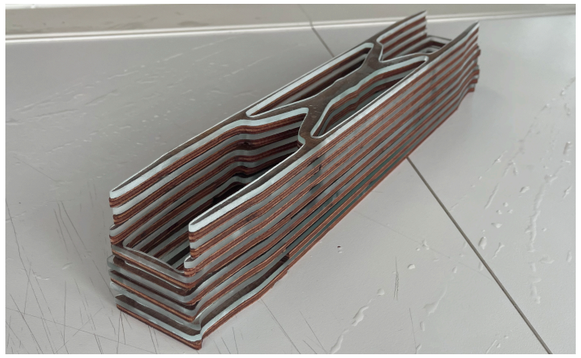
9.5 Waterjet cut glass

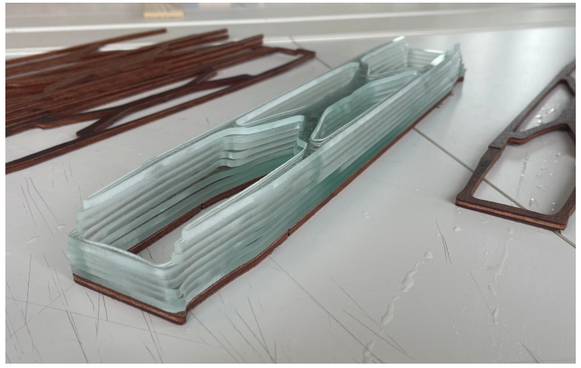
Due to the failure of the first two cast glass prototypes and the time constraints, a waterjet-cut glass prototype was created instead. This was accomplished in the glass lab at the Faculty of Architecture with the assistance of Hans. A sliced file was necessary for the waterjet cutter to accurately cut the float glass panels. Although initially challenging, the final prototype turned out impressive and matched the digital design in AutoCAD used for slicing the waterjet cutter path.

Panels of 4mm thickness were eventually chosen due to limitations with the 1mm panels. Hans highly recommended this change, as the thinner panels were more likely to fail and required significantly more work. Each cut line had to be carefully managed to prevent the 4000 bar pressure from lifting the glass off the bed and destroying the prototype.

Despite these challenges, I am very pleased with the results using the 4mm panels. After cutting, the panels were glued together, testing five different adhesives. Ultimately, UV glue, though slightly past its expiration date, was chosen as it worked perfectly for this project, as seen in the pictures.







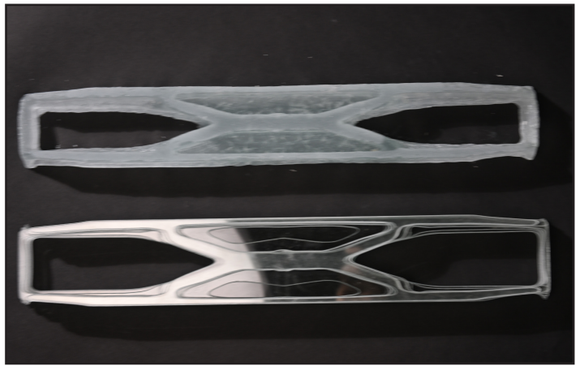




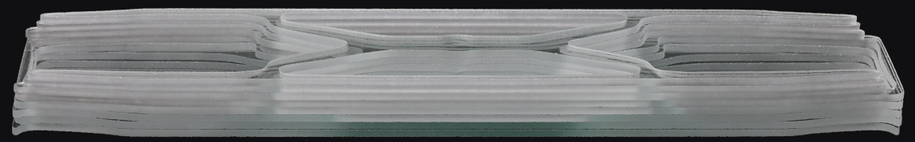


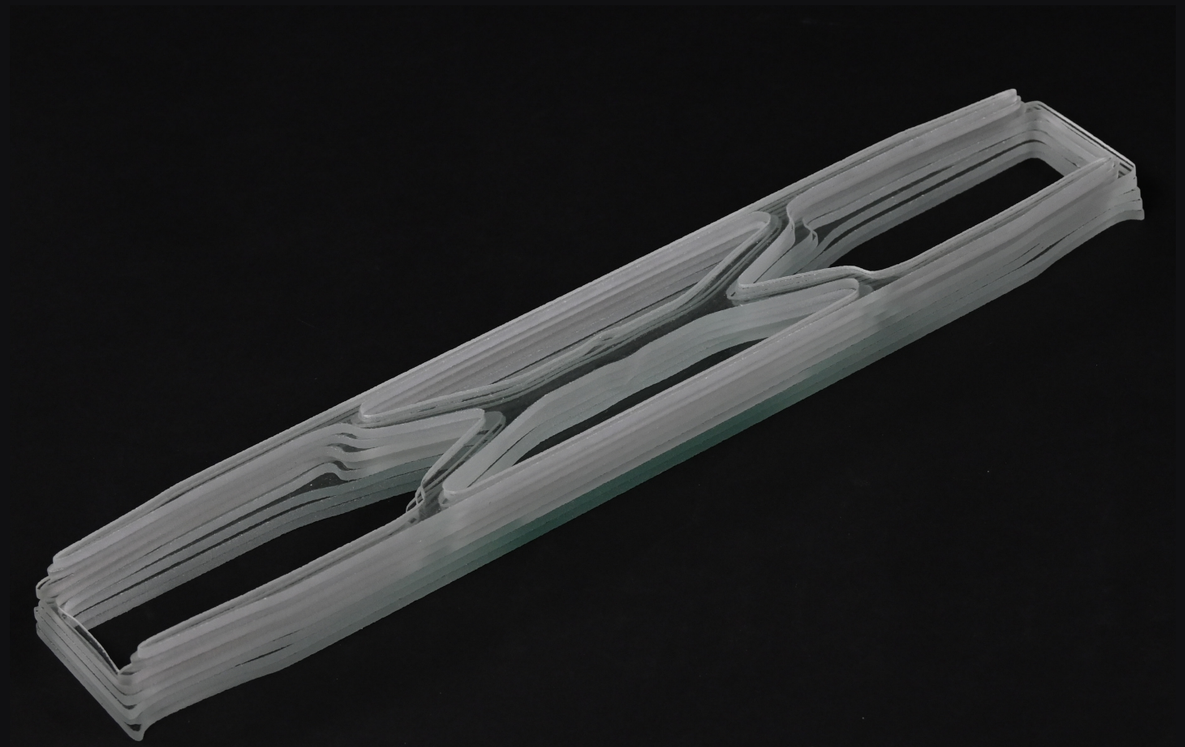


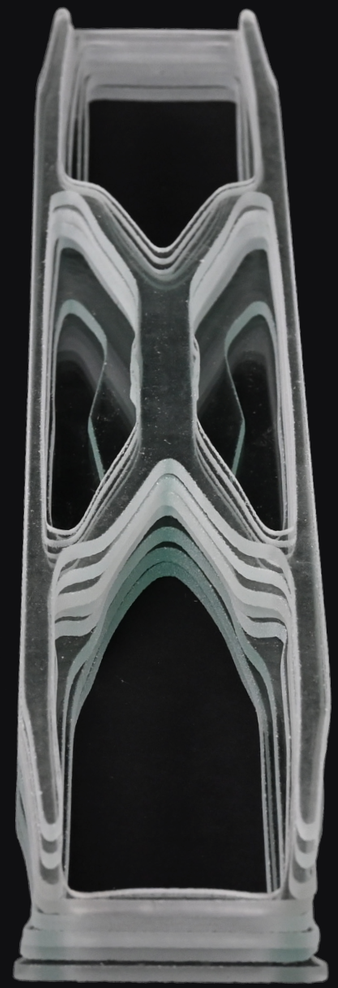




9.6 Final pictures waterjet cut glass







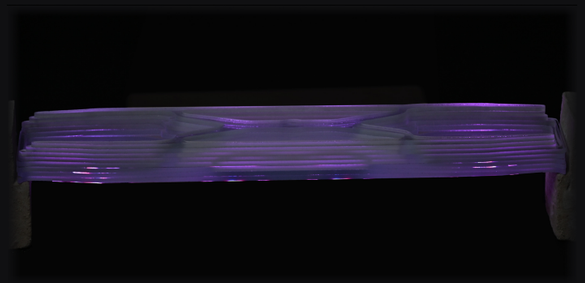


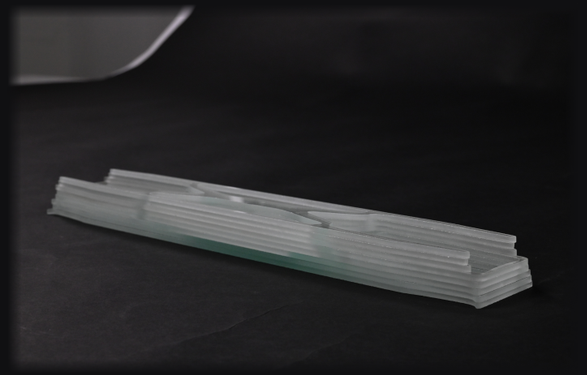
9.6 Final pictures waterjet cut glass

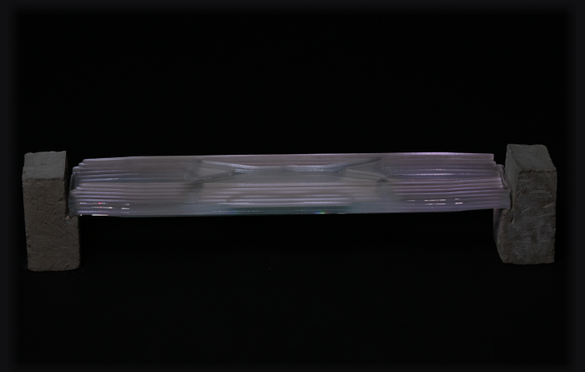




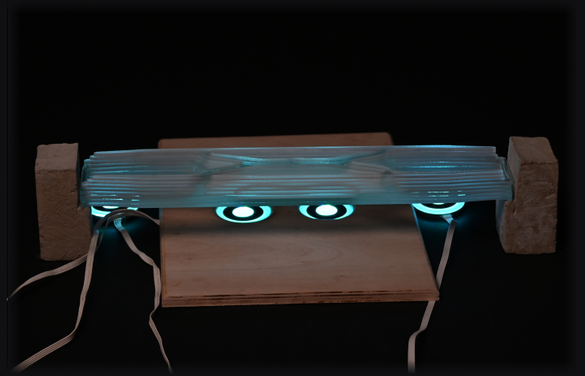
9.6 Final pictures waterjet cut glass



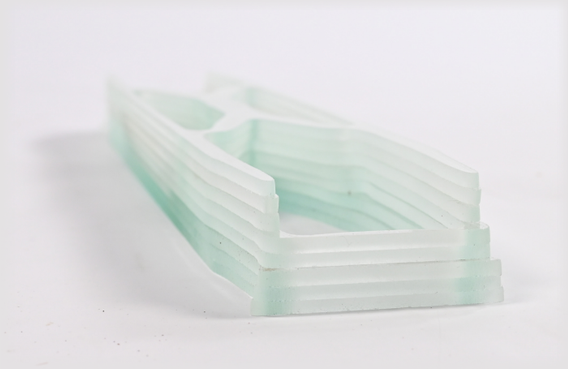


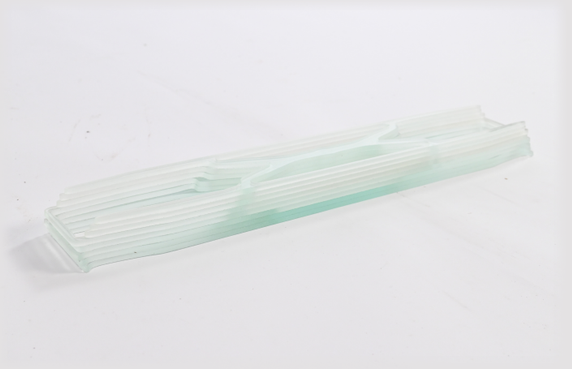


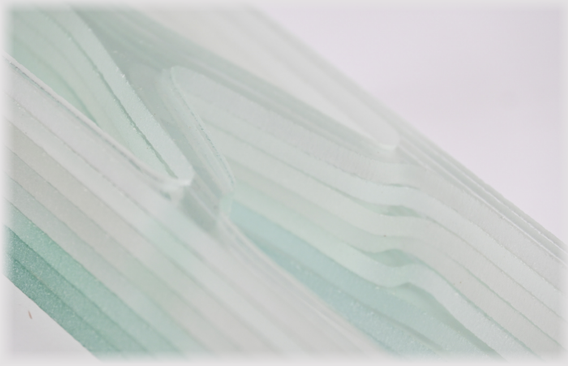


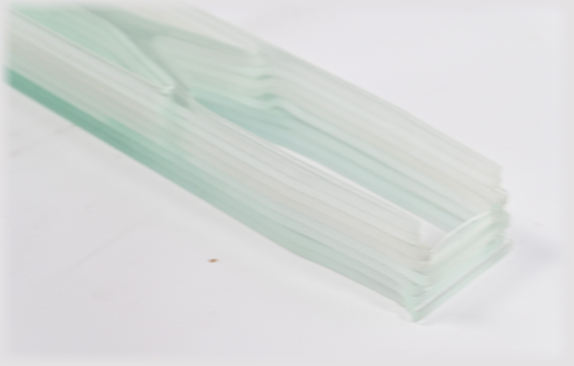




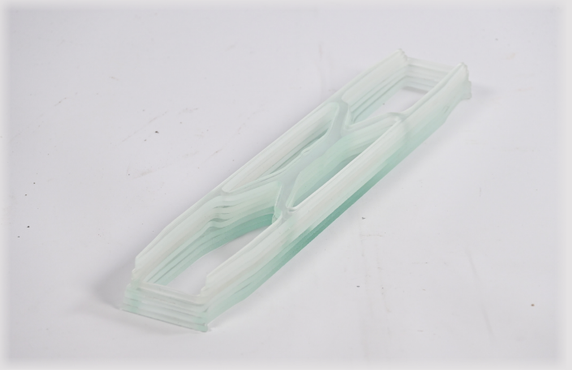




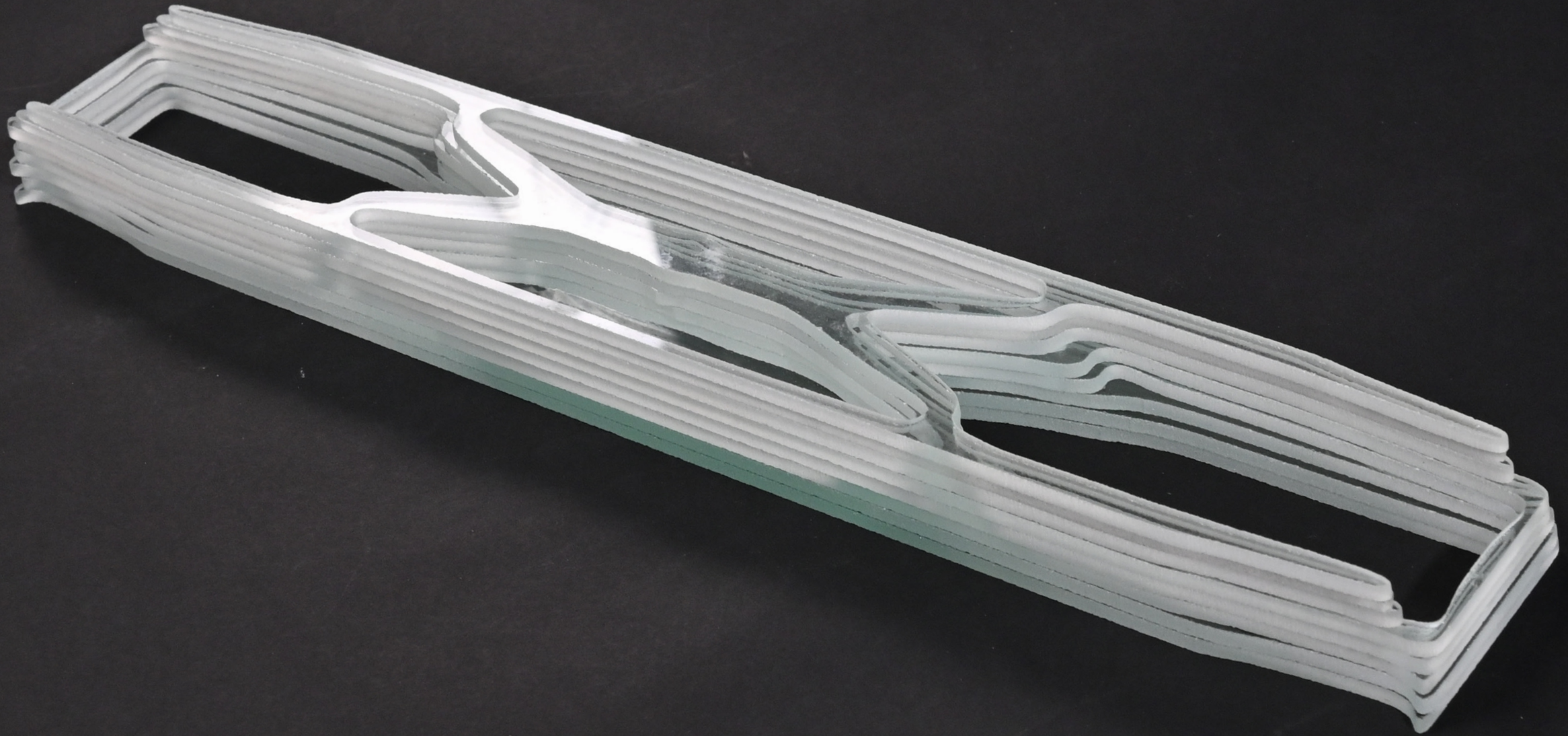








9.6 Final pictures waterjet cut glass



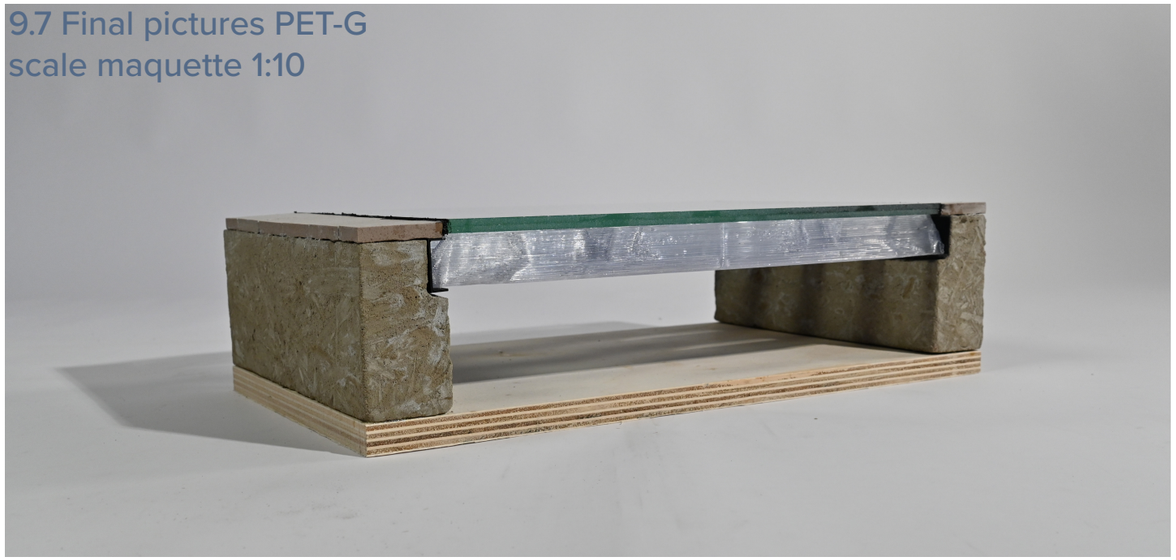
9.7 Final pictures PET-G
scale maquette 1:10

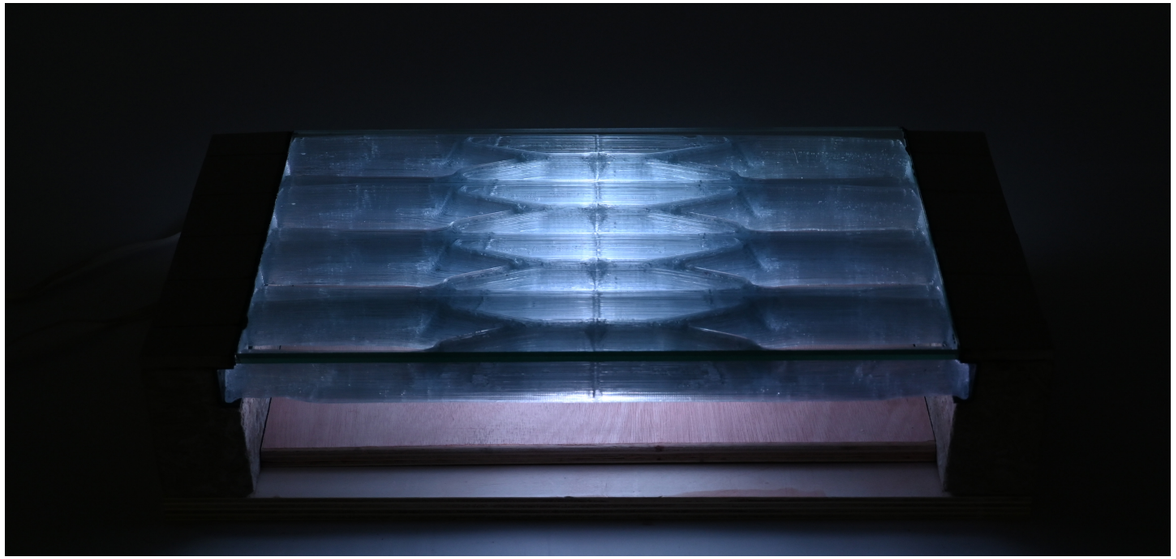


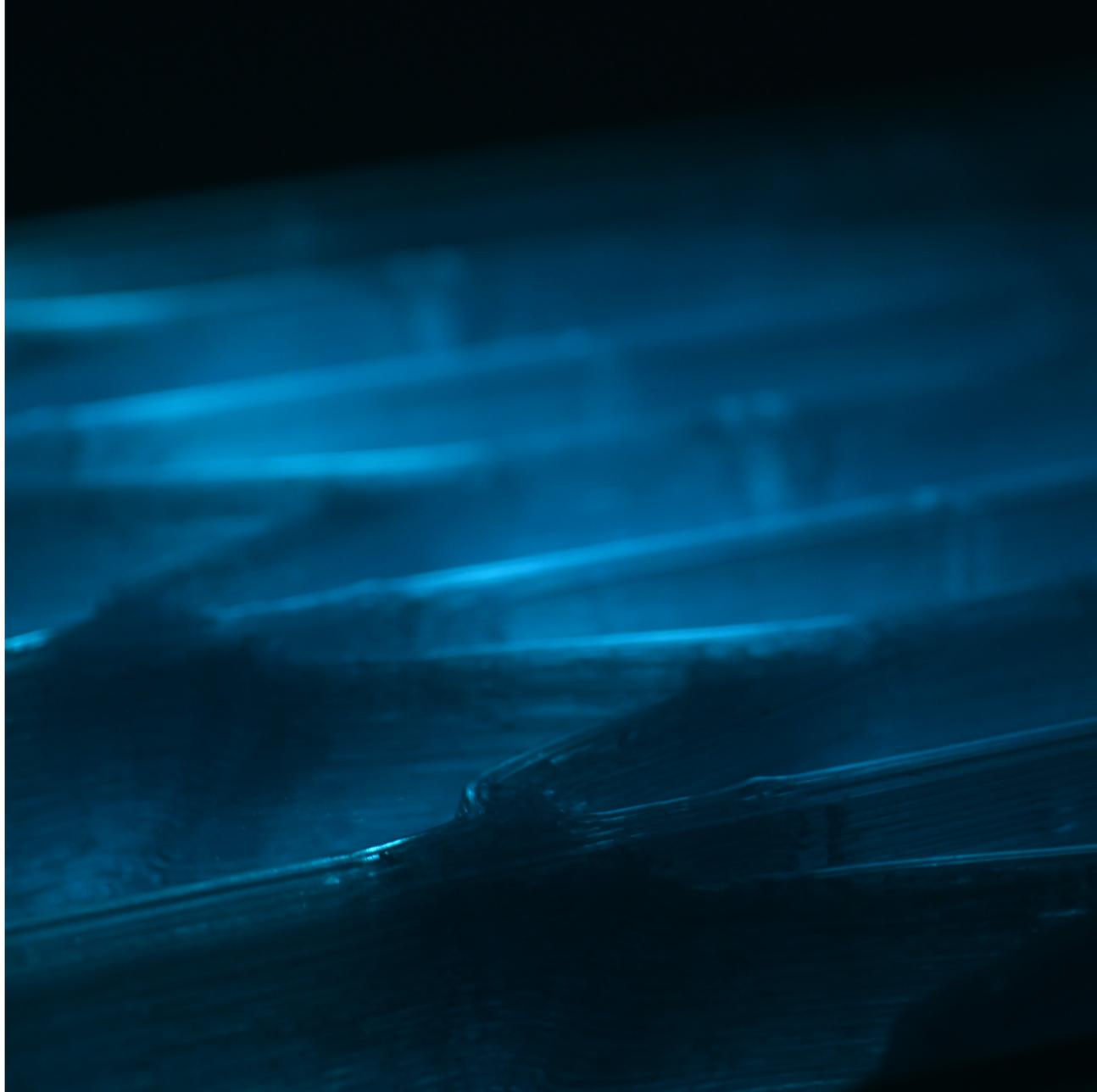




9.7 Final pictures PET-G
scale maquette 1:10

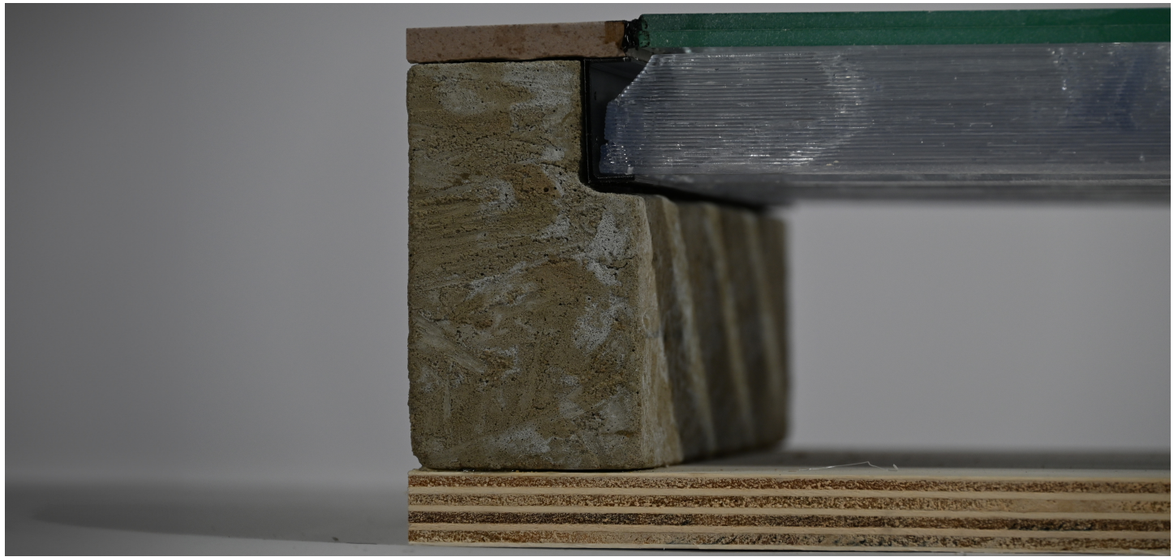


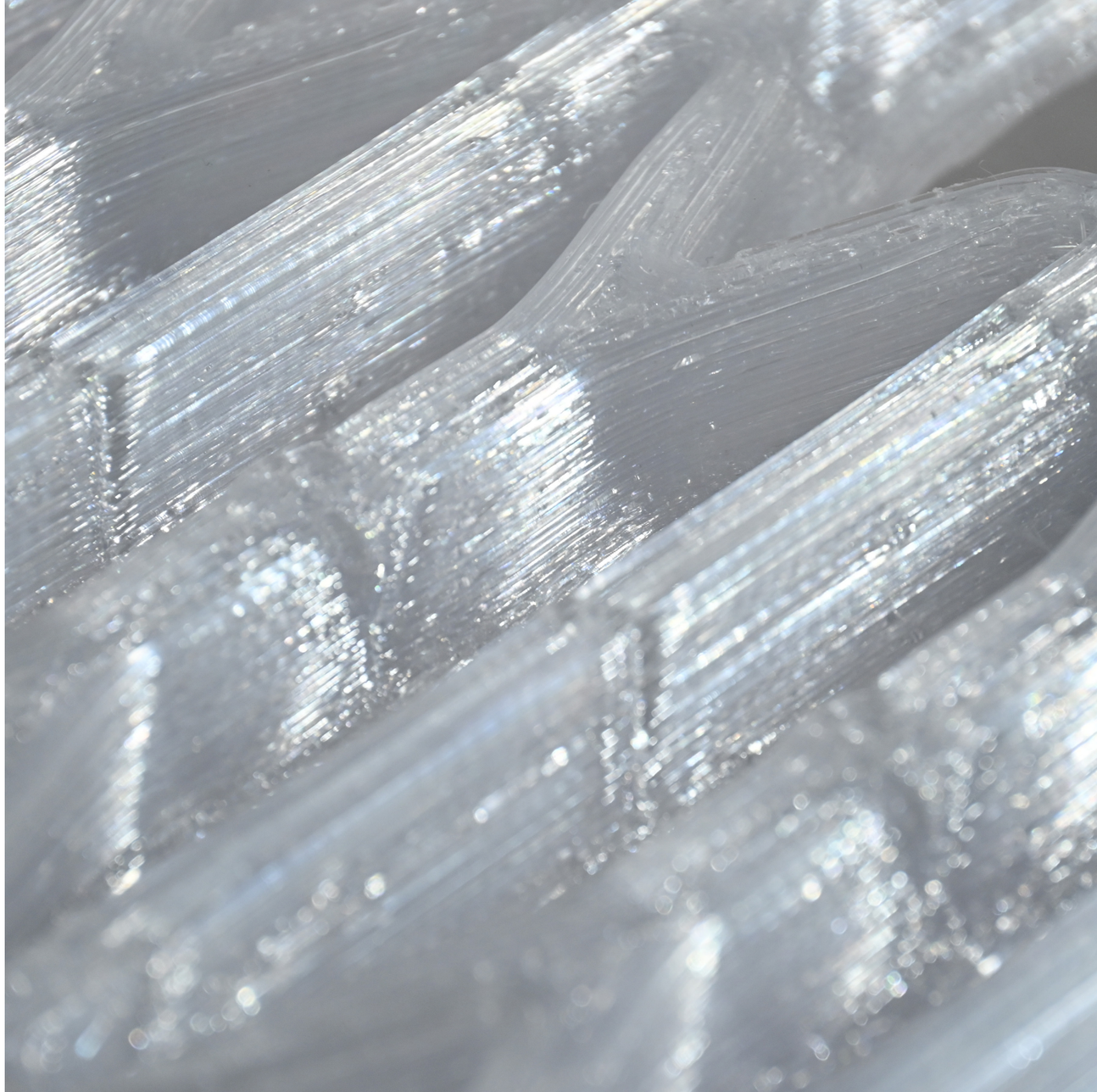




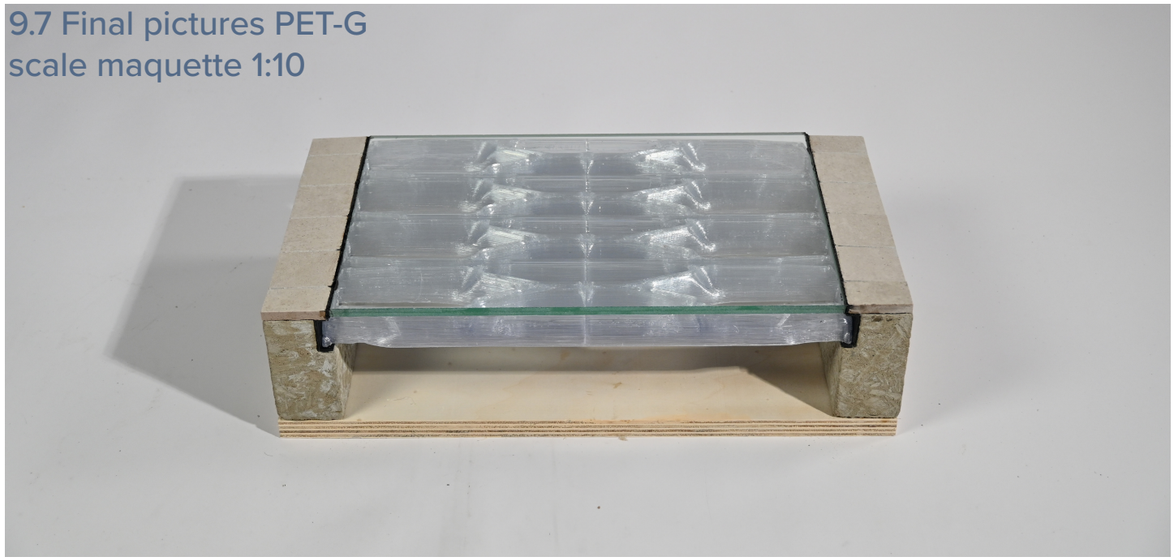
9.7 Final pictures PET-G
scale maquette 1:10

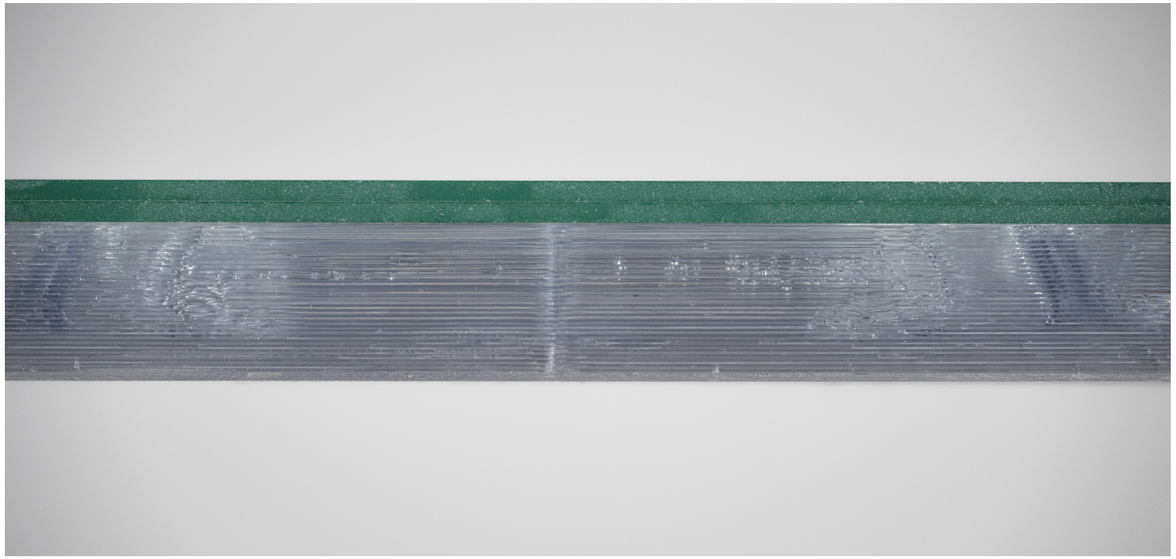


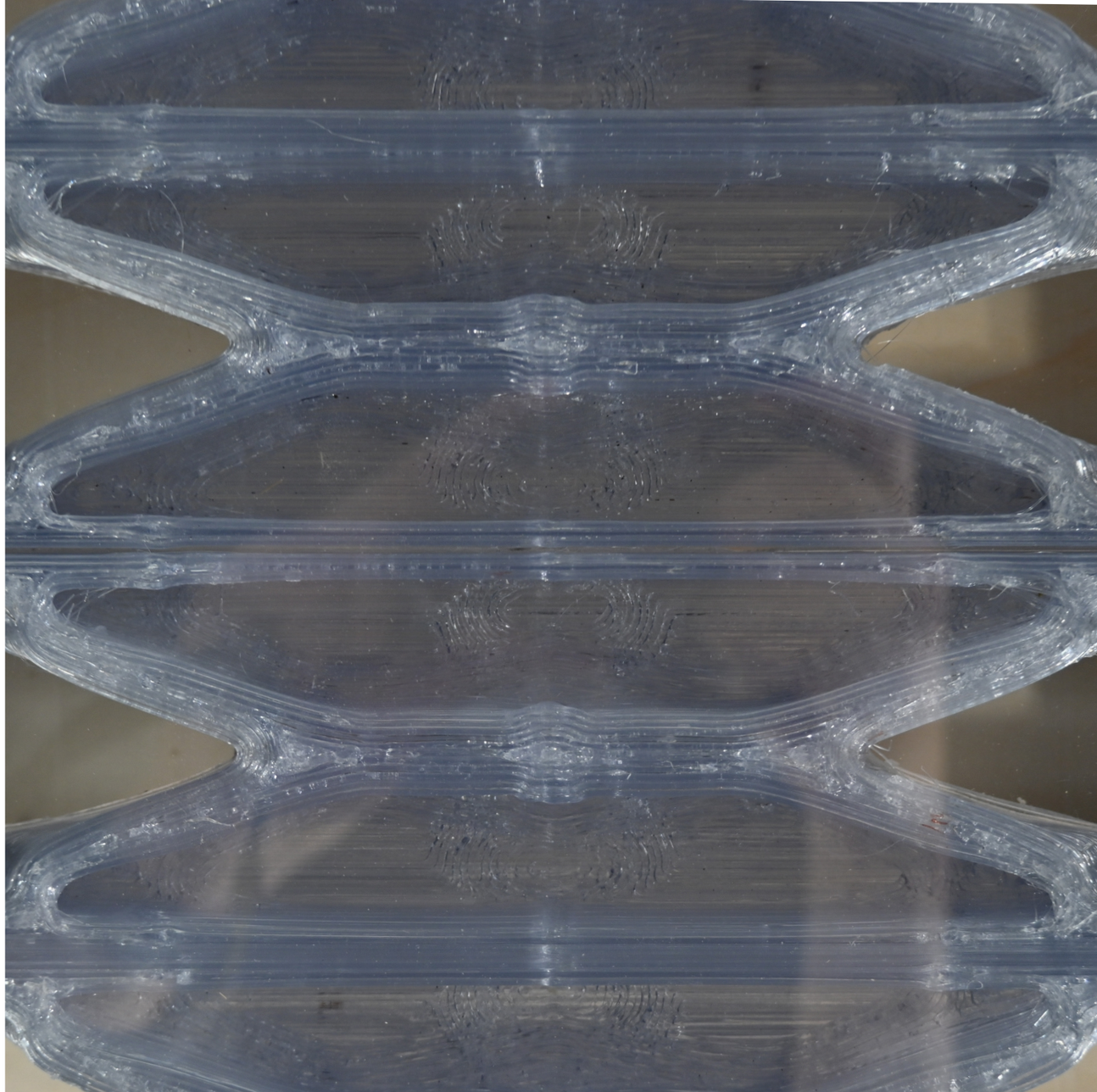




9.7 Final pictures PET-G
scale maquette 1:10

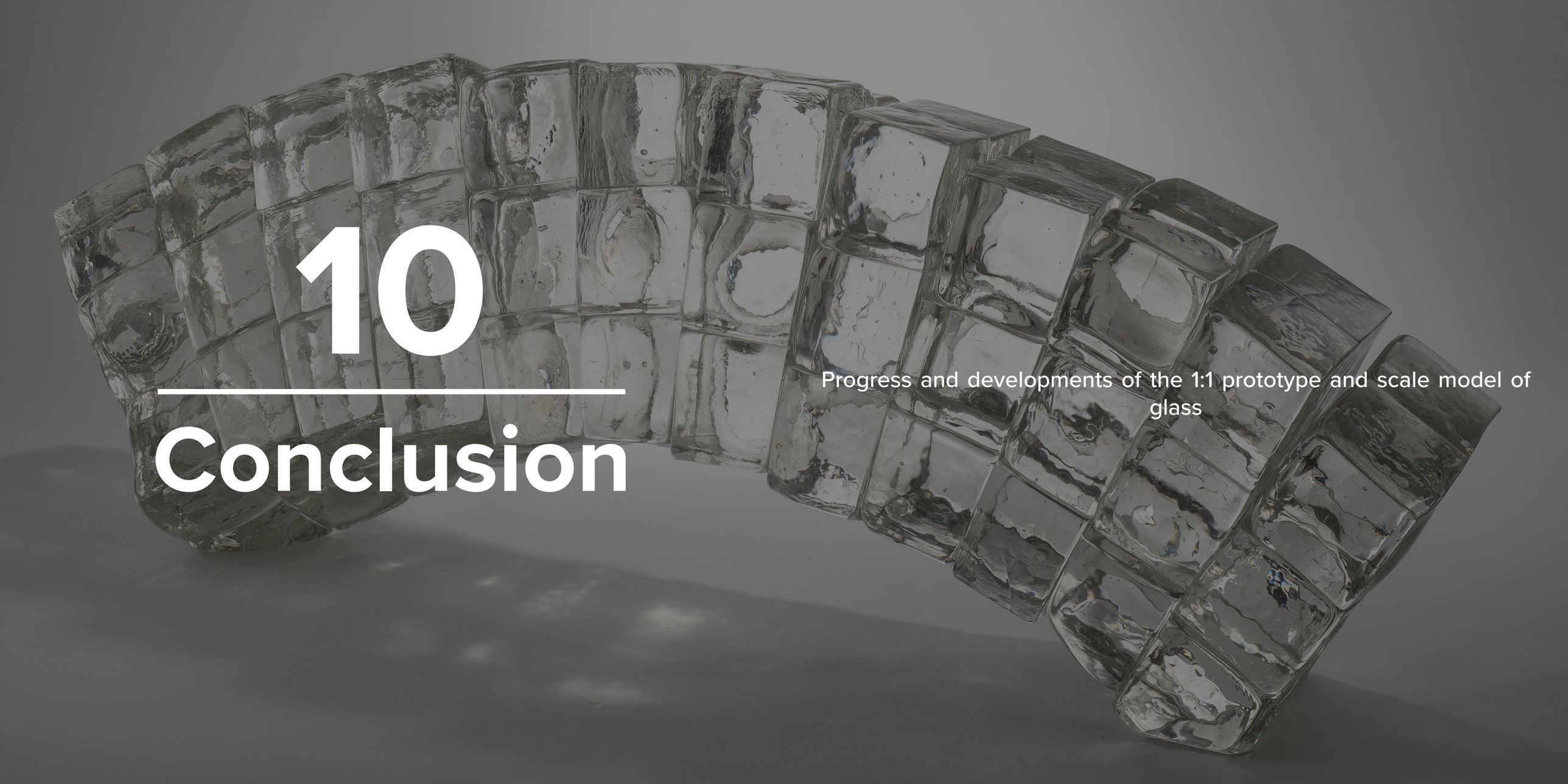






9.7 Final pictures PET-G
scale maquette 1:10 (light photoshop combined)





10
Conclusion

Progress and developments of the 1:1 prototype and scale model of glass

10 Conclusion

This master's thesis delved into the realm of 3D-printed, topology-optimized, large-scale glass structures, abbreviated as TO3DPGS. It served as the culmination of the Building Technology program at TU Delft, spanning a period of 30 weeks leading up to its final presentation.

In conclusion, the research journey unveiled several crucial steps in addressing the core question. A thorough review of literature established the initial parameters for algorithmic development. After the development the sub question could be anwered and with that the main research question can be tackled. Which is:

- How can the manufacturing process be tailored to fabricate a topology-optimized monolithic glass structure while ensuring effective production constraints?

This research question is answered at the end of the following part with the use of the sub-questions.

Material properties of glass

- What material properties of glass, should be tailored to create a topology-optimized structure using monolithic glass?

Extensive research was done looking in the material properies of glass. One of the main key parts in material specific porperties is that Glass exhibits notable weakness in tension but has exceptional strength under compression, a characteristic which is often highlighted in this thesis. Additionally, the prolonged annealing process was acknowledged, emphasizing the need for efficient heat radiation exposure to mitigate internal stresses.

Also the avoidance of sharp corners, and minimizing disparities in diameter, are important evaluations in the fabrication process of monolithic glass structures. One crucial factor to consider in manufacturing with casting or printing is the viscosity of glass at lower temperatures. Due to the extreme heat required for glass to reach its most fluid state, in 3D printing, the glass remains relatively 'thick,' achieving only a honey-like viscosity.

Prioritizing these constraints allows for attaining the desirable properties of glass, such as its exceptional transparency, impressive compressive strength, resistance to toxins, and complete recyclability.

Fabrication of topology optimized glass structures

- What fabrication options exist for constructing freeform giant glass structures, and what are the associated advantages, disadvantages, and potential applications in conjunction with topology optimization?

Consequently, various glass production methods were evaluated. These methods all fall within one of the following three categories; float, cast, and printed glass.
Float glass, despite its widespread use, posed limitations due to its maximum achievable layering and significant waste generation during cutting. At the time of writing the maximum applied structural glass has a layer thickness of 18 consisting out of 15mm float planes with special 'glueing' layers in between.

Casting initially seemed promising but was hindered by time-consuming polishing requirements and limitations on form freedom. The additional need for polishing and preparing the mould (3d-printed sand or the lost-wax technique) is time consuming. And at writing of this thesis their hasn't been an solution to create direct full transparenty when broken out of these molds.

Alternatively, 3D printing emerged as a solution, particularly through Direct Printing methods requiring minimal post-processing. Within the realm of 3D printing, the G3DP2, a machine developed by the MIT Media Lab, showcased significant promise. This innovative system directly deposits glass onto a bed within an annealing environment, offering ample form freedom and streamlining the fabrication process.

Fabrication limitations

- What are the constraints for this favored fabrication technique, and how must they be adapted to suit a topology-optimized structure?

However, like any fabrication method, 3D printing of glass comes with its limitation; the overhang angle, continuity, and print path. These key factors were examined using the G3DP2 developed by MIT as a baseline. In this thesis, size limitations and the minimum-maximum radius of the path were not considered, as discussions with professionals indicated that these limitations could be easily overcome with additional research and funding.

The overhang angle presents a challenge specific to the material. Similar issues arise with softer materials that have longer hardening times, due to the viscocity when printed. Consequently, the maximum overhang angle achievable from the bed is limited to over 60 degrees.

Continuity in the print path is crucial, ensuring uninterrupted printing. As with this printer it is not possible to stop-start controlleable, also it is highlighted that stop-starting has critical influence on the final strength of the print. Bridging is prohibited as the material remains viscous until post-printing annealing begins and would thereby fall directly on the lower parts.

The print path is essential for all 3D-prints, as issues like under and overfilling are common in 3D printing and can significantly impact the strength of the final object. Creating a boundary which can be fully 'filled', is of extreme need due to the problem of zig-zag's which worsens the feaseability, strength & astetical appearance of the structure.

10 Conclusion

Algorithms

- Which algorithms (implementations) are essential for producing a topology-optimized glass structure, particularly in the context of 3D printing?

As these boundaries are set for the implementations into the algorihm. Research into topology optimization revealed that utilizing SIMP optimization would be most suitable for this master's thesis. Hexahedral elements were ultimately selected for populating the domain.

The algorithm implementations include the overhang filter, the Layer to Layer filter (adjusted to 3D). Cubic hexahedra were scaled to have twice the height compared to width or depth, altering the overhang filter from 45 degrees to 63.435 degrees while maintaining user readability with rounded numbers in the domain.

The reconnection of islanding within print layers was Initially done by a MATLAB function which attempted for reconnecting islanding, but during algorithm testing, an adjusted version of the Dijkstra algorithm was adopted due to the lower addition of volume in the domain. Modifications within Dijkstra were centered on ensuring that it would encounter islands as start points rather than individual elements. Connections were established multiple times throughout the algorithm, assigning volume to nodes involved in connections. This elevated the significance of certain elements when the filter was active, thereby retaining them within the optimization loop.

The print path would be constructed by the nozzle offset, employing the medial axis and subsequently offsetting the boundaries of the topology optimization domain by two times the nozzle size. This post-processing step followed on the optimization algorithm. The other two adjustments were integrated into the algorithm.

The final object was exported using conversion to STL. The STL was triangulated via ISO export to ensure a smoothed, print-ready design could be obtained.

Fabrication

- Can the integration of glass constraints be achieved in the creation of a topology-optimized structure using additive manufacturing, mimicing with the use of large-scale plastic additive manufacturing?

Unfortunately, the principle with the (big 3D) Comau printer wasn't executed, so the real test with 3D printed PET-G wasn't applied. However, in the digital environment, it looked promising. One notable point is that the need for infill significantly altered the tool path. By adjusting the settings to ensure the tool path only travels within the geometry, visible bridging was prevented, potentially creating a seamless-looking prototype. Unfortunately, the utility technician at the lab didn't fix the extruder, so further experimentation could not be conducted.
Despite this setback, a glass prototype was made using the lost wax casting method. First, a scaled and adapted to a minimum feature size of 6mm flanges was produced. Whereafter the PLA model was 3D printed. Then, a silicon mold was created around it, allowing the PLA version to be released. Wax was then poured into the silicon mold. Once the wax model was set, silica plaster was poured around it. The wax was then steamed out of the silica plaster, which was subsequently placed in an oven with glass flower pots on top. After the firing process, the silica plaster cast was broken away to reveal the glass piece. The top and bottom of the glass were then polished, completing the prototype. Although this isn't a direct method of printing glass, it achieved the visual characteristics of glass, which was the primary goal of this experiment.
In the last week, a prototype made from 4mm float glass panels was produced using a waterjet cutter. Although the prototype was created for visual purposes, the experiment resulted in a high-quality finish and an impressive outcome.

Concluding

A monolithic glass structure optimized through topology can now be created via 3D printing. This thesis introduces an algorithm that considers overhang, continuity, and path planning to ensure the feasibility of printing a topology-optimized model.

Despite progress, achieving a full prototype remains a lengthy process, hindering thorough feasibility testing. However the algorihm shows potential for further exploration as the algorithm can be tweaked for other materials or even for casted glass. As the feasseability for cast glass is in most cases also increased due to the overhang filter which clusters the elements.
However, this research really advances the field of topology optimization, extending on current development within the research field. The addition of 3D printed glass and the reconnection path with the use of Dijkstra are new approaches for topology optimization. Also, the nozzle offset has not been found in the literature for smaller nozzles that are exclusively aimed at concrete printing (with large nozzles). Combining these different aspects within topology optimization broudens the field.

Therefore, the research objective positioned in the introduction has been accomplished:

"This thesis aims to push the boundaries of current research by developing a dedicated algorithm for crafting topology-optimized 3D-printed glass parts. This algorithm will not only account for the structural characteristics of glass but also tackle the limitations posed by 3D printing manufacturing. This endeavor adds to the wider investigation into 3D printing in the realm of topology optimization, particularly within glass structural applications."

11

Discussion

Discussion off the limitation of this master thesis for 3D printed topology optimized glass for structural applications in architecture.

11 Discussion

Mentioned by the start of this master thesis ‘This will become a difficult thesis with a lot of algorihm development involved‘. Also the lack of following the glass elective was adressed. As this would give an disadvantage over other students who followed the course.

However, after navigating/creating 1581 lines of MATLAB dedicated to optimization, 468 lines for 2D optimization, and a further 1262 lines of Python for the path offsetting, the discussion of this master’s thesis is reached. Along this thesis path, moments of both struggle and enlightenment were encountered.

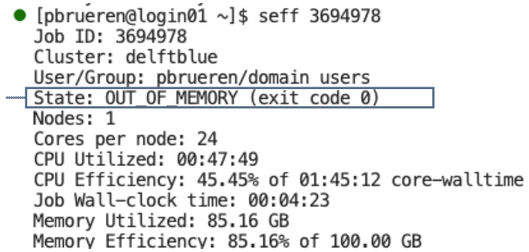
This thesis marks a significant stride forward in the creation of a customizable optimization tool tailored for the design intricacies of large-scale 3D printed glass structures. Despite these advancements, numerous challenges persist, leaving room for further research and exploration.

These will be adressed in different subsections were after a total recommendation for furhter research is given.

Evolution of the algorihm.

The evolution of the optimization algorithm began with the introduction of 8-node elements used within the SIMP approach utilizing the MMA solver principle. With considerable enhancements to the algorithm, the design domain could expand dramatically. Nevertheless, this expansion would entail a significant increase in RAM usage. Efforts were made to optimize allocation and explore alternative computational resources.
Further increments were imperative, particularly with the necessity of DelftBlue; without it, accommodating such a volume of elements within the domain would be unattainable. When not enought memory is assigned to the design domain the optimization loop will stop. As shown on the figure below. As the loop becomes more optimized, perhaps through the implementation of a heavy filter, it’s likely that the iterations within the loop could become lighter. This adjustment could render it more suitable for larger design domains.

Considering the rapid pace of computational advancements in our current era, it’s conceivable that in, say, five years, the optimization process could feasibly accommodate the creation of more detailed and expansive design domains.



Limitations for the overhang

To enhance the overhang filter, as discussed earlier in the subsection, expanding support to accommodate 9 nodes instead of the current 5 would represent a crucial advancement. This expansion would greatly enhance the flexibility in node generation across the design domain. Implementing an alternative selection method for determining the essential elements within the overhang loop would notably enhance the efficiency of the initial iterations. Currently, several iterations are required to achieve a shape that satisfies both the function and the overhang filter requirements.

Islanding Dijkstra and connecting the dots

The Dijkstra algorithm appears to be a suitable choice for extending this thesis. Each islanding layer introduced into the algorithm is connected via the shortest route, presenting a key area for improvement. As discussed with Langelaar, it would be advantageous to establish paths where volume already exists or is most likely to exist, essentially creating a blueprint underlying the optimization process. Although these paths may not always be the shortest, they could significantly enhance the optimization objective, as the significatns of these elements is allready determenid by the algorihm. Therefore including them in the final object would potentially be more beneficcial, then the ‘shortest path‘. Furthermore, integrating this concept within the overhang considerations would result in a comprehensive algorithm where islanding and overhang filtering are seamlessly combined.
Regarding this thesis, the presence of the islanding function occasionally leads to asymmetric results. One might ponder whether it would be advantageous to integrate it as a post-processing step alongside the nozzle offset.

Nozzle offsetting

Since the sizing of the nozzle was carried out separately in a Python algorithm from the main optimization script in MATLAB, a recommendation would be to integrate it into MATLAB. This integration could occur after the optimization loop, creating a closed-loop algorithm. In this setup, users could input variables, and the algorithm would generate a fully printable object, streamlining the process into one cohesive workflow.

Several limitations exist within the Python script, such as only selecting one layer at a time. A more user-friendly and efficient approach would involve offsetting the hole structures side by side. This could particularly enhance usability if the entire process could be initiated with the press of a single play button to start the algorithm.

Combining variables for printing

As previously suggested, combining various variables within a single algorithm—adjusting overhangs, establishing connections using blueprints, and offsetting the nozzle—could lead to the creation of a comprehensive algorithm capable of accommodating all 3D printed parts with material-specific characteristics. This innovation presents a significant opportunity for multiple departments within the additive manufacturing realm.

Such an algorithm has the potential to minimize construction waste on a large scale while also unlocking possibilities for greater design freedom in structural and architectural applications. By integrating material-specific considerations into the optimization process, this approach could revolutionize how we approach additive manufacturing, fostering innovation and efficiency across various industries.

11 Discussion

3D printing properties

In this thesis, the focus was primarily on the key aspects of preparing for the production of a 3D printed glass object. However, there is room for future improvements, particularly in addressing the non-isotropic characteristics of 3D printed glass. For instance, there's a notable difference in strength between horizontally direct printed planes and vertically or parallel printed lines. By incorporating these considerations into the optimization process, it's possible to anticipate better outcomes and achieve structural performance closer to that of the real produced item.

One approach to tackling this issue could involve slicing the optimization and treating the different layers as distinct connections between the nodes. This method allows for a more nuanced analysis of the structural integrity at various orientations, potentially leading to more robust designs that account for the inherent anisotropy of 3D printed glass.

Post processing

The slicing process currently relies on either CURA or SLIC3R, each with its own set of characteristics. SLIC3R stands out as an open-source application, meaning its development isn't primarily driven by economic profits. However, both slicing software options have their limitations when it comes to certain applications, including topology optimization.
To address these limitations, algorithmic tweaks could be implemented to tailor the slicing process more precisely for topology optimization requirements. This adjustment could involve optimizing slicing patterns to better suit the specific needs of complex structures generated through topology optimization.

Post processing additions

Non-planar slicing, which was experimented with in this thesis, presents a potential solution. However, further research is necessary to determine its effectiveness for these types of structures. By exploring alternative slicing approaches and conducting thorough research, it's possible to refine the slicing process and enhance its suitability for topology-optimized designs, ultimately improving the overall efficiency and quality of 3D printed objects.

3D printing and testing

One significant limitation of this thesis lies in the lack of proper testing for the final optimization results. While glass objects were fabricated, they were never subjected to structural integrity tests. Also the printer (G3DP2) has also two limitiations that were not incorperated within the algorihm, the minum radius and the minimum path length. However these two points were discussed with an expert as there were mentioned that these limitations could possible be resolved, therefore in further research they could be evaluated but are the ignored for this thesis.
Furthermore, The fabricated glass objects were not produced using 3D printing glass techniques but rather through the lost wax technique, which involves the casting of glass. This discrepancy raises questions about the applicability and validity of the optimization results in real-world scenarios.

In the case of the larger prototype made from PET-G, full printing was not executed; instead, infill was used to conserve material. This approach most likely result in worse performance compared to printing in full.
However, due to the scope of this research, conducting tests on full PET-G elements was deemed unnecessary.

To address these limitations and ensure the reliability of the optimization process, future work should include structural integrity testing of the optimized designs using appropriate 3D printing techniques.

Process

In conclusion, the process of this thesis was executed effectively, with the boundaries established after P2 providing a clear scope for what needed to be achieved within the allotted time frame. However, these boundaries also significantly influenced the trajectory of the graduation project, shaping the direction and focus of the research.

Moving forward, there is ample opportunity for more extensive research on the topic of 3D printed topology optimized glass structures. A potential avenue for further exploration could be pursuing a PhD research that delves into each of the three additions separately, with the final addition dedicated to concluding the topology optimization process.

Additionally, it would provide the ability to explore more advanced design domains or boundaries, as well as conduct evaluations and tests on both the algorithmic and physical elements involved. This comprehensive approach would contribute significantly to the advancement of knowledge in this field and open up new avenues for innovation in additive manufacturing and structural design.



Figure 59: AI generated full glass birdige in forrest with water



12

Reflection

Graduation reflection (P4) on design process and master track Building Technolody (Architecture, Urbanism and Building Sciences)

12 Reflection

1. What is the relation between your graduation project topic, your master track (A, U, BT, LA, MBE), and your master programme (MSc AUBS)?

The aim of this graduation project is to develop an algorithm for creating a 3D printed monolithic glass bridge, posing the research question: *How can the manufacturing process be tailored to fabricate a topology-optimized monolithic glass structure while ensuring effective production constraints?*
This question encompasses several aspects integrated within the master track Building Technology.

One of the primary focuses is on understanding structural elements and delving into technical aspects, supplemented by completed coding courses within the master track. The development of the algorithm is one of the main aspects of this graduation project, with significant alignment between the master track and the goals of the project. Additionally, research of structural performance and feasibility are central to the Building Technology master track. Material science is also one of the aspects tackled wihtin the master, as the material science from glass is one of the key parts within the design of the 3D printed glass topology optimization.

Concluding there are several aspects incorporated in the master track where the foundation of the graduation rests on, such as materials science, coding, structural performance, feasibility, 3D-printing and aesthetic quality. The final result of considering all these various factors would be a 3D printed design that's optimized for efficiency, aligning well with the principles of minimizing waste and using materials efficiently. These principles are emphasized not only in the building technology master track but also in both national and international guidelines.

Whereas the Architecture and Urbanism Master graduation tracks are for example focussed on user experience, programmatic studies, the goal of this graduation project is more scientific. The goal is to find the maximum opportunities of a material, in this case glass, making other context such as social context more irrelevant. It is a niche topic, in the larger scope of the MSc AUBS and the professional world related to architecture and building sciences, research like this is the foundation on which applied design is build. Making it possible to explore new glass-design in architectural projects.

The website from the TU-Delft building technolodgy track states: *'The emphasis of this programme is on the design of innovative and sustainable building components and their integration into the built environment.'*

At its core, this thesis focuses on all the key words of the above sentiment; its basis lies in an innovative way to create a component for the built environment with importance for sustainability. The main aspects in this thesis are, understanding the complexities of topology optimization, which include finite elements, materials science, extensive coding (using MATLAB/Python), and a thorough understanding of 3D printing techniques. With this exploration, the aim is to push the boundaries of what glass can achieve within architectural contexts, and promote innovation and creativity in design practices, while enhancing the sustainability and reusability of the materials used for the component, due to the achievement of a 3D printed topology optimized structural glass element .

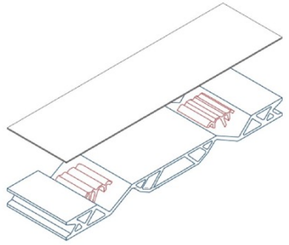
2. How did your research influence your design/ recommendations and how did the design/ recommendations influence your research?

Changes have been made through the addition of extra research, as can be seen in the picture above. Throughout the feasseabilities and the parameters that were set, the design changed significantly. The design is strictly coherend to the algorithm, but the algorithm can be tweaked until the user approves the quality of the outcome. Quality can be devided in this case into the different aspects incorperated within this graduation (see question 1).

The study pinpointed three key factors essential for achieving a viable 3D printed topology optimization: managing overhangs, determining nozzle size, and adapting printing paths. These factors are crucial for effective 3D printed topology optimization and differ from those for casted glass objects. To ensure optimal outcomes in terms of material use and strength, it is important to integrate these aspects into the optimization process. Neglecting one of these criteria could lead to failure in creating an optimized 3D printed glass element, resulting in issues like bridging (failed prints) and compromised strength.

The overhang angle notably impacted the outcome of the topology optimization. By adjusting parameters and the design domain, the user has influence on the final outcome of the topology optimalization (an experienced user could estimate a rough outcome of the topology optimalization when different settings are communicated).

At first, the use of linear quadrilateral elements was dismissed following advice from Charalampos. However, after receiving input from Matthijs Langelaar, it was acknowledged as the most straightforward approach for modifying the overhang angle, prompting its adoption. Altering the finite elements allowed for adjustments to the overhang angle during the topology optimization process, resulting in various angles being attained. For this thesis, an angle exceeding 60 degrees when viewed from the base plate is deemed significant due to the material characteristic of glass.



Anna Maria Koniari 2022



Eva Schoenmaker 2023



Pim Brueren 2024

12 Reflection

3. How do you assess the value of your way of working (your approach, your used methods, used methodology)?

The extensive research on the 3D printing limitations for topology optimization seemed to enhance the design in the next phase. clear boundaries were established for adaptations in the algorithm, the changes could be easely adressed one at a time and later be combined in the topology optimization. This specific type of topology optimization hasn't been previously undertaken.
To develop it, various existing algorithms have been modified and supplemented with additional ones. A new algorithm was created, considering the specific constraints of working with glass, thereby improving upon the original algorithm's efficiency. This new algorithm incorporates several crucial elements for glass, such as ensuring path continuity, adapting nozzle size, and adjusting overhang angles, all of which are vital considerations. Addressing each limitation separately enhances the algorithm's adaptability and speed of operation.
This approach significantly boosts the flexibility and efficiency of the workflow. Moreover, with minor adjustments, the final algorithm can be tailored to optimize concrete structures as well, increasing its functionality and applicability across different materials. Presently, this topology optimization is specifically tailored to accommodate the characteristics of 3D printed glass.
However, the limitations lie within the scarcity of published code. The papers explain therefore their methods but not converge them to a workable code. Therefore, elaborate research has to be done to configure the algorithm to a more efficient/functional one. The reflection at the end of each step, enhances the final outcome signiffanctly, as several changes throughout the hole code ultimately resulted in significant enhancements to the speed and efficiency of the topology optimization.

4. How do you assess the academic and societal value, scope and implication of your graduation project, including ethical aspects?

Ethical considerations play a significant role in this graduation project. The production and use of glass involve substantial CO2 emissions. However, the printing technique employed here emphasizes the potential for 100% recyclability of glass structures. Continuous research within topology optimization, with new findings emerging even during the graduation period, highlights the increasing importance of efficient material usage for the future. Topology optimization expands this potential by minimizing material usage, directing it only where necessary to form a stable structure. Furthermore, employing a printing approach for topology optimization maximizes this potential by eliminating material wastage during fabrication. In fact, in this scenario, recycled glass can even be utilized for constructing the architectural elements, further enhancing sustainability.

At an architectural scale, similar topology optimized structures are achieved using concrete. Yet, the transparency, compressive strength and recyclability of glass present a compelling alternative. Unlike concrete, which is often not entirely recyclable and involves CO2 emissions during granulate production, glass emerges as a more prestigious and renewable option. Furthermore, considering the challenges posed by fluctuating electricity demand, there is a significant opportunity in producing 3D printed glass structures during periods of low electricity demand and high stress in the grid.

5. How do you assess the value of the transferability of your project results?

The topic of transferability in the graduation project is inherently complex and technical, posing significant challenges. However, the limitations of the 3D printing values are quite comprehensive and understandable. As the outcomes show transformation within the topology optimization shape. And shape changes are for more people understandable than the understanding of the finitie elements. For someone used to topology optimization it is straightforward which changes have been adopted to the topology optimization. Consequently, the challenge for the final presentation lies in effectively explaining the technical intricacies of topology optimization using shapes that are coherent to individuals without prior experience in the field. If someone who would continue on the algorithm's development, I would recommend focusing on the anisotropic nature of the printed layering of the object. This could lead to outcomes that significantly diverge from the current projected feasibility of the object.

6. How did the existing research differ/extend from the findings and development of the project?

The existing research provided insights to configuring topology optimization for 3D printed structures. However, several methods and outcomes were tried for this graduation. Some methods were adapted directly in the code of the topology optimization as was the case for the overhang filter. To change the overhang angle several methods were investigated. Whereafter one was adapted, and the algorithm was changed accordingly. Also, the filter was adapted to expand from 2D to 3D applications.
For the path continuity, the literature method was assessed and found functional, a different approach was ultimately chosen. This shift had taken place from utilizing MATLAB's islanding connection function to employing the Dijkstra algorithm for ensuring continuity in topology optimization. This novel approach not only enhances efficiency but also reduces the additional material required to reconnect the 'islands' in each layer of the optimization process. Thereby extending on the current research and showing new potential in the combination and adaptation of existing methodologies.

7. If the project would develop further what would be your recommendations?

One of the key parts which could exstent this research is the integrating of an algorithm tailored for layer-specific use in 3D printing processes, considering variations in forces between print directions and vertical displacements. This enhancement is crucial due to the intricacies of 3D printing. The algorithm could incorporate distinct structural implementations for Finite Element Method (FEM) analysis in various directions or per layer. Building upon existing research on element slicing, it could be seamlessly integrated into the thesis's algorithm. Subsequent testing would validate its real-world applicability. Notably, numerous researchers have dedicated their doctoral studies to essential aspects like the overhang filter, highlighting the vast research opportunities in integrating 3D printing with topology optimization.

13

References

References

- Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. Struct Multidiscip Optim 51(3):565–572
- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., & Sigmund, O. (2011). Efficient topology optimization in MATLAB using 88 lines of code. Structural and Multidisciplinary Optimization, 43(1), 1–16. https://doi.org/10.1007/s00158-010-0594-7
- Baudet E, Ledemi Y, Larochelle P, Morency S, Messaddeq Y. 3D- printing of arsenic sulfide chalcogenide glasses. Optical Materials Express, 2019, 9(5): 2307
- Bendsøe, M. P., & Sigmund, O. (2004). Topology Optimization. Theory, Methods, and Applications. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-05086-6
- Bi, M., Tran, P. T., Xia, L., Ma, G., & Xie, Y. M. (2022). Topology optimization for 3D concrete printing with various manufacturing constraints. Additive Manufacturing, 57, 102982. https://doi.org/10.1016/j.addma.2022.102982
- Bi, M., Xia, L., Tran, P. T., Li, Z., Wan, Q., Wang, L., Shen, W., Ma, G., & Xie, Y. M. (2022). Continuous contour-zigzag hybrid toolpath for large format additive manufacturing. Additive Manufacturing, 55, 102822. https://doi.org/10.1016/j.addma.2022.102822
- Bendsøe, M. P., & Sigmund, O. (2004). Topol- ogy Optimization. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-05086-6
- Bruggi, M., & Duysinx, P. (2012). Topology optimization for minimum weight with compliance and stress constraints. Structural and Multidisciplinary Optimization, 46(3), 369–384. https://doi.org/10.1007/s00158-012-0759
- Bruns, T., & Tortorelli, D. (2001). Topology optimization of non-linear structures and compliant mech- anisms. Computer Methods in Applied Mechanics and Engineering, 190, 3443–3459. https://doi.org/10.1016/S0045-7825(00)00278-4
- Choi Y H, Kim C M, Jeong H S, Youn J H. Influenceofbed temperature on heat shrinkage shape error in FDM additive manufacturing of the ABS-engineering plastic. World Journal of Engineering and Technology, 2016, 4(3): 186–192
- Damen, W. (2019). Topologically Optimised Cast Glass Grid Shell Nodes. Exploring Topology Opti- mization as a design tool for Structural Cast Glass elements with reduced annealing time.
(Master's thesis). Technical University Delft. Delft.
- Delissen, A., Boots, E., Laro, D., Kleijnen, H., Van Keulen, F., & Langelaar, M. (2022). Realization and assessment of metal additive manufacturing and topology optimization for high-precision motion systems. Additive Manufacturing, 58, 103012. https://doi.org/10.1016/j.addma.2022.103012
- Dyer, T. (2014). Glass recycling. In Elsevier eBooks (pp. 191–209). https://doi.org/10.1016/b978-0-12-396459-5.00014-3
- Evenline (2023). https://www.evenline.co
- Engineering Niomta. Progress has been made in research on 3D printing technology and equipment for glass at Ningbo Institute of Materials Technique and Engineering. 2015
- Eckersley O'Callaghan. (2018). Glass Wippe. https://www.eocengineers.com/projects/glass- wippe-385/
- F. Kotz et al., "Three-dimensional printing of transparent fused silica glass," Nature, vol. 544, no. 7650, pp. 337–339, 2017.
- Giesecke, R., & Dillenburger, B. (2022). Three-dimensionally (3D) printed sand molds for custom glass parts. Glass Structures & Engineering, 7(2), 231–251. https://doi.org/10.1007/s40940-022-00176-y
- Inamura Chikara, Stern Michael, Lizardo Daniel, Houk Peter, & Oxman Neri. (2018b). Additive manufacturing of transparent glass structures. 3D Printing and Additive Manufacturing, 5(4), 269–283. https://doi.org/10.1089/3dp.2018.0157
- Inamura, C. (2017a). Towards a New Trans- parency: High Fidelity Additive Manufacturing of Transparent Glass Structures across Scales. MIT.
- Inamura, C. (2017b). GLASS II - APPLICA- TION. https://www.chikara-inamura.com/GLASS/ GLASS-II-APPLICATION
- Inamura, C. (2020). Glass Additive Manufac- turing. Paper presented at the Challenging Glass Ghent University.
- Inamura, C., Stern, M., Lizardo, D., Houk, P., & Oxman, N. (2018). Additive Manufacturing of Transparent Glass Structures. 3D Printing and Ad- ditive Manufacturing, 5(4), 269–283. https://doi. org/10.1089/3dp.2018.0157
- Ioannidis, M. (2023). Bringing glass giants to life. Master thesis TU-Delft
- Jet Edge. (n.d.). The Advantages of 5-Axis Waterjet Cutting Machines. Jet Edge Waterjets. Retrieved 13 January 2023, from https://blog.jet- edgewaterjets.com/water-jet-machines/advan- tages-of-5-axis-waterjet-cutting-machines
- Kang, H., Hong, S., Oh, S., Choi, J., & Song, M. (2002). Cutting glass by laser. Proceedings of SPIE. https://doi.org/10.1117/12.456849
- Klein J, Stern M, Franchin G, Kayser M, Inamura C, Dave S, Weaver J, Houk P, Colombo P, Yang M, Oxman N. Additive manufacturing of optically transparent glass. 3D Printing and Additive Manufacturing, 2015, 2(3): 92–105
- Klein, J. (2015). Additive Manufacturing of Op- tically Transparent Glass. http://www.liebertpub. com/doi/10.1089/3dp.2015.0021
- Klocke F, McClung A, Ader C. Direct laser sintering of borosilicate glass. In: Proceedings of the 15th Annual Symposium on Solid Freeform Fabrication. Austi: The University of Texas, 2004, 214– 219
- Kotz F, Arnold K, Bauer W, Schild D, Keller N, Sachsenheimer K, Nargang T M, Richter C, Helmer D, Rapp B E. Three-dimensional printing of transparent fused silica glass. Nature, 2017, 544(7650): 337–339
- Koniari, A.M., (2022). Just Glass. Master thesis TU-Delft
- Kuo, Y., & Cheng, C. C. (2019). Self-supporting structure design for additive manufacturing by using a logistic aggregate function. Structural And Multidisciplinary Optimization, 60(3), 1109–1121. https://doi.org/10.1007/s00158-019-02261-3
- Langelaar, M. (2016). Topology optimization of 3D self-supporting structures for additive manufacturing. Additive Manufacturing, 12, 60–70. https://doi.org/10.1016/j.addma.2016.06.010
- Langelaar, M. (2016a). An additive manufacturing filter for topology optimization of print-ready designs. Structural And Multidisciplinary Optimization, 55(3), 871–883. https://doi.org/10.1007/s00158-016-1522-2
- Liu, G. R., & Quek, S. S. (2014). Chapter 9 - FEM for 3D Solid Elements. In G. R. Liu & S. S.
Quek (Eds.), The Finite Element Method (Second Edition) (Second Edition, pp. 249–287). Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/B978-0-08-098356-1.00009-6
- Louw, H. (2015). The development of the window. In Routledge eBooks (pp. 7–96). https://doi.org/10.4324/9781315793832-2
- Luo, J., Gilbert, L. J., Bristow, D. A., Landers, R. G., Goldstein, J. T., Urbas, A., & Kinzel, E. C. (2016). Additive manufacturing of glass for optical applications. Proceedings of SPIE. https://doi.org/10.1117/12.2218137
- Moore, D. G., Barbera, L., Masania, K., & Studart, A. R. (2020). Three-dimensional printing of multi- component glasses using phase-separating res- ins. Nature Materials, 19(2), 212–217. https://doi. org/10.1038/s41563-019-0525-y

References

- Murphy, C. A., Lim, K. S., & Woodfield, T. B. F. (2022). Next evolution in Organ Scale biofabrication: Bioresin design for Rapid High Resolution VAT polymerization. Advanced Materials, 34(20). https://doi.org/10.1002/adma.202107759
- Musgraves, J. D., Hu, J., & Calvez, L. (Eds.).
- Nguyen, D. T., Meyers, C. D., Yee, T. D., Dudukovic, N. A., Destino, J. F., Zhu, C., Duoss, E. B., Baumann, T. F., Suratwala, T., Smay, J. E., & Dylla Spears, R. (2017). 3D Printed Transparent Glass. Advanced Materials, 29(26). https://doi.org/10.1002/adma.201701181
- NSG, “Pilkington and the Flat Glass Industry,” Glass Industry, 2010. [Online]. Available: https://www.pilkington.com/resources/pfgi20l0.pdf.
- Oikonomopoulou, F., Bristogianni, T., Barou, L., Jacobs, E., Veer, F. A., & Nijsse, R. (2018). In- terlocking cast glass components, Exploring a de- mountable dry-assembly structural glass system. Heron, 63.
- Oikonomopoulou, F., Bristogianni, T., Barou, L., Veer, F. A., & Nijsse, R. (2018). The potential of cast glass in structural applications. Lessons learned from large-scale castings and state-of-the art load-bearing cast glass in architecture. Journal of Building Engineering, 20, 213–234. https://doi. org/10.1016/j.jobe.2018.07.014
- Oikonomopoulou, F., Bristogianni, T., van der Velden, M., & Ikonomidis, K. (2022). The adhe- sively-bonded glass brick system of the Qaammat Pavilion in Greenland: From research to realization. Architecture, Structures and Construction, 2(1), 39–62. https://doi.org/10.1007/s44150-022- 00031-2
- Oikonomopoulou, F., Bristogianni, T., Veer, F. A., & Nijsse, R. (2018). The construction of the Crystal Houses façade: Challenges and innovations. Glass Structures & Engineering, 3(1), 87–108. https://doi. org/10.1007/s40940-017-0039-4
- Oikonomopoulou, F. (2019). Unveiling the third dimension of glass. Solid cast glass components and assemblies for structural applications. (PHD dissertation). Technical University Delft. Delft.
- Sasan, K., Lange, A., Yee, T. D., Dudukovic, N. A., Nguyen, D. T., Johnson, M., Herrera, O. D., Yoo, J., Sawvel, A. M., Ellis, M. E., Mah, C. M., Ryerson, R., Wong, L. L., Suratwala, T., Destino, J. F., & Dylla Spears, R. (2020). Additive Manufacturing of Optical quality Germania–Silica glasses. ACS Applied Materials & Interfaces, 12(5), 6736–6741. https://doi.org/10.1021/acsami.9b21136
- Oikonomopoulou, F. (2019). Unveiling the third dimension of glass: Solid cast glass components and assemblies for structural applications. A+BE: Architecture and the Built Environment, 9, 1–352. https://doi.org/10.7480/abe.2019.9.4088
- Oikonomopoulou, F., Bhatia, I. S., Damen, W., Van Der Weijst, F., & Bristogianni, T. (2020). Rethinking the cast glass mould. DOAJ (DOAJ: Directory of Open Access Journals). https://doi.org/10.7480/cgc.7.4662
(2019). Springer Handbook of Glass. Springer International Publishing. https://doi.org/10.1007/978- 3-319-93728-1
- Saint-Gobain. (n.d.). Glass manufacturing process. from https:// in.saint-gobain-glass.com/glass-manufactur- ing-process
- Sedak. (n.d.) https://www.sedak.com/en/
- Shelby, J. E. (2005). Introduction to glass sci- ence and technology (2nd ed). Royal Society of Chemistry.
- Schoenmaker, E., Adding a new dimension to glass (2023). Master thesis TU-Delft
- Sethian, J. A. (1999). Fast marching methods. Siam Review, 41(2), 199–235. https://doi.org/10.1137/s0036144598347059
- Shao, G. (2019). Comparison of BESO and SIMP to Do Structural Topology Optimization in Discrete Digital Design, and then Combine Them into a Hybrid Method. In Springer eBooks (pp. 199–209). https://doi.org/10.1007/978-981-13-8153-9_18
- Sigmund, O. (1994). Design of Material Structures Using Topology Optimization (PhD Thesis).
- Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. Structural and Multidisciplinary Optimization, 33(4), 401–424. https://doi.org/10.1007/s00158-006-0087-x
- Sigmund, O., & Maute, K. (2013). Topology optimization approaches. Structural and Multidisciplinary Optimization, 48(6), 1031–1055. https://doi.org/10.1007/s00158-013-0978-6
- Swain, B., Park, J. R., Shin, D. Y., Park, K., Hong, M. H., & Lee, C. G. (2015). Recycling of waste automotive laminated glass and valorization of polyvinyl butyral through mechanochemical separation. Environmental Research, 142, 615–623. https://doi.org/10.1016/j.envres.2015.08.017
- Today, Prototype, “Micron3DP Announces Improved Glass 3D Printer,” 2017.
- T. Gutowski et al., “Note on the Rate and Energy Efficiency Limits for Additive Manufacturing,” J. Ind. Ecol., vol. 21, pp. S69–S79, 2017.
- Van De Ven, E., Maas, R., Ayas, C., Langelaar, M., & Van Keulen, F. (2018). Continuous front propagation-based overhang control for topology optimization with additive manufacturing. Structural and Multidisciplinary Optimization, 57(5), 2075–2091. https://doi.org/10.1007/s00158-017-1880-4
- Van de Ven, E., Maas, R., Ayas, C., Langelaar, M., & Van Keulen, F. (2020). Overhang control based on front propagation in 3D topology optimization for additive manufacturing. Computer Methods in Applied Mechanics And Engineering, 369, 113169. https://doi.org/10.1016/j.cma.2020.113169
- Van de Ven, E., Ayas, C., Langelaar, M., Maas, R., & Van Keulen, F. (2021). Accessibility of support structures in topology optimization for additive manufacturing. International Journal For Numerical Methods in Engineering, 122(8), 2038–2056. https://doi.org/10.1002/nme.6611
- Van De Ven, E., Maas, R., Ayas, C., Langelaar, M., & Van Keulen, F. (2021). Overhang control in topology optimization: a comparison of continuous front propagation-based and discrete layer-by-layer overhang control. Structural and Multidisciplinary Optimization. Https://doi.org/10.1007/s00158-021-02887
- Yang, W., Li, W., Ma, G., & Feng, P. (2023). An integrated method of topological optimization and path design for 3D concrete printing. Engineering Structures, 291, 116435. https://doi.org/10.1016/j.engstruct.2023.116435
- Z. Lin, J. Fu, H. Shen, X. Yao, Smooth contour-parallel tool path generation for
high-speed machining through a dual offset procedure, Int. J. Adv. Manuf. Technol.
81 (2015) 1233–1245, https://doi.org/10.1007/s00170-015-7275-z.
- Zhang, D., Liu, X., & Qiu, J. (2021). 3D printing of glass by additive manufacturing techniques: A re- view. Frontiers of Optoelectronics, 14(3), 263–277. https://doi.org/10.1007/s12200-020-1009-z
-https://l1.nl/nieuw-glazen-kunstwerk-vertoont-scheuren-31332

14

Apendix

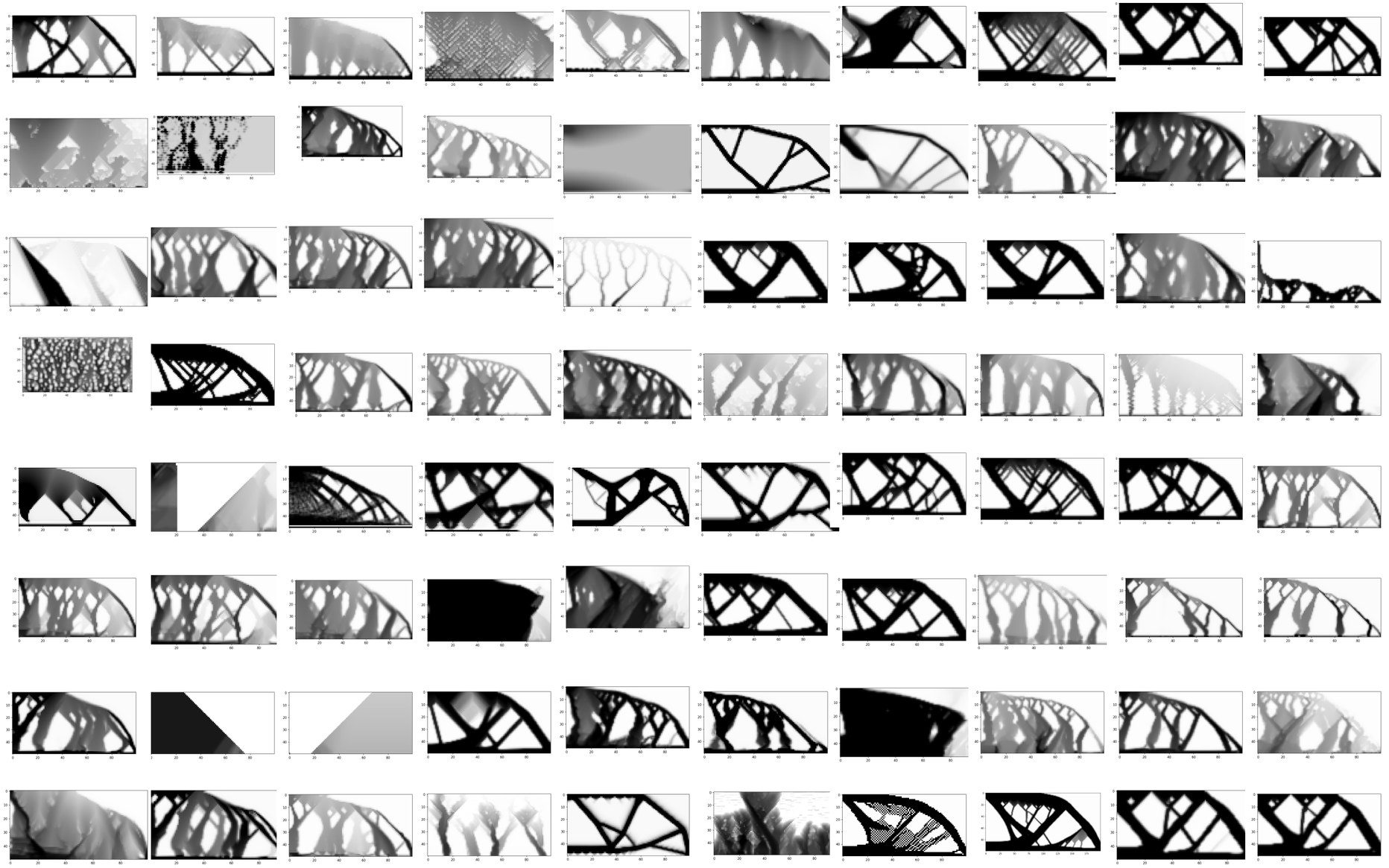
Getting Topology optimization & Overhang filter to work

To begin exploring topology optimization, the initial approach involved a 2D optimization process incorporating an overhang filter. This method served as a foundational step to understand the complexities and nuances of topology optimization. Various attempts were made to refine the overhang filter, such as modifying the interpolation methods or adjusting the nodes used for evaluation. However, none of these modifications yielded perfect results. The outcomes of these experiments are illustrated on the following pages. Despite the imperfect results, the insights gained from these trials were invaluable for the development of the final algorithm.

The 2D optimization was particularly advantageous due to its speed and flexibility, with 50 iterations taking a maximum of approximately 5 seconds. This efficiency allowed for rapid testing of different setups, significantly accelerating the learning process.

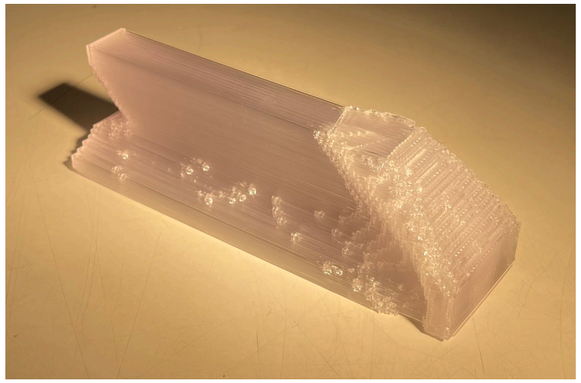
Eventually, the 2D method was abandoned in favor of a 3D optimization approach. The transition to 3D optimization required a different strategy for handling the overhang constraints. Instead of modifying the overhang filter itself, the new approach involved altering the sizes of the elements within the structure. This change addressed the limitations encountered with the 2D approach and proved to be more effective for 3D optimization.

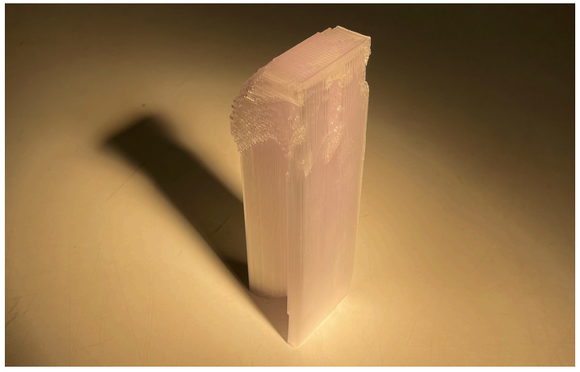
In summary, the journey from 2D to 3D optimization involved overcoming numerous challenges and learning from each iteration. Although the 2D method was ultimately set aside, the lessons learned were crucial in shaping the successful 3D optimization approach. The detailed outcomes of these trials and the evolutionary process of the algorithm are demonstrated on the subsequent pages.

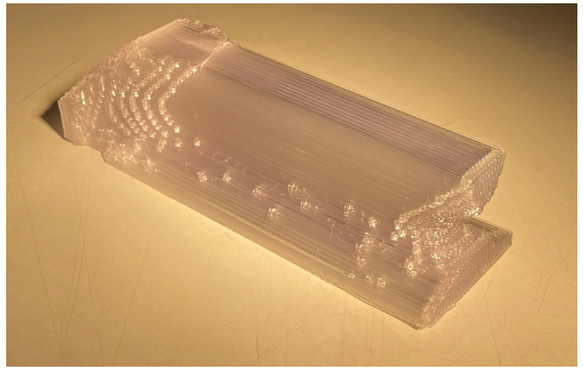


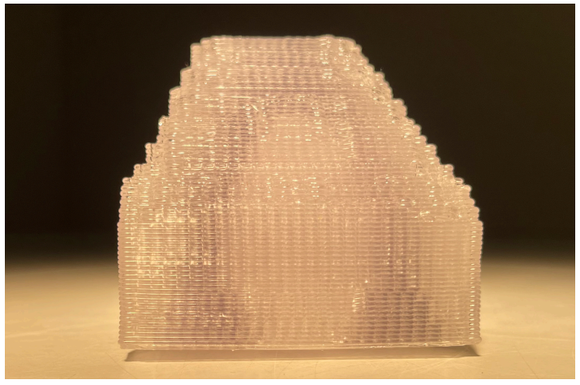
Printed TO in PET-G (P3)

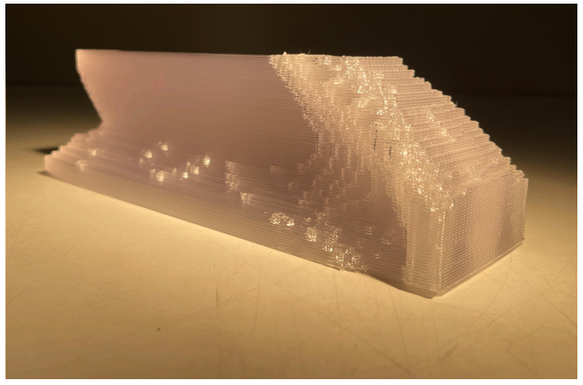
To discuss the progress, trials were conducted to print a final prototype for the P3. These trials served as an example of what the algorithm could achieve. Although some earlier attempts failed, valuable lessons were learned from these steps, informing subsequent prototype prints. Despite the optimization outcome looking completely different, the printing shape settings were maintained after these trials for P3.

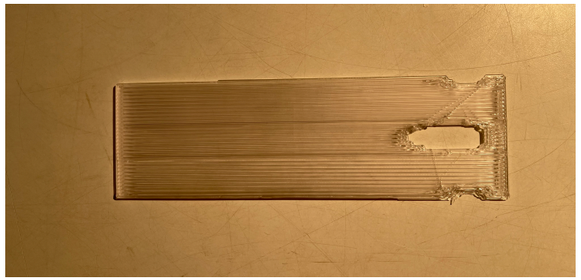


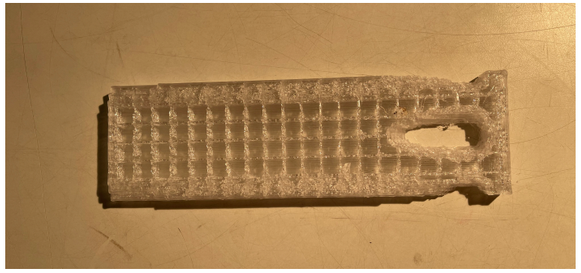


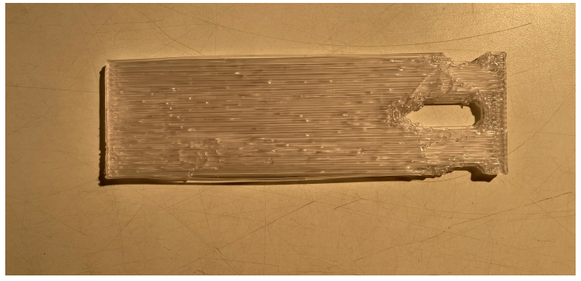


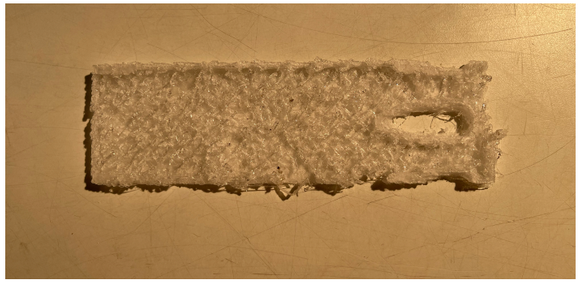


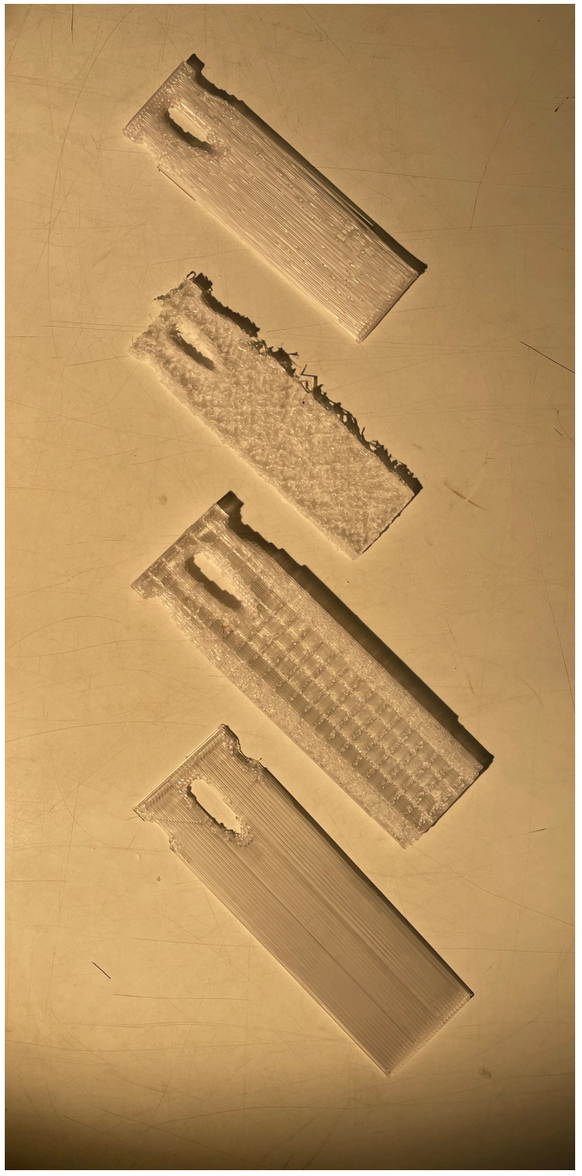




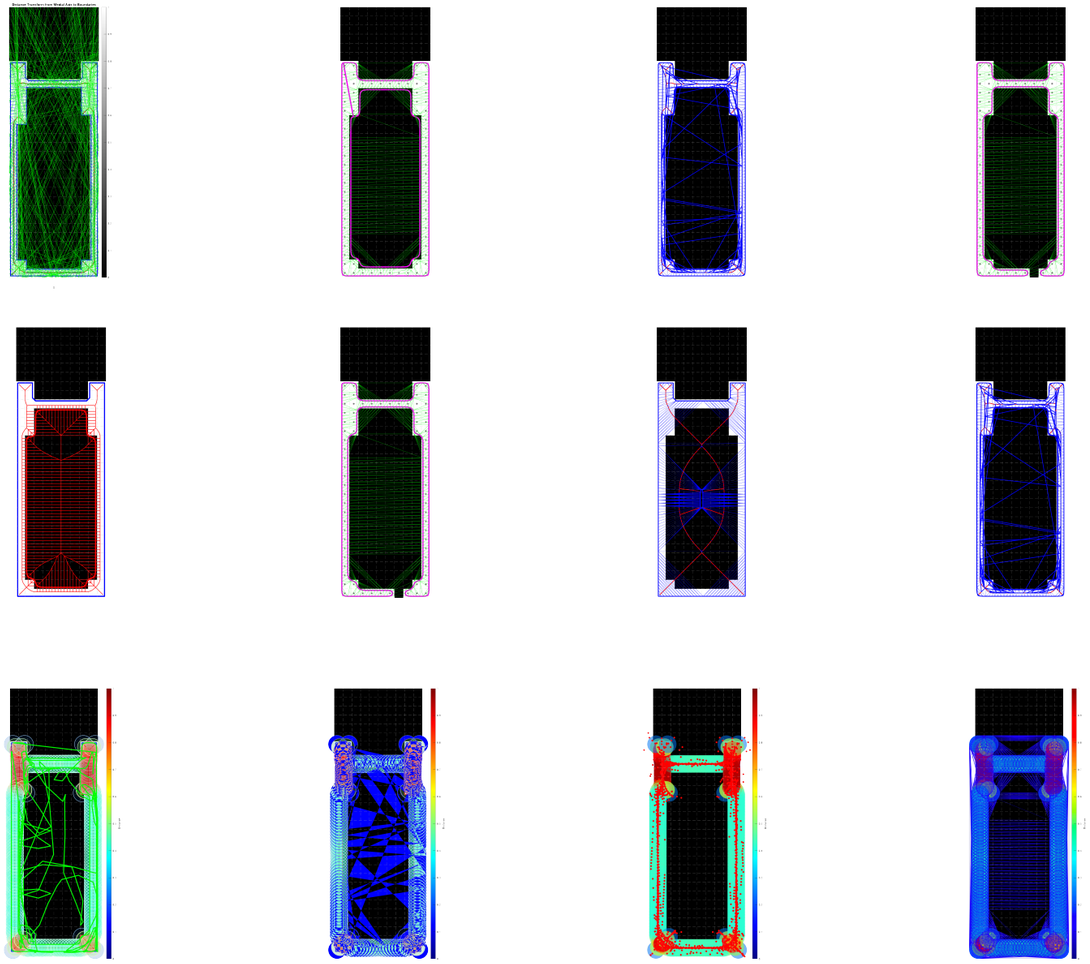


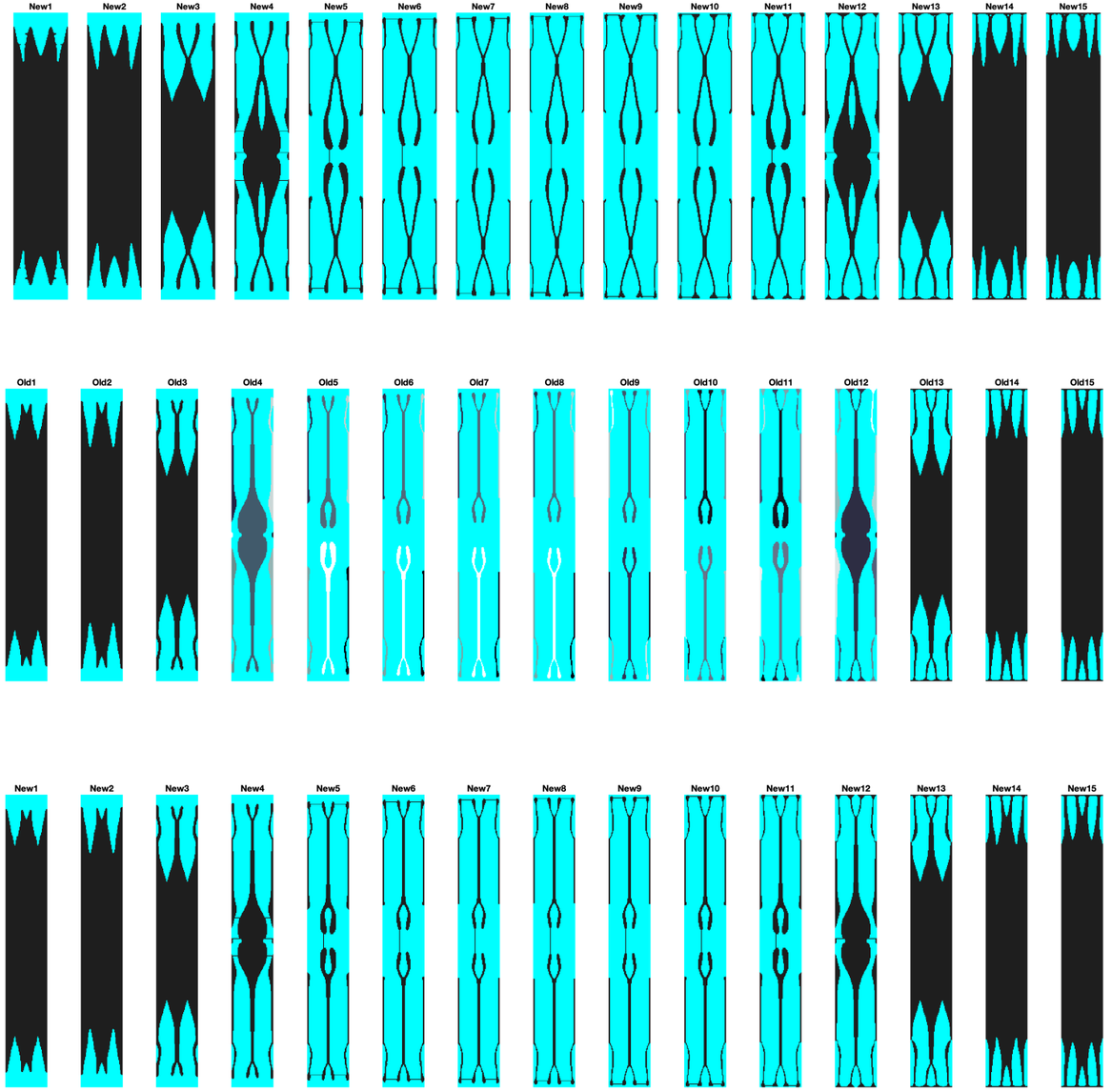




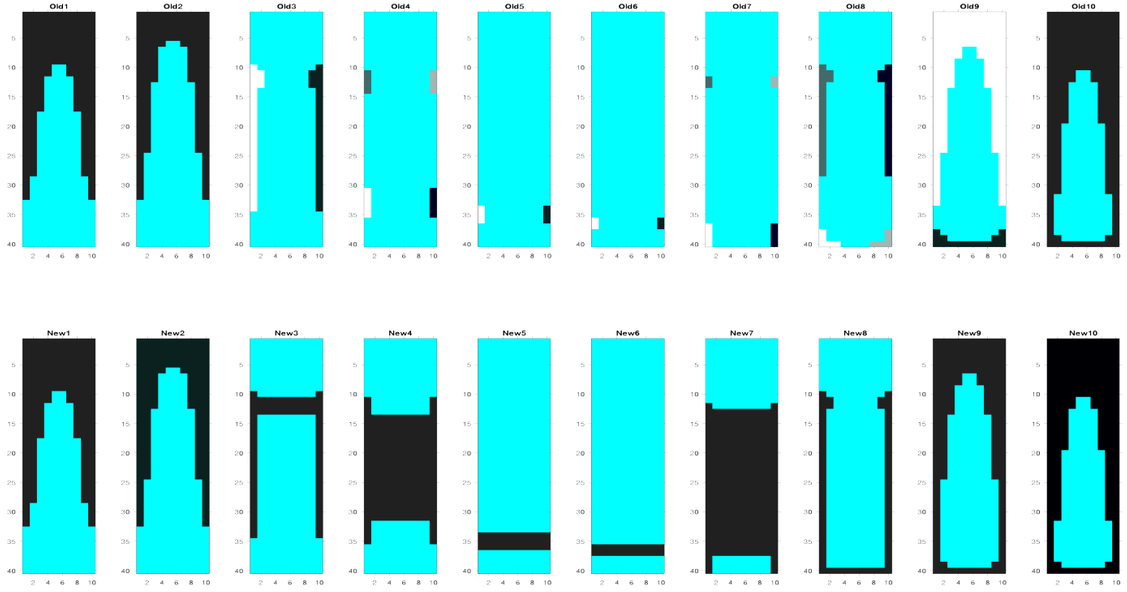


Path offset nozzle trials with Matlab

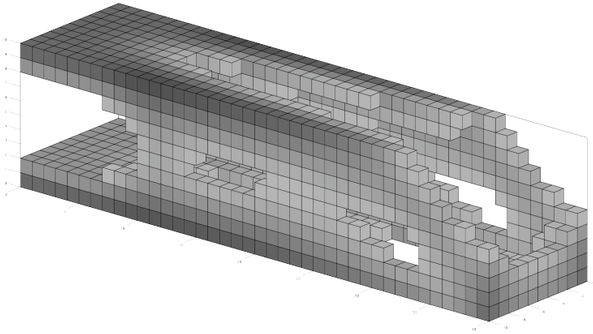




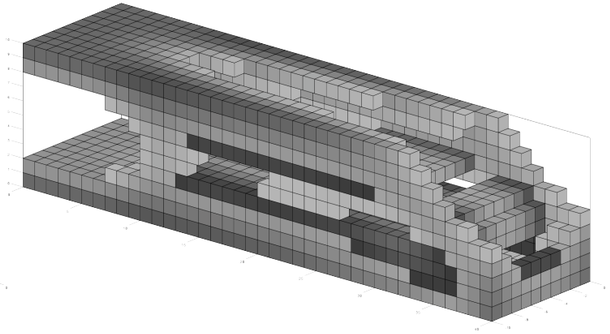
overview of connection with low volume by layers Topology Optimization (MBB beam) clearly visuable that this method is prone to filling the coomplete area 182% volume needed for reconnection



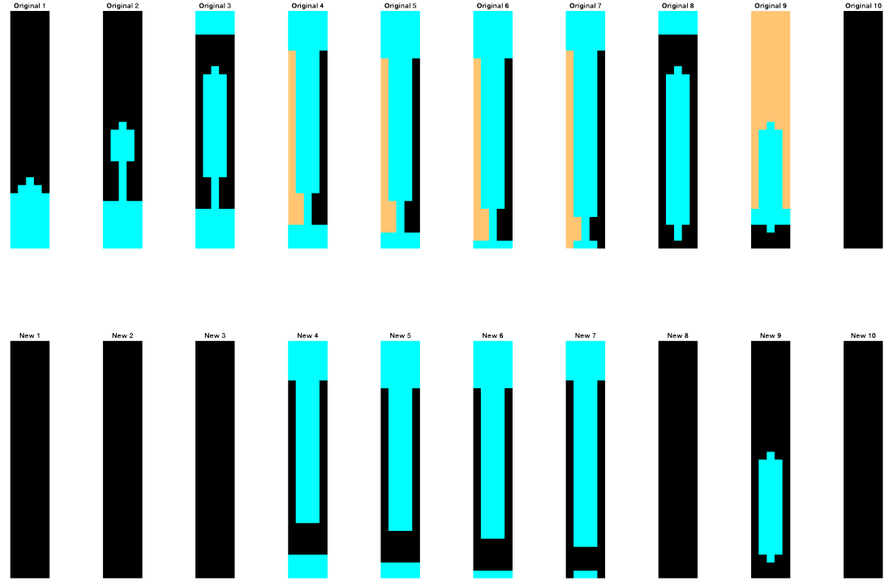
Not connected Topology Optimization (MBB beam) volume 0.3

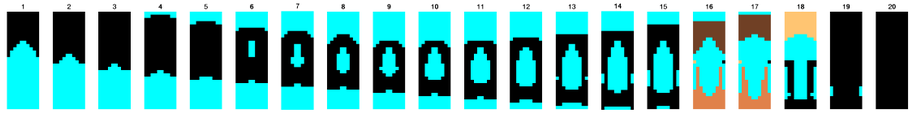


connected Topology Optimization Dijkstra (MBB beam) volume 0.364



Trial and error getting the MATLAB connection working, first it was connecting to all the ‘islands‘ in the layer therefore the becoming black of the items





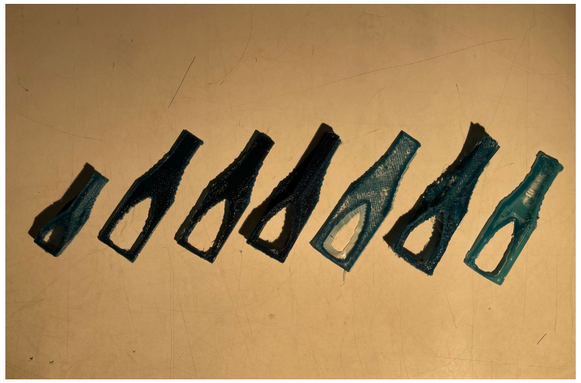
Printed objects wax model for glass model

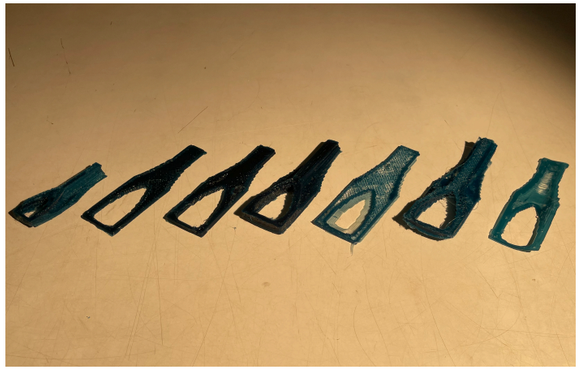
Lost-wax casting, an ancient technique for creating intricate metal objects, is often utilized with 3D-printed geometries due to the complexity and cost of 3D-printing metal. While PLA plastic can be used for this purpose, it evaporates during the ‘burnout’ process, potentially leaving residue in the mold. MachinableWax offers an alternative wax filament that claims to leave minimal residue and can be removed at lower temperatures. However, it requires careful printing due to its flexibility and tendency to shrink upon cooling, leading to adhesion issues and deformation. Printing settings needed considerable adjustment, with trial and error.

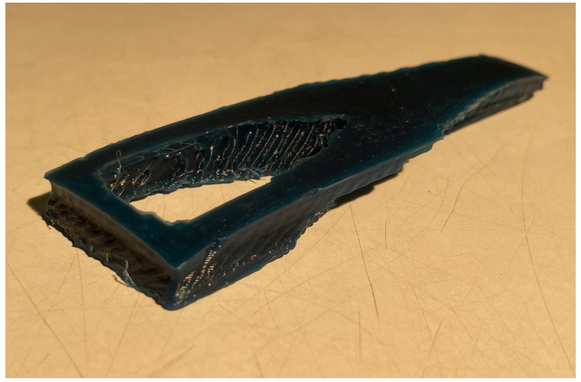
Test samples demonstrate the impact of different settings on print quality. The wax filament is best suited for printers with direct extruders to prevent jamming caused by its flexibility. A heated bed is crucial for adhesion and reducing shrinkage during printing.

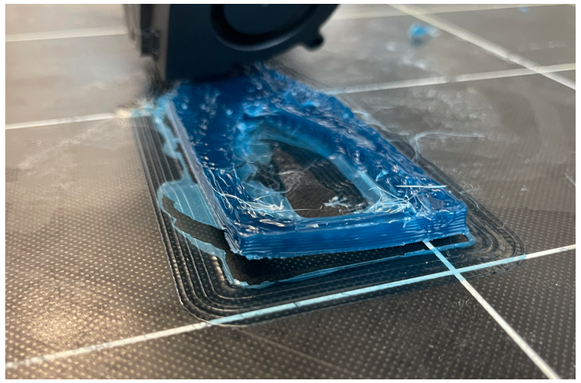
Ensuring proper adhesion between printing layers posed challenges, experimentation with different layer heights, speeds, and temperatures. The most effective combination was found to be 0.2 mm layer height using a 0.3 mm nozzle at 180°C, despite exceeding the recommended temperature. Utilizing 125% material extrusion and a low speed of 20 mm/s proved beneficial, while omitting cooling improved layer adhesion. However, model warping occured at every temperature, even with using a bed temperature of 100°C and several ‘glue‘ layers. Also with low infills and small outer edge, the shrinkage occurs.

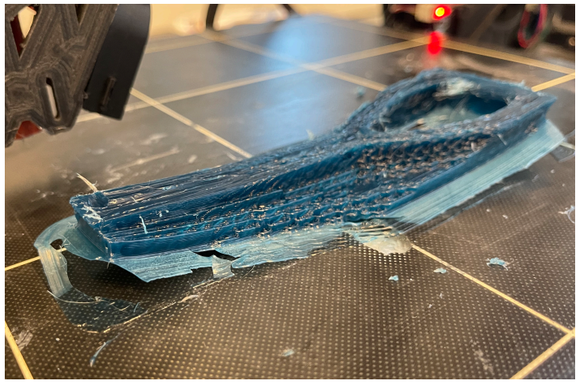
Therefore, this method was changed after these test to print in small steps in height and latter ‘glue‘ them together with candle wax to create so the total shape of the element.

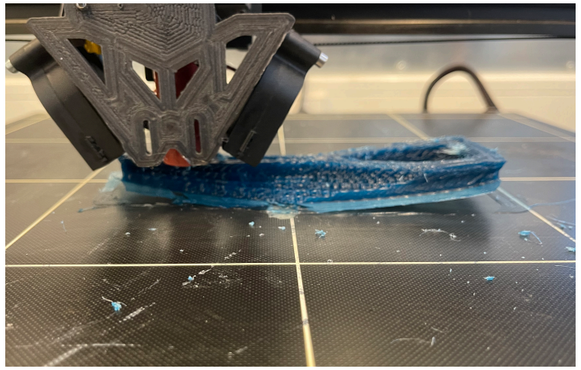


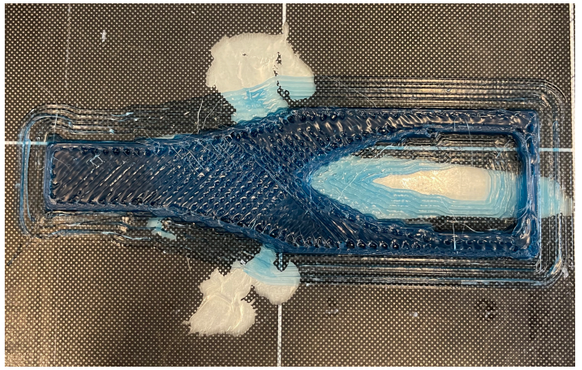


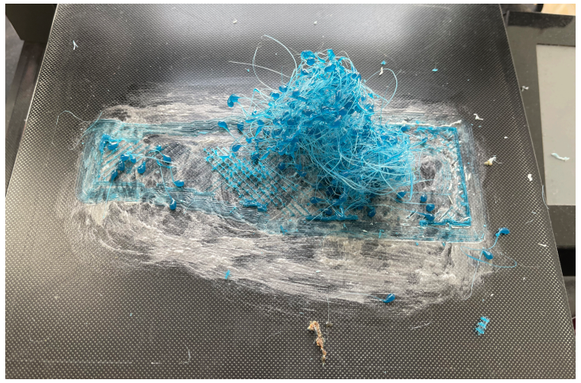


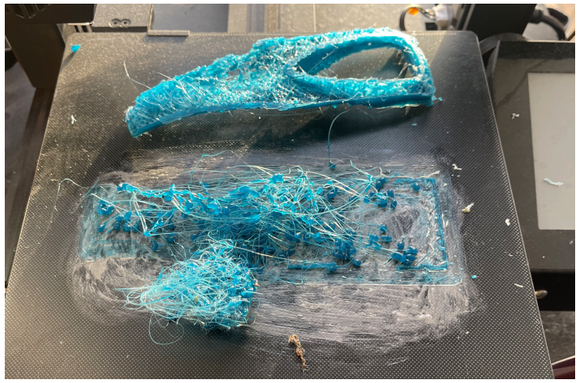




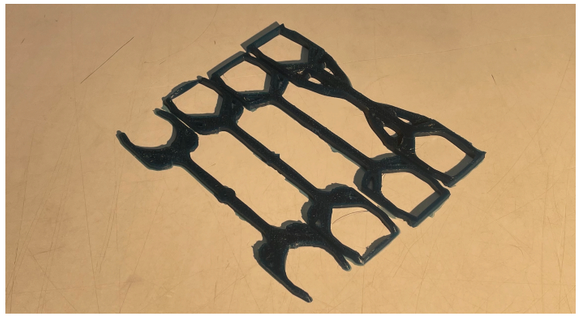
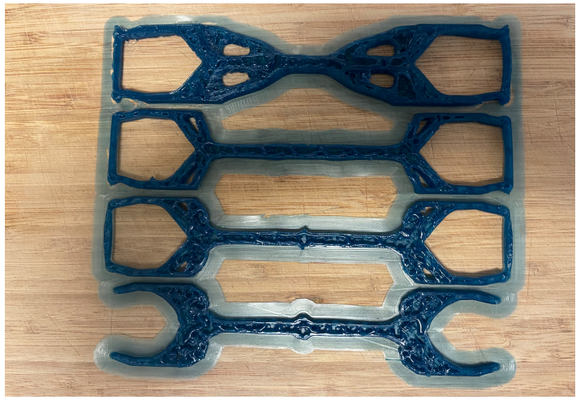
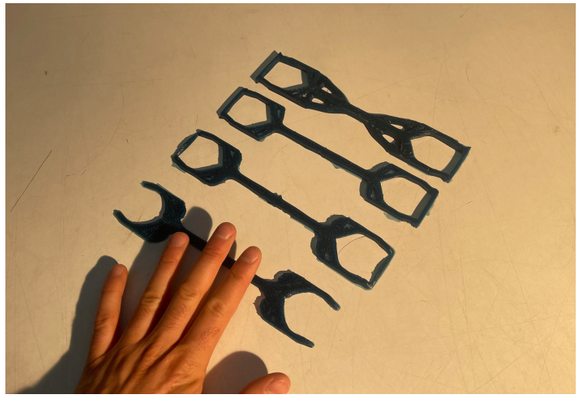
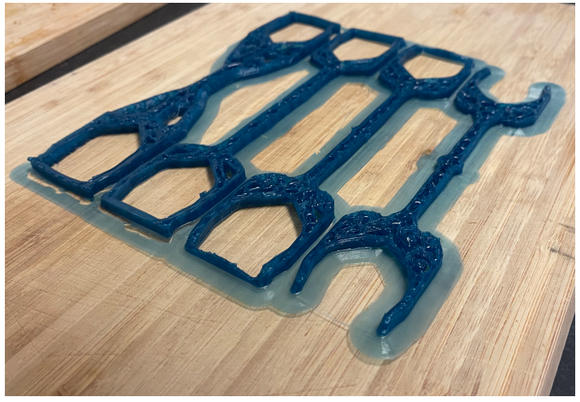


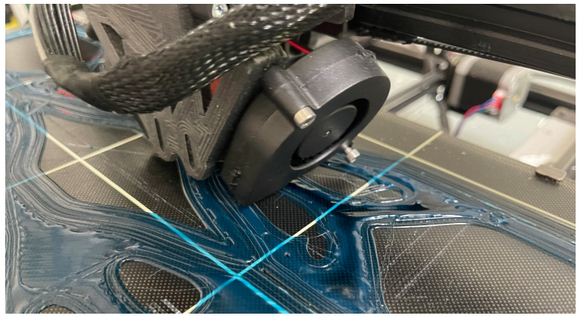
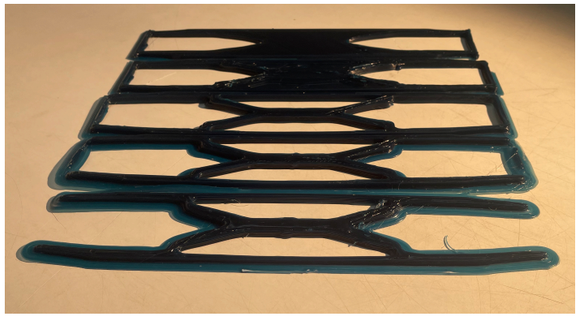
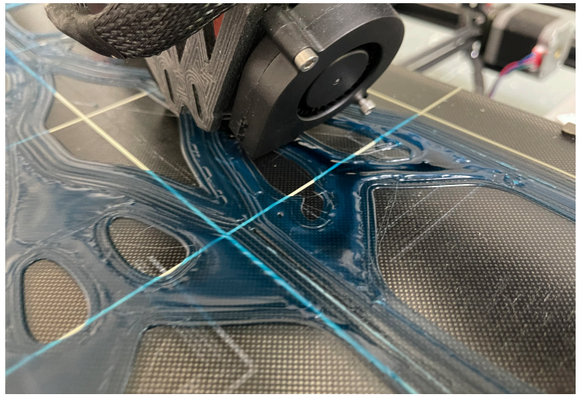
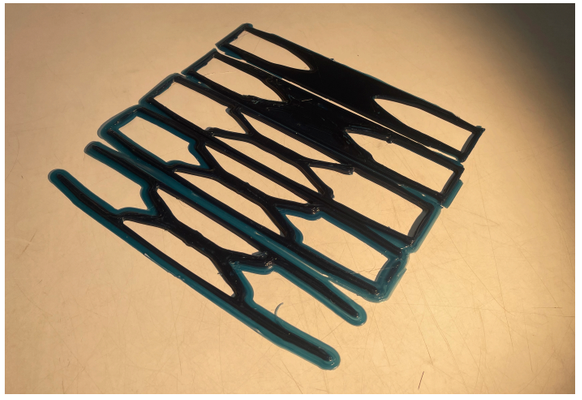


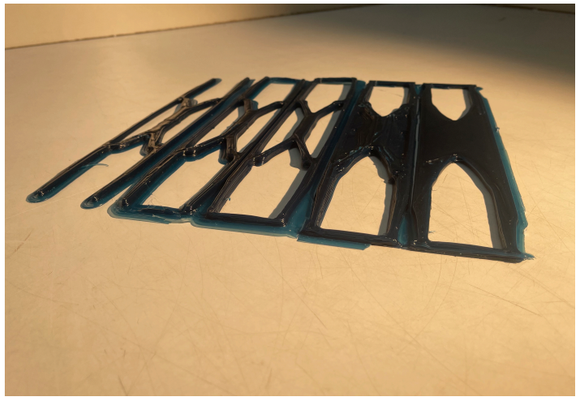




Printed objects wax model for glass model

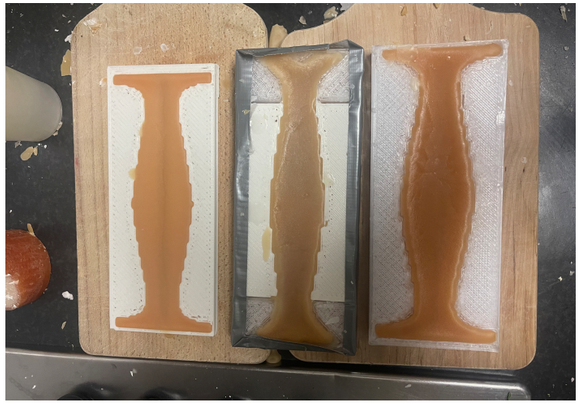



Printed candel wax in PLA molds

Additionally, an alternative method for casting wax was experimented with, utilizing PLA as a mold. After testing three different setups at home, this approach was discarded due to the shrinkage of the candles and the difficulty in releasing them from the rigid PLA mold. Interestingly, the pieces that were successfully cast revealed a remarkable level of detail, despite these challenges.

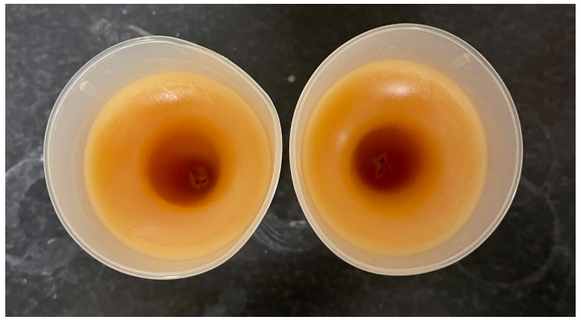




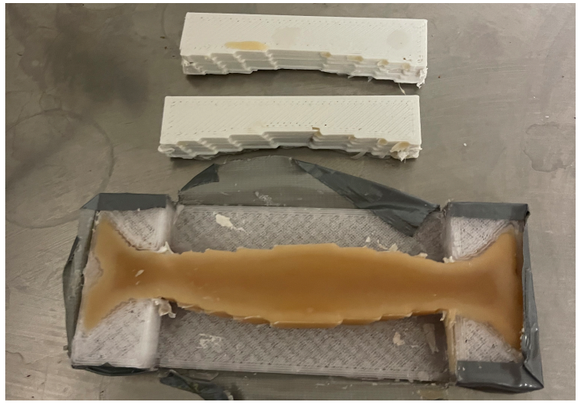




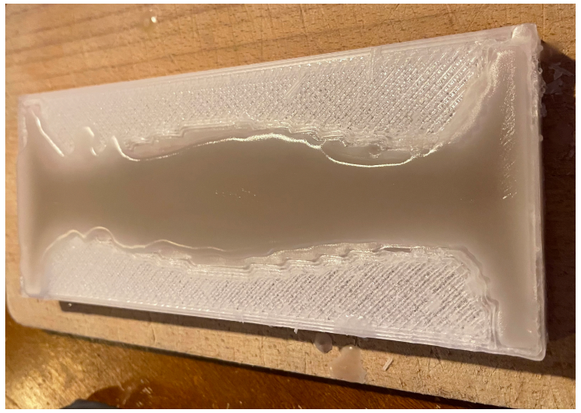








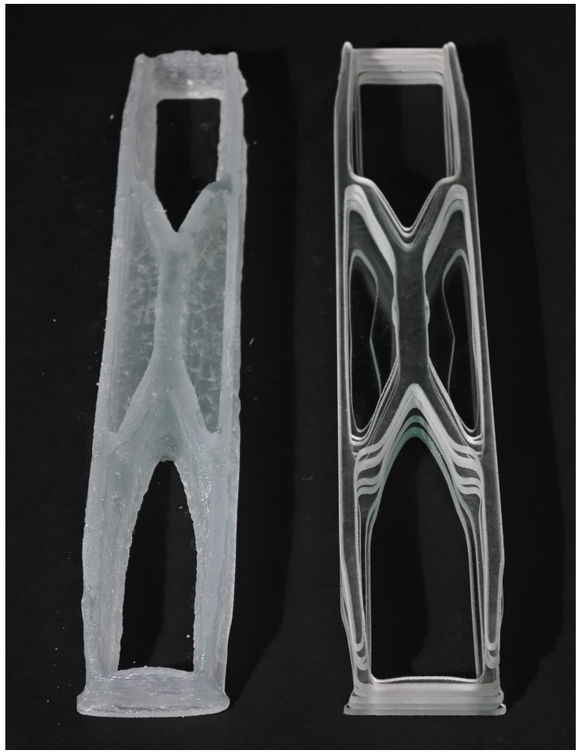


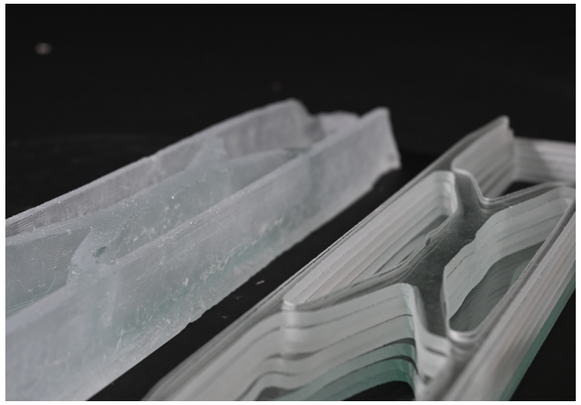




Glass





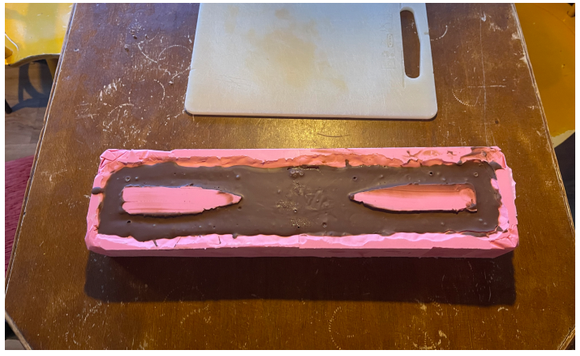






Chocolate in silicon mold

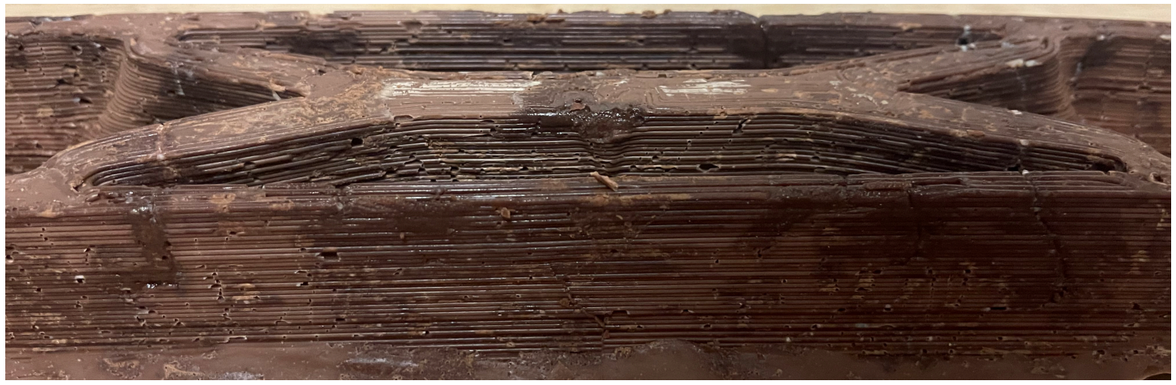






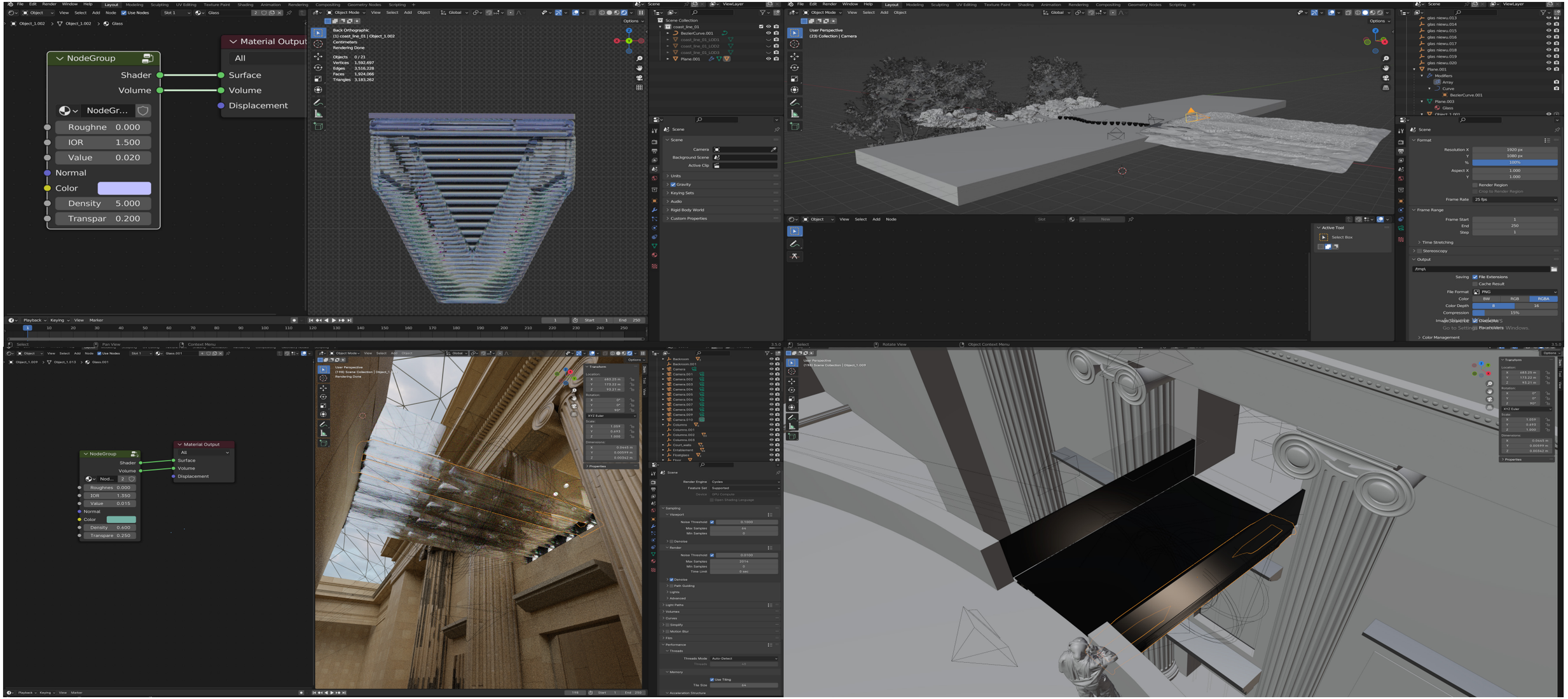




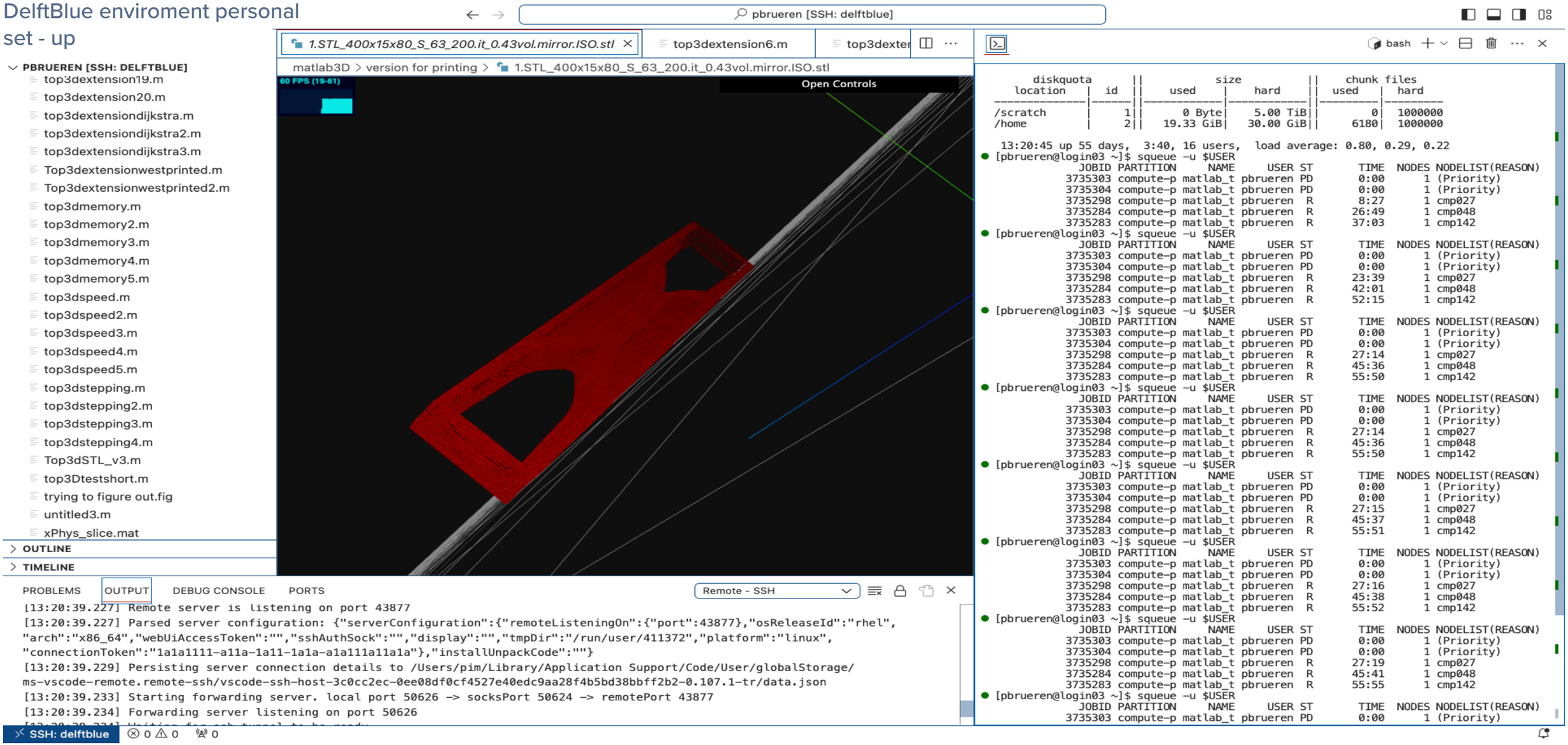




Blender (rendersoftware)



DelftBlue enviroment personal set - up



*Thanks for joining me on this **adventure** through the development of an **algorihm** for*
3D printed glass