

Topology Optimization of Concrete Structures for 3D Printing Applications

An Exploration of the Material-Structure Relationship in Design

A Master's Thesis

Seth Engel

University of Virginia

Department of Engineering Systems and Environment

April 20th, 2020

Abstract

Concrete 3D printing (3DCP) is an additive fabrication technique poised to revolutionize the construction industry. In order to realize the full potential of 3DCP, several unique challenges must be overcome, most notably, that of structural instability during the printing process. As an extrusion process, the cementitious building material used for 3DCP must display fluid characteristics when pumped through a nozzle, but must immediately behave as a solid upon deposition in order to support its self-weight. Therefore, the yield shear stress of the material, which defines the boundary between solid and fluid behavior, plays a significant role in the design of 3D printable structures, both in terms of material selection and viable design. The work presented here seeks to better understand these relationships between material, structure, and fabrication process through the use of a computational design tool known as topology optimization, which determines mathematically the optimal material distribution within a design space under given conditions and constraints. By simulating the self-weight only loading and fixed-based boundary conditions of the 3DCP fabrication process, and by considering material yield shear stress as a constraint on design, an optimization problem can be posed for which the solution is a structure that can be successfully fabricated without collapse. To that end, a novel shear stress-constrained topology optimization algorithm was developed and used to generate optimal structural designs for 3DCP fabrication. The resulting structures demonstrate the effect of material yield shear stress on optimal design and indicate an available range of material properties suitable for 3DCP fabrication. This development of material property ranges and design sensitivities can be used for material design, for determination of available or necessary structural design conditions, or for tailoring the fabrication process to achieve printable structures.

Keywords: Concrete 3D printing; topology optimization; yield shear stress; self-weight loading; shear stress constraint

Table of Contents

ABSTRACT	I
TABLE OF CONTENTS	II
FIGURES	IV
TABLES	V
ABBREVIATIONS	VI
ACKNOWLEDGEMENTS	VII
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: LITERATURE REVIEW	4
2.1 3D CONCRETE PRINTING	4
2.1.1 BACKGROUND	4
2.1.2 CURRENT PRINTING PROCESSES	5
2.1.4 DIGITAL METHODS	9
2.1.5 PHYSICAL METHODS	9
2.1.6 MATERIAL PROPERTIES	10
2.1.7 MIX DESIGN OF 3DCP CEMENT MORTAR	10
2.1.8 HARDENED STATE PROPERTIES OF CEMENT MORTAR FOR CONCRETE 3D PRINTING	12
2.1.9 FRESH STATE PROPERTIES OF 3DCP CEMENT MORTAR	12
2.1.10 YIELD SHEAR STRESS	16
2.1.11 CURRENT RESEARCH DIRECTIONS	17
2.2 TOPOLOGY OPTIMIZATION	18
2.2.1 SIMPLIFIED ISOTROPIC MATERIAL WITH PENALIZATION (SIMP)	19
2.2.2 EVOLUTIONARY STRUCTURAL OPTIMIZATION	21
2.2.3 LEVEL-SET	21
2.2.4 REQUIREMENTS	22
2.2.4.1 Geometry	22
2.2.4.2 Boundary Conditions	23
2.2.4.3 Loading	24
2.2.4.4 Define Material Properties	24
2.2.4.5 Iterations	25
2.2.4.6 Finite Element Analysis (FEA)	25
2.2.4.7 Objective Function	26
2.2.4.8 Constraint Functions	26
2.2.4.9 Sensitivity Analysis	27
2.2.4.10 Optimization	28
2.2.4.11 Filtering	30
2.2.4.12 Termination Criteria Checks	31
2.2.4.13 Results	31
2.2.5 APPLICATIONS	32
CHAPTER 3: POSING THE PROBLEM	34
CHAPTER 4: PROOF OF CONCEPT STUDY	37
4.1 ADDITION OF SELF-WEIGHT LOADING	38
4.2 SHEAR STRESS CONSTRAINTS AND NEW OPTIMIZATION CRITERIA	40
4.3 INCORPORATION OF SI UNITS	42

4.4	RESULTS AND DISCUSSION	42
4.4.1	EFFECT OF SELF-WEIGHT	42
4.4.2	VALIDATION OF OPTIMIZED RESULTS	44
4.4.3	OPTIMIZATION LIMITATIONS	48
4.4.4	THE MATERIAL-STRUCTURE RELATIONSHIP	50
CHAPTER 5: DEVELOPMENT OF SHEAR STRESS-CONSTRAINED TOPOLOGY OPTIMIZATION (SSTO) CODE		53
5.1	GEOMETRY	54
5.2	BOUNDARY CONDITIONS	56
5.3	LOADING	57
5.4	MATERIAL PROPERTIES	59
5.5	ITERATIONS	59
5.6	FEA	59
5.7	OBJECTIVE FUNCTION	65
5.8	CONSTRAINT FUNCTION	66
5.9	SENSITIVITY ANALYSIS	67
5.10	OPTIMIZATION	69
5.11	FILTERING	70
5.12	TERMINATION CRITERIA CHECKS	70
5.13	OTHER	71
5.14	BROAD OVERVIEW OF PROGRAM ORDER	72
CHAPTER 6: VALIDATION PROCEDURES		74
CHAPTER 7: RESULTS		76
7.1	HARDENED STATE	76
7.1.1	VMS-CONSTRAINED HARDENED STATE WALL (TABLE 10)	82
7.1.2	VMS-CONSTRAINED HARDENED STATE COLUMN (TABLE 11)	82
7.1.3	SS-CONSTRAINED HARDENED STATE WALL (TABLE 12)	83
7.1.4	SS-CONSTRAINED HARDENED STATE COLUMN (TABLE 13)	84
7.2	FRESH STATE CASE	85
CHAPTER 8: DISCUSSION		89
8.1	EFFECT OF YIELD SHEAR STRESS ON OPTIMAL DESIGN	89
8.2	AVAILABLE MATERIAL PROPERTY RANGE	90
8.3	DESIGN SENSITIVITY	90
CHAPTER 9: FUTURE WORK		93
9.1	MATERIAL PROPERTIES	93
9.2	REAL-WORLD APPLICATION AND TESTING	94
9.3	THE MATERIAL – STRUCTURE RELATIONSHIP IN DESIGN	95
CHAPTER 10: CONCLUSION		96
APPENDIX A: STIFFNESS MATRIX COMPARISON		97
APPENDIX B: TABLES OF NUMERICAL RESULTS		103
APPENDIX C: PROOF OF CONCEPT STUDY FULL MATLAB CODE		107
APPENDIX D: SSTO FULL MATLAB CODE		110
REFERENCES		124

Figures

Figure 1: Powder-based printing process.....	5
Figure 2: D-Shape printed structures.....	5
Figure 3: Contour Crafting process.....	6
Figure 4: Contour Crafting printed structures.....	7
Figure 5: Concrete Printing systems and structures in the academic setting.....	8
Figure 6: Concrete Printing systems and structures in the commercial setting.....	8
Figure 7: Digital design process (top) and preparation (bottom).....	9
Figure 8: Concrete Printing systems.....	10
Figure 9: Structural rebuilding rate shown as yield shear stress over time.....	13
Figure 10: Bingham Model for non-Newtonian fluids.....	14
Figure 11: Yield stress evolution.....	15
Figure 12: Yield shear stress graphical definition for viscoelastic materials.....	16
Figure 13: Cantilever beam subject to a uniform distributed load on the top surface.....	18
Figure 14: Cantilever beam cross sections showing progressive material removal.....	19
Figure 15: Example problem rectangular prism geometry.....	23
Figure 16: Example problem cantilever boundary conditions.....	23
Figure 17: Example problem loading at the bottom right edge of the design space.....	24
Figure 18: Example problem optimized structure.....	32
Figure 19: Nodal distribution of self-weight from each element.....	39
Figure 20: a. Shear stress constraint optimization logic; b. VMS constraint optimization logic.....	40
Figure 21: Geometry, loading, and boundary conditions for Case 1.....	42
Figure 22: Case 1 optimized geometry with varying material density.....	43
Figure 23: Case 2 optimized structures.....	44
Figure 24: Case 3 a. Optimized geometry and shear stress distribution; b. ANSYS model shear stress distribution.....	46
Figure 25: Case 3 selected locations for shear stress comparison.....	47
Figure 26: Case 4; a. Optimized geometry and shear stress distribution; b. ANSYS model shear stress distribution.....	48
Figure 27: Optimization progression for Case 5 at various iterations.....	49
Figure 28: Ashby diagram of Density vs. Tensile strength of engineering materials.....	50
Figure 29: Volume fraction of case 3 optimized structures vs. tensile strength for materials in Table 5.....	51
Figure 30: Chart of Young's Modulus vs. Tensile strength of engineering materials.....	52
Figure 31: SSTO element IDs.....	54
Figure 32: SSTO global node IDs.....	55
Figure 33: SSTO local node IDs.....	55
Figure 34: Boundary Conditions. a. BC1; b. BC2; c. BC3.....	56
Figure 35: Applied loading conditions. a. ALC1; b. ALC2; c. ALC3; d. ALC4.....	57
Figure 36: Finite element direct stiffness model.....	60
Figure 37: Local coordinate system.....	60
Figure 38: 2x2x2 Gauss Quadrature points.....	63
Figure 39: Optimization processes: a. Setup; b. Main iterative loop.....	73
Figure 40: MMA validation. Compliance minimization subject to 0.3 volume fraction constraint.....	74
Figure 41: VMS-constrained optimized structures. a. SSTO; b. PTO; c. ANSYS.....	75
Figure 42: Wall and column geometry, loading, and boundary conditions for hardened and fresh state cases.....	76
Figure 43: Comparison of VMS-constrained and SS-constrained results for a single loading and constraint.....	84
Figure 44: SS-constrained fresh state wall showing disconnectivity (Run 7).....	85
Figure 45: FEA results for SS-constrained fresh state optimization.....	88
Figure 46: Comparison of SS-constrained hardened state wall results and common bridge designs.....	89
Figure 47: Design sensitivity of shear stress-constrained hardened state wall.....	91
Figure 48: Comparison of design sensitivities for SSTO SS-constrained hardened state wall and PTO cantilever beam.....	91

Tables

Table 1: Numerical results from Case 1.....	43
Table 2: Case 2 numerical results.....	44
Table 3: Case 5 shear stress comparison. All values in MPa.....	47
Table 4: Numerical results for Case 5.....	49
Table 5: Properties of materials in Figure 28. Strength values and moduli given in MPa.....	50
Table 6: Gauss Quadrature Coordinates.....	64
Table 7: Local nodal coordinates.....	64
Table 8: Hardened concrete material properties (ANSYS).....	75
Table 9: VMS Test Results.....	75
Table 10: VMS-Constrained Hardened State Wall.....	78
Table 11: VMS-Constrained Hardened State Column.....	79
Table 12: Shear Stress-Constrained Hardened State Wall.....	80
Table 13: Shear Stress-Constrained Hardened State Column.....	81
Table 14: VMS-constrained hardened state wall loading and constraints.....	82
Table 15: VMS-constrained hardened state column loading and constraints.....	83
Table 16: SS-constrained hardened state wall loading and constraints.....	83
Table 17: SS-constrained hardened state wall case including unreachable constraint.....	83
Table 18: SS-constrained hardened state column loading and constraints.....	84
Table 19: SS-constrained fresh state wall loading and constraints.....	85
Table 20: SS-constrained hardened state wall with constant loading.....	91
Table 21: VMS-Constrained Hardened State Wall.....	103
Table 22: VMS-Constrained Hardened State Column.....	104
Table 23: Shear Stress-Constrained Hardened State Wall.....	105
Table 24: Shear Stress-Constrained Hardened State Column.....	105
Table 25: Shear Stress-Constrained Fresh State Wall.....	106

Abbreviations

3DCP	Concrete 3D Printing (Concrete Printing)
ACI	American Concrete Institute
BC	Boundary condition
BESO	Bi-directional Evolutionary Structural Optimization
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CC	Contour Crafting
CCSA	Conservative convex separable approximations
DOF	Degrees of Freedom
EID	Element Identification
ESO	Evolutionary Structural Optimization
FEA	Finite Element Analysis
FR	Fiber reinforcement
GCMMA	Globally Convergent Method of Moving Asymptotes
GFRC	Glass fiber reinforced concrete
GPa	Giga-Pascal
GQ	Gauss Quadrature
KKT	Karush, Kuhn, and Tucker
KPa	Kilo-Pascal
m	meter
MMA	Method of Moving Asymptotes
MPa	Mega-Pascal
N	Newton
NFRC	Natural fiber reinforced concrete
NID	Node identification
OC	Optimality Criteria
PTO	Proportional Topology Optimization
SFRC	Steel fiber reinforced concrete
SIMP	Solid Isotropic Material Penalization
SNFRC	Synthetic fiber reinforced concrete
SP	Superplasticizer
SPQ	Sequential quadratic programming
SS	Shear stress
SSTO	Shear Stress-Constrained Topology Optimization
STL	Stereolithography
TO	Topology Optimization
VMS	Von Mises stress

Acknowledgements

This work would not have been possible without the academic, financial, and emotional support of many people, to whom I am forever indebted. I would first and foremost like to thank my academic advisors, Dr. Ji Ma and Dr. Osman Ozbulut, for their confidence in my abilities, for their generous financial support, and for their thoughtful guidance throughout my Master's of Science research efforts. I would also like to thank Dr. Jose Gomez, my Master's of Engineering advisor, for his kindness, his careful instruction, and his dedication to his students.

I am honored to have worked with and learned from Dr. Marek-Jerzy Pindera, whose Strength of Materials course propelled me towards the field of structural engineering, and in whose office I spent countless hours discussing mechanics, rigorous mathematics, society, politics, and history. It is rare for one to come across a truly brilliant person, and I am privileged to have studied with such a man.

I am tremendously grateful to Dr. Bill Thurneck for his help and advice when deciding to pursue graduate studies in engineering, and to Dr. Brian Smith for his leadership, his openness, and his willingness to take a chance on a history major with no engineering experience. These two individuals opened the door to my future, through which I simply had to walk.

I want to recognize the students of structures group at the University of Virginia – a group that, while small, includes some of the brightest, most dedicated and friendly people I have had the pleasure of working with. I want to thank Heze Chen for his constant willingness to help with homework and for his love of basketball, Abdou Ndong for his inspiring dedication and positivity, Muhammad Sherif for his knowledge, experience, and generosity, and Ugur Kilic for his collaborative spirit. Heze, Abdou, Muhammad, and Ugur were the pillars of my academic experience, and I will be forever grateful to them.

It is to my family that I dedicate this work. They have, as a group, been unwavering in their support through all of my life's ups and downs, and especially during my transition into the field of civil engineering, and throughout my graduate studies. I want to thank my father, Daniel, and mother, Wendy, for their belief in me, for their love, and for their advice. I want to thank my step-parents, Carolyn and John for their support, both of me and of my parents. I want to thank my grandma Rozzy for her wisdom, her stories, and her warmth, which gave me strength on many occasions. And finally, I want to thank my sister, Laura Eve, for challenging me, for inspiring me, and for being better and funnier than me. She is, without doubt, my favorite person.

Chapter 1: Introduction

The conventional structural design process is typically linear and sequential. As part of this process, the structure is constrained by the properties of the chosen material, or the available materials are limited by the structure and function, with the fabrication method generally chosen as a result of material and structure combined. With the growing complexity of engineering components, the increased focus on efficiency of design and function, and the vast number of available engineering materials, this linear design process may no longer be sufficient to take full advantage of recent technological advances in materials and design. In many complex problems, the design material is not a static quantity, and can be considered as either an independent or dependent design variable. For example, compliant mechanisms and piezoelectric actuators can be designed and optimized using multiple materials and varying material placements (Gaynor and Guest, 2014; Luo, 2010). Additionally, the fabrication method can limit or expand the available design space for both structure and material. Additive manufacturing techniques have greatly expanded the range of complex geometries and tailored materials that could be incorporated into the fabrication process, while traditional methods such as casting and molding remain limited in scope. Structural design, then, becomes one part of a larger optimization problem, constrained by and optimized for fabrication method, geometry, and material selection.

The need for a new, nonlinear structural design process in the realm of civil engineering has emerged with the advent of additive construction. Additive manufacturing of concrete, or concrete 3D printing (3DCP) as it is commonly known, is a novel construction technique involving the layer-by-layer deposition of concrete or cement mortar. Many private companies and research institutions have invested heavily in this technology, as it is poised to revolutionize the construction industry.

The revolutionary nature of 3DCP can be seen from the potential economic benefits and from the design perspective. Economically speaking, 3DCP can provide significant cost savings for the owner of the structure. The two major costs incurred during the construction process, labor and materials, can both be dramatically reduced. Labor costs would be reduced due to the automation of the concrete pouring process, which is currently labor intensive. Reductions in material costs are a result of the potential for new designs offered by 3DCP. At its full potential, 3DCP could remove the need for formwork, significantly reduce the amount of rebar necessary, and allow for more complex geometries, including voids. The lack of formwork reduces the amount of wood necessary for construction, and complex geometries can reduce the amount of concrete required. The biggest cost savings, however, will come from the elimination of formwork, as steel and formwork currently accounts for the majority of material costs for concrete construction (Holt, 2019).

Though 3DCP has the potential to offer these benefits, major difficulties remain to be overcome before they can be realized. These challenges can be grouped into three main categories: 1. Materials; 2. The printing process; 3. Available design space.

1. Materials

The main issues with regards to materials used for 3DCP derive from the fact that any material used must be able to be pumped through a nozzle and must harden quickly upon being extruded. The cementitious mortar must display rheological properties allowing it to flow through the extruder, but must also maintain enough stiffness and strength in order to not deform under its

self-weight after extrusion. The time-dependent nature of the material properties of fresh state cement must also be considered, as stiffness, strength, and rheology can change over the duration of the printing process. With these potentially changing material properties, layer adhesion can become an issue, depending on how long each layer takes to print. The material properties will affect how each layer is bonded to an adjacent layer, either vertically or horizontally. Additionally, the effect of other chemical processes such as heat of hydration and shrinkage during the printing process is not well understood. Once the printing has completed, fresh state properties must then be related to the mechanical properties of the material in its hardened state in order to understand how the design function under applied loading.

2. The Printing Process

Many aspects of the printing process can be controlled by the user, and can either be selected based on a desired outcome, or can be tailored to the functioning of specific material. These parameters include layer geometry (height, width, thickness), layer cycle time, print speed, and print direction. The effects of setting or adjusting these parameters must be understood and generalized. Additionally, the size of the printer must be considered. Currently, most printers operate on a small scale, with only a few commercial operations capable of fabricating structures on the scale of small houses or bridges. In order to fabricate larger structures, larger printers must be designed.

3. Available Design Space

The design space that is currently available to be fabricated by the 3DCP process is limited to vertical walls or geometries assembled therefrom. This limitation is partly due to the necessity of the design to maintain its shape during the printing process. It is also partly due to fact that the structure must be able to carry applied loading when it is complete. In order to expand this design space, designs must be created that will not deform or buckle under their own self-weight during the printing process, and that do not require reinforcement even in the final structure. Another option that is being explored is to incorporate reinforcement and/or formwork into the 3DCP process.

The issues listed above represent the three components of the nonlinear structural design optimization problem stated in the opening paragraph. However, it is not any one individual component that represents the barrier to design, but the link between them. As will be discussed later, there are cementitious materials that have been shown to print successfully, there exist systems that have produced printed structures, and there is a growing library of designs that have been constructed using these materials and systems. But the link between these components is not yet understood. It is this link that will allow for the prediction of printable designs, the development and selection of proper materials, and the necessary adjustment of printing parameters to meet structural and material requirements, therefore opening the door for future material design, structural design, and codification for construction.

One design method capable of integrating structure, material, and fabrication is Topology Optimization (TO). Topology optimization is a computational method for determining optimal material placement in a specified design domain subject to given loading/boundary conditions and geometric or material constraints. The process involves minimizing an objective function subject to given constraints, and is expressed mathematically by the following set of statements and inequalities:

$$\left\{ \begin{array}{l} \text{minimize: } f(x) \\ \text{subject to: } g(x) \leq g^{max} \\ 0 \leq x_i \leq 1 \end{array} \right.$$

where $f(x)$ is the objective function, $g(x)$ is the constraint function that must be less than or equal to a maximum value g^{max} , and x_i represents the element density, taking a value of one when material is present and zero when material is absent.

The objective function $f(x)$ can vary, from minimizing the compliance of a component, minimizing weight or volume, or even optimizing heat transfer. Similarly, optimization can be performed under various constraint functions $g(x)$, such as geometry, volume fraction, or strength. The method is flexible and is capable of solving many different engineering design problems. Due to this flexibility, and to its design capabilities, topology optimization is now broadly utilized to refine complex designs, such as aircraft wing box ribs (Tucker, 2004), a Volvo rear lower control arm (Larsson, 2016), and turbomachinery impellers (Meli, 2019). As such, the process has been incorporated into many commercially available finite element analysis software packages, such as Abaqus, ANSYS, and Altair Optistruct.

With proper selection of geometric conditions and design constraints, topology optimization can also be used to predict the relationship between material properties, the fabrication process, and optimal geometry. It is the goal of this work to use topology optimization to develop this relationship for 3DCP applications, in order to design structures that can be successfully fabricated. This goal will be accomplished by identifying material properties that represent limitations imposed both by the fabrication process and on the final design. An optimization algorithm will then be created that incorporates those properties as constraints. The optimization process will then produce designs that can be successfully fabricated using 3DCP techniques without collapsing under its own weight.

Chapter 2: Literature Review

In order to accomplish the goal of this study, it is important first to lay the foundations of knowledge in the fields of 3DCP and TO. From this basis of knowledge, an understanding of fabrication processes, material properties, and the design method will emerge, allowing for identification of requirements and limitations inherent in the physical processes. These requirements and limitations can then be used to formulate, both conceptually and mathematically, a design problem, the solution to which will attempt to achieve the goal of this work.

2.1 3D Concrete Printing

2.1.1 Background

Traditional concrete construction is a complex, multi-step process that occurs both on and off the construction site. Off site, the concrete mix is designed for strength, weight, and workability using a variety of constituent materials. A mixture of cement, water, and fine aggregate comprises the concrete mortar, which is then mixed with coarse aggregate and other chemical additives to form the concrete for construction. This mixture is then produced in a concrete manufacturing facility and transported to the construction site in mixing trucks. Before the trucks arrive, on-site preparations have completed the installation of formwork and steel reinforcement to control concrete placement and to add tensile strength to the structure. The concrete is then pumped into place using a hose guided either manually or by crane, and finished manually. Individual concrete elements such as beams and columns can also be pre-fabricated offsite using similar methods, then transported to the construction site for placement. However, the recent development of Additive Manufacturing (AM) techniques, and their applications to the construction industry, offers the potential to dramatically change the concrete construction process.

The concept of AM began in 1986 with a patent for “rapid prototyping” developed by Chuck Hull (Hull, 1986). Extrusion-based techniques, involving the layer-by-layer deposition of material over time, emerged shortly thereafter, beginning with a patent by Scott Crump for filament deposition (Crump, 1992). Initially developed for polymer fabrication, AM techniques have quickly expanded into production of metals, ceramics, and aggregate-based materials. The seminal study on AM using cementitious materials, now known as 3D concrete printing (3DCP), was published by Joseph Pegna (1997). The experiment involved the deposition of alternating layers of sand and Portland cement, which was cured by steaming. This early method of concrete fabrication led to the development of the three processes in use today: D-Shape, Contour Crafting, and Concrete Printing.

2.1.2 Current Printing Processes

There are three major printing processes that have come to define the concrete 3D printing landscape, both in the commercial and academic realms. These processes are: 1. D-Shape; 2. Contour Crafting; and 3: Concrete Printing.

1. D-Shape

The D-shape process is a “dry” powder-based process that relies on a powder and binder deposition method to create successive layers of build material (Figure 1). Each layer of powder is deposited as a sheet and compacted, after which binder material is deposited through nozzles maneuvered by gantry cranes along the path of the solid component geometry. When the structure is complete, the excess powder is removed and the solid component remains.

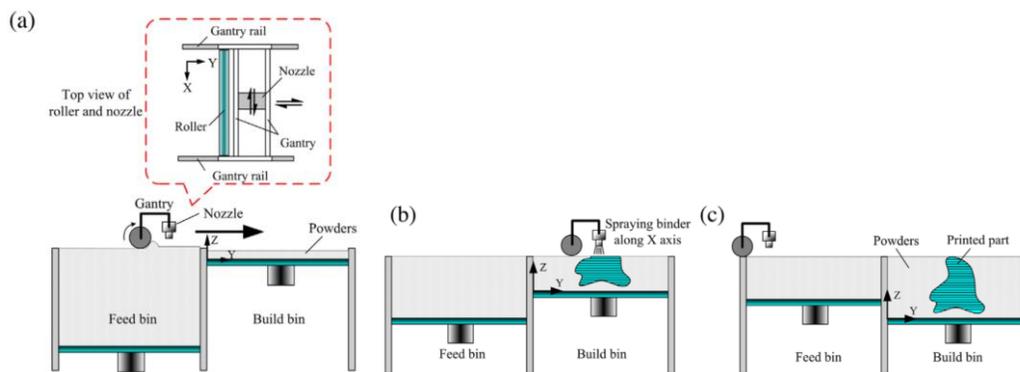


Figure 1: Powder-based printing process: a. before printing; b. during printing; c. after printing. (Feng, 2015)

D-Shape can produce very intricate geometries, more so than other methods (Figure 2), and it allows for the creation of overhangs and freeform features by using the excess powder as a support during the building process. Additionally, D-Shape can incorporate concrete of the highest strength due to the dry process, which does not employ concrete in fluid form. However, D-shape results in an overuse of build material, and suffers from an excess of clean up time. Other problems include penetration of the binder through multiple layers, inaccuracies in layer geometry due to unintended spread of binder material during deposition, and slower printing times. These problems aside, D-Shape has been used to design intricate architectural pieces, small houses, and artificial coral reefs.

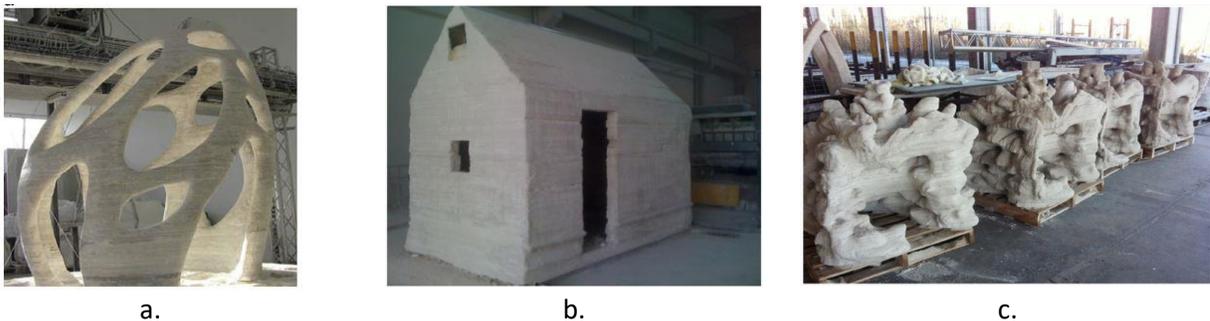


Figure 2: D-Shape printed structures. a. Radiolaria Pavilion (Lim, 2012) b. House (d-shape.com); c. Artificial reef (d-shape.com)

2. Contour Crafting

Contour Crafting (CC) was originally developed as a fabrication process for polymers or ceramic materials such as clay, but has been adapted for use with cement mortars (Khoshnevis, 2001, 2004, and 2006). The process involves the extrusion of slurry through a feeding tube. The surface of the extruded layer is then smoothed using top and side trowels attached to the extruder end (Figure 3).

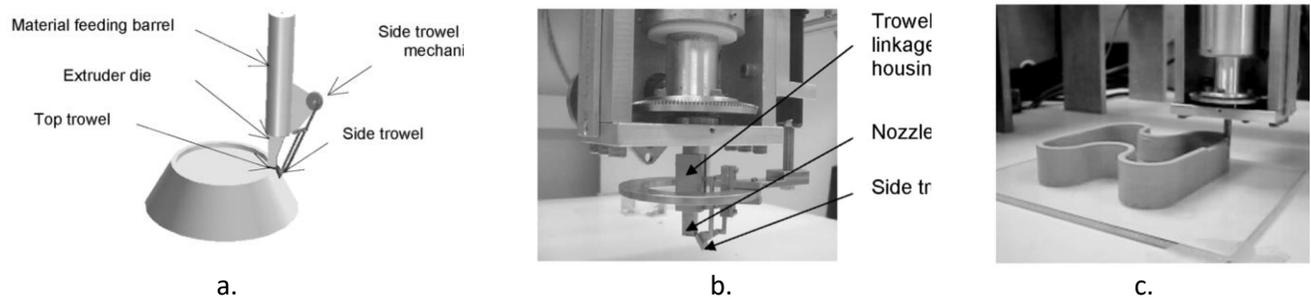


Figure 3: Contour Crafting process. a. CC schematic; b. CC printer head; c. CC printing. (Khoshnevis, 2001 and 2006)

The process allows for faster fabrication speeds, larger nozzle openings, and greater layer depths than other methods (up to 13mm), as the trowels are able to control the spread of the extruded material. These advantages make the CC method well suited for high-speed automated construction. However, complex geometries cannot yet be fabricated with CC due to the poor spatial resolution and significant weight of the layers, and the designs are limited to vertical elements in compression only. If a door or window is to be placed in a structure, a lintel must be used to support the concrete surface to be constructed.

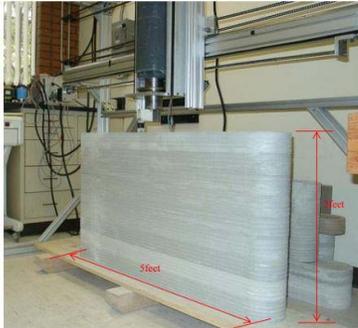
Several commercial enterprises have employed CC for digital concrete construction (Figure 4). WinSun is a Chinese company that has produced several 3DCP structures of note. They constructed an office building for the Dubai Future Foundation using pre-fabricated 3DCP components (Camacho, 2018), and similar methods were used to construct multiple houses and a five-story apartment building (Paul, 2018). Recently, WinSun fabricated isolation houses in a response to the COVID-19 outbreak in Wuhan using 3DCP technology (WinSun, 2020). CyBe Additive Industries is a Dutch company that has devised a six-axis robot-arm printer capable of extruding an in-house mortar with a hardening time of 5 minutes. This printer fabricated the R&Drone Laboratory in Dubai in 46 hours over 3 weeks (CyBe, 2020). San-Francisco-based Apis Cor has developed a crane printer capable of fabrication within a 58m² design base, with no height limitations (Paul, 2018), and has fabricated a 38m² house in Russia.



a.



b.



c.



d.

Figure 4: Contour Crafting printed structures. a. WinSun COVID-19 isolation house (winsun3d.com); b. Cybe R&Drone Laboratory (cybe.eu); c. CC printed wall (Khoshnevis and Hwang 2004); d. 3D-printed home (Apis-cor.com)

3. Concrete Printing

Concrete Printing (3DCP) is a similar process to CC, but does not involve trowels to shape the extruded material. As with CC, the cement mortar is extruded through a feeding tube, which follows the geometric path set by the design. Due to the lack of geometric constraints at the extruding end, the nozzle size is restricted to much smaller sizes, allowing for greater control of the design geometry. 3DCP is the most common printing process used in research settings, and therefore, throughout this work, 3DCP will refer to this printing process. Though it is the most common process, 3DCP does have its drawbacks. Due to the small nozzle, 3DCP requires the longest printing time for each layer, relative to D-Shape and CC. Additionally, the smaller nozzle limits mix design due to constituent particle size, as well as on rheological properties due to the restricted flow outlet.

Researchers at Loughborough University were able to fabricate a bench using the 3DCP technique (Lim, 2009). Also using 3DCP, Salet (2018) at the Eindhoven University of Technology, produced a 6.5x3.5m span bicycle bridge with a depth of 0.92m, by fabricating horizontal components on their side, building them vertically. Researchers at MIT developed a Digital Construction Platform comprised of a track-operated vehicle and several robotic arms, capable of fabricating entire structures, including a dome of 50 feet in diameter and 12 feet in height (Chandler, 2017).



a.



b.



c.



d.

Figure 5: Concrete Printing systems and structures in the academic setting. a. Robotic printing system developed at MIT; b. Wall artifact (S. Lim, 2009); c. and d. Bicycle bridge (Salet, 2018).

World's Advanced Savings Project developed a series of DeltaWASP printers, with the largest capable of fabrication within a 12mx7mx7m design space. These printers are designed to use local materials to print clay, but they have also been used to fabricate cement-based structural beams (Holt, 2019). XTreeE used a 6-axis robotic arm to fabricate a complex wall-element optimized for thermal insulation (Paolini, 2019).



Figure 6: Concrete Printing systems and structures in the commercial setting. Left: XTreeE (Buswell, 2018); Middle: DeltaWASP Printer (3dwasp.com); Right: WASP 3D-printed house (3dwasp.com).

2.1.4 Digital Methods

Common to all three printing methods are the physical and digital methods for depositing material in the desired geometry. Every 3D print begins with a digital design created using Computer Aided Design (CAD) software. The design then proceeds through a series of steps known as “Process Planning” (PP), which prepares the design for fabrication (Livesu, 2017). PP is generally performed by standalone Computer Aided Manufacturing (CAM) software such as Materialize Magics, but certain steps can also be performed by CAD software or by CAM software embedded in some printers. PP begins by checking the design for problem areas, such as those with geometries below the printing resolution. Once the design has been checked and/or fixed, it becomes tessellated divided it into a triangular mesh. The resulting design is then divided into horizontal slices of a thickness designated by the design engineer to correspond with the thickness of the printed material. The resulting contours will be fabricated by material deposition during the printing process. The final step of PP involves designating the machine tool path. This process generates a stereolithography (STL) file, which can be interpreted by embedded software in the 3D printer. Though this is the most common method, there are “direct slicing” methods that skip the tessellation step, and generate slice data in the form of Gcodes, which describe the machine tool path, and can be interpreted by filament (extruder) printers. The digital design process is displayed in Figure 7.

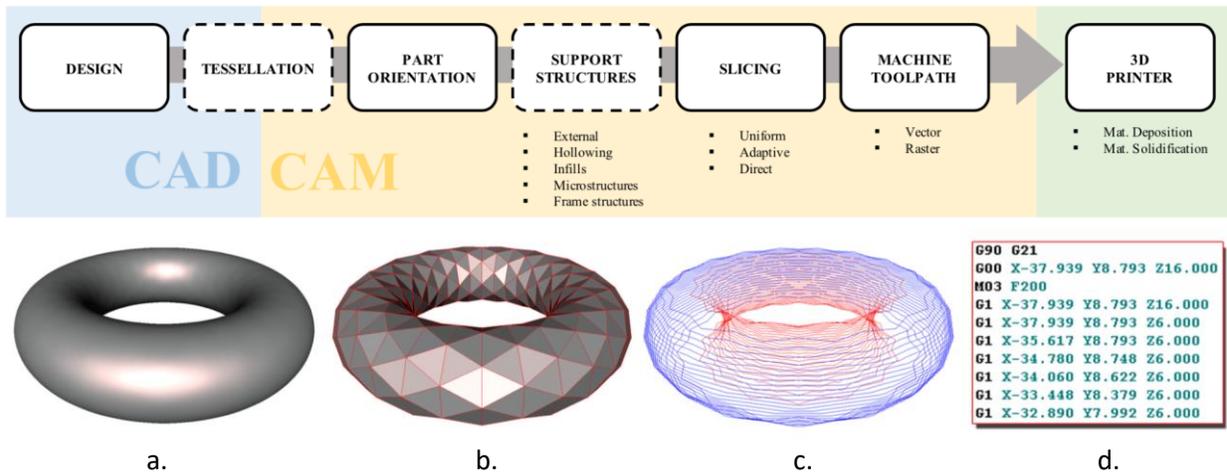


Figure 7: Digital design process (top) and preparation (bottom). a. Original design; b. Tessellated design; c. Sliced design; d. Machine instructions. (Livesu, 2017)

2.1.5 Physical Methods

The printer, or the physical machine that deposits the cementitious material, is composed of the following parts: 1. Intake, 2. Material chamber, 3. Pump, 4. Feeding tube, 5. Extruder, 6. Nozzle, and 7. Multi-axis guidance controller. The cement mortar enters the system through the intake and is stored in the material chamber until it is pumped through the feeding tube, into the extruder, and deposited through the nozzle, which is positioned at the desired location by the multi-axis guidance controller. Currently, there are three categories of guidance controller in use: crane systems, gantry systems, or robots (Figure 8). Crane systems, such as those used by Apis Cor, have the advantage of being vertically adjustable and can encompass very large design spaces. Gantry systems, though they are scalable, can only construct up to a fixed height. WinSun currently has a 40 x 10 x 6.6m gantry system, while TU

Eindhoven employs a system of 11 x 6 x 4m. Robots, such as those designed by Zhang (2018) at Nanyang Technological University, are the most restricted in terms of horizontal and vertical range, but with a potential six axes of movement, they can produce much more detailed designs much quicker than crane or gantry systems.



Figure 8: Concrete Printing systems. Left: Apis Cor crane printer diagram (apis-cor.com); Middle: Eindhoven University of Technology gantry printer (www.tue.nl); Right: Concrete printing robots (Zhang, 2019)

Each system allows the manufacturing engineer to control many print parameters that can affect the resulting structure. These parameters include print head speed, print direction, filament geometry, layer thickness, pump speed, and pump pressure.

Determination of these parameters depends heavily on the geometry of the structure and on the material properties of the fresh concrete mortar. Additionally, these parameters can dramatically affect the resulting mechanical properties of the structure in its hardened state. Therefore, the relationship between material and structure must be well understood in order to successfully construct concrete structures using additive manufacturing techniques.

2.1.6 Material Properties

During the printing process, the cement mortar must have the properties of a fluid in order to be extrudable, but immediately after printing, the material must solidify in order to support its self-weight and the weight of subsequent layers above. Without sufficient solidification or inherent strength, ground layers may compress with layer build-up, and overhangs may sag or collapse due to lack of support, leading to undesirable deformation and overall printing failure. Since the 3DCP technology is still in an early stage of development, the interaction among design variables, printing parameters, and material properties is still unknown. Research into these relationships is ongoing, and addresses three main issues: 1. Mix Design, 2. Fresh State Properties, and 3. Hardened State Properties. It is common for researchers to either investigate the properties of several different mix designs, to study in depth the properties and behavior of a single mix throughout the printing process.

2.1.7 Mix Design of 3DCP Cement Mortar

Concrete, and the cement mortar from which it is derived, comes in many different forms. The traditional ingredients of concrete are water, cement, a fine aggregate usually consisting of sand, and a coarse aggregate typically of rock. The materials engineer has complete control over these input ingredients and their respective volume ratios, and can tailor a specific mixture for a specific project.

For common construction projects, high compressive strength is the goal, with workability, coarse aggregate particle size, weight, and material availability operating as constraints on the design. Shear strength and tensile strength are not considered due to the presence of formwork and reinforcement, and the time-dependent nature of the material properties is not considered, as the concrete can remain restrained by formwork indefinitely. Additionally, rheological properties of fresh concrete are not considered again due to the restraint of formwork.

Mix design for 3DCP is a different problem altogether. Without formwork, rheological properties must be considered, and without steel reinforcement, tensile and shear strength become important. Coarse aggregates are generally not used, as they are often too large for effective extrusion through smaller nozzles and can cause inconsistencies in deposition, but other additives such as superplasticizers (SP), fiber reinforcement (FR), alternative binders, and chemical retarders or accelerators can be introduced in order to achieve the desired material behavior during and after printing.

Superplasticizers are chemical additives used to increase the fluidity of concrete. They function by separating cement particles that can become clumped together, and they are able to do so without increasing the water content. In fact, superplasticizers are often used to decrease water content, allowing for increased early-age strength of the material while maintaining or increasing workability. The increase in fluidity can be attributed to the reduction of yield stress the material, allowing it to flow more readily under the loading imposed by the weight of the material itself. Lignosulfonates, sulfonated naphthalene formaldehyde condensates, and polymelamine sulfonates are some of the more commonly used superplasticizers (Aitcin and Flatt, 2015).

Fiber reinforcement involves the introduction of thin fibers into the concrete mix in small volume fractions. Fibers can range in diameter from 0.0001 to 0.01 in. with an aspect ratio from 40:1 to 300:1. The fibers can be classified into one of four material categories: steel fibers (SFRC), glass fibers (GFRC), synthetic fibers (SNFRC), and natural fibers (NFRC). Many steel, glass, and synthetic fibers have tensile strengths significantly higher than that of hardened cement, and most have greater elastic moduli as well. The fibers are introduced into the mix to mitigate the very low tensile strength offered by unreinforced concrete (Zollo, 1997).

Alternative binders replace the functioning of the commonly used Portland Cement in the formation of the paste that serves as the glue holding together the various aggregates present in the concrete mixture. Alternate binders have the potential to affect strength, durability, and rheology and may be considered depending on the material properties of the binders and of the resulting mix. Four common types of alternative binders are 1. Calcium aluminate cements (CACs); 2. Calcium sulfoaluminate cements (CSAs), 3. Alkali-activated binders such as fly ash; and 4. Supersulfated cements (Juenger, 2011).

Chemical accelerators increase the speed at which the concrete hardens over time by increasing the hydration rate and by reducing the average distance between cement paste particles. This process increases early compressive strength, but may result in the loss of long-term strength. Common chemical accelerators include inorganic salts, soluble organic compounds, setting accelerators such as sodium silicate or aluminum chloride, or solid accelerators such as calcium aluminate or silicate (Aitcin and Flatt, 2015).

Chemical retarders are the opposite of accelerators. They are introduced in order to decrease the speed of hardening, allowing the mixture to remain workable for a longer period of time, resulting in a delayed increase in strength. Use of chemical retarders can delay concrete setting up to 24 hours or more without affecting strength or durability. Common retarders include lignosulfates, salts of carboxylic acid, or sugars (Aitcin and Flatt, 2015).

The effects of various additives on material properties and the printing process are discussed in detail below.

2.1.8 Hardened State Properties of Cement Mortar for Concrete 3D Printing

Hardened state research typically involves the fabrication and curing of 3DCP components and subjecting them to mechanical testing, typically in the form of compression and flexural tests. Most of this characterization is secondary to the development of printable materials, and is therefore presented as data for specific mix designs. A few studies have demonstrated the anisotropic nature of structures fabricated with powder-based techniques, by subjecting hardened samples different mixes to axial and transverse loading relative to print direction, and studying compressive and flexural strengths (Zhang, 2019, Feng, 2018). Feng, (2015) developed a novel stress-strain relationship described by a quadratic formula, and failure criterion were proposed considering the orthotropy resulting from the directional orientation of layers relative to geometric design and loading conditions. Research into the hardened mechanical properties of structures fabricated using CC techniques is not prevalent in the literature, and is an area that requires understanding. Additionally, links between fresh state properties and hardened state properties have yet to be established.

2.1.9 Fresh State Properties of 3DCP Cement Mortar

Fresh state research involves investigations into the behavior of the cement mortar in its fluid state, before it has been cured. It is in the fresh state that the material is pumped, fed, and extruded through the printer. Additionally, the material may continue to display fluid-like behavior for a time after extrusion. As such, both rheological properties and early-age mechanical properties of the cement mortar must be understood and linked to the success and/or failure of printing. Currently, the state of understanding the material properties of fresh state cement mortar is based on the novel concepts of flowability, extrudability, buildability, and printability (Rahul 2019, Paul 2018, Lim 2012). As will be shown below, these concepts have not been standardized as quantitative measures, either in terms of testing or in terms of verifiable quantities. Instead, they serve as qualitative descriptors of materials and their behavior during the printing process.

Buildability refers to how well the material can be layered on top of itself and maintain the desired geometry without collapse. Buildability is a function of thixotropy and cement hydration (Roussel, 2018; Rahul, 2019). Thixotropy refers to how the viscosity of a material decreases when stress is applied over a time interval, and how that viscosity *recovers* or increases back to the original value when the stress is removed. This increase and decrease in viscosity does not happen instantaneously, resulting in materials with *high* or *low* thixotropy, corresponding with how quickly or slowly viscosity recovers. Thixotropy can be characterized by the evolution of yield stress over time, which can be

modeled by equation 1, and visualized in Figure 9, which shows a reduction in shear strain over time as applied shear stress is increased, indicating rebuilding of yield stress.

$$\tau_y(t) = A_{thix} t_c \left(e^{\frac{t}{t_c}} - 1 \right) + \tau_y(t = 0) \quad (1)$$

where

- $\tau_y(t)$ = yield stress at time t
- A_{thix} = flocculation rate
- t_c = characteristic time

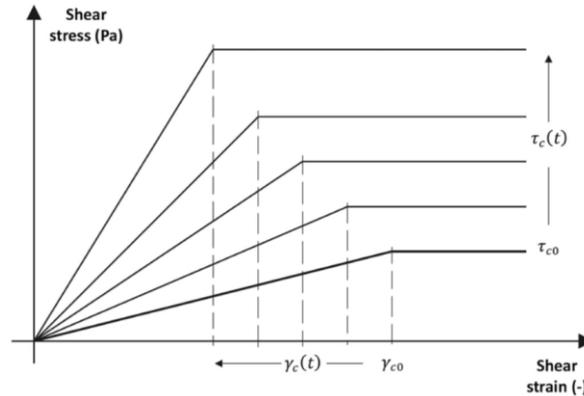


Figure 9: Structural rebuilding rate shown as yield shear stress over time: $\tau_{c0} \rightarrow \tau_c(t)$. (Roussel, 2018)

Materials promoting more rapid hydration allow for the faster development of both yield and green strength, or the early-age strength of the mixture. During the extrusion process, the stress applied by the pump must exceed the material yield shear stress to cause flow. Once deposited, the material must rapidly regain strength, such that the yield shear stress is no longer exceeded, and flow ceases. Highly thixotropic cementitious materials are therefore desirable for 3DCP applications because of their ability to regain structural stability and develop green strength quickly after extrusion. Calcined clay or other nanoclays have been found to increase thixotropy, but tend to reduce the rate of hydration over time due to their adsorption of water (Kawashima 2013, Heikal and Ibrahim 2016). Trigger (2013) demonstrated the relationship between green strength and increased thixotropy resulting from calcined clay, while also demonstrating a reduction in green strength with the presence of additives that reduce hydration rates, such as high-range water-reducing admixtures or fly ash. The addition of nanoclays was also found to increase stiffness of early-age concrete, which helped to decrease deformation during the printing process (Panda, 2019).

Flowability, also known as pumpability, describes how well the material moves through the printing system. A material with high flowability moves seamlessly through the system, while a material with low flowability is likely to clog, causing interruptions in the print or voids in the extruded layer. Voigt (2010) measured flowability using a drop table test, and found that the addition of fly ash improved flowability, while calcined clay reduced flowability. Yield stress and viscosity were also measured for all tested materials, and it was found that mixtures with low yield stress and low viscosity exhibited greatest flowability. However, these low yield stresses and viscosities also resulted in significant edge slump, indicating that highly flowable materials may not be suitable for fabrication, and that a balance must be struck between flowability and buildability of the mixture. In contrast, high yield stress and

high viscosity mixtures resulted in poor surface quality and honeycombing (clumping). Zhang (2019) identified a sand-cement ratio of 1.0-1.2 as optimal for flowability, and developed a high-thixotropy cement mixture that demonstrated high flowability and shape retention. Rheological properties of static yield stress, viscosity, and thixotropy were also measured and linked to flowability, with optimal material property ranges defined based on flowability and printability results. In both Voight and Zhang studies, yield stress and viscosity were obtained by considering the cement mixture as a Bingham fluid (Figure 10). Bingham fluids are a class of materials defined by exhibiting a defined yield point before flow begins, and a linear relationship between shear stress and shear strain rate thereafter, allowing yield strength to be calculated from equation 2. This model is commonly used to describe the fluid properties of cement mortar.

$$\tau = \tau_0 + \mu D \tag{2}$$

where

τ = shear stress

τ_0 = Shear yield stress

μ = plastic viscosity

D = Shear rate

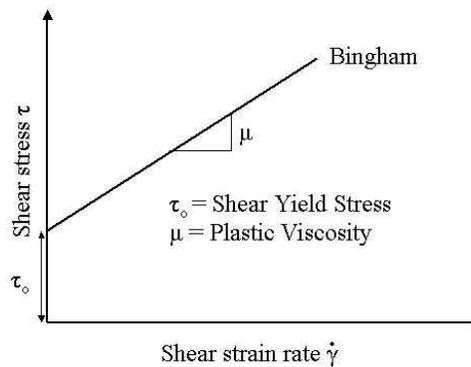


Figure 10: Bingham Model for non-Newtonian fluids.

In another study on flowability, Ma (2018) employed the slump test, the jump table test, and the V-funnel test to determine flowability for a mixture containing copper tailings. It was found that flowability increased by replacing natural sand with copper tailings up to 50%.

Extrudability refers to how well the material can maintain its shape after being formed and deposited through the printing nozzle. This characteristic can be evaluated by measuring the continuity and stability of a layer extruded through a nozzle of a designated size without resulting in a blockage, and is a pass or fail measure (Lee, 2012; Malabe, 2015). Blockages tend to be caused by the separation of liquid granular materials (Ma, 2018). Pumping pressures, then, cause the liquid to flow through the granular phase, leading to blockages as the grains build up behind the nozzle. Another method for evaluation involves extrusion rheology tests, which measure yield shear stress and viscosity, and are performed by using a ram extruder and load cell for force material through a die, and plotting pressure vs. distance moved by the piston. It was found that the presence of rheological modifiers with high fiber content, such as methyl cellulose or calcined clay, were necessary for successful extrusion. Malabe found that increasing the cement to sand ratio helps prevent these blockages and provides

better extrudability. Similar to pumpability and buildability, many of these studies measured rheological properties of the tested mixtures, and investigated the relationship between shear strength and extrudability, thereby determining optimal ranges for material properties of the mixtures for extrusion.

Printability is a combination of buildability and extrudability, i.e. how well the material prints and maintains the desired structure (Ma, 2018). Kazemian (2017) developed several test methods to measure print quality, shape stability, and the printability window for a given mix design. In this study, print quality was defined by layer dimension conformity and consistency, as well as by the presence of surface defects; shape stability was defined by measuring layer settlement; and printability window was defined as the timespan during which material could be extruded with acceptable quality. Roussel (2018) developed a set of rheological requirements for printable materials in terms of material properties, including minimum yield stress values to prevent flow, minimum elastic moduli to ensure buckling stability, and critical strain values to prevent surface cracking.

One of the major requirements posed for printability is that the yield stress of the material in each layer must remain greater than the stress applied on that layer by its self-weight and the weight of the layers above it. The yield stress is a function of time due to the continuous curing of concrete mortar, and the applied stress is a function of both time and height evolving throughout the printing process (Figure 11).

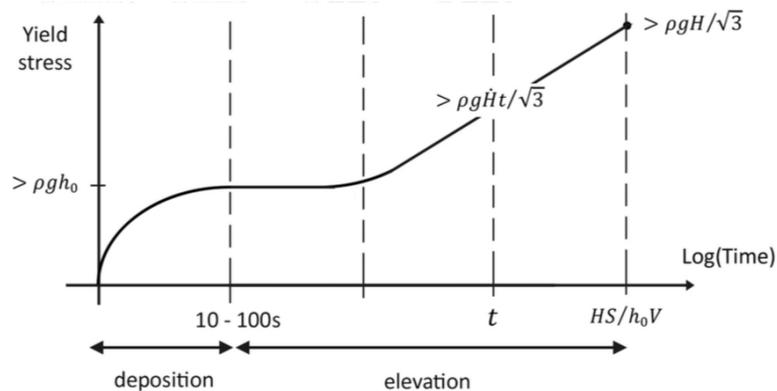


Figure 11: Yield stress evolution in terms of material density (ρ), layer thickness (h_0), final object height (H), contour length (S), nozzle velocity (V) and time (t). (Roussel, 2018)

Extrudability, buildability, flowability and printability are all affected by the time dependent nature of the material properties of cement mortar. In order to quantify this time dependence, the concept of *open time* was developed, and is used to describe the time span during which a material is extrudable, buildable, and therefore printable (Buswell, 2018). If printing is attempted too soon after mixing, the material may have excessive flowability and limited buildability. However, if too much time passes before printing is initiated, the material may no longer flow or be extrudable. Time-dependent behavior such as this may be attributable to thixotropy, hydration, or yield stress. The relationship between open time and material properties is not well understood at this time. However, Kruger (2019) developed an analytical model for buildability based on the time-dependence of yield stress and thixotropy, and used this model to define building rates and failure modes, leading to the creation of printable geometries (Kruger, 2020).

2.1.10 Yield Shear Stress

As shown in this section, one key material property emerges from many studies involving rheological properties of fresh cement mortar: yield shear stress, or simply yield stress (Kawashima, 2013; Paul, 2018; Kazemian, 2017; Buswell, 2018; Kruger 2019; Panda, 2018 and 2019; Roussel, 2010 and 2018). Yield stress is defined by the small region of applied shear stress values over which a viscoelastic material displays a rapid reduction in viscosity over several of magnitudes, as can be seen in Figure 12. From this small range, a single shear stress value is defined as the boundary between a material displaying solid properties and that same material displaying fluid properties when subjected to an increasing shear stress. This single value for concrete can be determined using the Shear Vane test (Omran, 2011; Rahul, 2019) and calculating yield shear using equation 3 relating applied torque to shear stress.

$$T = \frac{\pi}{2} D^3 \tau_y \left(\frac{H}{D} + \frac{1}{3} \right) \quad (3)$$

where

- T = measured maximum torque
- τ_y = yield stress
- H = vane height
- D = vane diameter

Time-dependence is an important aspect of yield shear stress in concrete as well as other viscoelastic materials. As concrete hardens over time, the yield shear stress of the material increases. In this way, yield shear stress is a dynamic property that will have a unique value depending on the time between mixing and measurement. In order to account for this time-dependence, Large Amplitude Oscillatory Shear tests can be performed to determine yield shear stress values at specified time intervals (Buswell, 2018).

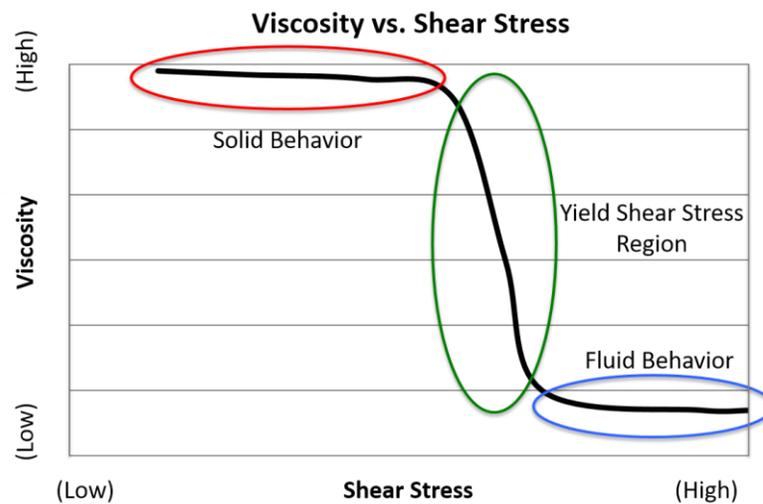


Figure 12: Yield shear stress graphical definition for viscoelastic materials. Yield shear stress is shown as the transition region between solid and fluid behavior corresponding to a range of applied shear stress values.

Buswell (2018) studied several problems related to printability, including geometry, open time, and self-weight, and was able to demonstrate the importance of yield stress in solving these problems. As yield stress of cement mortar changes with time, it governs the open time during which the mortar can be successfully extruded (lower yield stress) and support its own weight (higher yield stress). The study also demonstrated how minor changes in yield stress affect the printed layer thickness, as successive layers cause increased vertical displacement under self-weight. This deformation results in a gap between the previous printed layer and the layer currently being extruded, requiring continuous adjustment of the change in nozzle height between layers. As a result, the increment with which the printer head is raised for each layer must reduce throughout the printing process in order to maintain contact between successive layers. Rahul (2019) demonstrated that yield stress can be quantifiably linked to successful fabrication using 3D printing. Success in this case was defined in terms of extrudability, where measurements of printed layers conformed to those of the printer nozzle, and buildability, for which the vertical displacement of bottom layers were considered. The study identified an optimal range of values for yield stress, as values too low lead to failure of the printed element, and values too high impede extrusion.

2.1.11 Current Research Directions

The research and accomplishments presented here demonstrate the viability of 3DCP as a construction method, but also paint the picture of an industry in its infancy. Though many important structures have been fabricated, no standards have been set or proposed, either for measurement, testing, or design. The methods used to predict fabrication, structural performance, and material specification remain qualitative, and must be quantified in order to develop standards and make valid design predictions. These developments must take place before the 3DCP industry can fully develop. As 3DCP is still an emerging technology, there are many aspects that require further research and understanding.

Buswell (2018) addresses many of these challenges, and cites specific needs for further development within the field. Within the realm of fresh-state material development, Buswell discusses the need to relate pumpability, extrudability, and buildability to material properties through investigating open time, setting and layer cycle time, deformation during material build-up, and rheological measurements. Hardened state properties must also be investigated, including layer adhesion, shrinkage and durability, and the relationship between mechanical properties and printing parameters such as nozzle size, layer height, and print speed. Structural design for 3DCP is another area that needs further development. As the design is limited by materials and by the fabrication process, both areas must be well understood and linked to design through investigations of minimum feature sizes and tolerances, vertical printing angles, and modeling of both materials and the fabrication process. Much of the work presented in this section is geared toward development in these areas. However, further research is necessary and ongoing.

Incorporation of reinforcement into the 3DCP process or into design for 3DCP structures is also vital to the success of the industry, as methods compensating for the tensile weakness of concrete must be developed. Current methods under investigation involve fiber reinforcement, internal reinforcement, and external reinforcement. Additionally, for 3DCP to become an industry practice, design codes, standards, and testing methods must be developed to ensure quality and safety (Paul, 2018).

It is the author’s view that the most pressing need, and the largest area of potential research, involves relating material properties to printable designs. Therefore, the main thrust of this work is to help develop this link between materials, structure, and fabrication process.

2.2 Topology Optimization

The process of Topology Optimization (TO) attempts to mathematically determine the ideal placement of material within a given geometric volume. In order for an ideal material arrangement to exist, the geometry must be exposed to various loading states, boundary conditions, as well as constraints that define the limit of material functionality within that space. For example, consider Figure 13, which shows a cantilever beam subjected to a distributed load applied along the midline of the top face. In this design, the beam is solid. However, depending on the applied load and on various material properties such as strength and stiffness, it is possible that material could be removed and the remaining structure could still support the applied load. Now, imagine the beam is made of steel. Steel is a heavy and expensive material, so it may be desirable to minimize the amount of material used in this beam, as long as the beam continues to support the load. By removing a small amount of material from the mid-left and mid-right sections of the beam cross section, as in Figure 14a, an I beam with a very thick web is created. Removing the same amount again reveals a thinner I beam. Continuing this process over several iterations results in an I beam of standard dimensions used in construction – one that can still resist the applied distributed load (Figure 14d). This iterative process of material removal while maintaining functionality forms the basis of TO. To maximum the effect, material removal is generalized to material placement, allowing addition or subtraction of material at locations determined to be necessary under given loading, boundary, and material conditions.

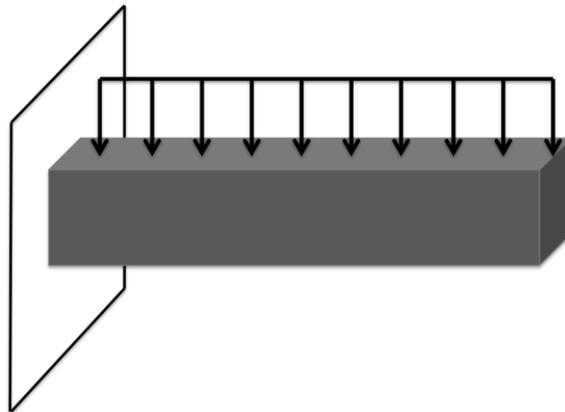


Figure 13: Cantilever beam subject to a uniform distributed load on the top surface.

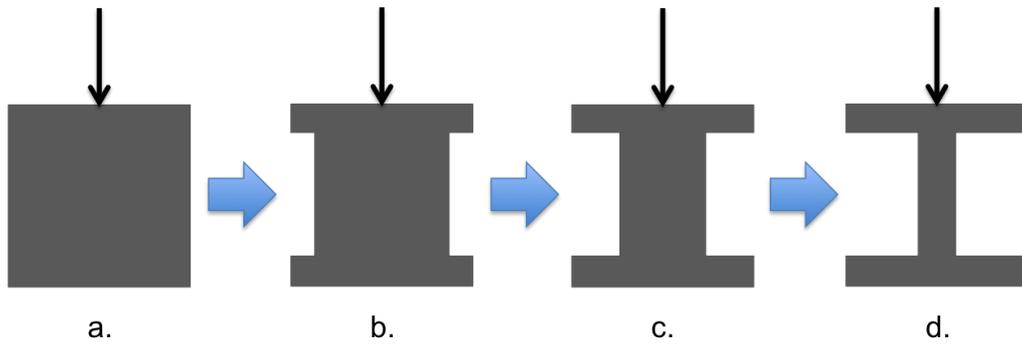


Figure 14: Cantilever beam cross sections showing progressive material removal.

TO was first developed by Bendsoe and Kikuchi (1988) as an extension of shape optimization. This initial method attempted to reduce the volume of a specimen by introducing infinitesimal, periodic holes into an otherwise continuous, homogeneous, and isotropic material medium, provided the resulting structure could still carry the applied loads and meet design constraints. Therefore, the design problem became a question of presence or absence of material at individual points within the geometry, and how the resulting porosity affected the material properties of the new “composite” material. The porosity of the composite material can then be considered as an artificial material density of a single, finite volume element, and the stiffness of that element becomes a function of that artificial density. The optimal artificial density can then be determined at each material point, allowing for the determination of the optimal distribution of material within the design space. This initial work then solved a compliance minimization problem subject to a minimum volume constraint, which has since become a standard textbook TO problem considered by many algorithms. However, this method was subject to the problem of intermediate material properties. It was common that, at a given point, the composite material was present, but held properties that differed greatly from those of the homogeneous material. Since material can either exist (1) or not exist (0), these intermediate regions were not physically viable.

Since this initial homogenization formulation, several other methods have been developed that have replaced homogenization as TO methods in use by researchers, commercial software, and design engineers. A few of the most widely used and tested methods are introduced here.

2.2.1 Simplified Isotropic Material with Penalization (SIMP)

The SIMP method was first developed by Bendsoe (1989) as an extension of the homogenization method, in an attempt to solve the problem of intermediate materials. Here, the material was no longer considered as a composite. Instead, the presence or absence of material at each element became the design variable, with each element assigned an artificial density ranging from 0 to 1, with 1 corresponding to the existence of material and zero corresponding to a void. This artificial density would scale the stiffness of the material, penalizing any intermediate density with a very low stiffness, and therefore driving the solution at that point to a value of 0 or 1. Penalization is performed using a power-law approach where E is Young’s Modulus, x_i is the artificial density, and p is the penalization factor (eqn 4).

$$E(x_i) = g(x_i)E_0 \quad \text{where} \quad g(x_i) = \rho_i^p \quad (4)$$

Bendsoe and Sigmund (1999) defined the penalization factor numerically and physically by conceptualizing the intermediate material densities as composite materials, and using the Voigt and Hashin-Shtrikman upper bounds to define engineering parameters. These bounds define an upper limit on stiffness of the imaginary composite material, consisting of a given volume fraction of solid material and a corresponding volume fraction of voids within a given representative volume element. The Voigt upper bound is estimated using the rule of mixtures for composite materials (eqn 5), and by assuming strain continuity between materials. In other words, both the void area and solid area will deform equally. This assumption results in an upper bound on stiffness due to the fact that strains are not continuous in reality, and that allowing the void areas to absorb some of the strain energy makes the composite material stiffer than the homogeneous material. The Hashin-Shtrikman upper bound is developed similarly from the limitations placed on composite bulk and shear moduli by considering the rule of mixtures and the theory of minimum potential energy (eqn 6). Lower bounds are not considered when defining the penalization factor, as the minimum stiffness is known to be zero (or approaching zero) for the void material.

$$E_{eff} = v_f E_f + v_m E_m \quad (5)$$

where

E_{eff} = Effective composite Young's Modulus

v_f = Fiber volume fraction

E_f = Fiber Young's Modulus

v_m = Matrix volume fraction

E_m = Matrix Young's Modulus

$$\Pi = U - W \quad (6)$$

where

Π = Total potential energy

U = Elastic strain energy

W = Work

$$U_e = \frac{1}{2} \int \{\varepsilon\}^T \{\sigma\} dV \quad (7)$$

where

ε = Strain

σ = Stress

Thus, the penalization method was related to physical phenomenon, and was no longer purely a mathematical construct. However, the choice of p depends on the physical problem and can have an effect on the final solution. Often p is relaxed, or varied from larger to smaller values, throughout the optimization process, proceeding from $p = 1$, until convergence is found.

Since its introduction in 1989, SIMP has become the most common method for TO. It has been used to solve compliance minimization problems (Sigmund, 2001; Talischi, 2012; Liu, 2014), stress-constrained problems (Bruggi and Duysinx, 2012; Holmberg, 2013; Gebremedhen, 2019), and problems considering multiple materials (Gaynor, 2014; Li and Kim, 2018; Zuo and Saitou, 2017), among others. Sigmund (1999) released a 99-line Matlab code for 2D SIMP-based TO that has become a standard algorithm for

researchers. This 99-line code was modified for efficiency by Andreassen (2011) and was expanded to solve 3D problems as well (Liu 2005, 2014). Programs were also developed in the Python language (Hunter 2009 from Liu) and for Mathematica (Sokol 2011 from Liu).

2.2.2 Evolutionary Structural Optimization

The original Evolutionary Structural Optimization (ESO) method functioned by incrementally removing material at locations with the lowest stresses or strain energy (Huang and Xie, 2010). As a unidirectional method, a high initial volume fraction was set and optimization would proceed only by removing material. In this way, the structure “evolves” with each iteration, as each material placement is based on the previous arrangement. Due to the unidirectional limitation, the method was later expanded to Bi-directional Evolutionary Structural Optimization (BESO), which allowed material to be added as well as removed (Chu, 1996). The original versions of ESO/BESO could only be used to solve compliance minimization problems subject to a volume constraint, and could not be easily adapted to other types of problems. Huang and Xie (2010) were able to pose and solve a displacement-constrained problem using BESO, but the method has not been adapted to solve more general problems as of yet.

A significant limitation of the directional nature of ESO/BESO is the problem of local vs. global solutions. Solutions result in a locally optimized solution due to the fact that the method proceeds directionally and with relation to previous iterations, instead of searching for optimal solutions with each iteration. However, ESO/BESO methods are widely used, both in both research and industry. (Huang and Xie. 2008)

2.2.3 Level-Set

The Level-Set method is another continuous TO method, which functions by tracking the boundaries that define the surface of the solid volume under consideration. Tracking is performed using algorithms developed by Osher and Sethian (1988, 1999) using the Hamilton-Jacobi equation:

$$\frac{\partial \phi}{\partial t} = -V\mathbf{n} \cdot \nabla \phi \quad (8)$$

where t is a time-like parameter representing the evolution of the surface, and V is a speed function that moves the level-set function ϕ relative to a “merit” or constraint function. Similar to SIMP, magnitude and direction of changes to the level-set function are driven by sensitivities, both of the level-set function itself, and of the constraint functions.

The initial work using level-set methods for TO was performed by Sethian and Wiegman (2000). They were able to present the optimized structure of a cantilever beam subject to a point load minimizing both weight and compliance separately. The initial design geometry in this case was a porous structure. This was due to the fact that the original level-set method could only optimize a single, solid geometry, as the geometry evolved from the initial surface boundary. Introducing pores into the initial design allowed optimization to proceed along several surface lines. Later methods were developed to

introduce holes into the interior of an initially solid geometry, allowing interior voids to be produced where necessary during optimization (Wang 2007, Yamada, 2010).

Further optimization work has been performed using the level-set method, including compliance minimization (de Gournay, 2008), stiffness maximization of multifunctional materials (Challis, 2008), and stress minimization of multi-material topology (Chu, 2017). Additionally, Challis (2010) and Vogaitzis (2017) have developed open-source Matlab codes for level-set optimization, the former based on Sigmund’s 99-line Matlab code referenced above.

Each of these TO methods shares a set of basic requirements that form the scaffolding of the optimization process. The requirements are discussed in the following section, and are presented in terms of a standard TO problem. The example problem involves minimizing the compliance of a cantilever beam under a unit point load and subject to a volume fraction constraint. This example problem follows that set up and solved by Liu (2014), demonstrating the functionality of the open-source TO software, Top3D. The problem is formulated as follows:

$$\left\{ \begin{array}{l} \text{minimize: } c(\mathbf{x}_i) = \mathbf{F}^T \mathbf{u}(\mathbf{x}_i) \\ \text{subject to: } \frac{\sum_{i=1}^n x_i V_i}{\sum_{i=1}^n V_i} \leq V_f \\ \mathbf{0} \leq x_i \leq \mathbf{1} \end{array} \right. \quad (9)$$

where

- $c(\mathbf{x}_i)$ = Elemental compliance
- \mathbf{F} = Applied force at nodes of element i
- $\mathbf{u}(\mathbf{x}_i)$ = Nodal displacement
- V_i = Volume of element i
- V_f = Designated maximum volume fraction

It is important to note that, in the following sections, specific attention will be paid to the formulation of requirements for the SIMP method, as the SIMP method will be used in this research. Additionally, TO will be discussed in the context of structural analysis and design, though it is not limited to that domain.

2.2.4 Requirements

2.2.4.1 Geometry

The first stage of any TO problem involves defining the initial geometry that is to be optimized. This process involves setting boundaries on the space that can potentially contain material. This space can be directly related to a real-world engineering problem with defined architectural and/or structural dimensions, or it can be a theoretical design space created by the design engineer. The boundaries are determined in terms of dimensionality (2D or 3D) as well as lengths of sides and selection of shapes. The space is then discretized into finite elements for analysis. Conceptually, the space can be thought of as initially empty or initially full of material, but the initial state of artificial material density can be defined on a range from 0-1 depending on the directionality of the optimization process. As such, each

element in this initial space is given an initial condition of material presence or absence from which optimization is to proceed.

Common geometries used throughout research to test the effectiveness and accuracy of new TO methods include the rectangular beam, L-shape, and T-shape. In design, however, the initial geometry can be as complicated as the available space for a Real Lower Control Arm of a Volvo automobile (Larsson, 2016) or the design space for a turbomachinery impeller (Meli, 2019).

The geometry given for the example problem is a beam, represented by a rectangular space (Figure 15).

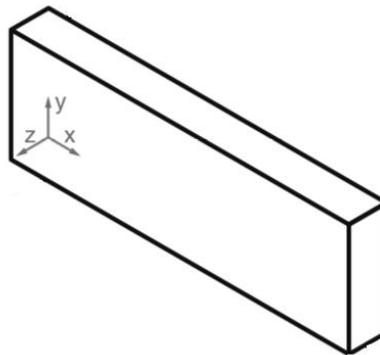


Figure 15: Example problem rectangular prism geometry.

2.2.4.2 Boundary Conditions

Once the geometry has been set in space, boundary conditions (BC's) must be defined. BC's set limitations on the movement of the component, and are derived from the physical problem. They are defined in terms of limitations on displacements and rotations at fixed points in the geometry. For example, the cantilever beam considered above will have fixed displacements in all directions, as well as fixed rotations at each node on the connected surface, while a simply-supported beam will have fixed displacements in one direction only at the bottom corner nodes. BC's must be defined in order to prevent full rigid body motion, which leads to an infinite number of possible displacements relative to a given loading condition.

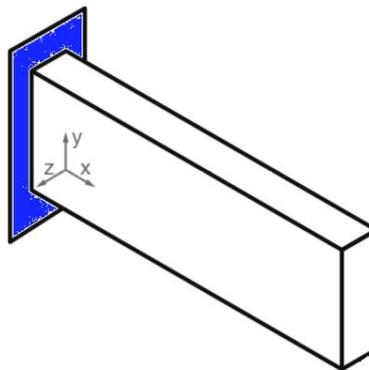


Figure 16: Example problem cantilever boundary conditions (blue area represents fixed edge).

2.2.4.3 Loading

The purpose of any TO process is to find the optimal solution to an engineering design problem. In the case of structural design, the problem involves the application of some design load. The load may be a point load, a distributed load, a design-dependent load, or a dynamic load. Design-dependent loads refer to loading that changes based on the design, a common example of which would be pressure loading acting on a surface contour – as the contour changes, so does the pressure (Lee, 2012). Additionally, multiple loads or multiple load cases may be considered. Loads are applied to the initial design space at the nodes of the discretized geometry corresponding to locations of applied force on the final design. Loading locations and values must therefore be known beforehand, the exception being the case of design-dependent load cases where loading and location are adjusted as the material placement changes.

The location of the point load in the example problem will be the bottom right corner, as seen in Figure 17.

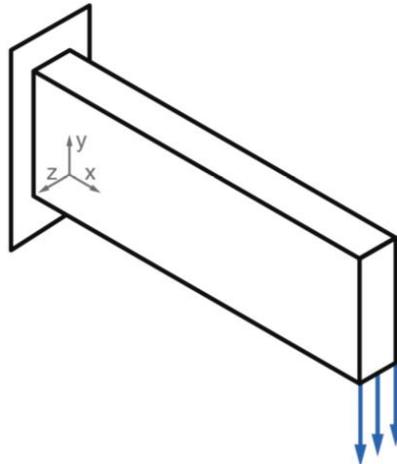


Figure 17: Example problem loading at the bottom right edge of the design space.

2.2.4.4 Define Material Properties

TO is a problem of material placement. As such, the specific material or materials under consideration must be defined in terms of their physical properties. Young's Modulus and Poisson's ratio must be defined in order to perform Finite Element Analysis, as discussed in section 2.2.4.6. Additionally, if the weight of the component is to be considered, the physical density of the material must be defined.

The example problem is unit-less. Therefore, material properties defined for the example problem are as follows:

Young's Modulus = 1
Poisson's Ratio = 0.1

2.2.4.5 Iterations

The subsequent sections, including Finite Element Analysis (FEA), objective and constraint function calculations, sensitivity analysis, optimization, and termination criteria checks are performed in order, and if the result of these steps does not end the optimization process, a new iteration will begin with FEA. Loading may also be included in the iterative process if the loads are design-dependent (Lee, 2012) or if self-weight is considered (Bruyneel and Duysinx, 2005), as is the case in this work. Figure 39 (section 5.14) shows the iterative processes involved in TO.

2.2.4.6 Finite Element Analysis (FEA)

With set boundary conditions and applied loads, FEA can be performed on the potential structure. FEA determines the displacements of each element, allowing for calculations of stress and strain as well. These calculations provide the information necessary to determine the relative importance of material presence at the location of each element. For example, if an element experiences very low stress, material placement at that location may not be necessary for structural functionality, and therefore, the material may be removed from the structure. The mathematical basis of FEA will be discussed later when presenting the development of the novel TO algorithm.

The method of FEA is dependent upon the dimensionality of the geometry, the physical problem being solved, and the required accuracy of the solution. Additionally, the computational cost of FEA is often considered in determining the method. In his 99-line code, Sigmund (2001) uses a square bi-linear 4-node element to perform 2D FEA, while Liu (2014) uses an 8-node hexahedral element in an expanded 3D version of the 99-line code. By using a single element type defined beforehand, both Sigmund and Liu are able to pre-populate the stiffness matrix in order to save computational time and resources. PolyTop is a TO algorithm developed for the purpose of expanding available mesh options by using isoparametric polygonal elements (Talischi, 2012).

An alternative method for performing FEA, avoiding the necessity of writing new code, is to link the TO program to a commercial FEA solver. Zuo and Xie (2015) developed a TO algorithm in Python that accesses the FEA solver in Abaqus, while Papazafeiropoulos (2017) developed a program for enabling quick and easy data transfer between Matlab and Abaqus specifically for problems involving optimization.

The example problem requires use of FEA equations to calculate displacements, and therefore, compliance. The FEA displacement equation is presented below (eqn 10) will be discussed in detail in section 5.6. Additionally, the FEA code developed for Top3D is used for FEA development and calculations.

$$\mathbf{Ku} = \mathbf{F} \tag{10}$$

where

\mathbf{F} = applied force vector

\mathbf{u} = displacement vector

\mathbf{K} = stiffness matrix

For this example problem, the element side length, and therefore volume, are set to unity.

2.2.4.7 Objective Function

The objective function defines the goal of the optimization problem. That goal is given in terms of maximizing or minimizing a specific quantity that can be calculated based on geometry, boundary conditions, applied loading, and/or FEA results. The goal must also be a differentiable function of the design variable (artificial density) such that as the material placement changes, so does the objective function. As such, the objective function is calculated for each iteration. If the objective function satisfies certain termination criteria, as discussed later in section 2.2.4.12, the optimization can be considered complete, and the process will end.

Defining the objective function depends on the physical problem being solved. The most common form of TO is the compliance minimization problem, where the goal is to minimize the displacement of the structure under loading. In this case, and in the case of the example problem, the objective function is a calculation of either A. the maximum compliance of the structure, or B. the sum of the compliance of each structural node. Case B is used in the example problem, and is expressed as:

$$\min c(\mathbf{x}_i) = \mathbf{F}^T \mathbf{u}(\mathbf{x}_i) \quad (11)$$

and calculated as:

$$c = \sum_{n=1}^i \mathbf{u}(\mathbf{x}_i) \quad (12)$$

Second to the compliance problem in terms of commonality is the volume minimization problem, which searches for the arrangement of material that results in the lowest volume. The objective function in this case can be calculated in terms of mass or volume.

Other objective functions include minimizing stress, strain energy, torque (Choi, 2011), vibrations (Du and Olhoff, 2007), or maximizing energy absorption (Soto, 2001).

2.2.4.8 Constraint Functions

A constraint function defines a limitation on the optimization process that reflects a physical restriction imposed as a result of the design problem. This function is expressed as a differentiable inequality in terms of the design variable, where the calculated value of the constraint in the current iteration is compared to a limiting value set by the design engineer. For example, a simply-supported beam may be optimized to minimize weight, but restricted to a maximum deflection of 2 cm due to serviceability considerations.

With each iteration of the TO process, deflection at each node is calculated during the FEA step, and compared with the maximum allowable displacement. If the constraint is not satisfied, such that deflection exceeds the maximum, the following iteration will attempt to place material in order to reduce deflection. If the constraint is satisfied, the process will consider improving the objective function by further minimizing or maximizing the target quantity. Without this constraint, the optimization process would continue to reduce the volume of the structure endlessly, resulting in physically unacceptable or unrealistic deflections. In this case, it is likely that the volume would reduce to a physical minimum or even zero volume, as the reduction process is unrestricted.

The constraint function is not limited to deflection, however. Depending on the physical problem, the optimization process can be constrained by stress, volume (Liu, 2014), connectivity (Li, 2016), geometry (Dapogny, 2017), or many other criteria. It is most common for the constraints above to be considered on a global level, where a single maximum/minimum value from the entire structure is considered as the only constraint imposed on the design. Considering the global max/min assures that all other elements will satisfy the constraint as well. However, global consideration may not result in a fully optimized structure, as local conditions may or may not significantly affect the overall structure. Therefore, researchers have developed algorithms to impose multiple constraint function, allowing for consideration of values at a local level, where each individual element is separately constrained (Duysinx and Bendsoe, 1998), or at a regional level, where values are averaged across several designated areas in the structure (Holmberg, 2013). Additionally, multiple constraints allow optimization problems to account for multiple criteria simultaneously, such as compliance and connectivity (Li, 2016).

In the example problem, the optimization process is constrained by a maximum volume fraction, expressed mathematically as:

$$\frac{\sum_{i=1}^n x_i V_i}{\sum_{i=1}^n V_i} \leq V_f \quad (13)$$

For this problem, the volume fraction constraint is set to: $V_f = 0.3$.

2.2.4.9 Sensitivity Analysis

With each iteration, the objective function and constraint functions are calculated using current values of the design variable. If these functions do not satisfy termination criteria, the optimization process will continue, and the material placement for the subsequent iteration will be determined. In order to determine the new material placement, sensitivity analysis must be performed.

Sensitivity analysis enables generation of a new material placement by calculating the effect of changing the design variable on the values of the objective and constraint functions. For example, considering a compliance minimization problem subject to a volume fraction constraint. If the volume fraction of the structure is less than the constraint, material may be added to the structure in order to decrease compliance. Sensitivity analysis calculates how adding or removing material at a given location will affect the displacement. If the result is reduced compliance, material will be added. If the result increased compliance, material may not be added.

Sensitivity analysis is performed by calculating the gradients of the objective and constraint functions with respect to the design variable at each element in the structure. These gradients can be calculated by either discrete methods or variational methods (Papadrakakis, 1996). Variational methods tend to be complex and expensive, and are much less common than discrete methods. Therefore, only discrete methods will be discussed here. Discrete methods involve calculating the derivatives of the target function, either analytically or numerically. The finite difference method offers a numerical solution through perturbation of the design variable by small increments and re-calculation of the target

function. The forward, backward, or central difference method may be employed. Mathematically, the finite (forward) difference method is represented as such:

$$\frac{\partial f(x_i)}{\partial x_i} = \frac{f(x_i + h) - f(x_i)}{h} \quad (14)$$

where $f(x_i)$ is a continuous, differentiable function of the design variable x_i , and h is a perturbation of the design variable. This formula can be used to calculate sensitivities of any objective or constraint function of x_i simply by changing the function $f(x_i)$, making the method simple to implement. However, since the derivative must be calculated for each element and for each function, the finite difference method quickly becomes computationally expensive.

The alternatives to numerical methods are analytical methods, which involve chain rule differentiation of finite element equations. In contrast to the finite difference method, derivatives for unique functions are unique themselves, so sensitivity equations must be formulated for target functions on an individual basis.

Analytical methods for determining derivatives is computationally more efficient, but can be challenging to implement due to the complex nature of many objective and constraint functions. Additionally, it is possible to encounter solutions that require inversion of the stiffness matrix, which may have multiple solutions at the local level. In this case, adjoint variables or pseudo-loads must be calculated globally, which can blur the distinction between local and global sensitivities. A potential solution here is to combine numerical and analytical methods into “semi-analytical” methods involving finite difference equations and derivatives of finite element equations (Papadrakakis, 1996).

For the example problem, the analytical sensitivities are presented here for both the minimum compliance objective function (eqn 15) and the volume fraction constraint function (eqn 16).

$$\frac{\partial c(x_i)}{\partial x_i} = -\mathbf{u}_i(x_i)^T [px_i^{p-1}(E_0 - E_{min})\mathbf{k}_i^0]\mathbf{u}_i(x_i) \quad (15)$$

$$\frac{\partial V(x_i)}{\partial x_i} = V_i \quad (16)$$

2.2.4.10 Optimization

The optimization algorithm is responsible for determining material placement for the subsequent iteration. There have been several proposed methods for optimization, each requiring different inputs, and each proceeding towards a solution in a different manner. Two of the most common methods for optimization are discussed here.

1. Method of Moving Asymptotes

The Method of Moving Asymptotes (MMA) was developed by Svanberg (1987), and while it can be generalized to many optimization problems, it was designed specifically for structural optimization. MMA belongs to a special class of algorithms known as conservative convex

separable approximations (CCSA) designed to solve non-linear inequality optimization problems. CCSA methods are applicable to minimization problems with *less-than or equal-to* constraints (Svanberg, 2002). These methods consist of two iterative loops, one inner and outer. The outer loop is represented by the iterative TO process, discussed further in section 5.5. The inner loop takes in the values of the objective and constraint functions and their derivatives, and then creates and solves a sub-problem of the same form as the original TO formulation that approximates the original functions. If the solution is considered optimal, it is accepted and the inner loop terminates. If the solution is rejected, a new sub-problem is created and solved, until an optimal solution is reached. MMA further develops this idea by employing the idea of “moving asymptotes” which form bounds on the sub-problem and the resulting function approximations. The bounds move with each iteration, allowing a more optimal solution to be reached.

MMA has quickly become a widely used method for optimization due to its ability to efficiently consider a large number of design variables. However, the method is not globally convergent, meaning it does not always result in a globally optimal solution. As a result, Svanberg (2002) developed a globally convergent MMA (GCMMA), which in practice is quite computationally expensive and time consuming due to the number of iterative parameters involved.

2. Optimality Criteria

Optimality Criteria (OC) is an older and simpler optimization method based on the conditions proposed by Karush, Kuhn, and Tucker (KKT) (eqn 16). These conditions define whether a given point represents an optimal value. In the context of TO, OC is similar in form to MMA, in that it requires values of the objective and compliance functions and their sensitivities in order to calculate the next material iteration. Optimization proceeds as a nested loop that either adds or subtracts material based on calculations of the optimality condition. The direction and magnitude of optimization (whether material is added or subtracted, and the amount thereof) is driven by the relationship of the optimality condition to the objective function of the current iteration. Consider the compliance minimization problem presented in the example, and in Bendsoe (1995) and Liu (2014). The OC is calculated as B_i (eqn 17) and the previous objective function value x_i is scaled by this value (eqn 18). If the new objective function value is greater than the previous value, material is added in proportion to B_i , whereas if the new value is less than the previous value, material is removed. A move limit is set to define the maximum value of change between iterations.

$$\frac{\partial c(x_i)}{\partial x_i} + \lambda \frac{\partial V(x_i)}{\partial x_i} = 0 \quad (16)$$

$$B_i = \frac{\partial c(x_i)}{\partial x_i} \left(\lambda \frac{\partial V(x_i)}{\partial x_i} \right)^{-1} \quad (17)$$

$$x_i^{new} = \begin{cases} \max(0, x_i - m), & \text{if } x_i B_i \leq \max(0, x_i - m) \\ \min(1, x_i + m), & \text{if } x_i B_i \geq \min(1, x_i + m) \\ x_i B_i, & \text{otherwise} \end{cases} \quad (18)$$

This relationship between OC and optimization direction depends on the problem being solved. In the case of compliance minimization with no self-weight, adding material will always reduce compliance and subtracting material will always increase compliance. This relationship does not exist under self-weight loading, or when considering material properties such as shear stress, as will be shown later in this work. In order to use OC, this directional relationship must be known and clearly defined, such that optimization direction can be known *a priori*.

3. Other Methods

Other optimization methods include Sequential Quadratic Programming such as CONLIN (Fleury, 1989), the Active Set algorithm (Nocedal and Wright, 2006), and Line Search methods (Duysinx, 2020).

OC is the optimization method selected for the example problem due to the singular nature of the constraint, and the prescribed optimization direction presented by the compliance minimization problem. Additionally, OC is employed by Top3D to solve the compliance minimization problem, so modification is not required.

2.2.4.11 Filtering

Many calculations in TO occur at the elemental level or result in values defined for each element. All FEA calculations are based on and/or produce nodal or elemental quantities. In the case of FEA, these quantities are based partly on continuity relationships between elements. This is an important concept, as no material element exists in a vacuum. However, when assigning material to an individual element, these relationships are not considered. This disconnected relationship can result in checkerboard patterns, or areas of alternating material and void placements, representing a physically impossible arrangement of material (Diaz and Sigmund, 1995). It is for this reason that filtering plays an important role in the TO process. Filtering involves weighting the design variable of an element based on values of the design variables of surrounding elements.

Several filtering methods have been devised to solve this issue. Sigmund initially proposed the density filtering method (1994, 1997 from Duysinx course lectures). This method was then simplified by Bruns and Tortorelli (2001). The simplified method is employed in current TO algorithms (Liu, 2014) and defines the “neighborhood” of each element based on a filter radius R , prescribing a weighting factor as a function of that distance. The weighted artificial densities are then calculated as follows, and later replace the initial density values:

$$x_i = \frac{\sum_{j=1}^n H_{ij} v_j x_j}{\sum_{j=1}^n H_{ij} v_j} \quad (19)$$

$$H_{ij} = R - \text{dist}(i, j) \quad (20)$$

where

n = Total number of elements

H = Weight factor

R = Filter radius

$\text{dist}(i, j)$ = distance between elements i and j

Another common filtering technique is the Heaviside Filter (Guest, 2014), which modifies the density filter with a Heaviside function, driving the artificial density directly to 0 or 1. This method eliminates “grey” elements of intermediate artificial density.

The filtering technique employed by Liu (2014) is used in the example problem.

2.2.4.12 Termination Criteria Checks

In order for the optimization process to end, termination criteria must be defined. These criteria can define a successful optimization or an incomplete optimization, or they can manually end the process when specific criteria are met. In the case of successful or manually completed optimization, the geometry produced by the most recent iteration becomes the optimized structure

A successful optimization results in a structure that satisfies all constraints and completes the goal stated in the objective function. In this case, successive iterations will converge on a solution, and a convergence tolerance is set to define when optimization is complete. For example, Liu (2014) defines a change variable, which is determined by calculating the artificial density change of each element from one iteration to the next, considering only the maximum absolute value of the change. If this change is less than a specified tolerance value, and the constraints are satisfied, the process is thought to have converged on a solution, and the program will end. Other convergence checks include a set KKT tolerance (Svanberg, 2002) or a resolution tolerance (Biyikli and To, 2015).

It is standard practice to define a maximum number of iterations to prevent the process from running indefinitely. This is commonly performed by embedding the iterative processes within a while loop, which records the number of iterations, and runs until the maximum number of is reached. If the maximum number of iterations is reached without convergence on a solution, the optimization can be considered incomplete.

It is also possible to define specific criteria that will manually end the process, such as a minimum or maximum volume being reached, as will be discussed later in the context of this work. These criteria can have a physical meaning, identifying if either the constraints cannot be met, or if the objective function cannot be satisfied.

The example problem solved by Top3D considers only two termination criteria: 1. Convergence based on incremental changes in artificial density between iterations, and 2. A set maximum number of iterations.

2.2.4.13 Results

The final optimized geometry is displayed graphically when the process is finished, and the final values of the objective and constraint functions are recorded. The geometry is presented in terms of individual elements, with the artificial density of the elements portrayed in gray scale. Common practice is to represent solid elements in black and void elements in white. Depending on the program,

an element of intermediate artificial density may be forced to black or white, or presented on a gray scale to indicate density value.

The results of the example problem are presented in Figure 18. Top3D employs a gray scale display function, allowing intermediate elements to be displayed in terms of their calculated artificial density values.



Figure 18: Example problem optimized structure.

2.2.5 Applications

TO is a flexible design method, capable of solving many different engineering design problems through the choice and implementation of objective and constraint functions, and their relationship to physical applications. As a result, TO is now broadly utilized to refine complex designs, such as aircraft wing box ribs (Tucker, 2004), a Volvo rear lower control arm (Larsson, 2016), and turbomachinery impellers (Meli, 2019). As such, the process has been incorporated into many commercially available finite element analysis software packages, such as Abaqus, ANSYS, and Altair Optistruct.

Despite the power of the TO process, its practical impact was initially limited, as many of the complex designs generated by the process could not be manufactured using traditional fabrication techniques. However, with the rapid advancement of additive manufacturing, TO has become an increasingly popular design method since previously “un-machinable” designs can now be fabricated (Wang 2013, Robbins 2016).

Much of the current research involving TO in additive manufacturing is focused on metals or polymers due to the ability to fabricate a component in its final form. There are limitations to these designs, however. Many designs contain overhanging regions of material that must be supported. Vanek (2014) and Dumas (2014) used TO to design removable structural supports necessary for the additive process. Leary (2014) and Gaynor and Guest (2016) link the additive fabrication method with structural design by using TO to design support-free components, allowing for fabrication without support removal post-printing.

With proper selection of geometric conditions and design constraints, TO can also be used to predict the relationship between material properties and optimal geometry. For example, Nomura (2015) developed an optimization method for anisotropic materials, and Zhang (2016) investigated the effects of mechanical anisotropy on structures optimized for additive manufacturing. Zuo and Saitou (2017) and Li and Kim (2018), among others, developed TO methods for multi-material designs, while Du

(2016 and 2018) developed a computational framework for TO of bi-modular materials. These works are important since many real engineering materials are not linear elastic, as required by the conventional TO codes, and designs often utilize multiple materials in tandem.

In the context of this work, it is important to understand the advances made in stress-constrained TO methods. Stress-constrained TO problems consider stress equations as constraint functions, and designate a maximum stress value for a particular design. Several methods for stress-constrained optimization have been developed, nearly all considering Von Mises stress (VMS) or failure criterion such as that proposed by Drucker-Prager (Luo and Kang, 2012). A vast majority of stress-constrained TO algorithms employ the SIMP method, and nearly all consider either mass or compliance minimization. Duysinx and Bendsoe (1998) provided the theoretical basis for inclusion of local stress constraints, but considering the large number of constraint functions, other less costly methods are more widely used. Global stress constraints have been employed (Gebremedhen, 2019), however, the presence of stress concentrations may skew the results in these cases. As a compromise, in an attempt to mitigate both stress concentrations and computational expense, techniques have been developed to cluster or regionalize the stress constraints (Holmberg, 2013; Lee, 2012; Le, 2009). Advances such as these have resulted in the incorporation of VMS stress constraints into commercially available FEA software (ANSYS).

Additionally, for the purposes of this work, the use of TO in concrete design must be understood. To date, only a few studies have investigated the application of TO in concrete design. Much of this work is geared towards finding the optimal placement of reinforcement within concrete components, (Amir, 2012; Bogomolny and Amir, 2011 and 2012; Amir and Sigmund, 2013) or external to concrete structures (Cunha and Chaves, 2014). These works apply principles of multi-material TO as the basis for design of reinforced concrete members. Both linear and non-linear material models have been used to define concrete behavior in these cases. Linear models were used when considering displacements only, while non-linear models were used when considering damage failure modes, such as cracking. Outside the realm of rebar placement, TO has also been used in one case to design pre-stressed concrete members (Amir and Shakour, 2017), and in another case to design a concrete slab, fabricated using 3D printed, stay-in-place formwork (Jipa, 2016). To the best of the author's knowledge, no studies have been performed using TO for 3DCP applications.

Chapter 3: Posing the Problem

3DCP fabrication dictates that, in order for a structure to be printable, the cement mortar should be flowable, extrudable, and buildable. Additionally, the design to be printed must lend itself to fabrication from the chosen cement mortar in its fluid state, as structural stability during the 3DCP process is determined by the fresh state properties of cement mortar as it is being extruded. In this way, to successfully fabricate the structure, both the material and the design must be printable. The material must display printable properties relative to the fabrication method, and the structure must be designed in such a way as to avoid excessive deformation and collapse during the printing process. Therefore, the goal of this work is to offer a connection between structural design, material properties, and fabrication technique, and, by employing TO, to use that connection to develop complex structural designs that can be successfully fabricated using 3DCP techniques.

As discussed in section 2.1.10, flowability, extrudability, and buildability are all functions of the material's yield shear stress. In order for the material to successfully move through the printing system and to be deposited through the nozzle, it must behave as a fluid when under pressure exerted by the pump. During these stages of the printing process, the applied stress must exceed the material's yield stress. Once the material has been deposited, the material must behave as a solid in order to bear loading. After extrusion, the only forces acting on the material is its own self-weight. This self-weight loading must result in stress distribution where, at every material point, the applied shear stress is beneath the material's yield stress, resulting in solid behavior. The yield stress of the material, then, functions as a design constraint, limiting the geometries that can be successfully fabricated using 3DCP.

From the physical problem presented above, a theoretical topology optimization problem can be developed that will result in a printable design. This problem will be referred to as the "concrete problem." The concrete problem is formulated in 5 parts:

1. Objective Function

The objective function is chosen to take advantage of the potential of 3DCP to produce complex structures without formwork, and to reduce the amount of material necessary in the building process. Therefore, the objective function is to minimize the volume of the structure.

$$\text{minimize: } \frac{\sum(x_i V_i)}{\sum V_i} \quad (21)$$

2. Constraint Function

The constraint is chosen based on material and fabrication limitations. Therefore, the constraint function is the material's yield shear stress. A passive constraint is also imposed on the problem in the form of finite element equations. Stress calculations are performed using displacement results from FEA. Therefore, the stress is constrained by the FEA equations.

$$|\sigma_{xy}|^{max} \leq \sigma_{yield} \quad (22)$$

$$\mathbf{Ku} = \mathbf{F} \quad (23)$$

3. Loading

The applied loading considers the loads acting on the material after it has been extruded. Therefore, the applied loading will be only the structural self-weight.

$$F_{sw_i} = \rho g V_i x_i \quad (24)$$

4. Geometry

The design space in which the structure is to be optimized depends on the architectural constraints of the problem. Initially, the program will be written to allow the user to input the geometry in terms of length, width, and depth of a rectangular prism. This rectangular prism may represent a column, a beam, or another object of the designer's choosing. This work considers both beam objects and column objects.

5. Boundary Conditions

Boundary conditions are set first to consider the surface upon which the structure is to be printed. The 3DCP process deposits material on a flat surface. Therefore, displacement of the bottom surface is to be fixed in the vertical direction. Additional boundary conditions are set and to consider architectural constraints and locations of "future loads," or applied loading on the design once it has been fabricated and cured. Full material placement is forced at each location of future loads, or at each architectural constraint. This work considers wall and column elements with fixed bases.

Combined into a mathematical expression, the TO problem can be formulated by equation 25. It is important to note that, though there are multiple shear stress components along different coordinate planes, for this problem, only shear stress on the x-y plane is considered, as this plane represents the plane of highest shear stress under vertical loading. The problem presented above will be developed in two stages. The first stage represents a proof of concept study, which serves to demonstrate the relationship between material and structural design, as well as the effectiveness of TO as a tool to develop and understand that relationship. To that effort, an open-source TO algorithm is modified to incorporate parts 1, 2, and 3 of the concrete problem. After demonstrating proof of concept, the second stage of this work develops an original TO algorithm designed to incorporate the full concrete problem.

$$\left\{ \begin{array}{l} \text{minimize:} \quad \frac{\sum(x_i V_i)}{\sum V_i} \\ \text{subject to:} \quad \mathbf{K}\mathbf{u} = \mathbf{F} \\ \quad \quad \quad |\sigma_{xy}|^{max} \leq \sigma_{yield} \\ \quad \quad \quad 0 \leq x_i \leq 1 \end{array} \right. \quad (25)$$

where

V_i = Volume of the i^{th} element

x_i = Element density

σ_{xy} = Shear stress in the element

σ_{yield} = Yield shear stress of the material.

\mathbf{F} = Applied force vector

\mathbf{u} = Displacement vector

\mathbf{K} = Stiffness matrix

Chapter 4: Proof of Concept Study

Though research has been done into stress-based optimization, the only open-source code available incorporating a stress constraint was published by Biyikli and To (2015), and was named Proportional Topology Optimization (PTO). This program solves the problem of volume minimization under a VMS constraint, considering an applied point load. Volume minimization in this case refers to the volume fraction of material in the optimized structure relative to the volume of the original rectangular geometry. In addition to being available for use, PTO was chosen as the basis of this work for the following reasons: 1) formulation of a volume minimization problem; 2) incorporation VMS stress constraints, which are similar in formulation to shear stress constraints; 3) utilization of the SIMP method; and 4) code simplicity, in the form of a non-sensitivity based optimization method.

Reasons 1 and 2 describe the similarities in the problems being solved in the PTO case and in the case of the work. This similarity can be seen when comparing the problem statements for each case:

Current Work

$$\left\{ \begin{array}{l} \text{minimize:} \\ \text{subject to:} \end{array} \right. \begin{array}{l} \frac{\sum(x_i V_i)}{\sum V_i} \\ \mathbf{K}\mathbf{u} = \mathbf{F} \\ |\sigma_{xy}|^{max} \leq \sigma_{ys} \\ 0 \leq x_i \leq 1 \end{array}$$

where

V_i = Volume of the i^{th} element
 x_i = Element density
 σ_{xy} = Shear stress in the element
 σ_{ys} = Yield shear stress of the material.

PTO

$$\left\{ \begin{array}{l} \text{minimize:} \\ \text{subject to:} \end{array} \right. \begin{array}{l} \frac{\sum(x_i V_i)}{\sum V_i} \\ \mathbf{K}\mathbf{u} = \mathbf{F} \\ \sigma'_{vms} \leq \sigma_{vms}^{max} \\ 0 \leq x_i \leq 1 \end{array}$$

where

σ'_{vms} = Maximum VMS in the domain
 σ_{vms}^{max} = VMS constraint

In addition to the material constraints, each case is subject to the finite element equation relating applied force \mathbf{F} to the nodal displacements \mathbf{u} through the stiffness matrix \mathbf{K} . \mathbf{K} is developed for 2D 4-node square elements using the stiffness equation:

$$\mathbf{K}_i = \int_{V_i} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (26)$$

where \mathbf{D} is the elasticity matrix and \mathbf{B} is the strain displacement matrix. For a 2D 4-node square element, \mathbf{D} and \mathbf{B} are given as:

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (27)$$

$$\mathbf{B} = \frac{1}{2L} \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix} \quad (28)$$

The elemental stresses are then calculated from

$$\boldsymbol{\sigma}(x_i) = \mathbf{D}(x_i)\mathbf{B}\mathbf{u}(x_i) \quad (29)$$

resulting in the stress vector $\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}$ from which the shear stresses σ_{xy} are extracted.

Incorporation of the SIMP method is desirable due to its effectiveness in stress-based topology optimization research (Duysinx, 1998; Lee 2012). PTO uses a modified form of SIMP developed by Andreassen (2011) where the stiffness is penalized as follows:

$$E(x_i) = E_{min} + x_i^P E_0 \quad (30)$$

Additionally, PTO attempts to simplify the optimization problem by introducing a non-sensitivity-based approach to generate a solution. PTO assigns material to each element based on the level of Von Mises stress in that element during the previous iteration. The amount of material available for spatial allocation in each iteration is determined by adding or subtracting a discrete, fixed amount (0.5% of the original design volume) to or from the amount available in the previous iteration. The placement of the available material is then determined based on the relative stress levels in each element. Constraints are then considered globally, and the entire structure is optimized based on the maximum elemental VMS.

In order to adopt this code to solve our proposed problem, three basic changes were necessary: (i) addition of self-weight loading; (ii) impose shear stress constraint and new optimization logic; (iii) incorporate physical quantities and material properties. These changes were implemented as follows (full code provided in Appendix C).

4.1 Addition of Self-Weight Loading

Lines 88-99 were added to the existing PTO code to incorporate self-weight loading into the optimization problem. Line 89 initiates an empty sparse matrix F_{sw} to store self-weight loading values at the nodes. Lines 90-95 add the self-weight loading to each node individually by locating the corresponding nodal Degree of Freedom (DOF) from the element degrees of freedom matrix $edofMat$.

```

89     F_sw = sparse((nelx+1)*(nely+1)*2,1);
90     for i = 1:nelx*nely
91         for j = 2:2:8
92             DOF = edofMat(i,j);
93             F_sw(DOF) = F_sw(DOF)+(.25*den*MN_Factor*V*g)*x(i);
94         end
95     end
96     F_total = F_sw+F_applied;
```

Since PTO is limited to 2D geometries, self-weight was simulated by assigning to each element a unit volume, and calculating the gravitational force acting at each node due to that volume. The nodal forces, defined in line 93, are applied to each node based on the self-weight (F_{sw}) of the surrounding elements, which is calculated as follows:

$$(F_{sw})_n = \sum_{i=1}^m \frac{1}{4} \rho g V_i x_i \quad (31)$$

where

- n = node
- m = number of nodes adjacent to node n
- ρ = physical density of the material
- g = gravitational constant
- V = volume
- x_i = element density

Corner nodes therefore carry only $\frac{1}{4}$ of the elemental self-weight, edge nodes carry $\frac{1}{4}$ of the self-weight of two adjacent elements, and interior nodes carry $\frac{1}{4}$ of the self-weight of four adjacent elements. Figure 19 shows how the self-weight is distributed from each element to each adjacent node. The resulting self-weight is converted to Mega-Newtons using the MN_Factor , as discussed later. Line 96 then calculates the total force vector \mathbf{F} by combining the applied and self-weight loadings as

$$\mathbf{F} = \mathbf{F}_{sw} + \mathbf{F}_{applied} \quad (32)$$

This section is nested at the beginning of the optimization loop (line 85), so that it runs before finite element analysis and after optimization step.

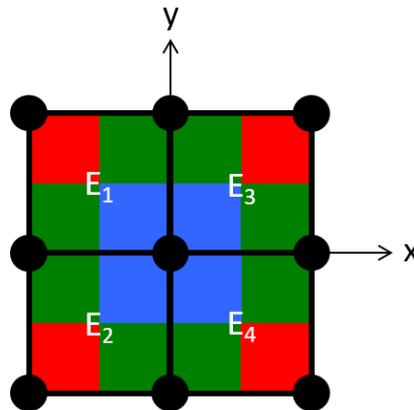


Figure 19: Nodal distribution of self-weight from each element. Red areas attributed to corner nodes, green areas to edge nodes, and blue areas to the central node.

4.2 Shear Stress Constraints and New Optimization Criteria

The original PTO code was optimized considering a VMS constraint, meaning it searched for the optimal arrangement of material that would minimize the volume fraction while making sure the VMS in each element was less than or equal to a designated value. The optimization process would end when the difference between the maximum calculated VMS and the constraint was below a certain value (0.001). Due to its nature as a positive summation of all stress components, and due to the lack of self-weight consideration, VMS would always rise when the volume fraction decreased, and decrease as the volume fraction increased, so the constraint would always be reached. Therefore, the optimization process would add material when VMS was higher than the constraint, and remove material when it was lower, and only one stopping condition was necessary.

On the other hand, the shear stress constraint functions quite differently. Depending on the loading and geometry, the shear stress could increase or decrease with addition or subtraction of material. As a result, changes in both shear stress and material volume must both be considered. Due to the lack of sensitivity analysis in PTO, simultaneous consideration was not possible. Therefore, new optimization criteria and stopping conditions were required. Figure 20 shows the optimization logic of the VMS case (a) and that developed for the shear stress constrained optimization (b). It is important to note that this optimization logic is applied to the global structure, determining material placement at every element during each iteration.

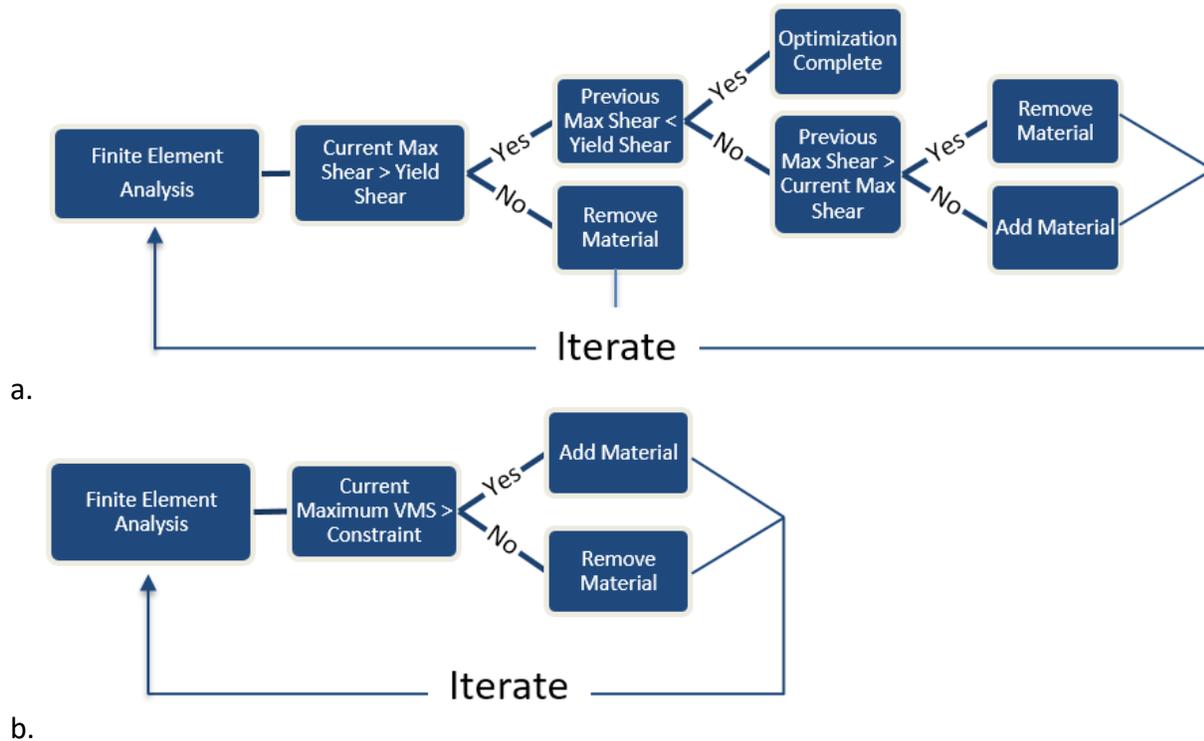


Figure 20: a. Shear stress constraint optimization logic; b. VMS constraint optimization logic.

Implementation of the shear stress optimization code begins as follows:

```
108 s = (U(edofMat)*(DE*B)')*repmat(E',1,3);
109 sxmat = reshape(s(:,1),nely,nelx);
110 symat = reshape(s(:,2),nely,nelx);
111 shear_mat = reshape(s(:,3),nely,nelx);
112 abs_shear_mat = abs(shear_mat);
113 cur_max_shear = max(abs_shear_mat(:));
```

Line 108 calculates the stress matrix resulting from finite element analysis. Lines 109 and 110 extract the normal stress matrices *sxmat* and *symat* from the stress matrix *s*, and line 111 extracts the shear stress matrix *shear_mat*. The shear matrix is then transformed into an absolute value matrix *abs_shear_mat* in line 112, as the directionality of shear is not under consideration. Line 113 then determines the global maximum shear value for the current iteration.

The shear stress constraint was implemented by adding stopping conditions that would end the optimization process based on the shear stress value and volume fraction. Lines 121-131 encompass the main optimization section. The amount of material added or subtracted during each iteration is proportional to the number of elements in the volume, and the proportion was increased from 0.001 to 0.005 in order to speed up the optimization process (lines 125, 127, and 130).

```
121 if (cur_max_shear > yield_shear)
122     if prev_max_shear < yield_shear
123         disp('Optimization Complete < Yield Shear'); break; end
124     if prev_max_shear > cur_max_shear
125         xTarget = sum(x(:))-0.005*numel(x);
126     else
127         xTarget = sum(x(:))+0.005*numel(x);
128     end
129 else
130     xTarget = sum(x(:))-0.005*numel(x);
131 end
```

Additional stopping conditions were added to the code as well. Lines 149-156 define the end conditions where the minimum possible volume fraction (0.01) is reached and the structure satisfies the shear stress constraint (lines 150-152) and where an empty volume results (line 153-155).

```
149 if cur_vol_fract < 0.01
150     if cur_max_shear < yield_shear
151         disp('Minimum Volume Reached');
152         break;
153     else
154         disp('Empty Volume');
155     end
156 end
```

After running the optimization process and checking the stop criteria, and before the next iteration, the previous values for volume fraction and maximum shear stress are updated with the current values for comparison during the next iteration. This update is performed in lines 171-172.

```
171     prev_vol_fract = cur_vol_fract;  
172     prev_max_shear = cur_max_shear;
```

4.3 Incorporation of SI Units

The original PTO code is unit-less. Therefore, while serving as a useful theoretical tool, it has no physical basis and cannot make quantitative predictions. In order to use this tool to solve physical problems, units must be added to allow inclusion of measurable quantities. The accompanying literature provides a suggestion for unit incorporation, however, when adopted, large values for Young's Modulus resulted in poorly scaled matrices. Therefore, the base units incorporated into the code were meters (m), Newtons (N), and Mega-Pascals (MPa). MPa was chosen for its common use as descriptor of material strength, and also due the poorly scaled matrix discussed above. The beam length and height, element size, and volume are given in meters, the applied load is given in Newtons, and the shear stress constraint and Young's Modulus are given in MPa. Material density is given in $\frac{kg}{m^3}$ and gravity in $\frac{m}{s^2}$. Both the applied load and self-weight loading are converted into Mega-Newtons before stress calculations using the *MN_Factor* of 10^{-6} .

4.4 Results and Discussion

4.4.1 Effect of Self-Weight

In order to demonstrate the effect of self-weight loading on the optimized design, the unit-less program was used to run a single load case, Case 1, for fictitious materials with varying densities. The optimization problem was defined for a rectangular domain of 100 x 50 unit elements. The geometry simulated a cantilever beam with a fixed left edge and a point load mid-height on the right edge. Figure 21 shows the problem set-up.

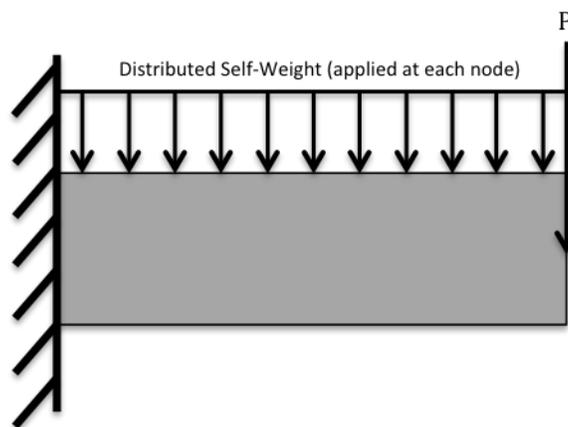


Figure 21: Geometry, loading, and boundary conditions for Case 1. Distributed self-weight applied at each node (refer to Fig. 18).

The applied load P was set to 150, and the material properties were kept as those set in the original code: Young's modulus was defined as 1, and Poisson's ratio was given as 0.3. The densities considered were 0, 0.5, and 1, with a density of zero corresponding to no self-weight. The structure was then optimized for volume minimization under a shear stress constraint. Figure 22 shows the resulting optimized designed for each density, and Table 1 displays the numerical results.

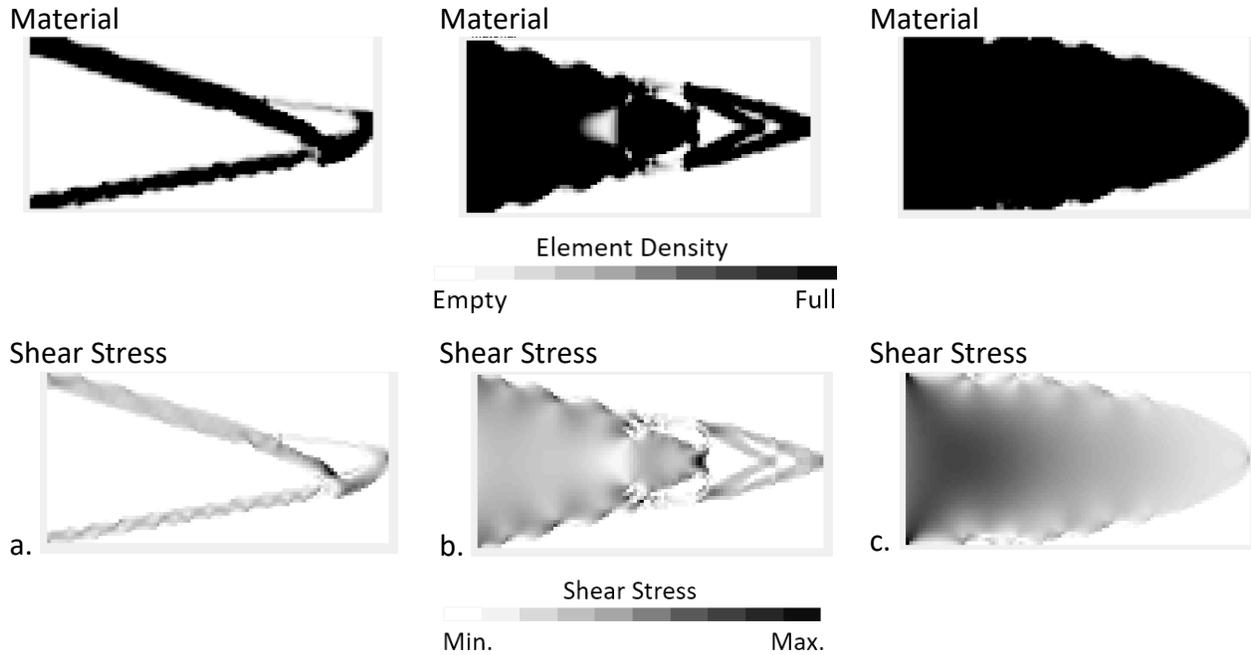


Figure 22: Case 1 optimized geometry with varying material density. a. Density = 0; b. Density = 0.5; c. Density = 1.0

Table 1: Numerical results from Case 1.						
Fig.	Material Density	Shear Stress Constraint	Iteration	Result	Maximum Shear Stress	Volume Fraction
a	0	100	135	Completed	68.68	0.222
b	0.5	100	74	Completed	91.27	0.529
c	1	100	51	Failed	127.801	0.814

From these results, it can be seen that the design with the least amount of material corresponds to the no self-weight case (density = 0, Case 1 (a)). This result was expected considering the additional self-weight load would result in higher shear stresses, requiring more material over which to distribute force to meet the constraint, as seen in Case 1 (b). It is also important to note that the optimization corresponding to a material density of 1 failed. The optimization process was designed to end after 50 iterations if the shear stress constraint was not satisfied, and if placement of additional material increases the shear stress instead of decreasing it. If this pattern occurred, the shear stress constraint could never be reached. Case 1 (c) with a material density of 1 could not be optimized with a maximum shear stress of 100, indicating there is no possible arrangement of material within the design volume that would satisfy the constraint. Since this simulation does not have a physical basis of real material properties and physical loads, the reason for this failure is best understood by comparing material property values and their relationship to the applied load. For Case 1 (c), the material density and material stiffness are of equal value. This equality translates to an extremely dense material. For

example, if the unit-less quantities used in this case were simulated with real units, a material with stiffness of 10 GPa would have a density of $10,000,000,000 \frac{kg}{m^3}$, the self-weight of which would cause very high shear stresses. Although this is clearly an unrealistic problem, such values in Case 1 (c) show that material properties place physical limits on the range of possible structural designs. This limit is important when considering the material properties of fresh concrete mortar used for 3D printing, as strengths of these materials are generally very low. Physical limits on structural designs will be discussed further in later sections.

In order to rule out algorithm error as the cause of optimization failure in Case 1 (c), a new problem was posed by slightly altering the geometry. Figure 23 shows the successfully optimized design of Case 2, with initial geometry 60 x 30, keeping all other variables the same as the Case 1 (c), including density = 1. These results show that the algorithm worked properly, and that optimization failure in Case 1 (c) resulted from a poorly posed problem.

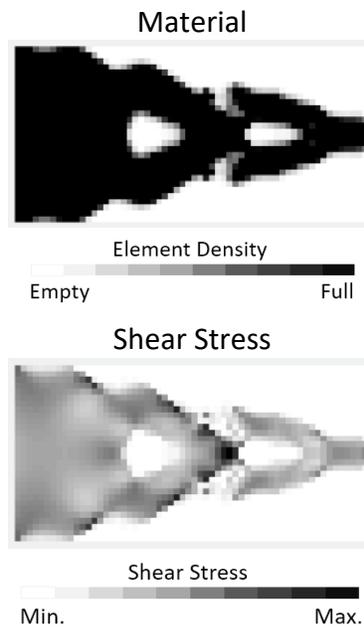


Figure 23: Case 2 optimized structures.

Table 2: Case 2 numerical results.					
Density	Constraint	Iteration	Result	Max Shear Stress	Volume Fraction
1	100	65	Completed	93.432	0.571

4.4.2 Validation of Optimized Results

In order to validate the optimized results, finite element analysis was performed on an optimized design case, Case 3, using ANSYS Mechanical software. Specifically, the optimized structure was reproduced in ANSYS, along with the applied loads, and the shear stress distribution from the code and from ANSYS were compared. Successful validation would show similar distribution and magnitude of

shear stress everywhere within the part, and all shear stress values in the simulated result would remain below the shear stress constraint. The process was as follows:

First, a material for consideration was chosen. Concrete was selected from the General Materials Library within the ANSYS material database, and the values below were incorporated into the optimization code.

Density = $2300 \frac{kg}{m^3}$,	Element Side Length = 0.01 m
Young's Modulus = 30 GPa	Beam Length = 1 m
Poisson's Ratio = 0.18	Beam Width = 0.5 m

The shear stress constraint was chosen based on the problem formulation as a cantilever beam with a point load, considered for a structural application. The American Concrete Institute (ACI) defines the maximum allowable shear stress for hardened structural concrete by the following equation

$$V_c = 0.17\lambda\sqrt{f'_c}bd \quad (\text{from ACI 318M-11 11.2.1.1 (11-3)}) \quad (33)$$

where

- f'_c = concrete compressive strength
- λ = weight factor, given as 1 for normal weight concrete
- b = beam height
- d = beam depth

The shear stress limit for Case 3 given by the above equation is 1.1 MPa. A load of 30 KN was applied to the same cantilever geometry as shown in Figure 21, while still considering the effect of self-weight. The structure was then optimized for minimum volume fraction using the new optimization code, and the resulting optimized design was drawn by hand using the ANSYS CAD tool, SpaceClaim. Since the optimization process does not attempt to smooth jagged edges or sharp curves, a design realization step was introduced before running FEA in order to analyze a more realistic component. Best efforts were made to keep the design geometry consistent.

The greatest deviation from optimized design to the ANSYS model was the addition of depth. In order for ANSYS to simulate the effects of gravity and incorporate self-weight, the geometry must be drawn in 3D. In contrast, the optimized design is only a 2-dimensional structure, with self-weight simulated as described in earlier sections. Therefore, a depth the size of one element was used, and forces were assigned symmetrically in order to minimize the effect of the addition of depth, and to simulate a thin, planar structure. The shear stress calculation for the 2D optimized design assumes the third dimension to have a unit length. However, a unit length in this case would be 1 meter, which is not physically applicable for this problem. When a physical depth of 0.01 m is added to the ANSYS model instead of unit depth, the amount of material available to carry load is reduced by a factor of 100. Therefore, the ANSYS shear stress results were expected to be higher than that of the optimized results by a factor of 100. Optimized and ANSYS results for Case 3 are displayed in Figure 24.

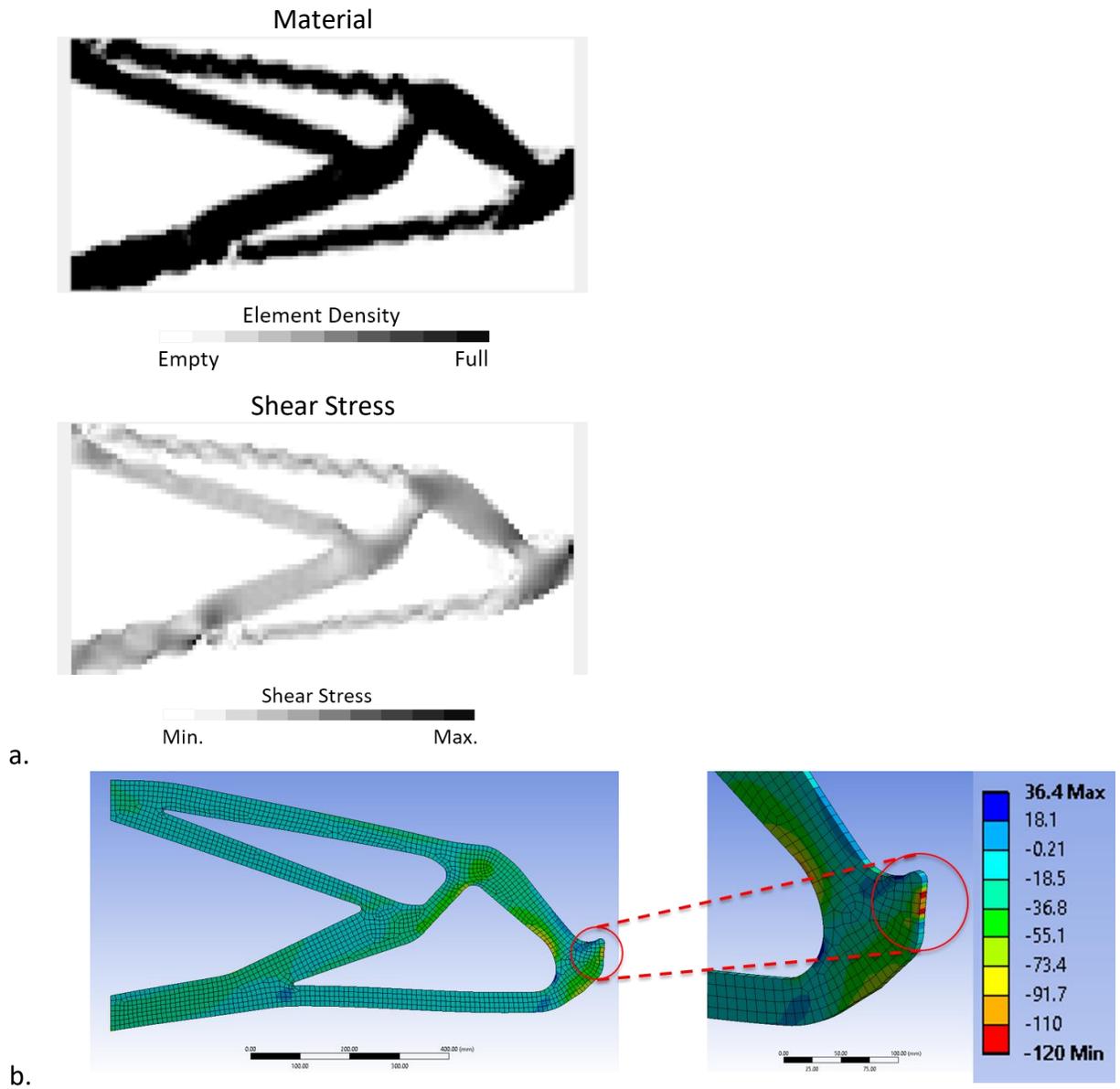


Figure 24: Case 3; a. Optimized geometry (top) and shear stress distribution (bottom); b. ANSYS model shear stress distribution. Red areas indicate shear stresses higher than constraint. All results in MPa.

Visual inspection of Figure 24 shows a consistent stress distribution when comparing the optimized result (Figure 24a, bottom) and the ANSYS result (Figure 24b). Dark spots in the optimized result correspond to higher relative stresses, while the ANSYS results display green/yellow (negative) or dark blue (positive) at locations of higher stress, and light blue for lower stress. The thin arms of the structure contain low stresses in both results, and thicker areas towards the left edge at top and bottom display relatively higher values. Additionally, higher stresses can be seen at areas of intersection and curvature.

Only one very small region was found where shear stress exceeded the constraint, corresponding to the area where the point load is applied. In the ANSYS model, the 30 KN load was divided equally over 6 nodes corresponding to three elements, so higher shear stresses at these nodes are artifacts due to point contact of the applied load. All other elements hold shear stresses below the constraint, indicating that the design has been successfully optimized for the shear stress constraint. Additionally,

a quantitative comparison of shear stresses is presented in Table 3, including adjustments for depth. Due to differences in mesh and the deformed geometry in ANSYS, stress values at specific locations could not be compared directly. Locations were defined by stress probes in ANSYS, and best attempts were made to locate the corresponding elements in the Matlab shear stress matrix. Locations in Table 3 correspond to those in Figure 25.

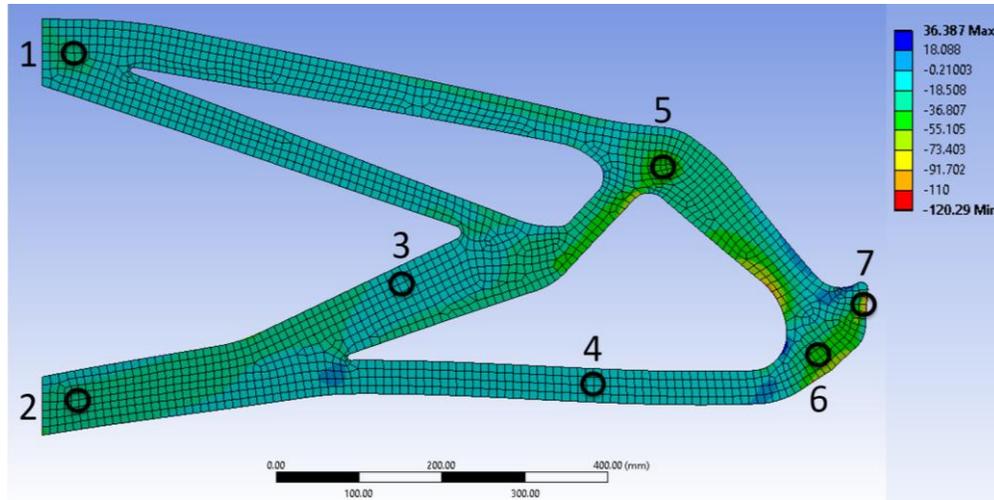


Figure 25: Case 3 selected locations for shear stress comparison. All results in MPa.

Location	Code	ANSYS	Adjusted ANSYS	Percent Difference
1	0.219	21.645	0.216	1%
2	0.164	15.839	0.158	3%
3	0.251	26.768	0.268	7%
4	0.084	6.230	0.062	26%
5	0.391	45.975	0.460	18%
6	0.476	53.371	0.534	12%
7	0.625	120.290	1.203	92%

Another simulation, Case 4, was performed to solve a problem identical to Case 3 while reducing the material density from $2300 \frac{kg}{m^3}$ to $23 \frac{kg}{m^3}$. Case 4 was run both to show the effect of self-weight and to confirm accuracy. Optimization and ANSYS results are displayed in Figure 26. Again, artifacts at the location of the applied point load result in shear stresses higher than the constraint. All other areas satisfy the shear stress constraint. The resulting design shows a volume fraction reduction of 6.5% from the original, from 0.358 to 0.293. With a density of $2300 \frac{kg}{m^3}$, the weight of the original structure is equivalent to a 40 N force, and the reduced density structure has a weight equivalent to a 0.33 N force. The reduction in density has a significant effect on the optimized structure considering the weight reduction to applied load ratio of 0.0013. This effect of density on optimized structure is an important result, especially for the case of concrete 3D printing. By reducing the density of the concrete mortar, and if the shear strength can be maintained, it may be possible to significantly reduce the amount of material used, and to create a wider range of printable geometries.

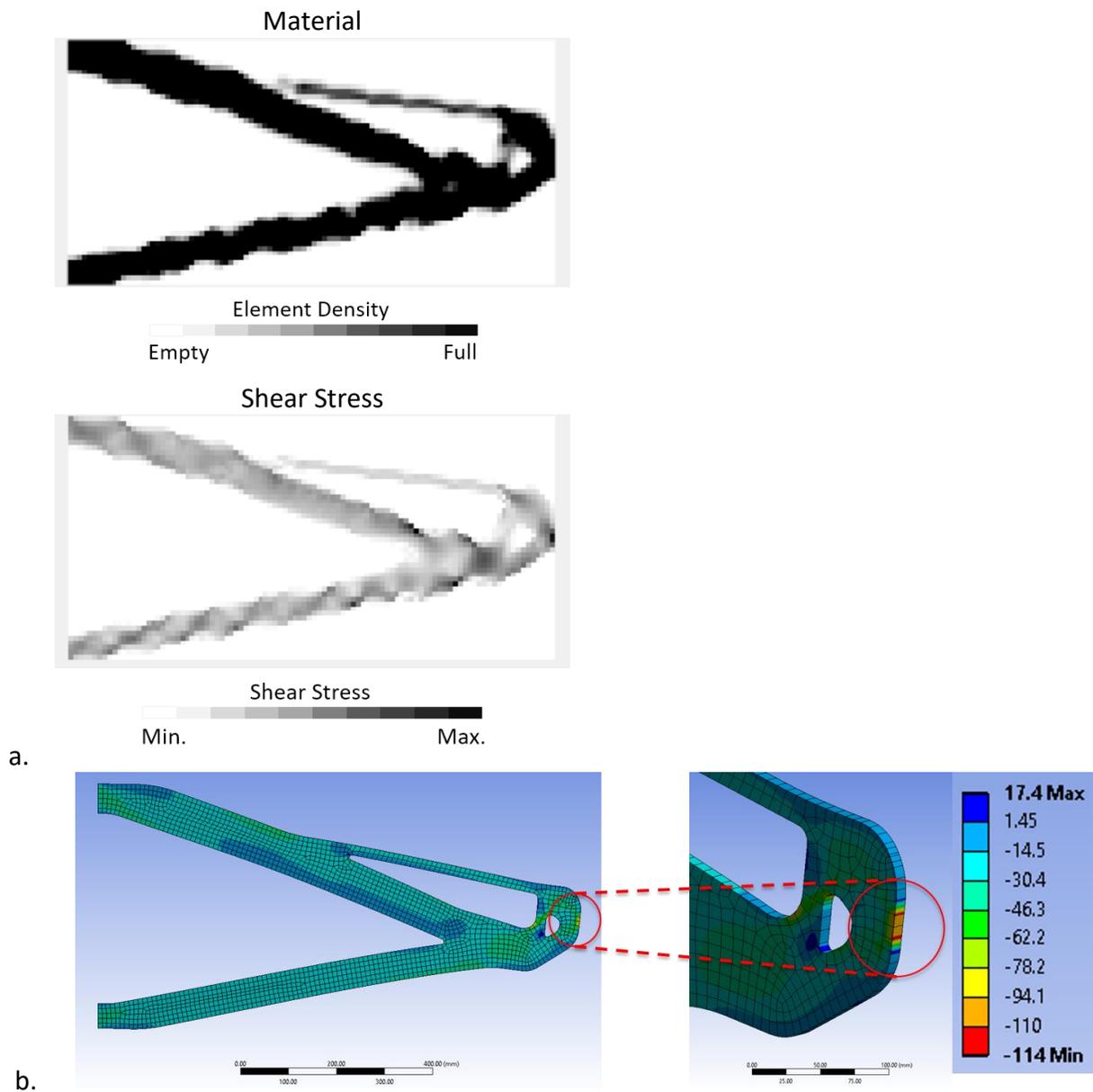


Figure 26: Case 4; a. Optimized geometry (top) and shear stress distribution (bottom); b. ANSYS model shear stress distribution.

4.4.3 Optimization Limitations

Combining the heuristic nature of PTO and the newly implemented optimization logic results in structures that are optimized only to a local minimum. This limitation can be understood by the lack of sensitivity analysis and by the constant optimization parameters: penalty coefficient, proportion exponent, and filter radius. The local minima behavior can also be demonstrated by manually relaxing the stress constraint, as presented in Case 5. Figure 27 below shows the optimization progression of Case 5, which mirrors Case 4, but with a shear stress constraint of 1.5 MPa as opposed to 1.1 MPa. Figure 27 (a) corresponds to the final iteration of Case 4.

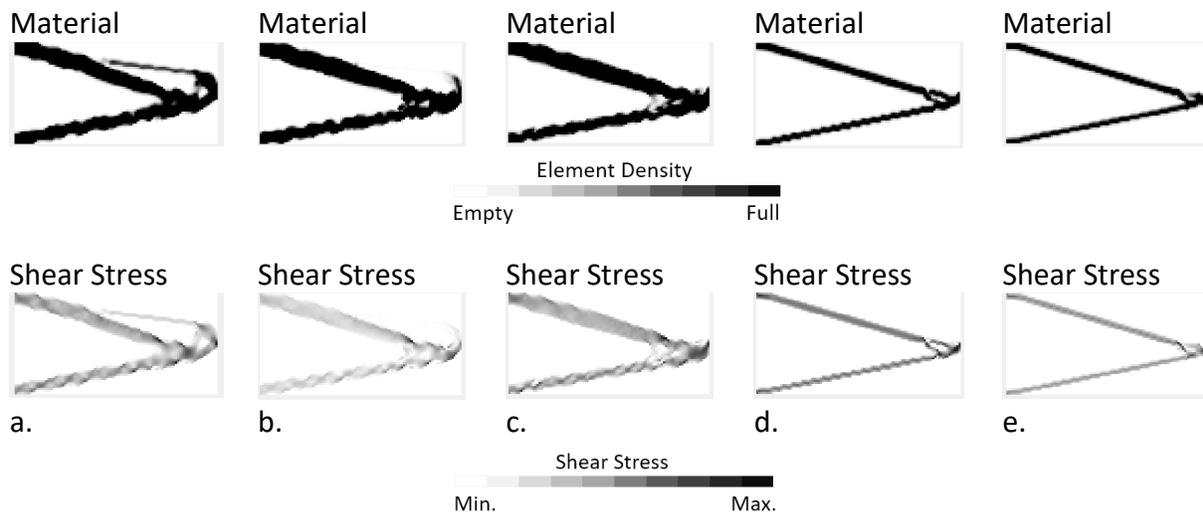


Figure 27: Optimization progression for Case 5 at various iterations (IT). Numerical results shown in Table 4.

Table 4: Numerical results for Case 5.					
Fig.	Constraint (MPa)	Iteration	Result	Max Shear Stress (MPa)	Volume Fraction
a	1.5	121	Running	0.738	0.293
N/A	1.5	122	Running	1.137	0.288
b	1.5	123	Running	1.398	0.283
c	1.5	124	Running	1.175	0.278
N/A	1.5	125	Running	0.732	0.273
d	1.5	151	Running	1.02	0.142
e	1.5	154	Completed	1.483	0.127

Between iterations 121 and 154, the volume fraction is constantly decreasing, as is the objective of the optimization problem. While the maximum shear stress increases above the original constraint of 1.1 MPa for iterations 122 – 124, it then reduces below that constraint between iterations 125 and 151, before arriving at the optimized solution for the 1.5 MPa constraint at iteration 154. Iteration 151 shows what would also be a minimum for Case 4. Therefore, resulting designs may not be fully optimized due to the optimization logic that was created to account for the lack of sensitivity analysis in the original code. Instead of ending the process after one instance of increased volume fraction, or after the first iteration exceeding the shear stress constraint, the optimization should be controlled by the rates at which these values change, relative to the constraint. As such, this work indicates the necessity of sensitivity analysis in shear stress constrained design. Sensitivity analysis incorporates the rate of change of design variables, and therefore allows convergence to be the determining factor when arriving at a solution, as opposed to a single value threshold. This convergence, as well as the ability to incorporate sensitivities of multiple variables, can lead to a better-defined solution.

Along with sensitivity analysis, further methods must be employed in order to reach a globally optimized result. Rasmussen (2008) adopts the cut-and-branch method, a variation on branch-and-bound, to solve a discreet truss optimization problem, while Sigmund (1998) concludes that continuation methods must be used. These continuation methods typically involve incrementally changing the optimization parameters until convergence. Due to the complex and large-scale nature of topology optimization problems, many are only solved to local minima.

4.4.4 The Material-Structure Relationship

The optimization code herein can be used to aid in the material selection process. Consider the Ashby Chart in Figure 28, relating density to tensile strength, and the problem set-up from Case 3. By defining material properties corresponding to various materials from the CES material database, unique optimized results can be obtained. If a range of values was given in the database, the average value was used for this case.

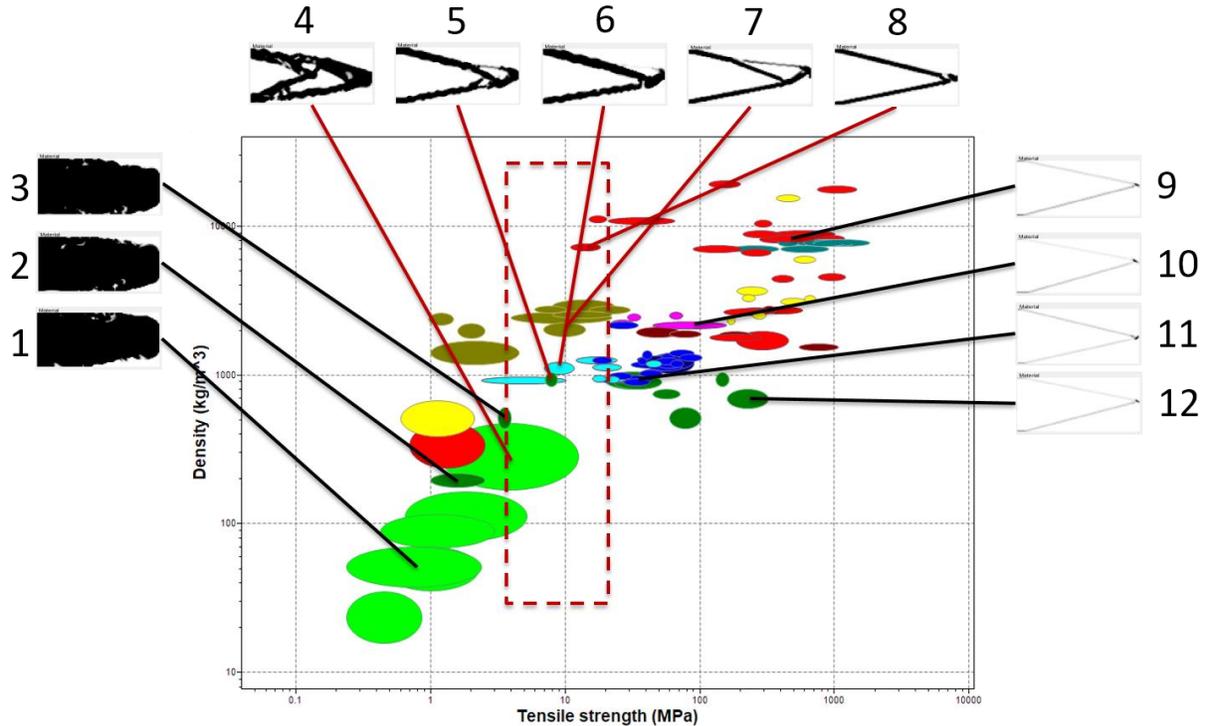


Figure 28: Ashby diagram of Density vs. Tensile strength of engineering materials (CES Edupack, 2019). Optimized structures from Case 3 shown for several selected materials. Average material property values were used if given a range of values in the software.

	Material	Tensile Strength	Yield Shear Stress	Young's Modulus	Density (kg/m ³)
1	Flexible Polymer Foam (LD)	1.30	0.65	2.00	54.00
2	Cork	1.75	0.88	37.50	197.50
3	Pine	3.55	1.78	750	520
4	Rigid Polymer Foam (HD)	6.80	3.40	0.34	320.00
5	Oak, across grain	7.9	3.95	5290	940.5
6	Silicone Elastomers	9.25	4.63	27.5	1120
7	Brick	10.46	5.23	23300.00	2035.00
8	Lead	17.5	8.75	13500	1130
9	Bronze	511.00	255.50	113500.00	8375.00
10	Silica Glass	100.00	50.00	71000.00	2195.00
11	Polypropylene (PP)	34.50	17.25	1223.00	900.00
12	Bamboo	239.50	119.75	17500.00	700.00

It is important to note here that this simulation assumes all materials to be linear, elastic, isotropic, and homogeneous. This assumption is not true for all materials simulated here, but will be used for demonstrative purposes. Therefore, using plane stress transformation equations, the yield shear stress can be calculated as follows:

$$\sigma_{ys} = \frac{\sigma_1}{2} \tag{34}$$

σ_{ys} = yield shear stress

σ_1 = principal stress = tensile stress

A total of 12 different materials or material categories were simulated, considering an applied load of 100 KN. Materials with tensile strength less than 3 MPa (Flexible Polymer Foams, Cork, Pine) cannot be optimized, as the process failed due to the constraint being unreachable. Materials with tensile strength greater than 12 MPa, however, result in structures with the minimum possible volume fraction (Low Alloy Steel, CFRP, Bronze, Tungsten Carbides, Bamboo, Silica Glass, Polypropylene). The results allow for consideration of a “sensitive range”, from 3 – 12 MPa (Rigid Polymer Foam, Oak, Silicone Elastomers, Brick, Lead), where minor differences in material properties can cause significant design changes. Materials with tensile strength less than 3 MPa are not suitable for design, while any material selected with a tensile strength greater than 12 MPa will result in the a similarly optimized structure. This region is therefore not sensitive to material property changes, where even significant increases in tensile strength will result in similar designs. Figure 29 demonstrates this relationship between tensile strength and volume fraction by plotting the results from Table 5. The design engineer may use this idea of a sensitive region to understand the sensitivity of the design with respect to material changes or improvements.

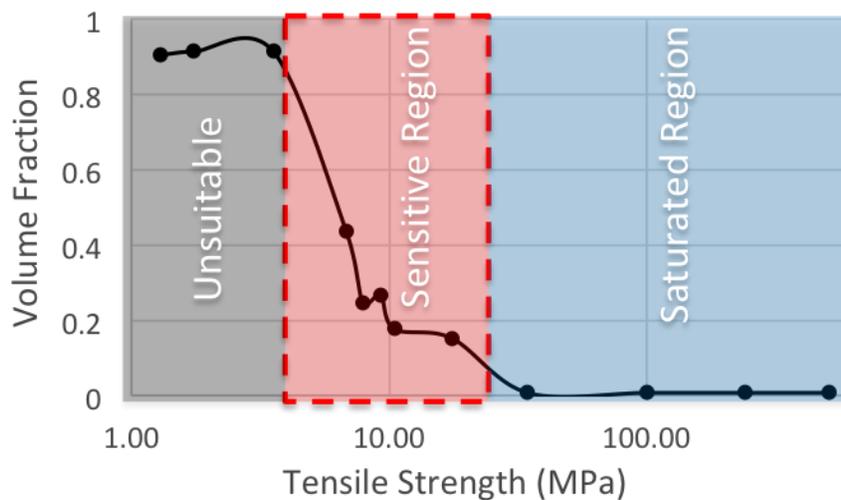


Figure 29: Volume fraction of case 3 optimized structures vs. tensile strength for materials in Table 5.

Results were also plotted showing the relationship between stiffness and tensile strength (Figure 30). Again, a sensitive region can be seen, demonstrating the structural dependence upon tensile strength. Though this simulation is an over-simplification of a complicated process, it demonstrates the potential usefulness of shear stress-constrained topology optimization in material selection and structural design.

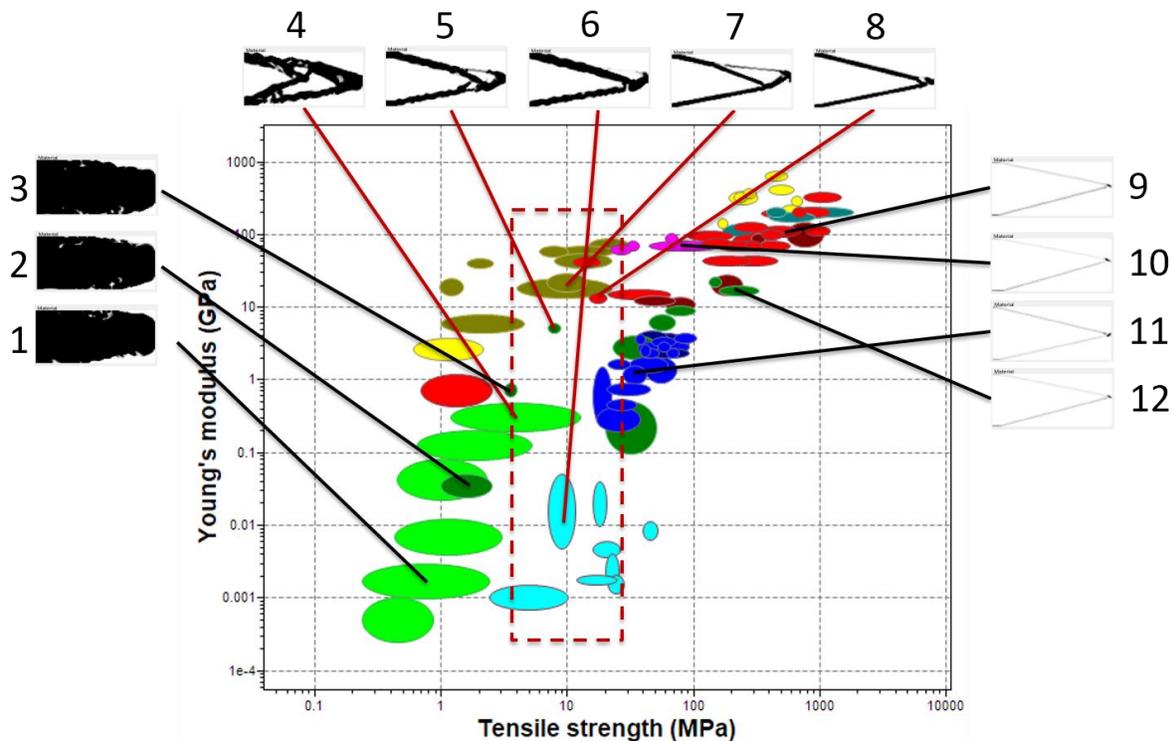


Figure 30: Chart of Young's Modulus vs. Tensile strength of engineering materials (CES Edupack, 2019).

This proof of concept study extends the existing topology optimization code PTO to solve the problem of volume minimization under a shear stress constraint and incorporated the effects of material self-weight into the optimization process. This new optimization process can be used to show general trends in optimized designs based on material properties and the design domain. It allows the user to understand the effects of changing material properties on structural design, and it gives insight into how geometry can limit the materials available for design. By performing parametric studies on the considered material properties of yield shear stress and density, the optimization process can show not only the material properties necessary to achieve a viable structure, but also the structures that can result from consideration of different materials. Materials could potentially be designed to match the density or shear stress constraint in the hopes of being used to create a certain structure, or conversely, structures could be optimized and designed considering the constraints of the material. In this way, the design engineer would better understand the material and design spaces available for a given problem.

Further work must be performed to refine the optimization process to produce fully optimized structures. Sensitivity analysis and methods for convergence at a global minimum must be incorporated into the optimization process. Code should also be developed to consider 3D geometry in order to more accurately model self-weight loading, and to fully realize geometric and material limitations. Additionally, the computational time of the optimization process should be accounted for, with an attempt made at time reduction. The code in its current state requires significant time to optimize geometries over 10,000 elements. This is due to the significantly larger force matrix that results from self-weight loading. Further investigation into implementation of self-weight loading may reduce the processing time, which will be necessary for consideration of real-world load cases. In order to optimize a design for 3D printable concrete, the problem set-up must also be changed to consider an initial design volume with a fixed base and no point load.

Chapter 5: Development of Shear Stress-Constrained Topology Optimization (SSTO) Code

As mentioned previously in section 2.2, there are many open-source TO programs available. None of these available codes, however, are suitable for the concrete problem as presented in this work, as they do not incorporate shear stress constraints, and often omit self-weight loading. Therefore, a new TO program was developed for the purposes of solving the concrete problem, and for demonstrating the relationship between materials, structure, and design. This new code is known as Shear Stress-Constrained Topology Optimization code, or SSTO. In order to develop this new code, an available TO program named Top3D (Liu, 2014) was chosen to serve as a guide.

As it is written, Top3D solves a compliance minimization problem subject to a volume fraction constraint. With minor modification, it is also able to solve heat conduction problems, which minimize or maximize the transfer of thermal loads, and compliant mechanism problems, which maximize output displacements relative to input forces.. The program employs a modified SIMP method for determining material presence (eqn 35), the analytical method for sensitivity analysis, a density filter, and the Optimality Criteria method for optimization. Additionally, the accompanying literature provides directions for incorporating different optimization techniques such as MMA or SQP, as well as different filtering methods. Geometry is defined in three dimensions, and FEA is performed using hexahedral elements and an analytically prescribed stiffness matrix.

$$E(x_i) = E_{min} + x_i^P (E_0 - E_{min}) \quad (35)$$

Top3D shares several of these requirements and methods with those chosen for concrete problem. The similarities between both programs are as follows:

1. 3D Geometry
2. SIMP Method
3. Analytical Sensitivity Analysis
4. Density Filter
5. MMA Optimization

These similarities allow for Top3D to be used as a guide to develop a new program. In the case of the density filter and MMA optimization implementation, sections of the Top3D code were inserted in their entirety into the new program, as will be discussed later. There are, however, several key differences between the functionality of Top3D and the concrete problem presented here, which require significant additions and modifications. These additions and modifications will be discussed in detail in the following sections, and include boundary and applied loading conditions, self-weight loading, formulation of the stiffness matrix, stress and strain calculations, shear stress constraints and sensitivity analysis, and termination criteria, among others.

Presentation of the new code will proceed in order of the TO requirements listed in section 2.2.4, and the full Matlab code can be found in Appendix D.

It is important to note here, that SSTO is capable of performing both shear stress-constrained and Von Mises Stress-constrained optimization. The user designates the type of constraint (lines 55 and 57,

respectively) and the respective constraint value (lines 56 and 58, respectively). Therefore, this section will discuss the formulation of both shear stress-constrained and VMS-constrained optimization code. Much of the code for these two problems is the same. However, where the code differs, the type of constraint will be indicated before presenting the formulation.

5.1 Geometry

As TO programs perform FEA calculations, the geometry of the design space must be defined in terms of nodes and elements. The number of elements in the x, y, and z directions are specified, as is the side length of each element. The initial design space is currently limited to rectangular prisms, defined by length, width, and height in terms of the number of elements along each coordinate direction. As will be discussed in the FEA section, each element is cubic in shape, so only one side length is defined, given in meters. Finally, the initial artificial density of each element is defined as a starting point from which optimization may proceed. For simplicity's sake, each element is assigned the same initial density. This initial density should not equal 0, and ideally be greater than or equal to 0.5 to allow for optimization to initiate properly. Otherwise, the structure may result in a minimum volume termination case. These values can all be changed by the user in lines 15 – 19.

The origin of the coordinate axes is placed at the bottom, left, rear corner of the design space, as shown in Figure 31. The elements are each assigned an element identification number (EID) starting with 1 at the top, left, rear corner. Each element in row z is numbered first, proceeding column-wise in the positive x direction, before moving to the subsequent row in the positive z direction. Each element is defined, not only by number, but also by eight nodes that correspond to the corners of the element. These nodes each have their own global and local node identification number (NID), as well as both global and local coordinates (Figure 32). Global NIDs are defined in the same manner and direction as EIDs, and the global coordinates are defined relative to the origin and are based on the element side length. The local NIDs are linked to the element to which they belong, and are assigned from 1-8 corresponding to the pattern in Figure 33. Local coordinates are given in terms of the ζ_1 , ζ_2 , and ζ_3 directions, with the local origin placed at the center of the element. Each face of the element is a distance of one length unit from the origin, perpendicular to one of the three coordinate axes.

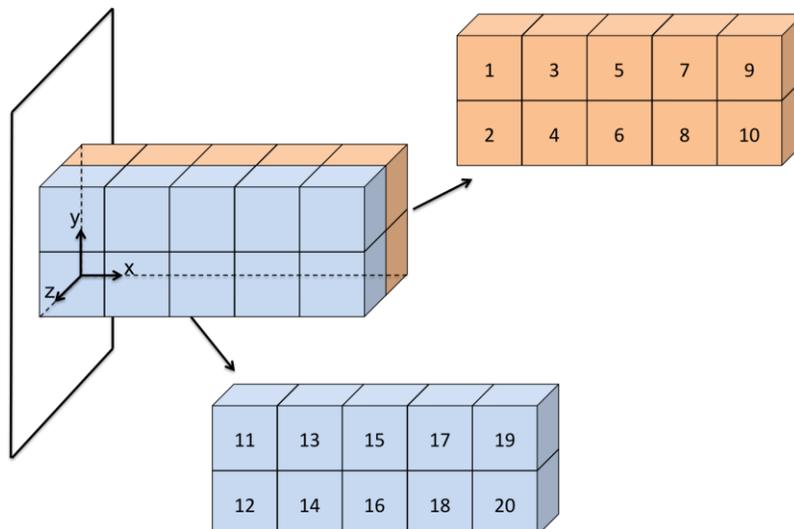


Figure 31: SSTO element IDs.

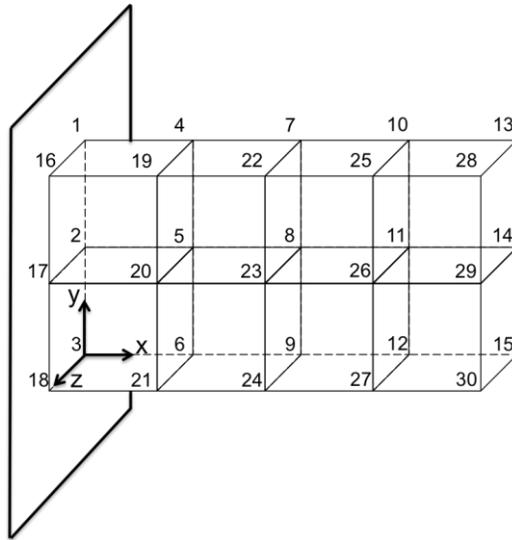


Figure 32: SSTO global node IDs.

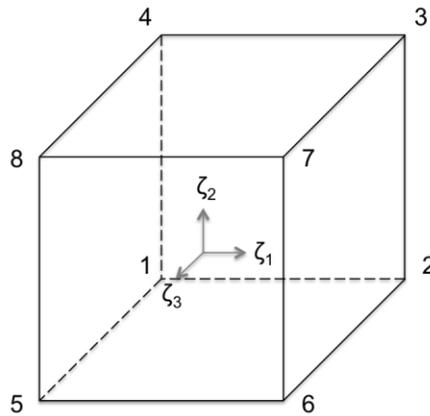


Figure 33: SSTO local node IDs.

In FEA, each node has the potential to be displaced in any of three directions x , y , and z . This directional freedom of movement in a specified direction is referred to as a degrees of freedom (DOF). Each of the three DOF assigned to each node is given a unique number corresponding to the node and direction of movement. Once numbered, these DOF are then assigned to each element for use in calculations.

Geometric information and IDs are assigned to nodes in the *nodal_data* function (lines 421 – 435) and elements in the *element_data* function (lines 438 – 477). In both functions, the information is stored in structure data types.

5.2 Boundary Conditions

The various boundary conditions that can be implemented in SSTO are given in lines 22 – 24 and implemented in the *BCs* function (lines 371 – 418). Any of these conditions may be turned on or off, and multiple conditions can be used simultaneously. The conditions are as follows:

- BC 1. Top perimeter fixed (lines 402 – 417) (Figure 34a)
- BC 2. Bottom perimeter fixed (lines 385 – 400) (Figure 34b)
- BC 3. Cantilever (lines 376 – 384). (Figure 34c)

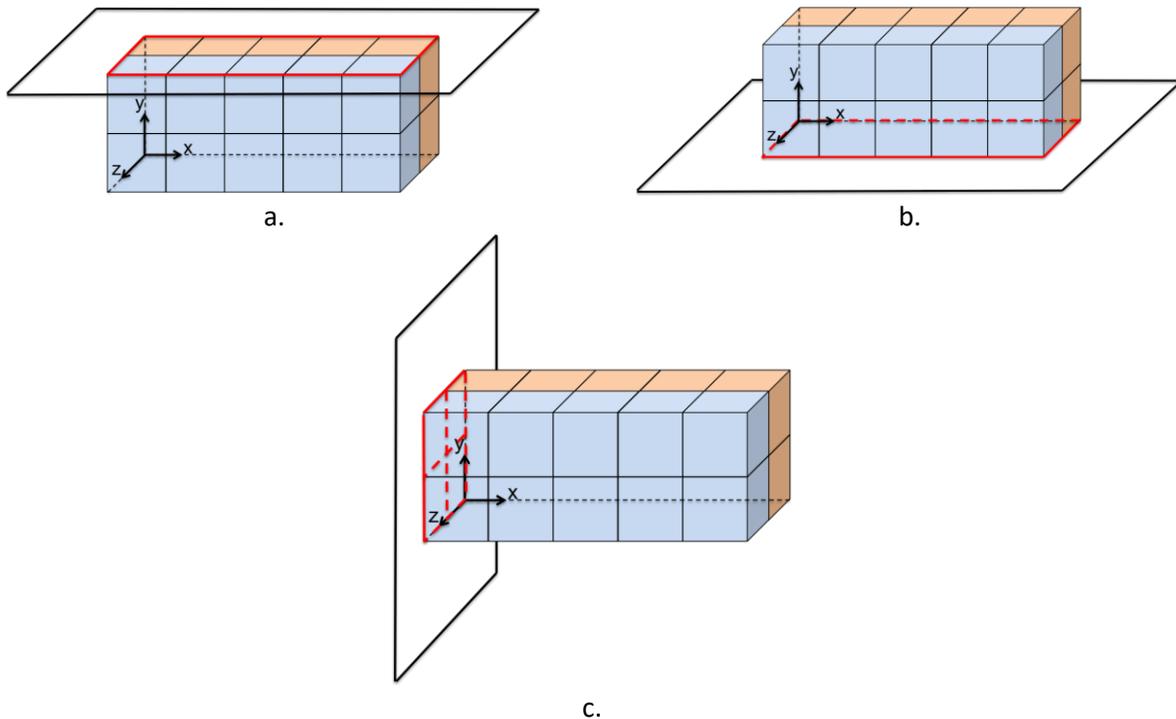


Figure 34: Boundary Conditions. a. BC1; b. BC2; c. BC3.

In all three cases, the EIDs of the elements along the boundary face or edge are determined, and the nodes corresponding to the proper locations are identified. For BC 1 the nodes along the left face of the selected elements are fixed, while for BCs 2, 3, and 4 the nodes along the top or bottom edge are “fixed”. Fixed, here, refers to the specification that the nodes cannot move in a given direction under applied loading. Therefore, the DOF associated with the given node and direction is removed from the FEA equations. This is done by generating a list of fixed DOF and removing those values from the full list of DOF. The result is a list of “free” DOF, or nodes that are permitted to move in the given directions (line 417). This list of free DOF will be used in FEA calculations.

5.3 Loading

Two types of loading may be applied to the structure: applied loading and self-weight loading. The type and location of loading for a given simulation is defined in lines 27-32, and multiple load cases can be considered simultaneously.

1. Applied Loading (Lines 311 – 368)

Applied loading is implemented by the function *applied_load*. The function begins by identifying the EIDs of elements that are subjected to the load. These EIDs are determined based on the selected load case, with available cases listed below:

- ALC 1. Point load at bottom right edge (lines 316 – 324) (Figure 35a)
- ALC 2. Point load at mid-height on right edge (lines 325 – 333) (Figure 35b)
- ALC 3. Point load at midpoint of top face (lines 334 – 346) (Figure 35c)
- ALC 4. Distributed load along the top face (lines 347 – 364) (Figure 35d)

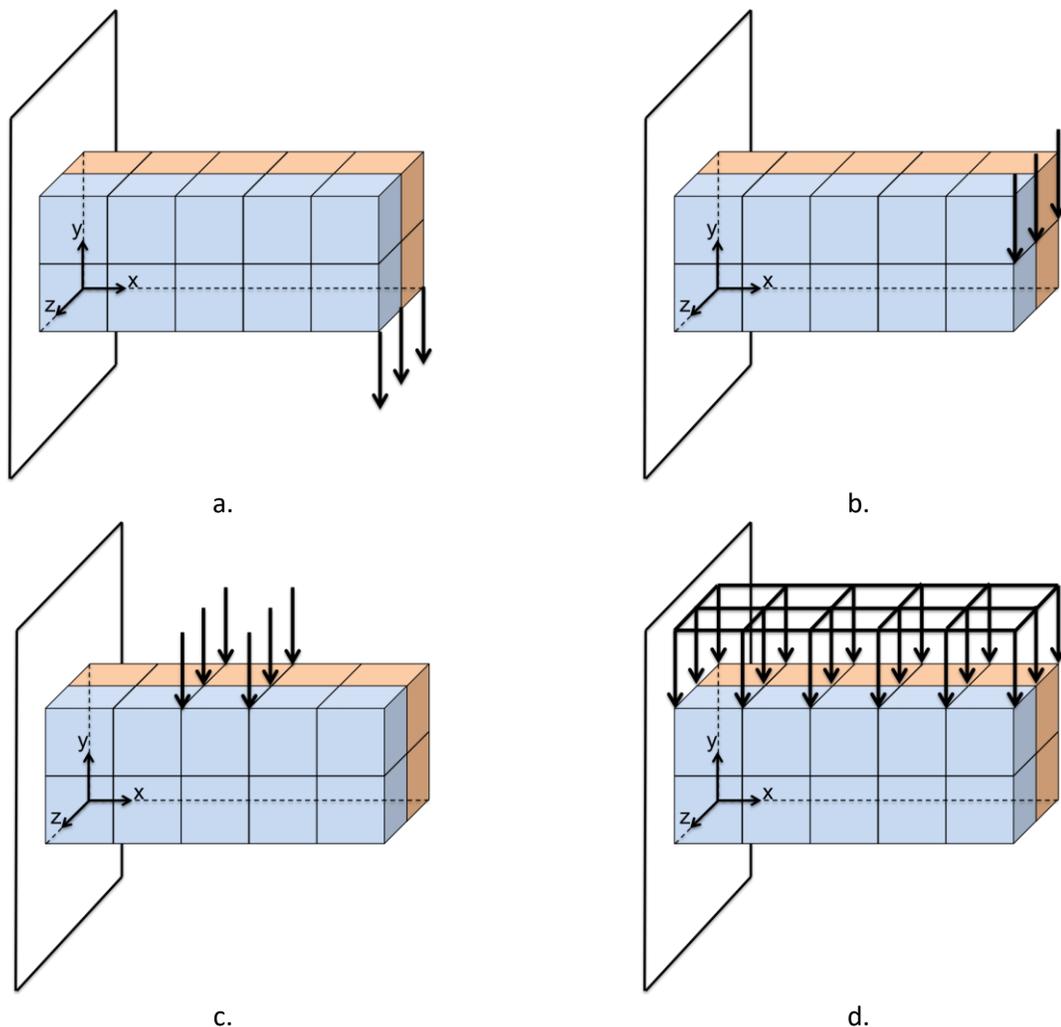


Figure 35: Applied loading conditions. a. ALC1; b. ALC2; c. ALC3; d. ALC4

In order to avoid stress concentrations that arise upon the application of a large load at a single node, point loads are divided and applied across two adjacent nodes, one on each side of the original loading location, as seen in Figure 35a/b. Additionally, if the geometric location of loading falls between nodes, such as ALC3 applied to Figure 35c, the load is similarly divided and applied to the two closest nodes. Figure 35c, then, applies the point load to 6 nodes in order to 1. Avoid stress concentrations and 2. Apply loading at the nodes, and not between them.

The elements subjected to loading are manually forced to maintain full artificial density, as material must exist at the location of the applied load.

In a method similar to that used for BCs, the appropriate nodes are selected where the loading will be applied. For ALC 1, 2, and 3, the magnitude of the point load is input in line 33. For the ALC 4 the magnitude of load applied to each node is input in line 34. In all cases, the magnitude of the applied load is input in Newtons and, in order to maintain unit consistency and well-scaled matrices, Newtons are converted to Mega-Newtons using an *MPa_Factor* of 1×10^6 . Additionally, all loading in these cases is applied in the y (vertical) direction, corresponding to DOF(2) for each node.

The applied loads are assigned to a forcing vector $F_{applied}$, the entries of which correspond to the appropriate node DOF.

2. Self-Weight Loading (Lines 480 – 502)

Self-weight loading is implemented by the function *Self_Weight*, and is done so in a manner similar to that used in the alteration to PTO presented in section 4.1. Due to the 3D nature of the geometry, each node is attributed $1/8^{\text{th}}$ of the self-weight of the element, and thus the loading formula changes to that shown in equation 36 (line 488). An *MPa_Factor* is applied resulting in units of Mega-Newtons, and the loads are stored in a self-weight loading vector F_{Sw} .

$$(F_{sw})_n = \sum_{i=1}^m \frac{1}{8} \rho g V_i x_i \quad (36)$$

where

n = node

m = number of elements adjacent to node n

ρ = physical density of the material

V = volume

x_i = element density

Finally, the full forcing vector F is created from the summation of $F_{applied}$ and F_{Sw} (line 501), allowing for the simultaneous application of external and self-weight loading. This summation is performed in the *Self_Weight* function due to the fact that *Self_Weight* is always implemented after *applied_load*. Additionally, self-weight loading is dependent upon the artificial element density. Since the artificial density of each element changes with each iteration, the self-weight loading must be recalculated with each iteration as well, while applied loads are set before the iterative process begins.

5.4 Material Properties

The material properties required for simulation are defined in lines 48 – 51 and 55, and are as follows:

1. Young's Modulus – Solid (E)
2. Young's Modulus – Void (E_{min})
3. Poisson's Ratio (ν)
4. Density (den)
5. Yield Shear Stress (γ_s)

Density is only required for self-weight loading cases. The use of Young's Modulus, both solid and void, as well as Poisson's Ratio will be discussed in section 5.6. Yield Shear Stress is a material property that is also defined as the constraint of the optimization process, and will be discussed further in section 5.8.

5.5 Iterations

The iterative process is governed by the while loop initiated on line 151 of SSTO, and ending on line 262. Within this loop, FEA is performed, the objective and constraint functions are calculated, sensitivity analysis and optimization are performed, and termination criteria are checked. With each iteration, important data resulting from these calculations are stored in data vectors, and specific key values are displayed as output (lines 205/208). This data is then stored in a data structure *BigData* when the loop is exited. Additionally, a plot of the material placement matrix is generated with each iteration by calling *display_3D* (lines 796 – 817), which is a plotting function taken directly from Top3D. If the termination criteria are met, as will be discussed in section 5.12, the program exits the while loop, and the data from the current iteration become the final optimization results. This process can be visualized in Figure 39 (section 5.14).

5.6 FEA

This section will develop the finite element equations that calculate displacements, strains, and stresses in a structure due to applied loads and subject to boundary conditions. First, the general FEA equations will be developed, and then their implementation in SSTO will be discussed.

FEA begins by dividing a structure into a set number of discrete elements. These elements are treated as springs connecting two nodes, with each spring having a defined stiffness, or resistance to deformation under loading. The stiffness is proportional to the displacement under a given load, and therefore if the stiffness and loading are known, displacements can be found according to Newton's 2nd Law (eqn 10). In the case of three-dimensional geometries, elements can connect multiple nodes, depending on the choice of element geometry and nodal placement.

Forces and displacements are measured and applied at the nodes, while the element links the nodes together, as can be seen in Figure 36.

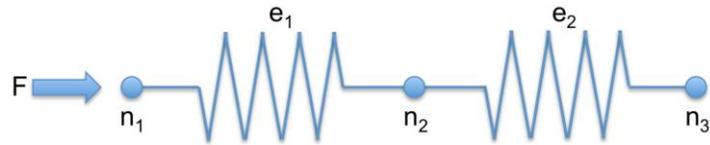


Figure 36: Finite element direct stiffness model. n = node, e = element, F = applied force.

The elements relate the displacements in one node to the displacements in adjacent nodes. However, as elements represent a physical quantity of material in a defined space, in order to define the relationship between nodal displacements, the displacement field within the element must be determined as well.

This displacement field is defined in terms of a shape function, also known as an interpolation function, which models how displacement varies within the element. Depending on the type and behavior of material, shape functions can be assumed as linear, quadratic, cubic, or even higher degree functions. Displacement at any node n, therefore, becomes proportional to the prescribed shape function, and displacement at any point within the element varies with position. As shape functions describe relationships between nodes of a single element, the functions can be expressed in terms of a coordinate system local to the element (zeta), and position within the element is given in this local coordinate system (Figure 37).

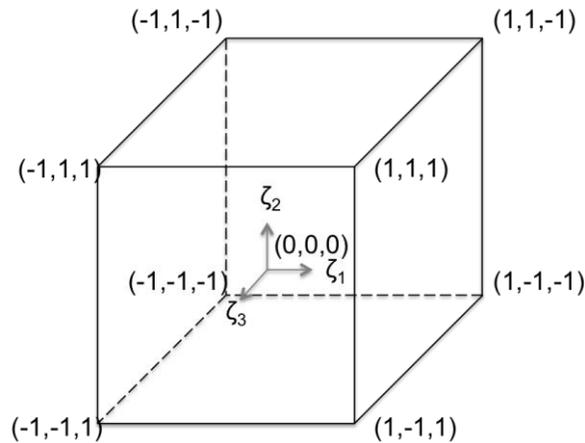


Figure 37: Local coordinate system.

Determining displacements using these shape functions requires the use of the Principle of Minimum Potential Energy. This principle states that any material, when subject to loading, will arrange itself in order to reach a state of minimum energy, resulting in some form of deformation or rigid body motion. Total potential energy in a structure can be expressed as:

$$\Pi = U - W \tag{37}$$

Where U represents elastic strain energy and W is the work done by an external force.

Returning to the elemental level, elastic strain energy in an element can be expressed in terms of stress and strain:

$$U_e = \frac{1}{2} \int \{\varepsilon\}^T \{\sigma\} dV \quad (38)$$

Stress and strain are both functions of displacement.

$$\varepsilon = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{bmatrix} \quad (39)$$

$$\sigma = E\varepsilon. \quad (40)$$

where E = Young's Modulus

With linear shape functions, displacement \mathbf{u} at each node is proportional to the shape functions:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{N} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_8 \\ v_8 \\ w_8 \end{bmatrix} = \mathbf{N} \quad (41)$$

Strain calculations, then, can be expressed in terms of the derivatives of shape functions. Compiling this relationship between strain and displacement at each node into matrix form, the strain-displacement matrix \mathbf{B} for each node is generated, and is formulated as follows:

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix}; \quad i = 1 - 8 \quad (42)$$

Strain in the element, therefore, can be represented as:

$$\boldsymbol{\varepsilon} = \mathbf{B}_{6 \times 24} \mathbf{q}_{24 \times 6} \quad (43)$$

where: double subscripts ($n \times m$) subscripts indicate matrix dimensions
single subscripts (i) indicate local node number

$$\begin{aligned} \mathbf{B}_{6 \times 24} &= [\mathbf{B}_1 \mathbf{B}_2 \dots \mathbf{B}_8] \\ \mathbf{q}_{24 \times 6} &= \text{Matrix of nodal degree of freedom displacements} \end{aligned}$$

and

$$\boldsymbol{\sigma} = E \mathbf{B}_{6 \times 24} \mathbf{q}_{24 \times 6} \quad (44)$$

Therefore, internal strain energy can be expressed as:

$$U_e = \frac{1}{2} \mathbf{u}^T \left(\int \mathbf{B}^T E \mathbf{B} dV \right) \mathbf{u} \quad (45)$$

In order to minimize the potential energy, the derivative is taken with respect to displacement:

$$\frac{\partial \Pi}{\partial \mathbf{q}} \rightarrow \frac{\partial U_e}{\partial \mathbf{u}} = \left(\int \mathbf{B}^T E \mathbf{B} dV \right) \mathbf{u} \quad (46)$$

The resulting integral, proportional to the displacement, is considered the stiffness of the element, and is defined as the elemental stiffness matrix \mathbf{K} . A complication arises from the fact that shape functions are given in terms of local coordinates and displacements are given in global coordinates. Therefore, partial derivatives of shape functions with respect to global displacements cannot be calculated directly. Instead, partial derivatives are calculated in the local coordinate system (ζ) and then converted from local to global coordinates using the Jacobian matrix, as shown in equation 47.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \zeta_1} & \frac{\partial y}{\partial \zeta_1} & \frac{\partial z}{\partial \zeta_1} \\ \frac{\partial x}{\partial \zeta_2} & \frac{\partial y}{\partial \zeta_2} & \frac{\partial z}{\partial \zeta_2} \\ \frac{\partial x}{\partial \zeta_3} & \frac{\partial y}{\partial \zeta_3} & \frac{\partial z}{\partial \zeta_3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_1} x_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_1} y_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_1} z_i \right) \\ \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_2} x_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_2} y_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_2} z_i \right) \\ \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_3} x_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_3} y_i \right) & \sum_{i=1}^8 \left(\frac{\partial N_i}{\partial \zeta_3} z_i \right) \end{bmatrix} \quad (47)$$

The elemental stiffness matrix can then be calculated locally (eqn 48), and converted to global coordinates using the inverse Jacobian.

$$\mathbf{K}_e = \int \mathbf{B}^T E \mathbf{B} dV = \iiint_{-1}^1 \mathbf{B}^T E \mathbf{B} |J| \delta \zeta_1 \delta \zeta_2 \delta \zeta_3 \quad (48)$$

The elasticity matrix \mathbf{E} can be decomposed into $\mathbf{E} = \mathbf{E}\mathbf{C}$ where \mathbf{C} is the constitutive matrix for the class of material used in the simulation. Similar to Top3D, for the purposes of this study, the materials are considered to be isotropic. Concrete mortar, though composed of a material mixture, can be considered homogeneous and isotropic at the continuum level, as discussed in section 2.1.10. Therefore, the constitutive matrix represented as:

$$\mathbf{C} = \frac{1}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} \end{bmatrix} \quad (49)$$

where ν = Poisson's Ratio

For 3D elements, equation 48 is difficult to calculate analytically. Therefore, numerical methods are often used. The most common method is Gauss-Quadrature (GQ), which calculates the integrand at a specified number of points within each element, prescribes a weighting coefficient to each value, and sums the result. The greater the number of GQ points, the more accurate the solution. However, there is a significant computational cost to increasing the number of GQ points. For this work, in order to evaluate the stiffness integral, three GQ methods are used and compared: 1. A 2x2x2 system of standard internal GQ points (Figure 38), the local coordinates of which are prescribed in line 67 (Table 6); 2. A 2x2x2 system of GQ points comprised coordinates corresponding to each node, and 3. A single GQ point at the center of the element. Results showed little to no difference between methods, so the center-point method is used for calculations.

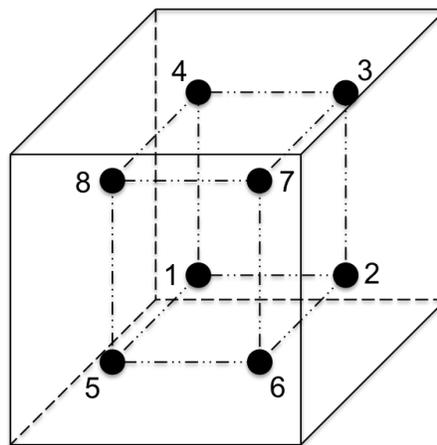


Figure 38: 2x2x2 Gauss Quadrature points.

GQ Point	ζ_1	ζ_2	ζ_3
1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
2	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
3	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
4	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
5	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
6	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
7	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
8	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$

Once the stiffness matrix has been calculated, the displacement at each node can be calculated using Newton's Second Law (eqn 10). From the resulting displacements, strains can be calculated using equation 43, and stresses from equation 44. It is important to note that these stresses and strains are not calculated at the nodes, but instead are average values over the element.

In SSTO, FEA is initiated by developing the stiffness matrix \mathbf{K} using the *stiffness* function (lines 505 – 594). Similar to Top3D, an 8-node hexagonal element is used here for FEA calculations. Local coordinates are defined as in Table 7, and for simplicity, the shape functions are assumed to be linear (eqn 48). Therefore, the partial derivatives of the shape functions in terms of local coordinates can be calculated using equation 49.

Node	ζ_1	ζ_2	ζ_3
1	-1	-1	-1
2	+1	-1	-1
3	+1	+1	-1
4	-1	+1	-1
5	-1	-1	+1
6	+1	-1	+1
7	+1	+1	+1
8	-1	+1	+1

$$N_i = \frac{1}{8}(1 + \zeta_{1i})(1 + \zeta_{2i})(1 + \zeta_{3i}) \quad i = 1 - 8 \quad (48)$$

$$\left. \begin{aligned} \frac{\partial N_i}{\partial \zeta_1} &= \frac{1}{8} \zeta_{1i} (1 + \zeta_{2i})(1 + \zeta_{3i}) \\ \frac{\partial N_i}{\partial \zeta_2} &= \frac{1}{8} \zeta_{2i} (1 + \zeta_{1i})(1 + \zeta_{3i}) \\ \frac{\partial N_i}{\partial \zeta_3} &= \frac{1}{8} \zeta_{3i} (1 + \zeta_{1i})(1 + \zeta_{2i}) \end{aligned} \right\} \quad i = 1 - 8 \quad (49)$$

The Jacobian can be calculated from these partial derivatives and from the global coordinates of the GQ points (line 554), and the partial derivatives of the shape functions with respect to global coordinates can then be calculated in line 564. Lines 567 – 584 develop the strain-displacement matrix, line 581 calculates the local stiffness matrix at each GQ point, and lines 587 – 590 assembles the full elemental stiffness matrix.

With the stiffness matrix fully assembled, FEA calculations can proceed. These calculations are performed by the *FEA* function (lines 625 – 731). The SIMP method is first employed at this point (line 630) with the stiffness of each element penalized according to its artificial density. The global stiffness matrix is then assembled by compiling all elemental stiffness matrices into one global stiffness matrix (lines 628 – 631). Line 641 calculates displacements from the global stiffness matrix and the global forcing vector. Nodal displacement results are then stored in the node structure, and placed in vector form to be stored in the element structure. Full field displacements in all three directions are placed in matrix form as well for visual inspection (lines 697 – 708).

Strains are calculated from displacements using the strain-displacement matrix at the center-point (line 715). Since stress calculations involve the elasticity matrix, the stresses are penalized according to the SIMP method, resulting in equation 50, calculated in line 716. Finally, stresses and strains are both stored in the element structure.

$$\boldsymbol{\sigma}(x_i) = (E_{min} + x_i^P (E_0 - E_{min})) \mathbf{B} \mathbf{q}(x_i) \boldsymbol{\eta}_s(x_i) \quad (50)$$

where $\boldsymbol{\eta}_s(x_i)$ is a stress penalization factor: $\boldsymbol{\eta}_s(x_i) = x_i^{SP}$

5.7 Objective Function

Shear Stress

As stated in Chapter 3, the goal of the concrete problem is to minimize the volume. Therefore, the objective function can be written as follows:

$$f(x_i) = \frac{\sum_1^i V_i x_i}{\sum_1^i V_i} \quad (51)$$

where

x_i = Artificial density of element i
 V_i = Volume of element i

The objective function is normalized by the total volume of the design space in order to constrain the values to within an acceptable limit, as per MMA guidelines (Svanberg, 2007). Calculation of the objective function is performed on line 735, and normalization takes place on line 169.

Von Mises Stress

Both the concrete problem and VMS-based optimization share the goal of volume minimization. Therefore, the VMS objective function is formulated in the same manner as the shear stress objective function.

5.8 Constraint Function

Shear Stress (Line 174)

The constraint function, as defined in Chapter 3, is represented by an inequality relating the maximum calculated shear stress value to the shear strength of the material.

$$|\sigma_{xy}|^{max} \leq \sigma_{yield} \quad (52)$$

Since directionality of shear stress does not play a role in failure due to self-weight loading, the absolute values of shear stress are considered when determining the maximum. The inequality formulation for MMA optimization requires the right side of the equation to be zero. Therefore, the inequality is rearranged to the form

$$\frac{|\sigma_{xy}|^{max}}{\sigma_{yield}} - 1 \leq 0 \quad (53)$$

The shear stress values for each iteration are calculated during FEA from equation (54).

$$\boldsymbol{\sigma}(x_i) = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} = \mathbf{E}(x_i)\mathbf{B}\mathbf{q}(x_i) \quad (54)$$

where they can be isolated from the 4th term in the resulting elemental stress vectors (line 717). The maximum of these shear stress values is determined and used to evaluate the constraint function as follows:

$$g(x_i)^{ys} = \frac{\left(\frac{|\sigma_{xy}|^{max}}{\sigma_{yield}} - 1 \right)}{\varepsilon} \quad (55)$$

where

- σ_{xy}^{max} = Maximum shear stress in the x-y plane
- σ_{yield} = Yield shear stress / shear strength
- ε = Relaxation parameter

The constraint function is normalized by a relaxation parameter ε as per Duysinx and Bendsoe (1998) in order to avoid the stress singularities that result when the artificial element density approaches zero.

Von Mises Stress

Similar to the shear stress case, the VMS constraint is represented by inequality 56, though replacing maximum and yield shear stresses with VMS. The resulting constraint equation is calculated on line 185 (eqn 57).

$$\sigma'_{VMS} \leq \sigma_{VMS}^{max} \quad (56)$$

$$g(x_i)^{VMS} = \frac{\left(\frac{\sigma'_{VMS}}{\sigma_{VMS}^{max}} - 1\right)}{\varepsilon} \quad (57)$$

where

- σ'_{vms} = maximum VMS in the domain
- σ_{vms}^{max} = VMS constraint
- ε = Relaxation parameter

VMS for each element is calculated during FEA using equation 58 on lines 720 – 727.

$$\sigma_{VMS} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - (\sigma_{xx}^2 * \sigma_{yy}^2) - (\sigma_{yy}^2 * \sigma_{zz}^2) - (\sigma_{xx}^2 * \sigma_{zz}^2) + 3(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{xz}^2)} \quad (58)$$

5.9 Sensitivity Analysis

Shear Stress

All sensitivity analysis calculations are performed using the sensitivity function (lines 734 – 793). Analytical sensitivities of both the objective function and the constraint function are found by differentiating the respective functions with respect to the design variable x_i . In the case of shear stress-constrained optimization, these sensitivities are formulated by equations 59 and 60.

$$\frac{\partial \sigma(x_i)}{\partial x_i} = \frac{\partial E(x_i)}{\partial x_i} \mathbf{B} \mathbf{q}(x_i) + E(x_i) \mathbf{B} \frac{\partial \mathbf{q}(x_i)}{\partial x_i} + E(x_i) \mathbf{B} \mathbf{q}(x_i) \frac{\partial \eta_s(x_i)}{\partial x_i} \quad (59)$$

where

$$\begin{aligned} \frac{\partial E(x_i)}{\partial x_i} &= p x_i^{p-1} (E - E_{min}) \\ \frac{\partial \mathbf{q}(x_i)}{\partial x_i} &= \mathbf{K}^{-1} \left(\frac{\partial \mathbf{F}(x_i)}{\partial x_i} - \frac{\partial \mathbf{K}(x_i)}{\partial x_i} \mathbf{q}(x_i) \right) \\ \frac{\partial \mathbf{K}(x_i)}{\partial x_i} &= \frac{\partial E(x_i)}{\partial x_i} \mathbf{K} \\ \frac{\partial V(x_i)}{\partial x_i} &= V_i \end{aligned} \quad (60)$$

However, applying stress sensitivity equation 59 resulted in optimization failure, indicated either by non-convergence, an empty volume solution, or a solution that was unpredictable, highly asymmetric, and non-intuitive. This failure may be due to the tensile and compressive nature of stresses, represented by positive or negative stress values for all stress components. These positive or negative values could misrepresent the appropriate search direction in the optimization step. The failure may also be caused by the local nature of the sensitivity calculations. Stress sensitivity equations require the calculation of the inverse elemental stiffness matrix \mathbf{K}_e^{-1} when determining the partial derivative of displacements. The local stiffness matrix \mathbf{K}_e is not invertible, since it is not subject to boundary conditions. In an attempt to circumvent this issue, the partial derivative of displacements was calculated on the global level using the invertible global stiffness matrix \mathbf{K}^{-1} . However, these global partial values do not necessarily reflect the rate of change of displacements on the local level.

Since the analytical method proved unsuccessful, a semi-analytical method was adopted (eqn 61) using chain rule differentiation. Analytical methods were used to determine $\frac{\partial E(x_i)}{\partial x_i}$ and $\frac{\partial \eta_s(x_i)}{\partial x_i}$, while the forward finite difference method was applied to calculate $\frac{\partial \mathbf{q}(x_i)}{\partial x_i}$ at each node, with a given perturbation value.

$$\frac{\partial \sigma(x_i)}{\partial x_i} = \frac{\partial E(x_i)}{\partial x_i} \mathbf{B} \mathbf{q}(x_i) \eta_s(x_i) + E(x_i) \mathbf{B} \frac{\partial \mathbf{q}(x_i)}{\partial x_i} \eta_s(x_i) + E(x_i) \mathbf{B} \mathbf{q}(x_i) \frac{\partial \eta_s(x_i)}{\partial x_i} \quad (61)$$

where

$$\frac{\partial \mathbf{q}(x_i)}{\partial x_i} = \frac{d\mathbf{q}(x_i)}{dx_i} = \frac{\mathbf{q}(x_i + \rho) - \mathbf{q}(x_i)}{\rho} \quad (62)$$

This semi-analytical method (line 758) requires FEA displacement calculations (\mathbf{q}) at both x_i and $x_i + \rho$ values (lines 662 – 678), where ρ is the perturbation factor, set in line 40. Due to the very small values resulting from this formulation (on the order of $10^{-3} - 10^{-5}$), sensitivity results were normalized such that the greatest value was equal to 1.

Von Mises Stress

Given that VMS- and shear stress-constrained optimization in this work share the same objective function, VMS sensitivity is also calculated using equations 51 and 56. However, the VMS constraint function requires a much different formulation for sensitivity analysis. For this work, VMS constraint sensitivity is calculated based on the work done by Holmberg (2013). Holmberg derives the following VMS sensitivity equation, implemented in SSTO in lines 765 - 782.

$$\frac{\partial \boldsymbol{\sigma}(x_i)}{\partial x_i} = \frac{\partial \boldsymbol{\sigma}(x_i)}{\partial \sigma_{VMS}} \left(\frac{\partial \sigma_{VMS}(x_i)}{\partial \boldsymbol{\sigma}} \right)^T \frac{\partial (\eta_s x_i)}{\partial x_i} \mathbf{E}(x_i) \mathbf{B} \mathbf{q}(x_i) - \eta_s x_i \mathbf{E}(x_i) \mathbf{B} \frac{\partial \mathbf{q}(x_i)}{\partial x_i} \quad (63)$$

where

$$\begin{aligned} \frac{\partial \mathbf{q}(x_i)}{\partial x_i} &= -\mathbf{K}^{-1} \left(\frac{\partial \mathbf{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i} \mathbf{q}(x_i) \right) \\ \frac{\partial \mathbf{K}}{\partial x_i} &= p x_i^{p-1} (E_0 - E_{min}) \mathbf{K} \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{xx}} &= \frac{1}{2 \sigma_{VMS}(x_i)} (2 \sigma_{xx}(x_i) - \sigma_{yy}(x_i) - \sigma_{zz}(x_i)) \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{yy}} &= \frac{1}{2 \sigma_{VMS}(x_i)} (2 \sigma_{yy}(x_i) - \sigma_{xx}(x_i) - \sigma_{zz}(x_i)) \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{zz}} &= \frac{1}{2 \sigma_{VMS}(x_i)} (2 \sigma_{zz}(x_i) - \sigma_{xx}(x_i) - \sigma_{yy}(x_i)) \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{xy}} &= \frac{3}{\sigma_{VMS}(x_i)} \sigma_{xy}(x_i) \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{yz}} &= \frac{3}{\sigma_{VMS}(x_i)} \sigma_{yz}(x_i) \\ \frac{\partial \sigma_{VMS}(x_i)}{\partial \sigma_{xz}} &= \frac{3}{\sigma_{VMS}(x_i)} \sigma_{xz}(x_i) \end{aligned}$$

5.10 Optimization

For this work, MMA was chosen as the method of optimization for the following reasons: 1. Common use in stress-based and non-stress based optimization (Bruggi and Duysinx, 2012; Gaynor, 2014; Li and Kim, 2018, and more); 2. Ability to handle a large number of constraints; 3. Straightforward implementation. For an in-depth discussion of the development of MMA equations, refer to Svanberg, 1998; Svanberg 2007.

For each iteration, in order to begin the MMA optimization process, certain variables and constants must be defined (lines 117 – 133) as per the Top3D instructional website and Svanberg (2007). The MMA program is a standalone open-source program provided by Svanberg, accessed by SSTO in lines 229-231. The MMA program requires the input of the stated initiate coefficients and variables, as well as the calculated objective and constraint functions and their sensitivities. The result of the optimization is a new material geometry, represented by a vector of x_i values.

5.11 Filtering

The filtering technique employed in SSTO follows that used in Top3D (eqn 19), based on the method developed by Bruns and Tortorelli (2001), and implemented in lines 597 – 622.

The weighting constants are used to scale objective and constraint functions and sensitivities (lines 790 – 791), as well as artificial densities (line 241).

5.12 Termination Criteria Checks

SSTO contains four unique termination criteria. Each criterion is checked at the end of every iteration, and if any one of the checks is satisfied, the optimization process will end. The termination criteria are listed below, as is a discussion of the significance of each.

1. Convergence

In order to determine if the solution has converged, SSTO checks difference in artificial densities between the previous and current iterations. If the maximum difference is less than a designated amount, the resulting structure is no longer changing significantly, and the solution is considered to have converged. The maximum difference is designated by the *change* variable, calculated on line 254, and compared to the tolerance variable *tolx* set by the user as an optimization parameter (line 41). The criterion check is made at the beginning of the main while loop, which is defined to run if only $change < tolx$ and if the continuation condition is also satisfied.

2. Continuation

To prevent the program from running indefinitely, a maximum number of iterations is set by the user (line 43). The main *while* loop runs only if the convergence criterion is satisfied and if the current iteration number is less than the maximum number of iterations. If the program reaches the maximum number of iterations, it cannot be determined whether the structure has been fully optimized. Therefore, the optimization is considered to be incomplete.

3. Minimum Volume (lines 192 – 196)

The optimization process is designed to end if a minimum volume fraction threshold is reached. The total volume of the structure is calculated at the end of each iteration, and compared to the designated minimum volume fraction. If the volume fraction is below the designated minimum, the while loop is manually broken, and the optimization process ends.

The minimum volume criterion exists in order to prevent the optimization process from resulting in a structure with no volume, or with a volume so small it cannot be realistically manufactured. This result may occur if the constraint is so high that the presence of even a small amount of material can support the design load while satisfying the constraint.

4. Maximum Volume (lines 197 – 201)

Similar to the minimum volume criterion, the maximum volume criterion is designed to end the optimization process if a maximum volume threshold is met.

Physically, this criterion signifies that the imposed constraint cannot be met by the available geometry. Volume will continually be added to the structure within the design space, but the structure will never be able to support the designed load and satisfy the constraint.

In addition to these termination criteria, a continuation criterion is also imposed (lines 259 – 261). The continuation criterion is designed to force the optimization to proceed in the event that the constraint has not been satisfied. This criterion is necessary due to the fact that the convergence criterion has no relationship to the constraint function. Therefore, it is possible for the solution to have converged to a certain tolerance, without satisfying the constraint. In this case, the structure is not optimized within the bounds of the problem, so the process must continue.

5.13 Other

1. Plotting

A 3D plot of element artificial densities is generated for each iteration using the *display_3D* function implemented by Liu (2014). This plot generates a gray-scale representation of each element based on the value of artificial density. Elements with x_i values less than 0.3 are not displayed, and considered to be void areas in the final design. This void threshold can be adjusted in line 807.

2. Parameterization

SSTO has the ability to perform parametric studies considering the following optimization parameters, designated in lines 34, 37-40, and 55: 1. Distributed load; 2. Stiffness penalization factor; 3. Stress penalization factor; 4. Relaxation parameter factor; 5. Perturbation factor; 6. Constraint value. These parameters can be set as vectors of several values, over which SSTO will run new optimization processes. Parametric studies are incorporated programmatically by nesting the main program within four sequential for loops, one for each parameter. Data and results from each run is stored as indicated in the following section.

3. Data Storage

For each run, several important data points are stored for analysis. This data is stored upon termination of the main while loop in a structure designated *BigData* with each row of the structure representing a unique run (lines 272 – 300). Additionally, several of the data points stored include values calculated at each iteration throughout the optimization process. This iterative data is stored in column vectors, with each row associated numerically with the corresponding iteration (lines 176 – 180; 187-188.).

4. Density Forcing

Due to the stress singularity problem, it is necessary to define and enforce a minimum artificial density value (Duysinx and Bendsoe, 1998). SSTO defines the minimum artificial density to be 0.1, and enforces this minimum by checking the density matrix for values less than 0.1 and manually forcing those values to zero (lines 243 – 246).

Additionally, material must be present at the location of applied loads. Therefore, SSTO forces the artificial density to be equal to 1 for these elements (lines 247-250).

5.14 Broad Overview of Program Order

The requirements presented above form the basic structure of the TO program SSTO. SSTO performs each of these requirements in a specified order as follows:

SSTO begins with sections defining the domain geometry, boundary and loading conditions, optimization parameters and constraints, material properties, and various constant values necessary for use throughout the program (lines 10 – 81).

Following these definitions, the program defines nodal and element data by calling *nodal_data* and *element_data* functions (lines 84, 87). The boundary conditions are then set using the *BCs* function (line 90), and the elemental stiffness matrix is developed using the *stiffness* function (line 93). The *filtering* function is then called in order to define the element weights (line 96).

The program then initiates the parameterization process, which is only run when specified parameters are entered as vector quantities instead of singular values (lines 99 – 112). If necessary, this parameterization includes the calculation of applied loads by calling the *applied_load* function (line 102). The first iteration is then initialized by setting various variable, constant, and data storage values necessary for optimization (lines 117 – 142).

The program then enters the main *while* loop, which checks two of the four termination criteria during each successive pass through the loop (criterion 1 and 2). Within the loop, self-weight loading may be applied by calling the *Self_Weight* function (lines 153 – 159). Then, FEA is performed using the *FEA* function (line 162), followed by sensitivity analysis, using the *sensitivity* function (line 165). At this point, all of the necessary information for optimization has been developed.

The MMA optimization process is initiated first by adjusting values for elements with artificial densities less than 0.1 or equal to 1. Next, the objective and constraint function and sensitivity values are defined for passage to the MMA program. The MMA program *mmasub* is then called, and the new material placement values are updated and stored. Checks are made again for elements with artificial densities less than 0.1 or equal to 1, and specified values are adjusted accordingly.

At this point, the remaining two termination criteria are checked (criterion 3 and 4), and a continuation condition is imposed. The program then prints key values to the command line and plots the artificial densities for the current iteration.

Once the program fails any of the termination criteria, the main while loop ends. Optimization data is then stored, and the command line displays the optimization result. If parametric studies are performed, a new run may be initiated. If not, the command line will display text indicating that the program has finished, and the program will end. The process explained here can be visualized in Figure 39.

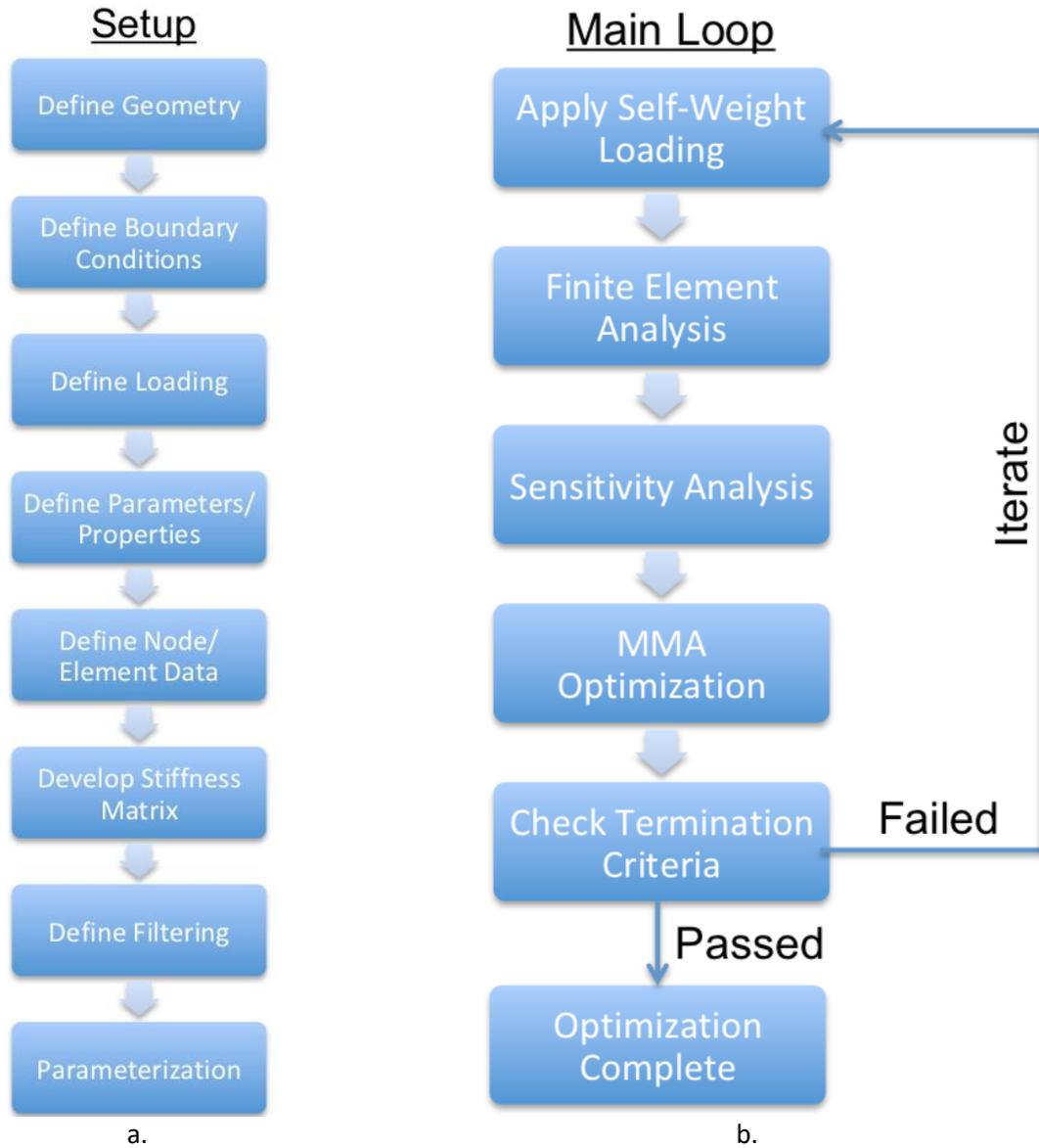


Figure 39: Optimization processes: a. Setup; b. Main iterative loop.

Chapter 6: Validation Procedures

Validation studies were performed throughout the development of SSTO to ensure that various components of the program were working properly and producing expected results. The first validation study was developed to determine the validity of the elemental stiffness matrix development. This test was performed by comparing the stiffness matrix produced by Top3D to that produced by SSTO. Values of Poisson's Ratio were set at 0.3 and 0.1 for two separate test cases. The resulting stiffness matrices are shown in Appendix A. With a % difference of zero for each element in each case, it was determined that the stiffness matrix had been properly formulated.

A second validation study was developed to ensure proper incorporation of the MMA optimization algorithm. In order to test the implementation of MMA optimization in SSTO, a compliance minimization problem was solved and compared with the results of Top3D. This problem was formulated in accordance with the example problem of a cantilever beam subject to a point load presented in section 2.2. The problem setup is shown in Figure 17. Top3D was altered to incorporate MMA optimization according to the included literature, so the results could be directly compared. Figure 40 shows that the resulting structures are almost identical, indicating proper implementation of the MMA optimization algorithm.

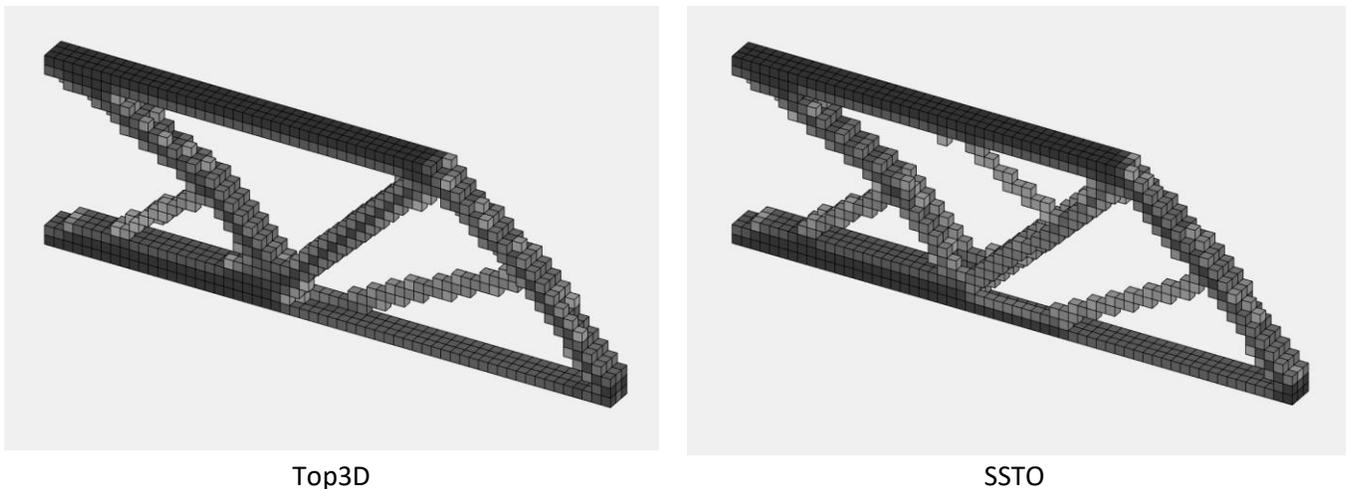


Figure 40: MMA validation. Compliance minimization subject to 0.3 volume fraction constraint.

A third validation study was developed to evaluate both the overall functioning of the completed SSTO in consideration of VMS-constrained optimization. This study was performed by posing a unit-less, VMS-constrained volume minimization problem, and comparing optimized structures resulting from SSTO, PTO, and the ANSYS topology optimization module. The posed problem considered a cantilever beam subject to a point load mid-height on the right edge. The VMS constraint was set at 2 MPa with an applied load of 100 Newtons. Material properties for concrete were selected from the ANSYS material database, and are given in Table 8. Optimized structures are displayed in Figure 41, and numerical results are given in Table 9.

Table 8: Hardened concrete material properties (ANSYS)	
Young's Modulus (E)	30 GPa
Poisson's Ratio (ν)	0.18
Physical Density (ρ)	$2300 \frac{kg}{m^3}$

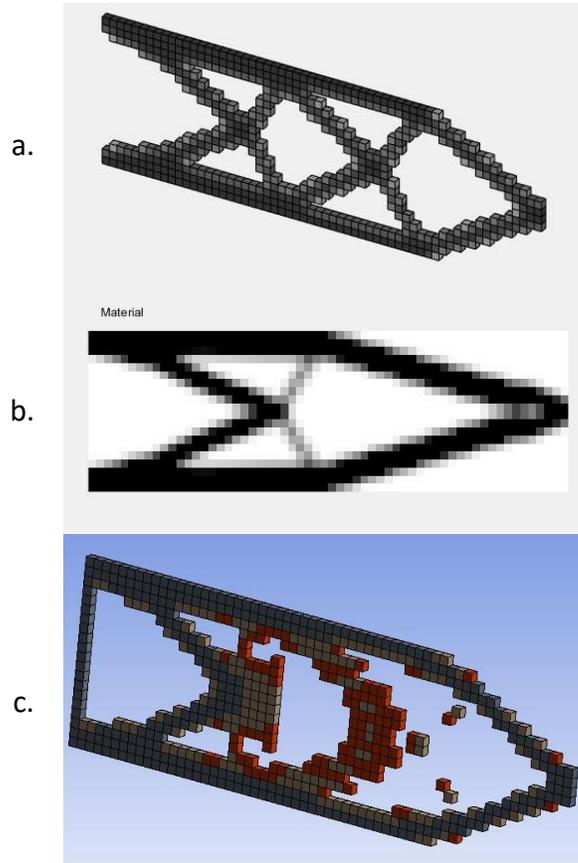


Figure 41: VMS-constrained optimized structures. a. SSTO; b. PTO; c. ANSYS. Red elements in ANSYS represent elements with 0.38-0.5 artificial density.

Table 9: VMS Test Results			
	Iteration	VMS	Volume
SSTO	132	2.00	496.08
PTO	109	1.90	480.00
ANSYS	86	1.55	558.00

SSTO and PTO showed similar values for final VMS and volume. The structures, however, took on different forms, with SSTO displaying two sets of cross-bracing elements in the beam interior while PTO included only one set, with longer diagonal elements approaching the location of the point load. The perimeters of the SSTO and ANSYS structures are almost identical, save for ANSYS including additional material at the fixed end. On the interior, the ANSYS structure forms half of a cross-bracing, though the red elements show the potential formation of two full sets cross-bracing elements. From this test, it was determined that the VMS formulation in SSTO, and the overall code, were both functioning properly. With this knowledge, the shear stress-constrained formulation could be implemented under the assumption that all other sections of the code were functioning as expected.

Chapter 7: Results

Parametric studies were performed to determine optimization parameters, considering both the VMS constraint and the SS constraint, with sensitivities calculated using the semi-analytical method. Both applied load and constraint values were varied in order to determine consistency of parameter performance. Stiffness penalization factors, stress penalization factors, perturbation factors, and relaxation parameters were the parameters evaluated through these studies. The optimal parameters were selected by considering success/failure rates, consistency of results, optimization trends, and an intuitive understanding of stress distributions in the resulting structures. The optimal parameters for both VMS- and SS-constrained optimization were determined to be:

Stiffness penalization = 2

Stress penalization = 8

Perturbation factor = 10^{-7}

Relaxation parameter = 0.1 (Hardened State); 0.01 (Fresh State)

7.1 Hardened State

After determining optimal parameters, initial studies were performed considering the shear strength of hardened state concrete material, evaluating both wall and column geometries under an applied distributed load along the top surface (Figure 42). Hardened state material properties are defined in Table 8. Load values were defined in the code in terms of Newtons per node, and ranged in magnitude from 100 to 1000 N/node for both the wall and column geometries. In SI units, these values correspond to 1.36 – 13.56 MPa for the wall and 1.21 – 12.10 MPa for the column. The hardened state case was examined for two reasons: First, it called for application distributed load, which was desirable due to the uncertain behavior of optimization under self-weight loading only. It was hypothesized that the structure may reduce to a minimum volume, as removing material would reduce the self-weight, and thus, the shear stress in the structure. If this process were to iterate, the structure could potentially reduce to zero volume, as removal of all material also leads to removal of all shear stress on the structure. By imposing a distributed load of significant size, however, the load path from top surface to bottom surface would be clearly defined from FEA, forcing material placement in areas of greater stress. Second, the hardened state case allowed for direct comparison of optimized structures resulting from a shear stress constraint vs. a VMS constraint. These structures would also correspond to real-world loading cases, though irrespective of fabrication method.

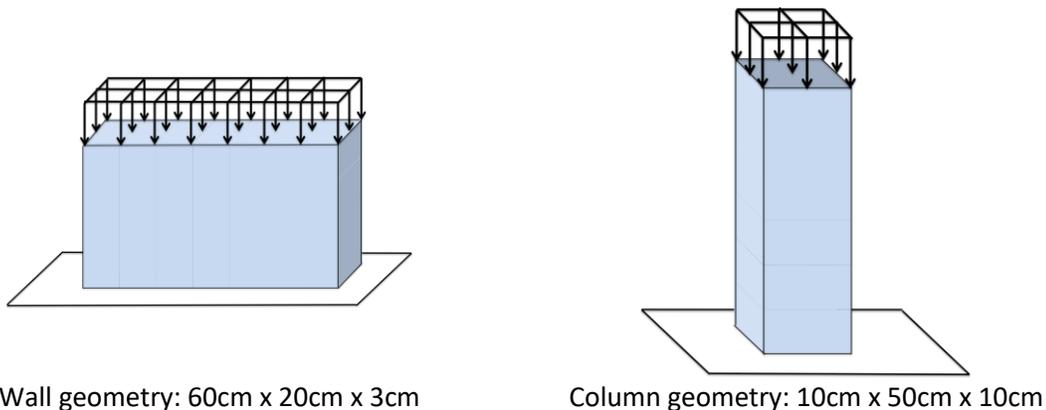


Figure 42: Wall and column geometry, loading, and boundary conditions for hardened and fresh state cases.

The range of constraint values for shear stress-constrained optimization was determined from experimental shear strength measurements of Portland cement (Moosavi and Bawden, 2003). These values were set at 5, 10, 15, and 20 MPa. The SS-constrained column was a special case, as the optimization proceeded much more slowly, and results were poor and inconsistent. Only a single constraint of 10 MPa was imposed in this case, with loads ranging from 8.47 to 18.15 MPa. Greater loads and a smaller constraint were applied to the column because the optimization process would result in either a separation of the column into two pieces, one at the top and one at the bottom of the structure, or the reduction to zero volume. For the VMS constraint, values were set to be demonstrative, and determined by trial and error under the same loading conditions. The resulting constraints were 10, 20, 30, and 50 MPa for the wall, and 20, 30, 40, and 50 MPa for the column. The resulting optimized structures are shown in Tables 10 – 13, with corresponding numerical results presented in Tables 20 – 23, Appendix B. Note that negative loading indicates downward directionality of the applied load.

Table 10: VMS-Constrained Hardened State Wall

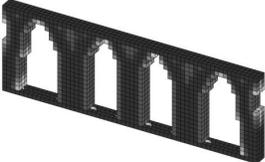
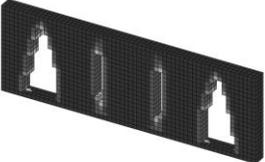
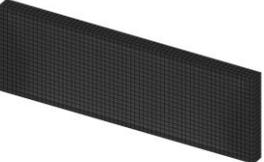
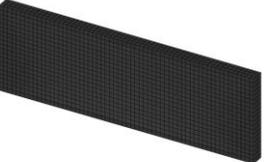
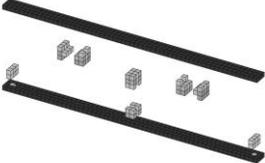
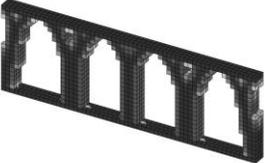
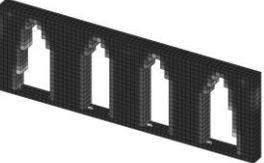
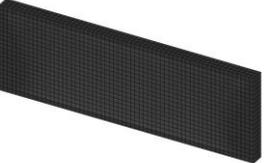
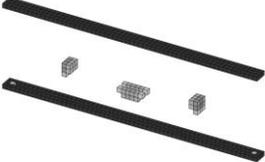
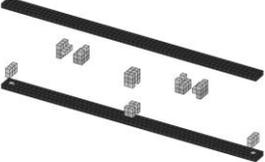
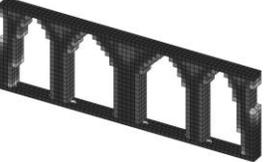
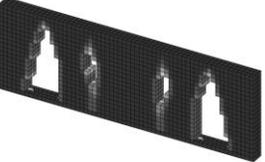
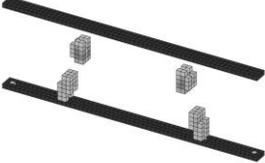
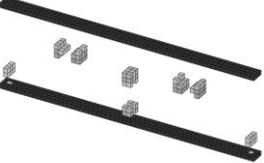
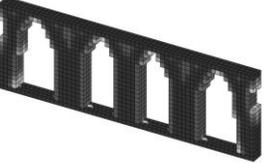
Table 10: VMS-Constrained Hardened State Wall				
	Load (MPa)			
Constraint (MPa)	-2.71	-4.07	-6.78	-13.56
10	 <p>Run 1</p>	 <p>Run 2</p>	 <p>Run 3</p>	 <p>Run 4</p>
20	 <p>Run 5</p>	 <p>Run 6</p>	 <p>Run 7</p>	 <p>Run 8</p>
30	 <p>Run 9</p>	 <p>Run 10</p>	 <p>Run 11</p>	 <p>Run 12</p>
50	 <p>Run 13</p>	 <p>Run 14</p>	 <p>Run 15</p>	 <p>Run 16</p>

Table 11: VMS-Constrained Hardened State Column

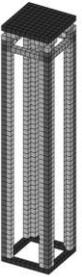
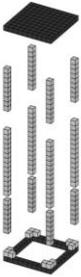
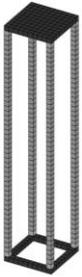
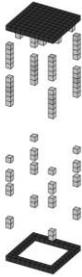
Table 11: VMS-Constrained Hardened State Column					
	Load (MPa)				
Constraint (MPa)	-1.21	-2.42	-3.63	-6.05	-12.10
20	 Run 1	 Run 2	 Run 3	 Run 4	 Run 5
30	 Run 6	 Run 7	 Run 8	 Run 9	 Run 10
40	 Run 11	 Run 12	 Run 13	 Run 14	 Run 15
50	 Run 16	 Run 17	 Run 18	 Run 19	 Run 20

Table 12: Shear Stress-Constrained Hardened State Wall

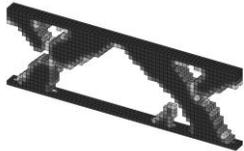
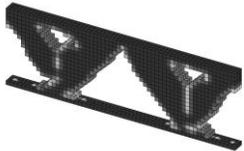
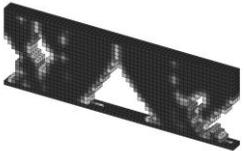
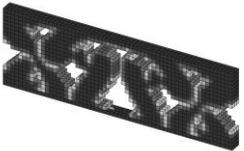
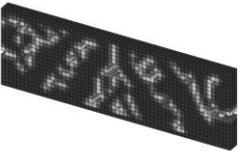
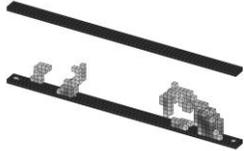
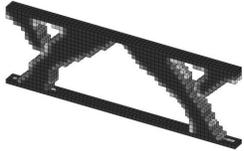
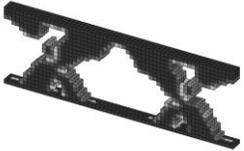
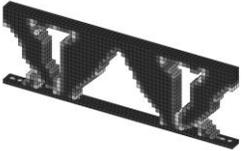
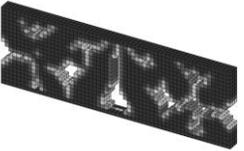
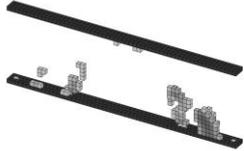
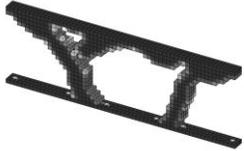
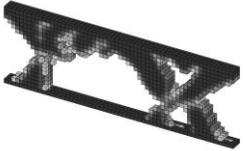
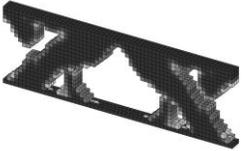
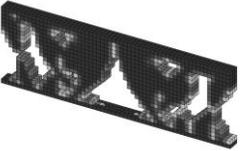
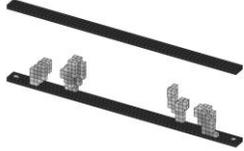
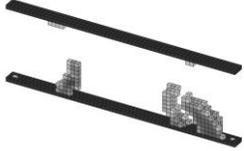
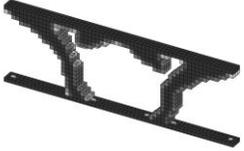
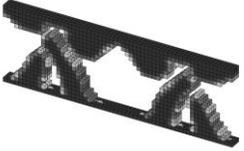
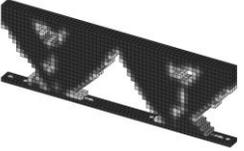
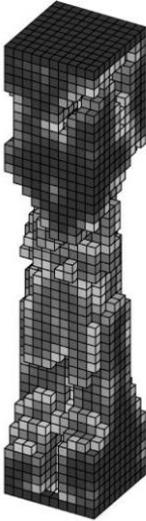
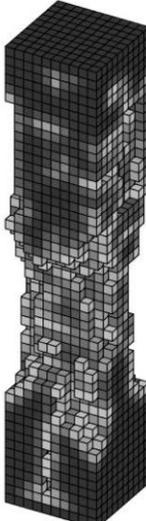
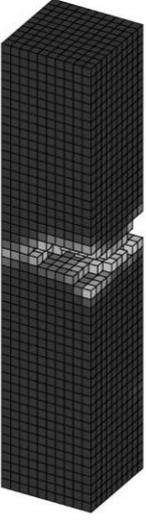
Table 12: Shear Stress-Constrained Hardened State Wall					
	Load (MPa)				
Constraint (MPa)	-1.36	-2.71	-4.07	-6.78	-13.56
5	 Run 1	 Run 2	 Run 3	 Run 4	 Run 5
10	 Run 6	 Run 7	 Run 8	 Run 9	 Run 10
15	 Run 11	 Run 12	 Run 13	 Run 14	 Run 15
20	 Run 16	 Run 17	 Run 18	 Run 19	 Run 20

Table 13: Shear Stress-Constrained Hardened State Column			
	Load (MPa)		
Constraint (MPa)	-8.47	-12.10	-18.15
10	 <p>Run 1</p>	 <p>Run 2</p>	 <p>Run 3</p>

7.1.1 VMS-Constrained Hardened State Wall (Table 10)

The range of applied loads and imposed constraints for the VMS-constrained hardened state wall case are shown in Table 14. Both visual and numerical analysis of the results show that, for a given constraint, the amount of material required increases as the applied load increases. This increase is due to the greater stresses induced in the material due to the increasing load. Addition of more material distributes the load throughout a greater material area, therefore reducing the overall stress level in each element. A similar trend can be seen under constant loading, as the constraint is relaxed from a tight (low-valued) constraint to a loose (high-valued) constraint. A loose constraint requires less material, as the material can be considered *stronger*, while a tight constraint represents a *weaker* material, thus requiring more material to support the load.

Load (MPa)	-2.71	-4.07	-6.78	-13.56
Constraint (MPa)	10	20	30	50

In cases of a large applied load and a tight constraint, this increase of material results in a maximum volume fraction (> 95%) and optimization failure (Runs 3, 4, and 8). This failure indicates that the imposed constraint is unreachable under the applied loading, and that there is no possible arrangement of material within the defined geometry that can support the load and satisfy the constraint, even if the structure is 100% solid. In cases of a small applied load and a loose constraint, the optimization proceeds until a minimum amount of material remains, defined by a volume fraction < 15% (Runs 5, 9, 10, 13, 14, and 15). This minimum indicates that the constraint is sufficiently high that only a minimal amount of material is required to support the load.

Successful optimization terminates or reaches the maximum iteration threshold between 15% and 95% of the original design volume, allowing for analysis of the resulting designs. In these cases, the algorithm reduces the VMS stress summation by forming a series of column-like objects primarily in compression, connected by a series of arches, which minimize the dominant tensile and shear stress components, therefore minimizing VMS. The successful resulting structures, therefore, resemble arched bridges with columns of varying thicknesses. The varying thicknesses correspond to the magnitude of the applied load, with more material required to support larger loads in the form of thicker columns, as can be seen through comparison of Runs 6 and 7.

7.1.2 VMS-Constrained Hardened State Column (Table 11)

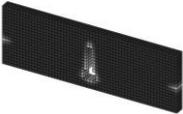
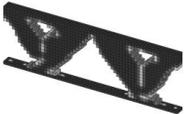
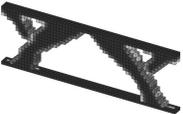
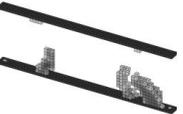
The range of applied loads and imposed constraints for the VMS-constrained hardened state column case are shown in Table 15. Similar to the VMS-constrained hardened state wall case, as the loading is increased, the amount of required material increases accordingly. Large loading and tight constraints result in optimization failure (Run 5) while smaller loads and loose constraints result in minimal volume designs (Runs 6, 11, 16, and 17). When optimization is successful or incomplete, the algorithm again seeks to minimize VMS by dividing the structure into smaller columns under compression at the four corners, connected to the top surface via a dome structure on the interior. The thickness of the columns again corresponds to the magnitude of the applied load. Additionally, if loading is held constant, and the constraint is relaxed from tight to loose, less material is required to support the load.

Load (MPa)	-1.21	-2.42	-3.63	-6.05	-12.1
Constraint (MPa)	20	30	40	50	-

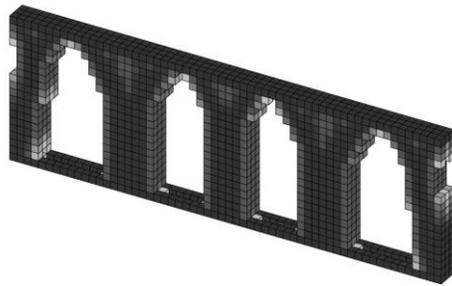
7.1.3 SS-Constrained Hardened State Wall (Table 12)

The range of applied loads and imposed constraints for the SS-constrained hardened state wall case are shown in Table 16. Similar to the VMS-constrained hardened state cases, SS-constrained results show that more material is required to support greater loads under a constant constraint, while if constraints are loosened, less material is required to meet the design specifications. Instances of minimum volume are seen (Runs 6, 11, 16, and 17), though unreachable constraints were not imposed for these loading conditions. Examination of numerical results show that in all cases, as the constraint is tightened or the applied load is increased, the resulting structure is approaching a maximum volume. In order to demonstrate this trend, an additional case was solved, holding the load of 81.3 KN/m constant, while tightening the constraint further to 0.5 MPa (Table 17, Run 21). As predicted, this run resulted in optimization failure by way of an unreachable constraint.

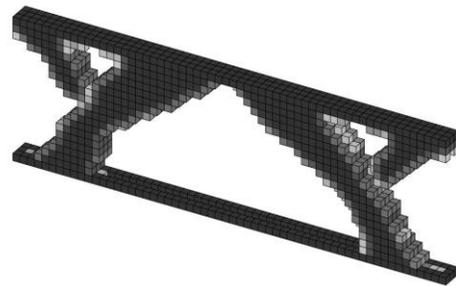
Load (MPa)	-1.36	-2.71	-4.07	-6.78	-13.56
Constraint (MPa)	5	10	15	20	-

	Constraint (MPa)				
Load (MPa)	0.5	5	10	15	20
-2.71	 Run 21	 Run 2	 Run 7	 Run 12	 Run 17
Volume Fraction	Constraint Unreachable	0.58	0.41	0.35	Minimum Volume

However, a major difference between the SS-constrained problem and the VMS-constrained problem can be seen in terms of the successfully optimized designs. While VMS-constrained results display arched bridge shapes of varying dimensions, SS-constrained results can vary dramatically from one constraint or loading condition to another. W-shapes are seen (Runs 2, 9, and 20), as are single arches (Runs 1 and 7), and even structures resembling the shape of a box girder (Runs 12 and 18). Additionally, Figure 43 shows that different structures result from VMS and SS optimization under the same loading conditions and constraints. This difference in optimal structure is due to the difference in stress distributions for VMS and shear stress, causing material placement in locations of greater respective stresses.



VMS Constraint



Shear Stress Constraint

Figure 43: Comparison of VMS-constrained and SS-constrained results for a single loading and constraint. Loading = 2.42 MPa, constraint = 10 MPa.

7.1.4 SS-Constrained Hardened State Column (Table 13)

The range of applied loads and imposed constraints for the SS-constrained hardened state wall case are shown in Table 18. As mentioned previously, only a single constraint is presented here, along with application of much higher applied loads. Results for the SS-constrained hardened state column were inconsistent, asymmetric, and not well formed. The reasons behind these unpredictable results are as yet unknown. It is hypothesized that, due to the region of very low shear stress that develops in the central part of the column, the numerical sensitivity method becomes unstable, and results in small deviations in sensitivity analysis, which compound with each iteration. These sensitivity deviations then cause unpredictable material placements. It is also possible that the optimization parameters used were not suitable for this combination of geometry and constraint. This case requires further work, beginning with further parametric studies of optimization parameters, and potentially extending to development of a new sensitivity formulation.

Table 18: SS-constrained hardened state column loading and constraints			
Load (MPa)	-8.47	-12.1	-18.15
Constraint (MPa)	10		

7.2 Fresh State Case

The next step was to perform studies considering SS constraints and self-weight loading only for the wall geometry. Initial values for SS constraints were set corresponding to experimental yield shear stress value determined through shear vane testing of fresh concrete mortar for 3DCP applications (Rahul, 2019). Values for Young's modulus (0.06 MPa) and Poisson's ratio (0.3) for 3DCP fresh cement mortar were found in Panda (2019) and Wolfs (2018), respectively. It was shown that in all cases, the optimized structure reduced to zero volume as hypothesized, due to the direct relationship between material removal and shear stress reduction under self-weight loading. In an attempt to provide structural continuity and additional boundary conditions, a very small distributed load, ranging from 1.36 – 3.39 KPa was imposed along the top surface of the wall. These loads were applied in the positive (upward) direction, and represented 30 – 75% of the self-weight of the initial volume. These small applied loads were necessary to create a boundary condition along the top surface of the structure, as a fixed condition resulted in negligible shear stress values throughout the structure due to support at both top and bottom surfaces. Application of this small distributed load allowed for successful optimization of SS-constrained wall geometries under very small distributed loads. Loads and constraints for the SS-constrained fresh state wall case are presented in Table 19. The resulting optimized structures are shown in Table 19, with corresponding numerical results are presented in Table 24, Appendix B.

Load (KPa)	1.36	2.03	2.71	3.39
Constraint (KPa)	1	1.5	2	2.5

As predicted, cases involving very small applied loads and loose constraints resulted in minimum volume designs (Runs 6, 9, 10, 13, and 14). Due to the low magnitude of the applied loads, the constraints never proved to be unreachable. Upon examination of the successful runs, a new problem emerged. There were many examples of *disconnectivity* present in the structures, or areas within the structures that appeared to have material suspended above a void space, as can be seen in Figure 44. In the context of a viscoelastic material, there would be no physical means for the material to behave as a solid and to be deposited above a void space. In these cases, there would be no load path for shear transfer from the top surface to the bottom surface through these void regions. Due to the fact that SSTO represents elements with an artificial density of < 0.3 as white (void) regions, the possibility exists that a small amount of material is in fact present. This possibility was refuted upon examination of the artificial material density matrix, which showed zero artificial density for elements in these regions. Further study is required to understand this resulting disconnectivity. Additional connectivity constraints may be required to mitigate this problem, which is present not only in the fresh state wall case, but also in the hardened state column case.

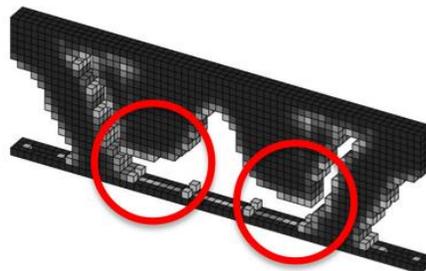
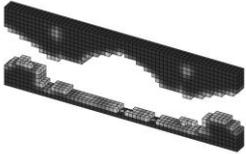
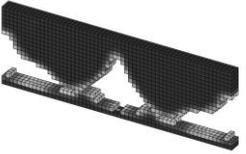
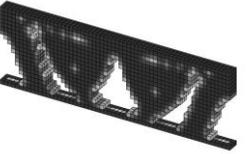
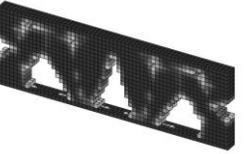
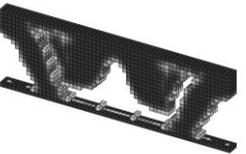
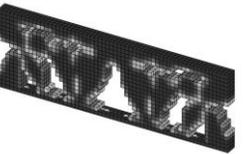
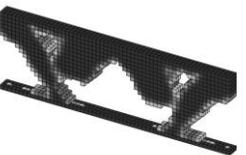
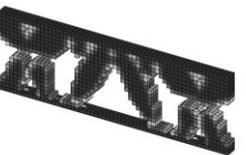
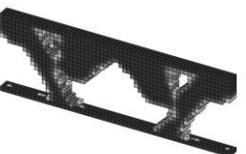
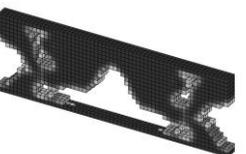


Figure 44: SS-constrained fresh state wall showing disconnectivity (Run 7).

Table 19: Shear Stress-Constrained Fresh State Wall

	Load (KPa)			
Constraint (KPa)	1.36	2.03	2.71	3.39
1	 Run 1	 Run 2	 Run 3	 Run 4
1.5	 Run 5	 Run 6	 Run 7	 Run 8
2	 Run 9	 Run 10	 Run 11	 Run 12
2.5	 Run 13	 Run 14	 Run 15	 Run 16

However, with proper selection of the applied load, it is possible to generate optimized structures that both satisfy the imposed constraint and intuitively represent valid structures consisting of well-defined load paths and full connectivity. Such structures can be seen resulting from Runs 3 and 4. The validity of this imposed distributed load boundary condition was determined by performing ANSYS FEA analysis on one of these optimized structures (Figure 45). The structure chosen for analysis was Run 3 (Table 19).

The results show that, when subjected to an upward distributed load (Figure 45b), only two small areas exceed the shear stress constrained imposed by the optimization process. These areas are adjacent to sharp corners, and therefore correspond to stress concentrations, which can be considered negligible. The remainder of the structure experiences a shear stress distribution below the constraint, indicating that the shear stress that develops in the structure is below the yield shear stress of the material. When the applied load is removed, and the structure is subjected only to its self-weight (Figure 45c), similar stress concentrations arise in other corner regions, but nearly the entire structure satisfies the shear stress constraint. These results indicate that the optimized structure in Figure 45a can be fabricated using 3DCP techniques without collapsing during the printing process. In order for the structure to maintain stability throughout printing, the material used must have a yield shear stress of 1KPa or greater.

Additionally, a downward distributed load was applied to the structure in order to determine the effect of self-weight only loading on the design. When subjected to the same low magnitude distributed load in the direction of gravity (Figure 45d), the structure experiences significant shear stresses originating at corners and radiating outwards, violating the shear stress constraint. This result shows that the application of even minor external loads causes the yield shear stress of the material to be exceeded, reinforcing the notion that structural design for 3DCP fabrication must consider self-weight only loading, as well as the yield shear stress of the concrete material.

Load: 2.71 KPa; Yield shear stress: 1 KPa

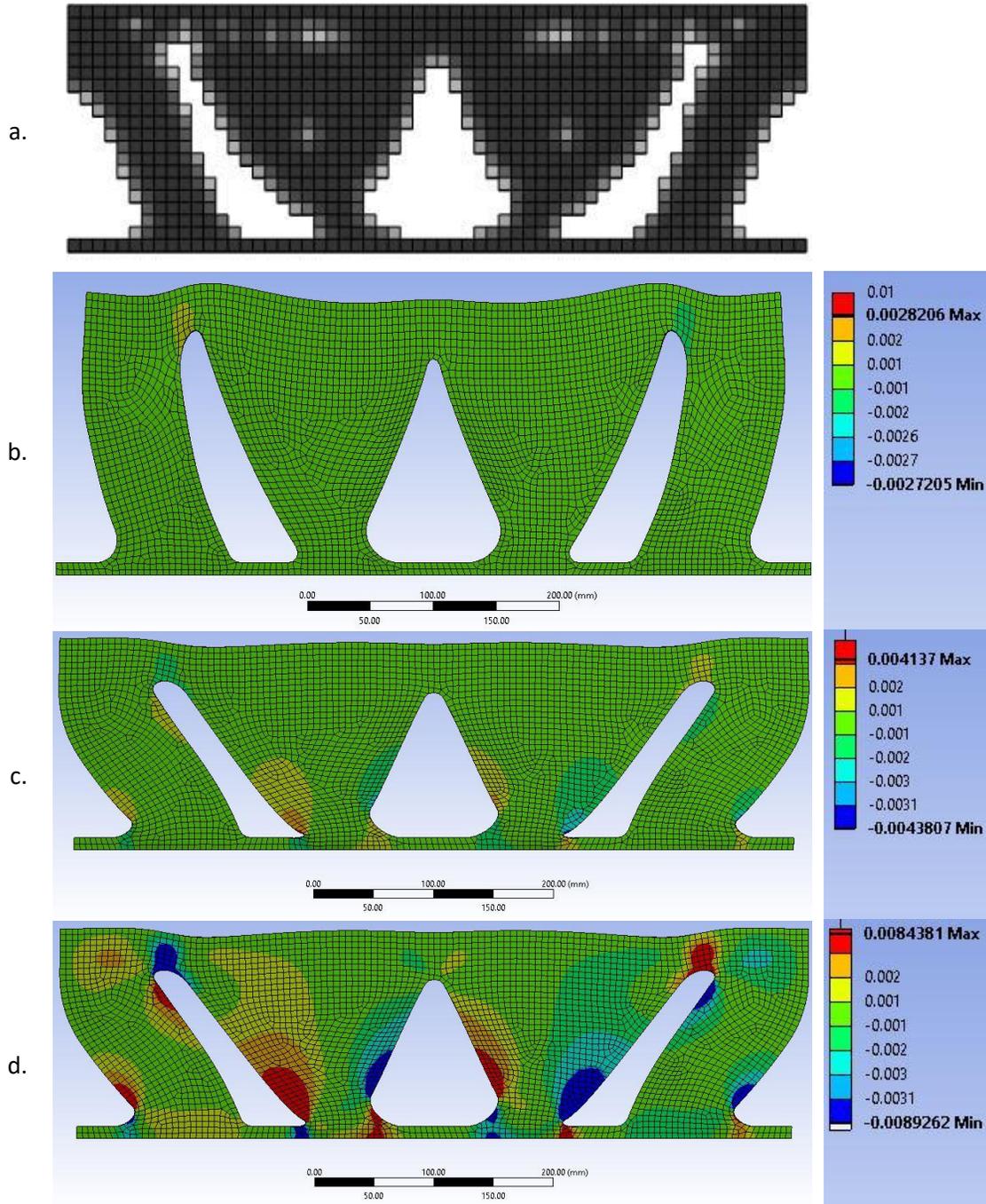


Figure 45: FEA results for SS-constrained fresh state optimization. Green region indicates satisfied constraint. a. SSTO optimized structure; b. ANSYS FEA results (deformed) including upward distributed load; c. ANSYS FEA results (deformed), self-weight only; b. ANSYS FEA results (deformed) including downward distributed load.

Chapter 8: Discussion

8.1 Effect of Yield Shear Stress on Optimal Design

The results of this work show that material yield shear stress or shear strength has a dramatic effect on optimal structural design, in the case of fresh state and hardened state concrete, respectively. In the SS-constrained hardened state wall case, increasing the shear strength of the concrete by a factor of 1.5 changed the optimal design from an arch shape (Table 12, Run 7) to something resembling a box girder (Table 12, Run 12). Both of these designs are commonly used in bridge construction (Figure 46). A similar dramatic change to a W-shape can be seen by reducing the shear strength (Table 12, Run 2). These results differ greatly from those produced by the often-considered VMS-constrained problem, which can be used to consider hardened state failure by the use of a VMS failure criterion. These differences serve to highlight the fact that a single material property can drive optimal structural design. Additionally, selection of a specified optimal design, or designing with a specific geometry in mind, can lead the designer to select the proper materials capable of serving the designated purpose. By studying this effect of yield shear stress or shear strength on optimal design, a connection can be made between an intrinsic material property and design characteristics, such as unsupported angles, overhangs, or cross-sectional dimension limits. This connection between material and structure can then serve to quantify design characteristics in terms of a defined material property, allowing for an understanding of the full design space corresponding to the use of particular materials. Predictions can then be made regarding viable structural designs as well as the materials available for use.



Figure 46: Comparison of SS-constrained hardened state wall results and common bridge designs. Top left: Table 12, Run 7; Top right: Table 12, Run 18; Bottom left: Bridge over the James River, Richmond VA (Photo by Seth Engel); Bottom right: Bonner Bridge box girder (Photo courtesy of HDR).

8.2 Available Material Property Range

As is the case with all structural applications, there exists a limited selection of materials that can serve the design function. This selection of materials available for use is limited by specific materials properties that govern the specific design function. In structural design, this limitation typically comes in the form of yield or ultimate tensile, compressive, or shear strengths, or stiffness, defined by Young's modulus. Materials are then selected for use based on these properties. Additional properties such as Poisson's ratio or shear moduli may also limit structural design. As such, the design engineer must understand the properties that govern design, as well as be able to quantify the properties that can be used. The topology optimization tool presented here can serve to define these bounds on potential materials used in structural design through interpretation of the resulting optimal structures, or more importantly, of the overall success or failure of the constrained optimization process. If the optimization process succeeds, the material constraints are met, and a structure can be designed within the design domain. If, however, the constraint is unreachable, then the material is not suitable for design, as there is no possible arrangement of material that can support the given loading conditions. Similarly, if the optimization process proceeds to a minimum volume, then the material is not optimal for design. In these cases, increasing the strength of the material does not result in design changes, and a weaker material can be used that may perform better in terms of other material properties, such as stiffness. SSTO is capable of defining these material property bounds for shear strength and yield shear stress by imposing a shear stress constraint on the design process.

This available material property range can be visualized by re-examining Table 17. The tight constraint of 0.5 MPa results in a design that cannot serve the required function, whereas the loose constraint of 20 MPa results in a design of minimum volume. These results indicate a range of optimal shear strengths between 0.5 and 20 MPa for the given loading and boundary conditions. Within this range, materials may either be selected or designed in order to serve the desired function and support the applied load. Outside of this range, the material would either not serve the required function or not be optimal for use. From these results, then, an effective material property range emerges, which can guide the design engineer in selecting materials for the given application, providing another link between materials and structures in the design process. Additionally, this range can help material scientists develop materials with desired material properties.

8.3 Design Sensitivity

By altering the material property within the available range, the resulting optimal design may or may not vary significantly. This variance can be considered *design sensitivity*. High design sensitivity corresponds to large changes in the design variable (volume) when the constraint (shear stress) is changed by only a small amount. Similarly, low design sensitivity corresponds to minor changes in the design variable over a wide range of constraints. This sensitivity can be visualized graphically by plotting the design variable vs. the constraint (Figure 47).

Consider the SS-constrained hardened state case presented in Table 17 and defined numerically in Table 20. Runs 2 and 7 show a dramatic difference in volume, indicating high design sensitivity, while Runs 7 and 12 display low design sensitivity, as the volume does not change drastically over a much larger range of imposed constraints. In this way, design sensitivity itself may vary throughout the

available material property range, with high or low gradients across small material property ranges. Understanding this design sensitivity can provide insight into material design, structural function, and the fabrication process. In the fresh state case, if an experimentally determined yield shear stress value for concrete mortar lies within a high design sensitivity range, any experimental or measurement errors may substantially change the printability of the structure. It would therefore be beneficial to use a material with low design sensitive properties in order to achieve a printable structure.

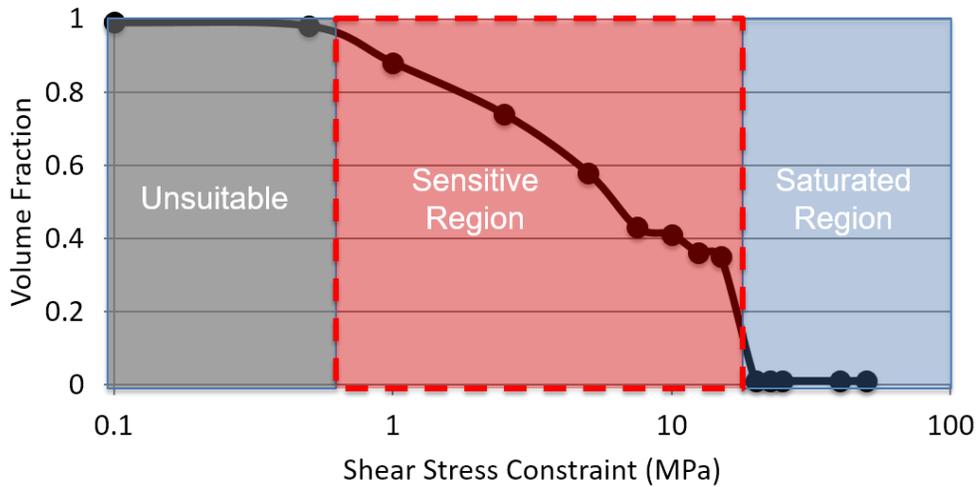


Figure 47: Design sensitivity of shear stress-constrained hardened state wall (Table 12).

Run	Load (MPa)	Constraint (MPa)	Result	Volume	Max Shear (MPa)
21	-2.71	0.5	constraint unreachable	> 95%	> 0.5
2	-2.71	5	max iterations	57.69%	4.99
7	-2.71	10	max iterations	41.42%	9.92
12	-2.71	15	max iterations	34.49%	14.86
17	-2.71	20	minimum volume reached	< 15.00%	6.45

It should be noted that this design sensitivity was also seen in the case of the shear stress-constrained problem solved by PTO. Though presented in terms of tensile strength, the problem considered shear strength due to the use of linear plane strain transformation equations. A side-by-side comparison of the design sensitivity curves shows similar behavior when geometry and loading are held constant, but the constraint is varied (Figure 48).

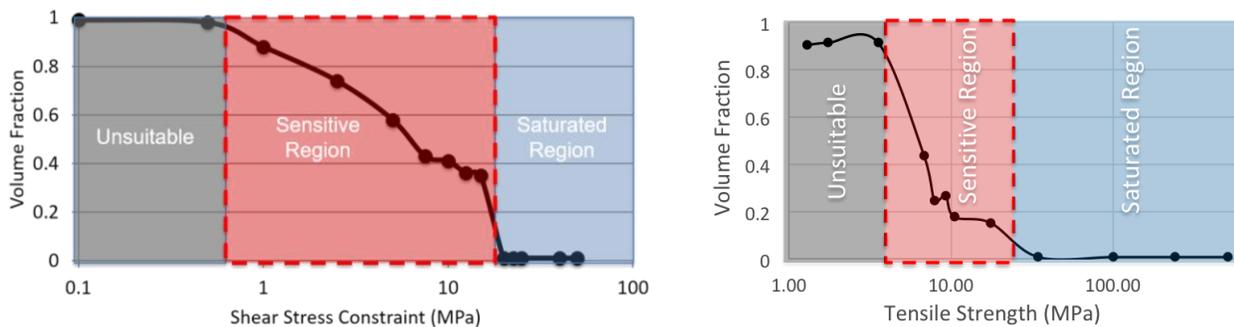


Figure 48: Comparison of design sensitivities for SSTS SS-constrained hardened state wall (left) and PTO cantilever beam (right).

Design sensitivity may arise from varying the material property in question, or by changing other material properties such as density or stiffness. It can also be set in terms of any dependent quantity, such as compliance, stress, or strain. Design sensitivity is therefore another example of the material-structure relationship at work, with optimal structures dictating preferred material selection or design, and sensitive material properties limiting or expanding those available designs.

Chapter 9: Future Work

The work presented here represents a jumping off point for further research into the relationship between materials, structural design, and fabrication process. There are several directions for future research that could develop this relationship, approached from all three angles. Ideas for these future research directions are presented here.

9.1 Material Properties

Yield shear stress is but one of the material properties that imposes limitations on design due to the fabrication process, and its inclusion here as a static, singular quantity does not represent the complete yielding behavior of the material. Several factors relating to yield shear stress were not considered here, and future research could include these factors for a more complete representation of material behavior. To begin, the yield shear stress here is considered only in one coordinate plane. However, in 3 dimensional structures, shear stress occurs on three material planes. Inclusion of multiple planes of shear stress in the constrained optimization process could be achieved in one of two ways. First, multiple constraints could be included in the optimization process. MMA has the ability to incorporate multiple constraints, and could therefore consider shear stress constraints on all three planes simultaneously. Second, an equivalent shear stress representation could be developed, similar to the equivalent stress value represented by VMS. An equivalent value would not require the addition of multiple constraints.

Multiple constraints could also be incorporated to optimize the structure considering local stress values. In its current form, SSTO optimizes the structure based on a global shear stress maximum. However, by constraining the shear stress in each individual element, a locally optimized structure would emerge, allowing for a greater potential design space. The reason for not including multiple constraints in SSTO is the increased computational expense, and the significant time required for optimization.

Another potential use of multiple constraints in the TO process may help address the major problem experienced with column geometry: structural connectivity. If the loading was too small, the structure would separate and often reduce to zero volume. In order to prevent this from happening, and force the structure to maintain a load path through the entire geometry, a connectivity constraint could be developed. Ideas for connectivity constraints have been proposed (Li, 2016; Du, 2018) and could be incorporated as an additional constraint in the optimization process.

Yield shear stress can also be further developed by considering time-dependence. As discussed in section 2.1.11, yield shear stress is a time-dependent property. As time progresses, yield shear stress increases, allowing for the structure to take on additional load without deformation. This time dependence could be included in the optimization process as a reflection of the printing process. If the geometry and printing parameters are known, the time it takes to print a single layer can be calculated. If the thixotropy of the material is also known, the yield shear stress in each layer can be adjusted based on print time. Models of this material-fabrication relationship have been developed (Kruger, 2020) and can be incorporated into the TO algorithm.

Yield shear stress is not the only material property that limits structural design. Stiffness, tensile and compressive strength, density, viscosity, among others all have a bearing on structural design. Often, these properties work in tandem, either as a result of the fabrication process or as a result of the indented structural application. Modification of the objective and/or constraint functions and their sensitivities allow for investigation into the limitations of these other material properties. Additionally, multiple material properties could be considered simultaneously, either through the use of multiple constraints, or by creating a singular constraint function as a linear combination of constraint functions representing the determination of multiple material properties.

9.2 Real-World Application and Testing

The work presented here is theoretical in nature. Though validation tests have been performed using ANSYS FEA software, no physical experiments have been conducted to validate the printability of the optimized structures. Therefore, future physical experiments should be designed to test the application of this work in the real world. To that effort, an experimental program should be designed to include all aspects investigated here: materials, design, and fabrication. After selecting an initial geometry representative of a common structural component, initial parametric studies performed using SSTO would develop a range of desired yield shear stress values for the concrete mortar to be used in the printing process. Then, concrete mortar 3DCP mixes would be designed and tested for yield shear stress. The experimentally determined material properties would be input into SSTO and an optimized structure would be developed. This optimized structural design would then be converted into STL or Gcode for printing. Finally, the print would be attempted, and the results analyzed. This process could be repeated for different geometries, mixes, and print parameters.

Currently, the initial geometries easily programmable in SSTO are limited to solid rectangular shapes. Modifications can be made to allow for initial geometries containing voids, or for those of non-rectangular shape. Incorporating complex initial geometries will allow for optimization of more real-world structural elements corresponding to architectural constraints. Additionally, complex initial geometries allow for development of intricate structures otherwise not considered by design engineers.

The purpose of this work is to design structures optimized for the printing process. To that end, it considers only the fresh state material properties and self-weight loading. This work does not consider the properties of the structure in its hardened state, nor the structural performance under applied loading. The relationship between printable structures and structural performance has yet to be established. It is suggested here that TO may be able to help develop this relationship by considering material properties of both fresh and hardened concrete, as well as by considering multiple loading states. This process could be iterative in nature, where an initial design is determined using fresh state properties, followed by successive FEA analyses of that design, considering hardened state properties under varying loads. The process could also be direct, by incorporating multiple constraints and/or objective functions and performing FEA calculations under multiple loading states (self-weight only and applied). It may also be possible to develop a relationship between shear stresses developed for very small distributed loads (section 7.2) and real-world applied loads, and to use this relationship in design.

9.3 The Material – Structure Relationship in Design

This work represents the first step on the path toward integrated design. It shows a direct connection between material and structure, and how the fabrication process can serve to limit or expand the design space created through this connection. The coupling of material, structure, and fabrication process has ramifications far beyond concrete 3D printing. Within the greater realm of additive manufacturing with any material, understanding design in terms of material properties and fabrication process is even more pronounced, as the processes are more precise, the designs more complex, and the applications more specialized than those of traditional methods of fabrication. However, even traditional fabrication methods can be analyzed in such a way. Every fabrication process has inherent material and structural design limitations. Every material has limitations on structural application due to inherent material properties. Every structural design has a set of material property and fabrication requirements. This study demonstrates that a full understanding of these limitations and requirements can be developed quantitatively through a thorough understanding of underlying material properties, and how they interact with the fabrication process and structural design requirements. Quantifying design in terms of material properties, then, allows a full range of design options to be explored, and optimal materials selected or designed to serve the purpose. With this work, the stage has been set for future exploration into the world of integrated design. Similar to the work done in this study, such investigations could begin by developing novel methods for quantifying design, or they could begin by altering current design methods to explore the material-structure interaction. Whatever the means for development, once harnessed, integrated design will be a powerful tool, both for design engineers and researchers alike.

Chapter 10: Conclusion

Engineering design is the intersection of materials, structure, and fabrication process. The structure must be designed to serve a function, must be composed of a material capable of serving that function, and must be fabricated using techniques capable of working with the source material and converting it to a finished product. The engineering and economic challenges of the modern world require innovative solutions that take full advantage of the vast array of available engineering materials and fabrication methods, both of which continue to expand. In this environment, optimal solutions are not only desired, they are necessary. To meet that end, no single piece of the design puzzle can be prioritized over the others, as it is the symbiotic relationship between them that creates truly optimal designs.

This work sought to begin the development of a symbiotic material – structure relationship contextualized by fabrication process with the goal of optimal structural design in mind. The relationship was developed in the context an emerging fabrication technique, 3DCP, which has the potential to revolutionize the construction industry. This construction revolution, however, can only take place if material, structure, and fabrication method each inform each other throughout the design process.

In order to accomplish this goal, the fabrication process was examined from a process, design, and material perspective. A single material property, yield shear stress, was identified as being a vital and limiting factor in both successful design and fabrication. This material property was then contextualized as a constraint on design, and a computational design method was employed to create optimal designs based on this imposed constraint. This design method involved development of a novel topology optimization algorithm, and the optimization problem was posed to simulate the 3DCP environment. Specifically, this environment required consideration of a design domain with a fixed base, a geometry under self-weight loading only, and a design constrained by a maximum shear stress as given by material yield shear stress.

Several optimization problems were solved, considering both hardened concrete design as well as fresh state design. From the resulting optimized structural designs, a picture of this material – structure relationship in design began to emerge through varying the yield shear stress in the material as well as the loading applied to the structure. This relationship can be used in the future to quantify available design spaces, geometric limits, structural applications, and fabrication requirements in terms of material properties, giving design engineers a better understanding of how to solve design problems, and giving material scientists a means to target material development for specific applications. Through further development of this relationship, the design process can become truly holistic in approach, which will prove vital when solving the increasingly complex engineering challenges of the future.

Appendix A: Stiffness Matrix Comparison

2x2x2 GQ: Poisson's Ratio = 0.3

0.24	0.08	0.08	-0.11	0.02	0.02	-0.09	-0.08	0.01	0.05	-0.02	0.04	0.05	0.04	-0.02	-0.09	0.01	-0.08	-0.06	-0.04	-0.04	-0.01	-0.01	-0.01
0.08	0.24	0.08	-0.02	0.05	0.04	-0.08	-0.09	0.01	0.02	-0.11	0.02	0.04	0.05	-0.02	-0.01	-0.01	-0.01	-0.04	-0.06	-0.04	0.01	-0.09	-0.08
0.08	0.08	0.24	-0.02	0.04	0.05	-0.01	-0.01	-0.01	0.04	-0.02	0.05	0.02	0.02	-0.11	-0.08	0.01	-0.09	-0.04	-0.04	-0.06	0.01	-0.08	-0.09
-0.11	-0.02	-0.02	0.24	-0.08	-0.08	0.05	0.02	-0.04	-0.09	0.08	-0.01	-0.09	-0.01	0.08	0.05	-0.04	0.02	-0.01	0.01	0.01	-0.06	0.04	0.04
0.02	0.05	0.04	-0.08	0.24	0.08	-0.02	-0.11	0.02	0.08	-0.09	0.01	0.01	-0.01	-0.01	-0.04	0.05	-0.02	-0.01	-0.09	-0.08	0.04	-0.06	-0.04
0.02	0.04	0.05	-0.08	0.08	0.24	-0.04	-0.02	0.05	0.01	-0.01	-0.01	0.08	0.01	-0.09	-0.02	0.02	-0.11	-0.01	-0.08	-0.09	0.04	-0.04	-0.06
-0.09	-0.08	-0.01	0.05	-0.02	-0.04	0.24	0.08	-0.08	-0.11	0.02	-0.02	-0.06	-0.04	0.04	-0.01	-0.01	0.01	0.05	0.04	0.02	-0.09	0.01	0.08
-0.08	-0.09	-0.01	0.02	-0.11	-0.02	0.08	0.24	-0.08	-0.02	0.05	-0.04	-0.04	-0.06	0.04	0.01	-0.09	0.08	0.04	0.05	0.02	-0.01	-0.01	0.01
0.01	0.01	-0.01	-0.04	0.02	0.05	-0.08	-0.08	0.24	0.02	-0.04	0.05	0.04	0.04	-0.06	-0.01	0.08	-0.09	-0.02	-0.02	-0.11	0.08	-0.01	-0.09
0.05	0.02	0.04	-0.09	0.08	0.01	-0.11	-0.02	0.02	0.24	-0.08	0.08	-0.01	0.01	-0.01	-0.06	0.04	-0.04	-0.09	-0.01	-0.08	0.05	-0.04	-0.02
-0.02	-0.11	-0.02	0.08	-0.09	-0.01	0.02	0.05	-0.04	-0.08	0.24	-0.08	-0.01	-0.09	0.08	0.04	-0.06	0.04	0.01	-0.01	0.01	-0.04	0.05	0.02
0.04	0.02	0.05	-0.01	0.01	-0.01	-0.02	-0.04	0.05	0.08	-0.08	0.24	0.01	0.08	-0.09	-0.04	0.04	-0.06	-0.08	-0.01	-0.09	0.02	-0.02	-0.11
0.05	0.04	0.02	-0.09	0.01	0.08	-0.06	-0.04	0.04	-0.01	-0.01	0.01	0.24	0.08	-0.08	-0.11	0.02	-0.02	-0.09	-0.08	-0.01	0.05	-0.02	-0.04
0.04	0.05	0.02	-0.01	-0.01	0.01	-0.04	-0.06	0.04	0.01	-0.09	0.08	0.08	0.24	-0.08	-0.02	0.05	-0.04	-0.08	-0.09	-0.01	0.02	-0.11	-0.02
-0.02	-0.02	-0.11	0.08	-0.01	-0.09	0.04	0.04	-0.06	-0.01	0.08	-0.09	-0.08	-0.08	0.24	0.02	-0.04	0.05	0.01	0.01	-0.01	-0.04	0.02	0.05
-0.09	-0.01	-0.08	0.05	-0.04	-0.02	-0.01	0.01	-0.01	-0.06	0.04	-0.04	-0.11	-0.02	0.02	0.24	-0.08	0.08	0.05	0.02	0.04	-0.09	0.08	0.01
0.01	-0.01	0.01	-0.04	0.05	0.02	-0.01	-0.09	0.08	0.04	-0.06	0.04	0.02	0.05	-0.04	-0.08	0.24	-0.08	-0.02	-0.11	-0.02	0.08	-0.09	-0.01
-0.08	-0.01	-0.09	0.02	-0.02	-0.11	0.01	0.08	-0.09	-0.04	0.04	-0.06	-0.02	-0.04	0.05	0.08	-0.08	0.24	0.04	0.02	0.05	-0.01	0.01	-0.01
-0.06	-0.04	-0.04	-0.01	-0.01	-0.01	0.05	0.04	-0.02	-0.09	0.01	-0.08	-0.09	-0.08	0.01	0.05	-0.02	0.04	0.24	0.08	0.08	-0.11	0.02	0.02
-0.04	-0.06	-0.04	0.01	-0.09	-0.08	0.04	0.05	-0.02	-0.01	-0.01	-0.01	-0.08	-0.09	0.01	0.02	-0.11	0.02	0.08	0.24	0.08	-0.02	0.05	0.04
-0.04	-0.04	-0.06	0.01	-0.08	-0.09	0.02	0.02	-0.11	-0.08	0.01	-0.09	-0.01	-0.01	-0.01	0.04	-0.02	0.05	0.08	0.08	0.24	-0.02	0.04	0.05
-0.01	0.01	0.01	-0.06	0.04	0.04	-0.09	-0.01	0.08	0.05	-0.04	0.02	0.05	0.02	-0.04	-0.09	0.08	-0.01	-0.11	-0.02	-0.02	0.24	-0.08	-0.08
-0.01	-0.09	-0.08	0.04	-0.06	-0.04	0.01	-0.01	-0.01	-0.04	0.05	-0.02	-0.02	-0.11	0.02	0.08	-0.09	0.01	0.02	0.05	0.04	-0.08	0.24	0.08
-0.01	-0.08	-0.09	0.04	-0.04	-0.06	0.08	0.01	-0.09	-0.02	0.02	-0.11	-0.04	-0.02	0.05	0.01	-0.01	-0.01	0.02	0.04	0.05	-0.08	0.08	0.24

Top3D: Poisson's Ratio = 0.3

0.24	0.08	0.08	-0.11	0.02	0.02	-0.09	-0.08	0.01	0.05	-0.02	0.04	0.05	0.04	-0.02	-0.09	0.01	-0.08	-0.06	-0.04	-0.04	-0.01	-0.01	-0.01
0.08	0.24	0.08	-0.02	0.05	0.04	-0.08	-0.09	0.01	0.02	-0.11	0.02	0.04	0.05	-0.02	-0.01	-0.01	-0.01	-0.04	-0.06	-0.04	0.01	-0.09	-0.08
0.08	0.08	0.24	-0.02	0.04	0.05	-0.01	-0.01	-0.01	0.04	-0.02	0.05	0.02	0.02	-0.11	-0.08	0.01	-0.09	-0.04	-0.04	-0.06	0.01	-0.08	-0.09
-0.11	-0.02	-0.02	0.24	-0.08	-0.08	0.05	0.02	-0.04	-0.09	0.08	-0.01	-0.09	-0.01	0.08	0.05	-0.04	0.02	-0.01	0.01	0.01	-0.06	0.04	0.04
0.02	0.05	0.04	-0.08	0.24	0.08	-0.02	-0.11	0.02	0.08	-0.09	0.01	0.01	-0.01	-0.01	-0.04	0.05	-0.02	-0.01	-0.09	-0.08	0.04	-0.06	-0.04
0.02	0.04	0.05	-0.08	0.08	0.24	-0.04	-0.02	0.05	0.01	-0.01	-0.01	0.08	0.01	-0.09	-0.02	0.02	-0.11	-0.01	-0.08	-0.09	0.04	-0.04	-0.06
-0.09	-0.08	-0.01	0.05	-0.02	-0.04	0.24	0.08	-0.08	-0.11	0.02	-0.02	-0.06	-0.04	0.04	-0.01	-0.01	0.01	0.05	0.04	0.02	-0.09	0.01	0.08
-0.08	-0.09	-0.01	0.02	-0.11	-0.02	0.08	0.24	-0.08	-0.02	0.05	-0.04	-0.04	-0.06	0.04	0.01	-0.09	0.08	0.04	0.05	0.02	-0.01	-0.01	0.01
0.01	0.01	-0.01	-0.04	0.02	0.05	-0.08	-0.08	0.24	0.02	-0.04	0.05	0.04	0.04	-0.06	-0.01	0.08	-0.09	-0.02	-0.02	-0.11	0.08	-0.01	-0.09
0.05	0.02	0.04	-0.09	0.08	0.01	-0.11	-0.02	0.02	0.24	-0.08	0.08	-0.01	0.01	-0.01	-0.06	0.04	-0.04	-0.09	-0.01	-0.08	0.05	-0.04	-0.02
-0.02	-0.11	-0.02	0.08	-0.09	-0.01	0.02	0.05	-0.04	-0.08	0.24	-0.08	-0.01	-0.09	0.08	0.04	-0.06	0.04	0.01	-0.01	0.01	-0.04	0.05	0.02
0.04	0.02	0.05	-0.01	0.01	-0.01	-0.02	-0.04	0.05	0.08	-0.08	0.24	0.01	0.08	-0.09	-0.04	0.04	-0.06	-0.08	-0.01	-0.09	0.02	-0.02	-0.11
0.05	0.04	0.02	-0.09	0.01	0.08	-0.06	-0.04	0.04	-0.01	-0.01	0.01	0.24	0.08	-0.08	-0.11	0.02	-0.02	-0.09	-0.08	-0.01	0.05	-0.02	-0.04
0.04	0.05	0.02	-0.01	-0.01	0.01	-0.04	-0.06	0.04	0.01	-0.09	0.08	0.08	0.24	-0.08	-0.02	0.05	-0.04	-0.08	-0.09	-0.01	0.02	-0.11	-0.02
-0.02	-0.02	-0.11	0.08	-0.01	-0.09	0.04	0.04	-0.06	-0.01	0.08	-0.09	-0.08	-0.08	0.24	0.02	-0.04	0.05	0.01	0.01	-0.01	-0.04	0.02	0.05
-0.09	-0.01	-0.08	0.05	-0.04	-0.02	-0.01	0.01	-0.01	-0.06	0.04	-0.04	-0.11	-0.02	0.02	0.24	-0.08	0.08	0.05	0.02	0.04	-0.09	0.08	0.01
0.01	-0.01	0.01	-0.04	0.05	0.02	-0.01	-0.09	0.08	0.04	-0.06	0.04	0.02	0.05	-0.04	-0.08	0.24	-0.08	-0.02	-0.11	-0.02	0.08	-0.09	-0.01
-0.08	-0.01	-0.09	0.02	-0.02	-0.11	0.01	0.08	-0.09	-0.04	0.04	-0.06	-0.02	-0.04	0.05	0.08	-0.08	0.24	0.04	0.02	0.05	-0.01	0.01	-0.01
-0.06	-0.04	-0.04	-0.01	-0.01	-0.01	0.05	0.04	-0.02	-0.09	0.01	-0.08	-0.09	-0.08	0.01	0.05	-0.02	0.04	0.24	0.08	0.08	-0.11	0.02	0.02
-0.04	-0.06	-0.04	0.01	-0.09	-0.08	0.04	0.05	-0.02	-0.01	-0.01	-0.01	-0.08	-0.09	0.01	0.02	-0.11	0.02	0.08	0.24	0.08	-0.02	0.05	0.04
-0.04	-0.04	-0.06	0.01	-0.08	-0.09	0.02	0.02	-0.11	-0.08	0.01	-0.09	-0.01	-0.01	-0.01	0.04	-0.02	0.05	0.08	0.08	0.24	-0.02	0.04	0.05
-0.01	0.01	0.01	-0.06	0.04	0.04	-0.09	-0.01	0.08	0.05	-0.04	0.02	0.05	0.02	-0.04	-0.09	0.08	-0.01	-0.11	-0.02	-0.02	0.24	-0.08	-0.08
-0.01	-0.09	-0.08	0.04	-0.06	-0.04	0.01	-0.01	-0.01	-0.04	0.05	-0.02	-0.02	-0.11	0.02	0.08	-0.09	0.01	0.02	0.05	0.04	-0.08	0.24	0.08
-0.01	-0.08	-0.09	0.04	-0.04	-0.06	0.08	0.01	-0.09	-0.02	0.02	-0.11	-0.04	-0.02	0.05	0.01	-0.01	-0.01	0.02	0.04	0.05	-0.08	0.08	0.24

2x2x2 GQ: Poisson's Ratio = 0.1

0.21	0.05	0.05	-0.06	-0.03	-0.03	-0.07	-0.05	-0.01	0.03	0.03	0.02	0.03	0.02	0.03	-0.07	-0.01	-0.05	-0.05	-0.02	-0.02	-0.02	0.01	0.01
0.05	0.21	0.05	0.03	0.03	0.02	-0.05	-0.07	-0.01	-0.03	-0.06	-0.03	0.02	0.03	0.03	0.01	-0.02	0.01	-0.02	-0.05	-0.02	-0.01	-0.07	-0.05
0.05	0.05	0.21	0.03	0.02	0.03	0.01	0.01	-0.02	0.02	0.03	0.03	-0.03	-0.03	-0.06	-0.05	-0.01	-0.07	-0.02	-0.02	-0.05	-0.01	-0.05	-0.07
-0.06	0.03	0.03	0.21	-0.05	-0.05	0.03	-0.03	-0.02	-0.07	0.05	0.01	-0.07	0.01	0.05	0.03	-0.02	-0.03	-0.02	-0.01	-0.01	-0.05	0.02	0.02
-0.03	0.03	0.02	-0.05	0.21	0.05	0.03	-0.06	-0.03	0.05	-0.07	-0.01	-0.01	-0.02	0.01	-0.02	0.03	0.03	0.01	-0.07	-0.05	0.02	-0.05	-0.02
-0.03	0.02	0.03	-0.05	0.05	0.21	-0.02	0.03	0.03	-0.01	0.01	-0.02	0.05	-0.01	-0.07	0.03	-0.03	-0.06	0.01	-0.05	-0.07	0.02	-0.02	-0.05
-0.07	-0.05	0.01	0.03	0.03	-0.02	0.21	0.05	-0.05	-0.06	-0.03	0.03	-0.05	-0.02	0.02	-0.02	0.01	-0.01	0.03	0.02	-0.03	-0.07	-0.01	0.05
-0.05	-0.07	0.01	-0.03	-0.06	0.03	0.05	0.21	-0.05	0.03	0.03	-0.02	-0.02	-0.05	0.02	-0.01	-0.07	0.05	0.02	0.03	-0.03	0.01	-0.02	-0.01
-0.01	-0.01	-0.02	-0.02	-0.03	0.03	-0.05	-0.05	0.21	-0.03	-0.02	0.03	0.02	0.02	-0.05	0.01	0.05	-0.07	0.03	0.03	-0.06	0.05	0.01	-0.07
0.03	-0.03	0.02	-0.07	0.05	-0.01	-0.06	0.03	-0.03	0.21	-0.05	0.05	-0.02	-0.01	0.01	-0.05	0.02	-0.02	-0.07	0.01	-0.05	0.03	-0.02	0.03
0.03	-0.06	0.03	0.05	-0.07	0.01	-0.03	0.03	-0.02	-0.05	0.21	-0.05	0.01	-0.07	0.05	0.02	-0.05	0.02	-0.01	-0.02	-0.01	-0.02	0.03	-0.03
0.02	-0.03	0.03	0.01	-0.01	-0.02	0.03	-0.02	0.03	0.05	-0.05	0.21	-0.01	0.05	-0.07	-0.02	0.02	-0.05	-0.05	0.01	-0.07	-0.03	0.03	-0.06
0.03	0.02	-0.03	-0.07	-0.01	0.05	-0.05	-0.02	0.02	-0.02	0.01	-0.01	0.21	0.05	-0.05	-0.06	-0.03	0.03	-0.07	-0.05	0.01	0.03	0.03	-0.02
0.02	0.03	-0.03	0.01	-0.02	-0.01	-0.02	-0.05	0.02	-0.01	-0.07	0.05	0.05	0.21	-0.05	0.03	0.03	-0.02	-0.05	-0.07	0.01	-0.03	-0.06	0.03
0.03	0.03	-0.06	0.05	0.01	-0.07	0.02	0.02	-0.05	0.01	0.05	-0.07	-0.05	-0.05	0.21	-0.03	-0.02	0.03	-0.01	-0.01	-0.02	-0.02	-0.03	0.03
-0.07	0.01	-0.05	0.03	-0.02	0.03	-0.02	-0.01	0.01	-0.05	0.02	-0.02	-0.06	0.03	-0.03	0.21	-0.05	0.05	0.03	-0.03	0.02	-0.07	0.05	-0.01
-0.01	-0.02	-0.01	-0.02	0.03	-0.03	0.01	-0.07	0.05	0.02	-0.05	0.02	-0.03	0.03	-0.02	-0.05	0.21	-0.05	0.03	-0.06	0.03	0.05	-0.07	0.01
-0.05	0.01	-0.07	-0.03	0.03	-0.06	-0.01	0.05	-0.07	-0.02	0.02	-0.05	0.03	-0.02	0.03	0.05	-0.05	0.21	0.02	-0.03	0.03	0.01	-0.01	-0.02
-0.05	-0.02	-0.02	-0.02	0.01	0.01	0.03	0.02	0.03	-0.07	-0.01	-0.05	-0.07	-0.05	-0.01	0.03	0.03	0.02	0.21	0.05	0.05	-0.06	-0.03	-0.03
-0.02	-0.05	-0.02	-0.01	-0.07	-0.05	0.02	0.03	0.03	0.01	-0.02	0.01	-0.05	-0.07	-0.01	-0.03	-0.06	-0.03	0.05	0.21	0.05	0.03	0.03	0.02
-0.02	-0.02	-0.05	-0.01	-0.05	-0.07	-0.03	-0.03	-0.06	-0.05	-0.01	-0.07	0.01	0.01	-0.02	0.02	0.03	0.03	0.05	0.05	0.21	0.03	0.02	0.03
-0.02	-0.01	-0.01	-0.05	0.02	0.02	-0.07	0.01	0.05	0.03	-0.02	-0.03	0.03	-0.03	-0.02	-0.07	0.05	0.01	-0.06	0.03	0.03	0.21	-0.05	-0.05
0.01	-0.07	-0.05	0.02	-0.05	-0.02	-0.01	-0.02	0.01	-0.02	0.03	0.03	0.03	-0.06	-0.03	0.05	-0.07	-0.01	-0.03	0.03	0.02	-0.05	0.21	0.05
0.01	-0.05	-0.07	0.02	-0.02	-0.05	0.05	-0.01	-0.07	0.03	-0.03	-0.06	-0.02	0.03	0.03	-0.01	0.01	-0.02	-0.03	0.02	0.03	-0.05	0.05	0.21

Top3D: Poisson's Ratio = 0.1

0.21	0.05	0.05	-0.06	-0.03	-0.03	-0.07	-0.05	-0.01	0.03	0.03	0.02	0.03	0.02	0.03	-0.07	-0.01	-0.05	-0.05	-0.02	-0.02	-0.02	0.01	0.01
0.05	0.21	0.05	0.03	0.03	0.02	-0.05	-0.07	-0.01	-0.03	-0.06	-0.03	0.02	0.03	0.03	0.01	-0.02	0.01	-0.02	-0.05	-0.02	-0.01	-0.07	-0.05
0.05	0.05	0.21	0.03	0.02	0.03	0.01	0.01	-0.02	0.02	0.03	0.03	-0.03	-0.03	-0.06	-0.05	-0.01	-0.07	-0.02	-0.02	-0.05	-0.01	-0.05	-0.07
-0.06	0.03	0.03	0.21	-0.05	-0.05	0.03	-0.03	-0.02	-0.07	0.05	0.01	-0.07	0.01	0.05	0.03	-0.02	-0.03	-0.02	-0.01	-0.01	-0.05	0.02	0.02
-0.03	0.03	0.02	-0.05	0.21	0.05	0.03	-0.06	-0.03	0.05	-0.07	-0.01	-0.01	-0.02	0.01	-0.02	0.03	0.03	0.01	-0.07	-0.05	0.02	-0.05	-0.02
-0.03	0.02	0.03	-0.05	0.05	0.21	-0.02	0.03	0.03	-0.01	0.01	-0.02	0.05	-0.01	-0.07	0.03	-0.03	-0.06	0.01	-0.05	-0.07	0.02	-0.02	-0.05
-0.07	-0.05	0.01	0.03	0.03	-0.02	0.21	0.05	-0.05	-0.06	-0.03	0.03	-0.05	-0.02	0.02	-0.02	0.01	-0.01	0.03	0.02	-0.03	-0.07	-0.01	0.05
-0.05	-0.07	0.01	-0.03	-0.06	0.03	0.05	0.21	-0.05	0.03	0.03	-0.02	-0.02	-0.05	0.02	-0.01	-0.07	0.05	0.02	0.03	-0.03	0.01	-0.02	-0.01
-0.01	-0.01	-0.02	-0.02	-0.03	0.03	-0.05	-0.05	0.21	-0.03	-0.02	0.03	0.02	0.02	-0.05	0.01	0.05	-0.07	0.03	0.03	-0.06	0.05	0.01	-0.07
0.03	-0.03	0.02	-0.07	0.05	-0.01	-0.06	0.03	-0.03	0.21	-0.05	0.05	-0.02	-0.01	0.01	-0.05	0.02	-0.02	-0.07	0.01	-0.05	0.03	-0.02	0.03
0.03	-0.06	0.03	0.05	-0.07	0.01	-0.03	0.03	-0.02	-0.05	0.21	-0.05	0.01	-0.07	0.05	0.02	-0.05	0.02	-0.01	-0.02	-0.01	-0.02	0.03	-0.03
0.02	-0.03	0.03	0.01	-0.01	-0.02	0.03	-0.02	0.03	0.05	-0.05	0.21	-0.01	0.05	-0.07	-0.02	0.02	-0.05	-0.05	0.01	-0.07	-0.03	0.03	-0.06
0.03	0.02	-0.03	-0.07	-0.01	0.05	-0.05	-0.02	0.02	-0.02	0.01	-0.01	0.21	0.05	-0.05	-0.06	-0.03	0.03	-0.07	-0.05	0.01	0.03	0.03	-0.02
0.02	0.03	-0.03	0.01	-0.02	-0.01	-0.02	-0.05	0.02	-0.01	-0.07	0.05	0.05	0.21	-0.05	0.03	0.03	-0.02	-0.05	-0.07	0.01	-0.03	-0.06	0.03
0.03	0.03	-0.06	0.05	0.01	-0.07	0.02	0.02	-0.05	0.01	0.05	-0.07	-0.05	-0.05	0.21	-0.03	-0.02	0.03	-0.01	-0.01	-0.02	-0.02	-0.03	0.03
-0.07	0.01	-0.05	0.03	-0.02	0.03	-0.02	-0.01	0.01	-0.05	0.02	-0.02	-0.06	0.03	-0.03	0.21	-0.05	0.05	0.03	-0.03	0.02	-0.07	0.05	-0.01
-0.01	-0.02	-0.01	-0.02	0.03	-0.03	0.01	-0.07	0.05	0.02	-0.05	0.02	-0.03	0.03	-0.02	-0.05	0.21	-0.05	0.03	-0.06	0.03	0.05	-0.07	0.01
-0.05	0.01	-0.07	-0.03	0.03	-0.06	-0.01	0.05	-0.07	-0.02	0.02	-0.05	0.03	-0.02	0.03	0.05	-0.05	0.21	0.02	-0.03	0.03	0.01	-0.01	-0.02
-0.05	-0.02	-0.02	-0.02	0.01	0.01	0.03	0.02	0.03	-0.07	-0.01	-0.05	-0.07	-0.05	-0.01	0.03	0.03	0.02	0.21	0.05	0.05	-0.06	-0.03	-0.03
-0.02	-0.05	-0.02	-0.01	-0.07	-0.05	0.02	0.03	0.03	0.01	-0.02	0.01	-0.05	-0.07	-0.01	-0.03	-0.06	-0.03	0.05	0.21	0.05	0.03	0.03	0.02
-0.02	-0.02	-0.05	-0.01	-0.05	-0.07	-0.03	-0.03	-0.06	-0.05	-0.01	-0.07	0.01	0.01	-0.02	0.02	0.03	0.03	0.05	0.05	0.21	0.03	0.02	0.03
-0.02	-0.01	-0.01	-0.05	0.02	0.02	-0.07	0.01	0.05	0.03	-0.02	-0.03	0.03	-0.03	-0.02	-0.07	0.05	0.01	-0.06	0.03	0.03	0.21	-0.05	-0.05
0.01	-0.07	-0.05	0.02	-0.05	-0.02	-0.01	-0.02	0.01	-0.02	0.03	0.03	0.03	-0.06	-0.03	0.05	-0.07	-0.01	-0.03	0.03	0.02	-0.05	0.21	0.05
0.01	-0.05	-0.07	0.02	-0.02	-0.05	0.05	-0.01	-0.07	0.03	-0.03	-0.06	-0.02	0.03	0.03	-0.01	0.01	-0.02	-0.03	0.02	0.03	-0.05	0.05	0.21

Appendix B: Tables of Numerical Results

Table 21: VMS-Constrained Hardened State Wall				
Constraint: 10 MPa				
Run	Load (KN/m)	Result	Volume	Max VMS (MPa)
1	-81.3	complete	56.36%	9.97
2	-122.0	complete	79.16%	9.99
3	-203.3	constraint unreachable	> 95.00%	12.80
4	-406.7	constraint unreachable	> 95.00%	25.65
Constraint: 20 MPa				
Run	Load (KN/m)	Result	Volume	Max VMS (MPa)
5	-81.3	minimum volume reached	< 15.00%	12.79
6	-122.0	complete	45.67%	19.85
7	-203.3	complete	65.91%	19.95
8	-406.7	constraint unreachable	> 95.00%	25.61
Constraint: 30 MPa				
Run	Load (KN/m)	Result	Volume	Max VMS (MPa)
9	-81.3	minimum volume reached	< 15.00%	14.03
10	-122.0	minimum volume reached	< 15.00%	19.15
11	-203.3	complete	46.53%	29.80
12	-406.7	complete	79.54%	29.98
Constraint: 50 MPa				
Run	Load (KN/m)	Result	Volume	Max VMS (MPa)
13	-81.3	minimum volume reached	< 15.00%	28.35
14	-122.0	minimum volume reached	< 15.00%	35.42
15	-203.3	minimum volume reached	< 15.00%	31.91
16	-406.7	complete	56.37%	49.82

Table 22: VMS-Constrained Hardened State Column				
Constraint: 20 MPa				
Run	Load (MPa)	Result	Volume	Max VMS (MPa)
1	-1.21	complete	6.13%	19.16
2	-2.42	complete	26.33%	19.73
3	-3.63	max iterations	32.02%	19.66
4	-6.05	complete	43.08%	19.68
5	-12.10	constraint unreachable	> 95.00%	25.47
Constraint: 30 MPa				
Run	Load (MPa)	Result	Volume	Max VMS (MPa)
6	-1.21	minimum volume reached	< 5.00%	19.04
7	-2.42	complete	10.94%	28.80
8	-3.63	complete	26.17%	29.58
9	-6.05	max iterations	31.71%	29.39
10	-12.10	complete	58.13%	29.72
Constraint: 40 MPa				
Run	Load (MPa)	Result	Volume	Max VMS (MPa)
11	-1.21	minimum volume reached	< 5.00%	17.56
12	-2.42	complete	6.34%	38.20
13	-3.63	complete	13.02%	38.61
14	-6.05	complete	27.80%	39.29
15	-12.10	complete	43.06%	39.38
Constraint: 50 MPa				
Run	Load (MPa)	Result	Volume	Max VMS (MPa)
16	-1.21	minimum volume reached	< 5.00%	22.43
17	-2.42	minimum volume reached	< 5.00%	34.08
18	-3.63	complete	9.51%	48.28
19	-6.05	complete	26.16%	49.29
20	-12.10	complete	39.38%	49.04

Table 23: Shear Stress-Constrained Hardened State Wall				
Constraint: 5 MPa				
Run	Load (KN/m)	Result	Volume	Max Shear (MPa)
1	-40.7	max iterations	46.46%	4.88
2	-81.3	max iterations	57.69%	4.99
3	-122.0	max iterations	74.30%	5.05
4	-203.3	complete	73.94%	4.99
5	-406.7	max iterations	85.16%	5.01
Constraint: 10 MPa				
Run	Load (KN/m)	Result	Volume	Max Shear (MPa)
6	-40.7	minimum volume reached	< 15.00%	5.89
7	-81.3	max iterations	41.42%	9.92
8	-122.0	max iterations	47.27%	9.97
9	-203.3	max iterations	56.48%	9.40
10	-406.7	max iterations	79.24%	9.86
Constraint: 15 MPa				
Run	Load (KN/m)	Result	Volume	Max Shear (MPa)
11	-40.7	minimum volume reached	< 15.00%	10.06
12	-81.3	max iterations	34.49%	14.86
13	-122.0	max iterations	44.76%	13.95
14	-203.3	max iterations	57.39%	14.67
15	-406.7	max iterations	65.69%	16.36
Constraint: 20 MPa				
Run	Load (KN/m)	Result	Volume	Max Shear (MPa)
16	-40.7	minimum volume reached	< 15.00%	3.90
17	-81.3	minimum volume reached	< 15.00%	6.45
18	-122.0	max iterations	32.75%	19.66
19	-203.3	complete	49.35%	17.69
20	-406.7	max iterations	66.55%	19.79

Table 24: Shear Stress-Constrained Hardened State Column				
Constraint: 10 MPa				
Run	Load (MPa)	Result	Volume	Max Shear (MPa)
1	-8.47	max iterations	58.54%	13.05
2	-12.10	max iterations	69.20%	11.53
3	-18.15	constraint unreachable	> 95.00%	104.67

Table 25: Shear Stress-Constrained Fresh State Wall				
Constraint: 1 KPa				
Run	Load (KN/m)	Result	Volume	Max Shear (KPa)
1	0.047	complete	48%	1.00
2	0.0610	complete	68%	1.00
3	0.0813	complete	70%	1.00
4	0.1017	complete	72%	0.99
Constraint: 1.5 KPa				
Run	Load (KN/m)	Result	Volume	Max Shear (KPa)
5	0.047	complete	23%	1.49
6	0.0610	minimum volume reached	15%	1.47
7	0.0813	complete	61%	1.50
8	0.1017	complete	67%	1.49
Constraint: 2 KPa				
Run	Load (KN/m)	Result	Volume	Max Shear (KPa)
9	0.047	minimum volume reached	17%	0.78
10	0.0610	minimum volume reached	46%	1.28
11	0.0813	complete	55%	1.99
12	0.1017	complete	60%	1.99
Constraint: 2.5 KPa				
Run	Load (KN/m)	Result	Volume	Max Shear (KPa)
13	0.047	minimum volume reached	18%	1.00
14	0.0610	minimum volume reached	46%	1.29
15	0.0813	complete	52%	2.49
16	0.1017	complete	63%	2.50

Appendix C: Proof of Concept Study Full Matlab Code

```

1  % Proportional Topology Optimization stress (PTOs) - Cant Beam - (2014)
2  % Modified by Seth Engel
3  % CURRENT STATE: Cantilever left edge, point load, shear stress constraint
4  clc;clear;
5  %% Inputs
6  E0 = 30000;           % Young's Modulus of solid (MPa)
7  Emin = 1e-9;         % Young's Modulus of void (MPa)
8  L = 1;               % Length of geometry (m)
9  H = 0.5;             % Height of geometry (m)
10 size_el = 0.01;      % Length of element side (m)
11 lv = 30000;          % Load value (N)
12 ld = 3;              % Number of elements with loading/displacements
13 nelx = L/size_el;    % Number of elements in x direction
14 nely = H/size_el;    % Number of elements in y direction
15 nu = 0.18;           % Poisson's ratio
16 penal = 3;           % Penalization factor
17 q = 2;               % Proportion exponent
18 rmin = 1.5;          % Filter radius
19 yield_shear = 1.1;   % Yield shear (MPa)
20 xlim = [0,1];        % Lower and upper bounds on element density
21 den = 2300;          % Material density
22 V = size_el^3;       % Element volume (m^3)
23 g = -9.81;           % Gravitational acceleration (kg/m^2)
24 MPa_Factor = 10^-6; % MPa adjustment factor
25 abs_shear_mat = [];  % Initiates absolute shear matrix
26 prev_vol_fract = 1;  % Sets initial volume fraction to 1
27 prev_max_shear = 1000; % Initiates maximum shear value
28 SW = true;           % Includes self-weight if true
29
30 %% Setup Finite Element Analysis
31 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
32 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
33 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
34 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
35 KE = 1/(1-nu^2)/24*( [A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
36 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
37 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
38 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
39 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
40 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
41
42 %% Define Loads and Supports
43 iF = 2*(nelx*(nely+1)+nely/2+1+(-(ld-1)/2:(ld-1)/2));
44 jF = ones(1,ld);
45 sF = -lv/ld*ones(ld,1);
46 F_applied = sparse(iF,jF,sF,2*(nely+1)*(nelx+1),1)*MPa_Factor;
47
48 %% Define Displacement and DOF Sets
49 U = zeros(2*(nely+1)*(nelx+1),1);
50 fixeddofs = 1:2*(nely+1);
51 alldofs = 1:2*(nely+1)*(nelx+1);
52 freedofs = setdiff(alldofs,fixeddofs);
53
54 %% Setup Stress Analysis
55 B = (1/2/size_el)*[-1 0 1 0 1 0 -1 0; 0 -1 0 -1 0 1 0 1; -1 -1 -1 1 1 1 1 -1];
56 DE = (1/(1-nu^2))*[1 nu 0; nu 1 0; 0 0 (1-nu)/2];
57
58 %% Setup Filter
59 iW = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
60 jW = ones(size(iW));
61 sW = zeros(size(iW));
62 k = 0;
63 for il = 1:nelx

```

```

64     for j1 = 1:nely
65         e1 = (i1-1)*nely+j1;
66         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
67             for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
68                 e2 = (i2-1)*nely+j2;
69                 k = k+1;
70                 iW(k) = e1;
71                 jW(k) = e2;
72                 sW(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
73             end
74         end
75     end
76 end
77 w = sparse(iW,jW,sW);
78 W = bsxfun(@rdivide,w,sum(w,2));
79
80 %% Initialize Iteration
81 x = repmat(0.9,nely,nelx);
82 loop = 0;
83
84 %% Run Iteration
85 while loop<500
86     %% Define Self-Weight Loading
87     loop = loop+1;
88     if SW == true
89         F_sw = sparse((nelx+1)*(nely+1)*2,1);
90         for i = 1:nelx*nely
91             for j = 2:2:8
92                 DOF = edofMat(i,j);
93                 F_sw(DOF) = F_sw(DOF)+(.25*den*V*g)*x(i)*MPa_Factor;
94             end
95         end
96         F_total = F_sw+F_applied;
97     else
98         F_total = F_applied;
99     end
100
101     %% Finite Element Analysis
102     E = Emin+x(:)'.^penal*(E0-Emin);
103     sK = reshape(KE(:)*E,64*nelx*nely,1);
104     K = sparse(iK,jK,sK); K = (K+K')/2;
105     U(freedofs) = K(freedofs,freedofs)\F_total(freedofs);
106
107     %% Stress Calculation
108     s = (U(edofMat)*(DE*B)').*repmat(E',1,3);
109     sxmat = reshape(s(:,1),nely,nelx); % SigmaX matrix
110     symat = reshape(s(:,2),nely,nelx); % SigmaY matrix
111     shear_mat = reshape(s(:,3),nely,nelx); % SigmaXY matrix
112     abs_shear_mat = abs(shear_mat); % Absolute value shear stress matrix
113     cur_max_shear = max(abs_shear_mat(:)); % Defines current maximum shear stress value
114     vms = reshape(sqrt(sum(s.^2,2)-s(:,1).*s(:,2)+2.*s(:,3).^2),nely,nelx);
115
116     %% Compliance Calculation
117     ce = E'.*sum((U(edofMat)*KE).*U(edofMat),2);
118     C = reshape(ce,nely,nelx);
119
120     %% Optimization Algorithm
121     if (cur_max_shear > yield_shear)
122         if prev_max_shear < yield_shear
123             disp('Optimization Complete < Yield Shear'); break; end
124         if prev_max_shear > cur_max_shear
125             xTarget = sum(x(:))-0.005*numel(x);
126         else
127             xTarget = sum(x(:))+0.005*numel(x);
128         end
129     else
130         xTarget = sum(x(:))-0.005*numel(x);

```

```

131     end
132
133     xRemaining = xTarget;
134     yield_shear_proportion = abs_shear_mat.^q/sum(sum(abs_shear_mat.^q));
135     yield_shear_proportion(:) = W*yield_shear_proportion(:);
136     x(:) = 0;
137     %%
138     while (xRemaining > 0.1)
139         xDist = xRemaining.*yield_shear_proportion;
140         x(:) = x(:)+xDist(:);
141         x = max(min(x,xlim(2)),xlim(1));
142         xRemaining = xTarget-sum(x(:));
143     end
144     cur_vol_fract = sum(x(:))/(nelx*nely);
145     %% Results
146     % Print Results
147     fprintf('It:%5i  MaxShearStress: %5.3f  VolumeFraction: %5.3f\n',...
148         loop,cur_max_shear,cur_vol_fract);
149     if cur_vol_fract < 0.01
150         if cur_max_shear < yield_shear
151             disp('Minimum Volume Reached');
152             break;
153         else
154             disp('Empty Volume');
155         end
156     end
157
158     % Plot Results
159     colormap(flipud(gray));
160     subplot(2,1,1); imagesc(x); axis equal off; text(2,-2,'Material');
161     subplot(2,1,2); imagesc(abs(shear_mat)); axis equal off; text(2,-2,'Shear Stress');
162     drawnow;
163     %subplot(4,1,3); imagesc(sxmat); axis equal off; text(2,-2,'X-axis Stress'); drawnow;
164     %subplot(4,1,4); imagesc(symat); axis equal off; text(2,-2,'Y-axis Stress'); drawnow;
165     if x == ones(nely,nelx); disp('Full Volume'); break; end
166
167     %% Check Stop Criteria
168     if cur_vol_fract > prev_vol_fract && loop > 50
169         disp('Max Yield Shear Stress Cannot Be Reached'); break; end
170     prev_vol_fract = cur_vol_fract;
171     prev_max_shear = cur_max_shear;
172 end
173 %
174 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
175 % Copyright (C) 2014 University of Pittsburgh. All rights reserved.
176 % Modified by Seth Engel, University of Virginia (2019)
177
178 % Any person who obtained a copy of this software can (in part or whole) copy,
179 % modify, merge, publish, and distribute the software on condition of retaining
180 % this license with the software. The user is allowed to utilize the software
181 % for all purposes but commercial. Also, appropriate credit must be provided.
182 %
183 % The software is provided "as is", without warranty of any kind, express or
184 % implied, including but not limited to the warranties of merchantability,
185 % fitness for a particular purpose and noninfringement. In no event shall the
186 % authors or copyright holders be liable for any claim, damage or other
187 % liability, whether in an action of contract, tort or otherwise, arising from,
188 % out of or in connection with the software or the use or other dealing in the
189 % software.
190 %
191 % The software is coded by Emre Biyikli (biyikli.emre@gmail.com) and Albert C.
192 % To (albertto@pitt.edu). The software can be downloaded from www.ptomethod.org.

```

Appendix D: SSTO Full Matlab Code

```
1  %%% Shear Stress-constrained Topology Optimization (SSTO)
2  %%% Developed by Seth Engel (2020)
3  %%% University of Virginia Department of Engineering Systems and Environment
4
5  %%% Current problem setup: SS-constrained fresh state wall
6  %%%     Bottom-fixed wall geometry subject to self-weight loading and
7  %%%     distributed loading across the top surface.
8
9
10 % Constants, Definitions, and Initializations
11 clc; clear; % Clears contents
12 vid_set = false; % Initializes video recording if true
13
14 % Domain Geometry
15 nelx = 60; % # of elements in x
16 nely = 20; % # of elements in y (vertical direction)
17 nelz = 3; % # of elements in z
18 L = 0.01; % element side length (m)
19 volfrac = 0.9; % Initial element density
20
21 % Boundary Conditions
22 cantilever = false; % Sets cantilever boundary condition
23 bottom_edge_fixed = true; % Sets bottom-fixed boundary condition
24 top_edge_fixed = false; % Sets top-fixed boundary condition
25
26 % Loading
27 Applied = true; % Initializes applied loading (if true)
28 Sw = true; % Initializes self-weight loading (if true)
29 top_distributed = true; % Initiates distributed load along top of structure
30 bottom_pl = false; % Sets location of point load (bottom right edge)
31 mid_pl = false; % Sets location of point load (mid right edge)
32 top_mid_pl = false; % Sets location of point load (top mid-span)
33 Pu = -30000; % Applied point load (N)
34 Distributed = [0.1]; % Applied distributed load on each node (N)
35
36 % Optimization Parameters
37 k_penal_vect = [2]; % Stiffness Penalization factor (SIMP)
38 penalization = [8]; % Stress Penalization factor (Holmberg)
39 relax = [0.01]; % Minimum density for constraint function
40 perturb = [10^-6, 10^-7]; % Sensitivity FDM perturbation
41 tol_x = 0.01; % Optimization tolerance
42 rmin = 1.5; % Filter radius
43 maxloop = 500; % Maximum number of iterations
44 minvol = 0.15; % Minimum volume fraction termination criteria
45 maxvol = 0.95; % Maximum volume fraction termination criteria
46
47 % Material Properties
48 E = 0.06; % Young's Modulus of solid material (MPa)
49 Emin = 1e-9; % Young's modulus of voided material (MPa)
50 nu = 0.3; % Poisson's Ratio
51 den = 2300; % material density (kg/m^3)
52
53 % Constraint
54 shear = true; % Initiates yield shear stress constraint
55 ys_vect = [0.002]; % Shear Stress constraint (MPa)
56 VMS = false; % Initiates VMS constraint (MPa)
57 VMSy = 10; % Von Mises Stress constraint (MPa)
58
59 % Constants
60 g = -9.81; % Gravitational acceleration (m/s^2)
61 V = L^3; % Element volume (m^3)
62 MPa_Factor = 10^-6; % Mega-Pascal Conversion Factor
63 nele = nelx*nely*nelz; % Total number of elements
```

```

64 nnode = (nelx+1)*(nely+1)*(nelz+1); % Total number of nodes
65 ndof = nnode*3; % Total number of degrees of freedom
66 run = 1; % Initiates first run
67 GQ_matrix = [-1/sqrt(3) -1/sqrt(3) -1/sqrt(3); % 2x2x2 Gauss Quadrature Coordinates
68 -1/sqrt(3) -1/sqrt(3) 1/sqrt(3);
69 -1/sqrt(3) 1/sqrt(3) -1/sqrt(3);
70 -1/sqrt(3) 1/sqrt(3) 1/sqrt(3);
71 1/sqrt(3) -1/sqrt(3) -1/sqrt(3);
72 1/sqrt(3) -1/sqrt(3) 1/sqrt(3);
73 1/sqrt(3) 1/sqrt(3) -1/sqrt(3);
74 1/sqrt(3) 1/sqrt(3) 1/sqrt(3)];
75
76 C = (1/((nu+1)*(1-2*nu)))*[1-nu nu nu 0 0 0; % Formulates Elasticity Tensor (C)
77 nu 1-nu nu 0 0 0;
78 nu nu 1-nu 0 0 0;
79 0 0 0 (1-2*nu)/2 0 0;
80 0 0 0 0 (1-2*nu)/2 0;
81 0 0 0 0 0 (1-2*nu)/2];
82
83 % Formulates Nodal Data
84 [nodes,NIDgrid] = nodal_data(nelx,nely,nelz,L);
85
86 % Formulates Element Data
87 [edofmat,element,EIDgrid] = element_data(nelx,nely,nelz,nele,nodes,L);
88
89 % Boundary Conditions
90 [freeDOFs,fixedDOFs,element] =
91 BCs(element,nelx,nely,nelz,ndof,nodes,cantilever,bottom_edge_fixed,top_edge_fixed,EID
92 grid);
93
94 % Develop Element Stiffness Matrix
95 [element] = stiffness(nele,element,GQ_matrix,C);
96
97 % Prepare Filtering
98 [H,Hs] = filtering(nele,rmin,nelx,nely,nelz);
99
100 % Parameterization
101 for load = 1:length(Distributed) % Distributed loading parameterization
102 Wu = Distributed(load);
103 % Applied Loading
104 [F_applied,LoadEID,element] =
105 applied_load(nelx,nely,nelz,nodes,Pu,nnode,Applied,MPa_Factor,element,bottom_pl,mid_p
106 l,top_mid_pl,top_distributed,Wu);
107 for v = 1:length(perturb) % Perturbation factor parameterization
108 delta = perturb(v);
109 for e = 1:length(relax) % Relaxation factor parameterization
110 eps = relax(e);
111 for stiff = 1:length(k_penal_vect) % Stiffness penalization factor
112 parameterization
113 k_penal = k_penal_vect(stiff);
114 for s = 1:length(penalization) % Stress penalization factor
115 parameterization
116 s_penal = penalization(s);
117 for con = 1:length(ys_vect) % Shear stress constraint
118 parameterization
119 ys = ys_vect(con);
120 % Start clock for each run
121 tic
122
123 % Initialize Iteration
124 x = repmat(volfrac,[nely,nelx,nelz]); % Defines initial
125 value of design variables
126 xPhys = x; % Defines initial
127 value of filtered design variables
128 change = 1; % Initiates
129 change variable
130 iter = 1; % Initial

```

```

iteration number
121 of active elements active = setdiff(EIDgrid(:),LoadEID(:)); % Defines the set
122 general constraints. m = 1; % The number of
123 design variables x_j. n = nele; % The number of
124 with the lower bounds for the variables x_j. xmin = zeros(n,1); % Column vector
125 with the upper bounds for the variables x_j. xmax = ones(n,1); % Column vector
126 iteration ago (provided that iter>1). xold1 = x(:); % xval, one
127 iterations ago (provided that iter>2). xold2 = x(:); % xval, two
128 with the lower asymptotes from the previous iteration (provided that iter>1). low = ones(n,1); % Column vector
129 with the upper asymptotes from the previous iteration (provided that iter>1). upp = ones(n,1); % Column vector
130 a_0 in the term a_0z. a0 = 1; % The constants
131 with the constants a_i in the terms a_iz. a = zeros(m,1); % Column vector
132 with the constants c_i in the terms c_iy_i. c_const = 10000*ones(m,1); % Column vector
133 with the constants d_i in the terms 0.5d_i(y_i)^2. d = zeros(m,1); % Column vector
134
135 % Initiates data storage
136 maxshear_store = [];
137 maxshear_sense_store = [];
138 maxVMS_store = [];
139 maxVMS_sense_store = [];
140 fval_store = [];
141 change_store = [];
142 objective_store = [];
143 % Initiate Video
144 if vid_set == true
145     vid = VideoWriter(char('Run '+string(run)), 'Motion JPEG
146     AVI');
147     vid.FrameRate = 5;
148     vid.Quality = 100;
149     open(vid);
150 end
151 % Iterations
152 while (change > tolx) && (iter < maxloop)
153     % Defines Self-Weight Loading
154     if Sw == true
155         [F,F_Sw_partial,element,nodes] =
156         Self_Weight(nele,element,nodes,den,g,V,F_applied,nnode,MPa_Factor,xPhys);
157     else
158         F = F_applied;
159         F_Sw = zeros(nnode*3,1);
160         F_Sw_partial = zeros(nnode*3,1);
161     end
162     % Finite Element Analysis
163
164     [U,UxMat,UyMat,UzMat,element,nodes,KE,VMS_mat, shearXY_mat] =
165     FEA(nelx,nely,nelz,element,nodes,Emin,E,k_penal,s_penal,nele,edofmat,freeDOFs,C,ndof,
166     F,F_Sw_partial,nnode,xPhys,Sw,delta);
167     % Sensitivity Analysis
168     [element,VMS_sensitivity_mat,vol,d_vol,pshearXY_mat] =
169     sensitivity(nelx,nely,nelz,xPhys,element,H,Hs,C,nele,k_penal,s_penal,E,Emin,delta);
170     % Define MMA functions

```

```

168         xval = x(:);           % Design variable values x
169         f0val = vol/nele;      % Objective function value at x
170         df0dx = d_vol(:);     % Derivative of objective function at
171         x
172         if shear == true
173             sense_scalar = max(abs(pshearXY_mat(:))); %
174             Sensitivity analysis normalization
175             fval = (max(abs(shearXY_mat(active)))/(ys - 1)/eps;
176             % Constraint function value at x
177             dfdx = -abs(pshearXY_mat(:)/sense_scalar)'; % Sets
178             constraint sensitivity at x
179             % Data storage
180             maxshear_store =
181             [maxshear_store;max(abs(shearXY_mat(active)))];
182             maxshear_sense_store =
183             [maxshear_sense_store;max(abs(pshearXY_mat(:)))];
184             fval_store = [fval_store;fval];
185             change_store = [change_store;change];
186             objective_store = [objective_store;f0val];
187         end
188         if VMS == true
189             sense_scalar = max(abs(VMS_sensitivity_mat(:))); %
190             Sensitivity analysis normalization
191             fval = (max(VMS_mat(active))/(VMSy - 1)/eps; %
192             VMS formulation from Holmberg
193             dfdx = (VMS_sensitivity_mat(:)/sense_scalar)'; %
194             Sets constraint sensitivity at x
195             % Data storage
196             maxVMS_store =
197             [maxVMS_store;max(abs(VMS_mat(active)))];
198             maxVMS_sense_store =
199             [maxVMS_sense_store;max(abs(VMS_sensitivity_mat(:)))];
200         end
201
202         % Min/Max volume fraction termination conditions
203         if f0val < minvol
204             disp("Minimum Volume Reached")
205             terminate = "minimum volume reached";
206             break;
207         end
208         if f0val > maxvol
209             disp("Constraint Unreachable")
210             terminate = "constraint unreachable";
211             break;
212         end
213
214         % Print results to command window
215         if shear == true
216             fprintf(' It.:%5i Obj.:%9.5f Shear.:%7.5f
217             Shear_Sense.:%7.3f MaxFval.:%7.3f
218             ch.:%7.3f\n', iter, f0val, max(abs(shearXY_mat(active))), max(abs(dfdx(:))), fval, change);
219         end
220         if VMS == true
221             fprintf(' It.:%5i Obj.:%11.4f VMS.:%7.5f
222             VMS_Sense.:%7.3f MaxFval.:%7.3f
223             ch.:%7.3f\n', iter, f0val, max(VMS_mat(active)), max(abs(dfdx(:))), fval, change);
224         end
225
226         % Plot densities
227         clf; display_3D(xPhys);
228         iter = iter+1;
229
230         % Write to video
231         if vid_set == true
232             writeVideo(vid,getframe);
233         end
234     end

```

```

220         % Defines results for complete/incomplete optimization
221         if f0val <= maxvol && f0val >= minvol && iter < maxloop
222             terminate = "complete";
223         end
224         if iter == maxloop
225             terminate = "max iterations";
226         end
227
228         % Call MMA for optimization
229         [xmma, ymma, zmma, lam, xsi, eta, mu, zet, s, low, upp] = ...
230             mmasub(m, n, iter, xval, xmin, xmax, xold1, xold2, ...
231                 f0val, df0dx, fval, dfdx, low, upp, a0, a, c_const, d);
232
233         % Update MMA Variables
234         xnew = reshape(xmma, nely, nelx, nelz);
235         for EID = 1:nele
236             if element(EID).fixed == 1 % Hold material at fixed
237                 elements
238                     xnew(EID) = 1;
239                     xPhys(EID) = 1;
240                 end
241             end
242             xPhys(:) = (H*xnew(:))./Hs; % Apply density filtering
243             for EID = 1:nele
244                 if xPhys(EID) < 0.1 % Define minimum element density
245                     of 0.1
246                         xnew(EID) = 0;
247                         xPhys(EID) = 0;
248                     end
249                     if element(EID).fixed == 1 % Hold material at fixed
250                         elements
251                             xnew(EID) = 1;
252                             xPhys(EID) = 1;
253                         end
254                     end
255                     xold2 = xold1(:);
256                     xold1 = x(:);
257                     change = max(abs(xnew(:)-x(:))); % Calculate tolerance
258                     x = xnew;
259                     new_volfrac = sum(xPhys(:))/nele; % Calculate new
260                     objective function
261
262                     % Continuation condition
263                     if fval > 0
264                         change = 1;
265                     end
266                 end
267             end
268             if vid_set == true
269                 close(vid);
270             end
271             disp('Run '+string(run)+' finished')
272             clf; display_3D(xPhys);
273             % Save/Store Final Optimized Design Results
274             figname = 'Run '+string(run)+'.fig';
275             savefig(string(figname));
276             fprintf(' It.:%5i Obj.:%9.5f Shear.:%7.3f Shear_Sense.:%7.3f
277                 MaxFval.:%7.3f ch.:%7.3f
278                 \n', iter, f0val, max(abs(shearXY_mat(active))), max(abs(pshearXY_mat(active))), fval, chan
279                 ge);
280
281             Bigdata(run).time = toc;
282             Bigdata(run).Wu = Wu;
283             Bigdata(run).eps = eps;
284             Bigdata(run).k_penal = k_penal;
285             Bigdata(run).s_penal = s_penal;
286             Bigdata(run).perturb = delta;
287             Bigdata(run).end = terminate;
288             Bigdata(run).ys = ys;

```

```

280         Bigdata(run).obj_final = f0val;
281         Bigdata(run).maxshear_final = max(abs(shearXY_mat(:)));
282         Bigdata(run).maxshear = maxshear_store;
283         Bigdata(run).maxshearsense = maxshear_sense_store;
284         Bigdata(run).VMSy = VMSy;
285         Bigdata(run).maxVMS_final = max(abs(VMS_mat(:)));
286         Bigdata(run).maxVMS = maxVMS_store;
287         Bigdata(run).maxVMS_sense = maxVMS_sense_store;
288         Bigdata(run).it = iter - 1;
289         Bigdata(run).fval_final = fval;
290         Bigdata(run).ch_final = change;
291         Bigdata(run).fval = fval_store;
292         Bigdata(run).obj = objective_store;
293         Bigdata(run).ch = change_store;
294         Bigdata(run).xPhys_final = xPhys;
295         Bigdata(run).shear_final = shearXY_mat;
296         Bigdata(run).shearsense_final = pshearXY_mat;
297         Bigdata(run).VMS_final = VMS_mat;
298         Bigdata(run).VMSsense_final = VMS_sensitivity_mat;
299         Bigdata(run).init_vol = volfrac;
300         save('Bigdata.mat','Bigdata');
301         run = run + 1;
302     end
303 end
304 end
305 end
306 end
307 end
308 disp("All Runs Finished")
309
310 %% Applied Loading Condition
311 function [F_applied,LoadEID,element] =
312     applied_load(nelx,nely,nelz,nodes,Pu,nnode,Applied,MPa_Factor,element,bottom,mid,top_
313     mid,top_distributed,Wu)
314     F_applied = zeros(nnode*3,1); % Applied load vector
315     LoadEID = [];
316     if Applied == true
317         for z = 1:nelz
318             if bottom == true % Point load bottom right edge
319                 EID = nelx*nely*z;
320                 element(EID).fixed = 1;
321                 LoadEID = [EID,EID-1];
322                 Load_NID1 = element(EID).nodeIDs(2);
323                 Load_NID2 = element(EID).nodeIDs(6);
324                 F_applied(nodes(Load_NID1).dof(2)) = Pu*MPa_Factor/(nelz+1);
325                 F_applied(nodes(Load_NID2).dof(2)) = Pu*MPa_Factor/(nelz+1);
326             end
327             if mid == true %Point right edge mid-height
328                 EID = nelx*nely*z-(nely/2);
329                 element(EID).fixed = 1;
330                 LoadEID = [EID,EID+1];
331                 Load_NID1 = element(EID).nodeIDs(2);
332                 Load_NID2 = element(EID).nodeIDs(6);
333                 F_applied(nodes(Load_NID1).dof(2)) = Pu*MPa_Factor/(nelz+1);
334                 F_applied(nodes(Load_NID2).dof(2)) = Pu*MPa_Factor/(nelz+1);
335             end
336             if top_mid == true % Point load top face midspan
337                 EID = (nely*(nelx/2)+1)*z;
338                 element(EID).fixed = 1;
339                 LoadEID = [EID];
340                 Load_NID1 = element(EID).nodeIDs(4);
341                 Load_NID2 = element(EID).nodeIDs(8);
342                 Load_NID3 = element(EID).nodeIDs(3);
343                 Load_NID4 = element(EID).nodeIDs(7);
344                 F_applied(nodes(Load_NID1).dof(2)) = Pu*MPa_Factor/(nelz+1)/2;
345                 F_applied(nodes(Load_NID2).dof(2)) = Pu*MPa_Factor/(nelz+1)/2;
346                 F_applied(nodes(Load_NID3).dof(2)) = Pu*MPa_Factor/(nelz+1)/2;
347                 F_applied(nodes(Load_NID4).dof(2)) = Pu*MPa_Factor/(nelz+1)/2;

```

```

345         F_applied(nodes(Load_NID4).dof(2)) = Pu*MPa_Factor/(nelz+1)/2;
346     end
347     if top_distributed == true % Distributed load top surface
348         EIDs = [];
349         for x = 1:nelx
350             EID = nely*(x-1)+nelx*nely*(z-1)+1;
351             element(EID).fixed = 1;
352             EIDs = [EIDs,EID];
353             Load_NID1 = element(EID).nodeIDs(3);
354             Load_NID2 = element(EID).nodeIDs(4);
355             F_applied(nodes(Load_NID1).dof(2)) = Wu*MPa_Factor;
356             F_applied(nodes(Load_NID2).dof(2)) = Wu*MPa_Factor;
357             if z == nelz
358                 Load_NID3 = element(EID).nodeIDs(7);
359                 Load_NID4 = element(EID).nodeIDs(8);
360                 F_applied(nodes(Load_NID3).dof(2)) = Wu*MPa_Factor;
361                 F_applied(nodes(Load_NID4).dof(2)) = Wu*MPa_Factor;
362             end
363         end
364     end
365 end
366 end
367 F_applied = sparse(F_applied);
368 end
369
370 %% Boundary Conditions (fixed DOF's)
function [freeDOFs,fixedDOFs,element] =
BCs(element,nelx,nely,nelz,ndof,nodes,cantilever,bottom_edge_fixed,top_edge_fixed,EID
grid)
371
372     fixedNID = []; % Initializes vector of fixed NIDs
373     fixedDOFs = []; % Initializes vector of fixed DOFs
374
375     new_fixedDOFs = [];
376     if cantilever == true % Fixed left face
377         for z = 1:nelz+1
378             for y = 1:nely+1
379                 new_NID = y+(nelx+1)*(nely+1)*(z-1);
380                 fixedNID = [fixedNID,new_NID]; % Vector of fixed NIDs
381                 fixedDOFs = [fixedDOFs;nodes(new_NID).dof]; % Defines DOFs of fixed
nodes
382             end
383         end
384     end
385     if bottom_edge_fixed == true % Fixed bottom perimeter
386         EIDs = [];
387         EIDouter = EIDgrid(nely,1:nelx,1:nelz);
388         EIDinner = EIDgrid(nely,2:nelx-1,2:nelz-1);
389         EIDs = setdiff(EIDouter,EIDinner);
390         for i = 1:length(EIDs)
391             EID = EIDs(i);
392             element(EID).fixed = 1;
393             fixed_NID1 = element(EID).nodeIDs(1);
394             fixed_NID2 = element(EID).nodeIDs(2);
395             fixed_NID3 = element(EID).nodeIDs(5);
396             fixed_NID4 = element(EID).nodeIDs(6);
397             new_fixedDOFs =
[nodes(fixed_NID1).dof,nodes(fixed_NID2).dof,nodes(fixed_NID3).dof,nodes(fixed_NID4).
dof];
398             fixedDOFs = [fixedDOFs,new_fixedDOFs];
399         end
400     end
401     if top_edge_fixed == true % Fixed top perimeter
402         EIDs = [];
403         EIDouter = EIDgrid(1,1:nelx,1:nelz);
404         EIDinner = EIDgrid(1,2:nelx-1,2:nelz-1);
405         EIDs = setdiff(EIDouter,EIDinner);
406         for i = 1:length(EIDs)

```

```

407         EID = EIDs(i);
408         element(EID).fixed = 1;
409         fixed_NID1 = element(EID).nodeIDs(3);
410         fixed_NID2 = element(EID).nodeIDs(4);
411         fixed_NID3 = element(EID).nodeIDs(7);
412         fixed_NID4 = element(EID).nodeIDs(8);
         new_fixedDOFs =
[ nodes(fixed_NID1).dof, nodes(fixed_NID2).dof, nodes(fixed_NID3).dof, nodes(fixed_NID4).
413 dof];
414         fixedDOFs = [fixedDOFs, new_fixedDOFs];
415     end
416 end
417     freeDOFs = setdiff(1:ndof, fixedDOFs); % column vector of free DOFs
418 end
419
420 %% Formulates Nodal Data Structure
421 function [nodes, NIDgrid] = nodal_data(nelx, nely, nelz, L)
422     for z = 1:nelz+1
423         for x = 1:nelx+1
424             for y = 1:nely+1
425                 NID = (nely+1)*(x-1) + ((nelx+1)*(nely+1)*(z-1)) + y; % Individual node
426                 ID
427                 NIDgrid(y,x,z) = NID;
428                 nodes(NID).Gcoords = [(x-1)*L (nely-y+1)*L (z-1)*L]; % Stores global
429                 coordinates
430                 dof_x = (NID-1)*3+1;
431                 dof_y = (NID-1)*3+2;
432                 dof_z = (NID-1)*3+3;
433                 nodes(NID).dof = [dof_x; dof_y; dof_z]; % Stores nodal degrees of freedom
434             end
435         end
436     end
437
438 %% Formulates Element Data Structure
439 function [edofmat, element, EIDgrid] = element_data(nelx, nely, nelz, nele, nodes, L)
440     % Sets local coordinates of 8 corner nodes
441     n1_Lcoord = [-1 -1 -1];
442     n2_Lcoord = [1 -1 -1];
443     n3_Lcoord = [1 1 -1];
444     n4_Lcoord = [-1 1 -1];
445     n5_Lcoord = [-1 -1 1];
446     n6_Lcoord = [1 -1 1];
447     n7_Lcoord = [1 1 1];
448     n8_Lcoord = [-1 1 1];
449     edofmat = zeros(nele, 24); % Initializes element DOF matrix
450     for z = 1:nelz
451         for x = 1:nelx
452             for y = 1:nely
453                 EID = y+nely*(x-1) + (nely)*(nelx)*(z-1); % Determines element ID
454                 EIDgrid(y,x,z) = EID;
455                 % Relates local node IDs (n_) to global
456                 n4 = y + (nely+1)*(x-1) + (nely+1)*(nelx+1)*(z-1);
457                 n1 = n4+1;
458                 n8 = n4 + (nely+1)*(nelx+1);
459                 n2 = n1+nely+1;
460                 n3 = n4+nely+1;
461                 n5 = n8+1;
462                 n6 = n5+nely+1;
463                 n7 = n8+nely+1;
         element(EID).nodeIDs = [n1 n2 n3 n4 n5 n6 n7 n8]; % Defines node IDs
464     corresponding to each element
         element(EID).Lcoord = [n1_Lcoord; n2_Lcoord; n3_Lcoord; n4_Lcoord;
465     n5_Lcoord; n6_Lcoord; n7_Lcoord; n8_Lcoord]; % Stores local coordinates of all nodes
         associated with EID
         element(EID).CP = [nodes(n4).Gcoords(1)+L/2 nodes(n4).Gcoords(2)-L/2

```

```

nodes(n4).Gcoords(3)+L/2]; % Stores global coordinate of element center point
466     element(EID).Gcoords = [];
467     element(EID).DOF = [];
468     element(EID).fixed = 0;
469     for i = 1:8 % Stores global coordinates of each node of the element
        element(EID).DOF = [element(EID).DOF;
470 nodes(element(EID).nodeIDs(i)).dof];
        element(EID).Gcoords = [element(EID).Gcoords;
471 nodes(element(EID).nodeIDs(i)).Gcoords(:)'];
472     end
473     edofmat(EID,:) = element(EID).DOF;
474     end
475     end
476     end
477 end
478
479 %% Self-Weight Loading
function [F,F_Sw_partial,element,nodes] =
480 Self_Weight(nele,element,nodes,den,g,V,F_applied,nnode,MPa_Factor,xPhys)
481     F = zeros(nnode*3,1); % Combined load vector
482     F_Sw = zeros(nnode*3,1); % Self-Weight load vector
483     F_Sw_partial = zeros(nnode*3,1); % Self-Weight load vector for partial
484     displacement calculation
485     for EID = 1:nele
486         xi = xPhys(EID);
487         for n = 1:8
488             % Assigns self-weight loading
489             F_Sw(nodes(element(EID).nodeIDs(n)).dof(2)) =
490             F_Sw(nodes(element(EID).nodeIDs(n)).dof(2))+(1/8)*xi*den*g*V*MPa_Factor;
491             % Partial derivative of self-weight loading vector
492             F_Sw_partial(nodes(element(EID).nodeIDs(n)).dof(2)) =
493             F_Sw_partial(nodes(element(EID).nodeIDs(n)).dof(2))+(1/8)*den*g*V*MPa_Factor;
494         end
495     end
496     % Stores load vectors with associated elements
497     element(EID).g = [];
498     for EID = 1:nele
499         for n = 1:8
500             element(EID).g =
501             [element(EID).g;full(F_Sw(nodes(element(EID).nodeIDs(n)).dof(2)))];
502             nodes(element(EID).nodeIDs(n)).SW =
503             F_Sw(nodes(element(EID).nodeIDs(n)).dof(2));
504         end
505     end
506     F = sparse(F_applied+F_Sw);
507 end
508
509 %% Develop Stiffness Matrix
function [element] = stiffness(nele,element,GQ_matrix,C)
510     for EID = 1:nele
511         element(EID).K = zeros(24,24); %Initialize stiffness matrix
512         for t = 1:2
513             for point = 1:8 % Local formulations at GQ points or nodes
514                 % Formulate Matrix of Partial Derivatives of Shape Functions
515                 if t == 1 % GQ points
516                     element(EID).BGQ(point).B = []; % Initializes local B matrix for
517                     each GQ point
518                     sail = GQ_matrix(point,1); sai2 = GQ_matrix(point,2); sai3 =
519                     GQ_matrix(point,3);
520                     end
521                 if t == 2
522                     element(EID).BCP = []; % % Initializes local B matrix for
523                     centerpoint
524                     sail = 0; sai2 = 0; sai3 = 0;
525                 end
526             end
527         end
528     end

```

```

521     % Calculates partial derivatives of shape functions
522     pN = zeros(8,3);
523     pN(1,1) = -(1/8)*(sai2-1)*(sai3-1);
524     pN(1,2) = -(1/8)*(sai1-1)*(sai3-1);
525     pN(1,3) = -(1/8)*(sai1-1)*(sai2-1);
526     pN(2,1) = (1/8)*(sai2-1)*(sai3-1);
527     pN(2,2) = (1/8)*(sai1+1)*(sai3-1);
528     pN(2,3) = (1/8)*(sai1+1)*(sai2-1);
529     pN(3,1) = -(1/8)*(sai2+1)*(sai3-1);
530     pN(3,2) = -(1/8)*(sai1+1)*(sai3-1);
531     pN(3,3) = -(1/8)*(sai1+1)*(sai2+1);
532     pN(4,1) = (1/8)*(sai2+1)*(sai3-1);
533     pN(4,2) = (1/8)*(sai1-1)*(sai3-1);
534     pN(4,3) = (1/8)*(sai1-1)*(sai2+1);
535     pN(5,1) = (1/8)*(sai2-1)*(sai3+1);
536     pN(5,2) = (1/8)*(sai1-1)*(sai3+1);
537     pN(5,3) = (1/8)*(sai1-1)*(sai2-1);
538     pN(6,1) = -(1/8)*(sai2-1)*(sai3+1);
539     pN(6,2) = -(1/8)*(sai1+1)*(sai3+1);
540     pN(6,3) = -(1/8)*(sai1+1)*(sai2-1);
541     pN(7,1) = (1/8)*(sai2+1)*(sai3+1);
542     pN(7,2) = (1/8)*(sai1+1)*(sai3+1);
543     pN(7,3) = (1/8)*(sai1+1)*(sai2+1);
544     pN(8,1) = -(1/8)*(sai2+1)*(sai3+1);
545     pN(8,2) = -(1/8)*(sai1-1)*(sai3+1);
546     pN(8,3) = -(1/8)*(sai1-1)*(sai2+1);
547     pN = pN'; % Transposes matrix for multiplication
548
549     % Calculate Jacobian
550     Jacobian = zeros(3,3);
551     for c = 1:3 % # of columns
552         for r = 1:3 % # of rows
553             for i = 1:8 % # of nodes
554                 Jacobian(r,c) = Jacobian(r,c) + pN(r,i)*element(EID).Gcoords(i,c); % Determines Jacobian at first GQ
555                 point
556             end
557         end
558     end
559     if t == 1
560         element(EID).Jac(point).J = Jacobian; % Stores Jacobian at each
561         GQ point
562     end
563     if t == 2
564         element(EID).JacCP = Jacobian; % Stores Jacobian at each
565         centerpoint
566     end
567     pN_global = Jacobian\pN; % Creates global partial derivatives matrix
568     3x8 (coord x node)
569
570     % Formulate Strain Displacement Matrix
571     for i = 1:8
572         B1 = zeros(6,3); % Initializes local B matrix for each point
573         l_pN_vect = [pN_global(1,i),pN_global(2,i),pN_global(3,i)]'; %
574         Local coordinate vector of partial derivatives of N
575         % Assignes PDE values to B matrix
576         B1(1,1) = l_pN_vect(1);
577         B1(2,2) = l_pN_vect(2);
578         B1(3,3) = l_pN_vect(3);
579         B1(4,1) = l_pN_vect(2);
580         B1(4,2) = l_pN_vect(1);
581         B1(5,2) = l_pN_vect(3);
582         B1(5,3) = l_pN_vect(2);
583         B1(6,1) = l_pN_vect(3);
584         B1(6,3) = l_pN_vect(1);
585     end
586     if t == 1 % Creates B matrix at each GQ point
587         element(EID).BGQ(point).B = [element(EID).BGQ(point).B,B1];

```

```

end
582         if t == 2 % Creates B matrix at centerpoint
583             element(EID).BCP = [element(EID).BCP,B1]; end
584         end
585
586         % Calculate stiffness matrix for each element using GQ points
587         if t == 1
588             KGQ =
589             element(EID).BGQ(point).B'*C*element(EID).BGQ(point).B*det(Jacobian); % Stiffness
590             matrix at each GQ point
591             element(EID).K = element(EID).K+KGQ; % Creates global stiffnes
592             matrix for the element
593         end
594     end
595
596     %% Prepare Filter (From TOP3D)
597     function [H,Hs] = filtering(nele,rmin,nelx,nely,nelz)
598         iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
599         jH = ones(size(iH));
600         sH = zeros(size(iH));
601         k = 0;
602         for k1 = 1:nelz
603             for i1 = 1:nelx
604                 for j1 = 1:nely
605                     e1 = (k1-1)*nelx*nely + (i1-1)*nely+j1;
606                     for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz)
607                         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
608                             for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
609                                 e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
610                                 k = k+1;
611                                 iH(k) = e1;
612                                 jH(k) = e2;
613                                 sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
614                             end
615                         end
616                     end
617                 end
618             end
619         end
620         H = sparse(iH,jH,sH);
621         Hs = sum(H,2);
622     end
623
624     %% Finite Element Analysis
625     function [U,UxMat,UyMat,UzMat,element,nodes,KE,VMS_mat,shearXY_mat] =
626     FEA(nelx,nely,nelz,element,nodes,Emin,E,k_penal,s_penal,nele,edofmat,freeDOFs,C,ndof,
627     F,F_Sw_partial,nnode,xPhys,Sw,delta)
628         % Formulate Global Stiffness Matrix: unperturbed
629         KE = element(1).K;
630         iK = reshape(kron(edofmat,ones(24,1))',24*24*nele,1);
631         jK = reshape(kron(edofmat,ones(1,24))',24*24*nele,1);
632         sK = reshape(KE(:)*(Emin+xPhys(:)'.^(k_penal-1))*(E-Emin)*k_penal),24*24*nele,1);
633         K = sparse(iK,jK,sK); K = (K+K')/2; % Sparse global stiffness matrix
634
635         % Formulate Partial Derivative of Global Stiffness Matrix (unpertubed)
636         iK2 = reshape(kron(edofmat,ones(24,1))',24*24*nele,1);
637         jK2 = reshape(kron(edofmat,ones(1,24))',24*24*nele,1);
638         sK2 = reshape(KE(:)*((xPhys(:)'.^(k_penal-1))*(E-Emin)*k_penal),24*24*nele,1);
639         K_partial = sparse(iK2,jK2,sK2); K_partial = (K_partial+K_partial')/2; % Sparse
640         global stiffness matrix
641
642         % Finite Element Analysis
643         U = zeros(ndof,1);
644         U(freeDOFs,:) = K(freeDOFs,freeDOFs)\F(freeDOFs,:);

```

```

642
643     % Calculate global partial derivatives of displacement
        partial_u = zeros(ndof,1); % Initiates vector of partial derivatives of
644 displacement
645     if Sw == true % Includes Force Vector
        partial_u(freeDOFs) = K(freeDOFs,freeDOFs)\(F_Sw_partial(freeDOFs,:)-
646 K_partial(freeDOFs,freeDOFs)*U(freeDOFs,:));
647     else % No Force Vector (applied load only)
        partial_u(freeDOFs) = K(freeDOFs,freeDOFs)\(-
648 K_partial(freeDOFs,freeDOFs)*U(freeDOFs,:));
649     end
650
651     % Store Nodal Displacement (u) in Node Structure (unperturbed)
652     for i = 1:nnode
653         nodes(i).ux = U(nodes(i).dof(1));
654         nodes(i).uy = U(nodes(i).dof(2));
655         nodes(i).uz = U(nodes(i).dof(3));
656         % Partial displacements
657         nodes(i).qx = partial_u(nodes(i).dof(1));
658         nodes(i).qy = partial_u(nodes(i).dof(2));
659         nodes(i).qz = partial_u(nodes(i).dof(3));
660     end
661
662     % Formulate Global Stiffness Matrix: perturbed
663     KE = element(1).K;
664     iKd = reshape(kron(edofmat,ones(24,1))',24*24*nele,1);
665     jKd = reshape(kron(edofmat,ones(1,24))',24*24*nele,1);
666     sKd = reshape(KE(:)*(Emin+(xPhys(:)+delta).^k_penal*(E-Emin)),24*24*nele,1);
667     Kd = sparse(iKd,jKd,sKd); Kd = (Kd+Kd')/2; % Sparse global stiffness matrix
668
669     % Finite Element Analysis
670     Ud = zeros(ndof,1);
671     Ud(freeDOFs,:) = Kd(freeDOFs,freeDOFs)\F(freeDOFs,:);
672
673     % Store Nodal Displacement (Ud) in Node Structure (perturbed)
674     for i = 1:nnode
675         nodes(i).uxd = Ud(nodes(i).dof(1));
676         nodes(i).uyd = Ud(nodes(i).dof(2));
677         nodes(i).uzd = Ud(nodes(i).dof(3));
678     end
679
680     % Store Nodal Displacement in Element Structure
681     for EID = 1:nele
682         element(EID).u = [];
683         element(EID).ud = [];
684         element(EID).q = [];
685         for n = 1:8
686             NID = element(EID).nodeIDs(n);
        element(EID).u = [element(EID).u, [nodes(NID).ux nodes(NID).uy
687 nodes(NID).uz]];
        element(EID).ud = [element(EID).ud, [nodes(NID).uxd nodes(NID).uyd
688 nodes(NID).uzd]];
        element(EID).q = [element(EID).q, [nodes(NID).qx nodes(NID).qy
689 nodes(NID).qz]];
690         end
691         element(EID).q = element(EID).q';
692         element(EID).u = element(EID).u';
693         element(EID).ud = element(EID).ud';
694     end
695
696     % Displacement Matrices
697     Ux = [nodes(:).ux];
698     UxMat = reshape(Ux,[nely+1,nelx+1,nelz+1]);
699     Uy = [nodes(:).uy]; % Vertical displacement
700     UyMat = reshape(Uy,[nely+1,nelx+1,nelz+1]);
701     Uz = [nodes(:).uz];
702     UzMat = reshape(Uz,[nely+1,nelx+1,nelz+1]);

```

```

703     Uxd = [nodes(:).uxd];
704     UxMatd = reshape(Uxd, [nely+1, nelx+1, nelz+1]);
705     Uyd = [nodes(:).uyd]; % Vertical displacement
706     UyMatd = reshape(Uyd, [nely+1, nelx+1, nelz+1]);
707     Uzd = [nodes(:).uzd];
708     UzMatd = reshape(Uzd, [nely+1, nelx+1, nelz+1]);
709
710     % Stress Calculation (at centerpoint)
711     VMS_vect = zeros(nele,1);
712     shearXY_vect = zeros(nele,1);
713     for EID = 1:nele
714         element(EID).x = xPhys(EID); % Assign element density to each element
715         element(EID).strainCP = element(EID).BCP*element(EID).u; % Calculate element
strains
716         element(EID).stressCP = (Emin+(E-
Emin)*element(EID).x^k_penal)*C*element(EID).strainCP*element(EID).x^s_penal; %
Calculate element stress
717         shearXY_vect(EID) = element(EID).stressCP(4); % Isolate shear stress
718
719         % Calculate VMS
720         s11 = element(EID).stressCP(1);
721         s12 = element(EID).stressCP(6);
722         s13 = element(EID).stressCP(5);
723         s22 = element(EID).stressCP(2);
724         s23 = element(EID).stressCP(4);
725         s33 = element(EID).stressCP(3);
726         element(EID).VMS = sqrt(s11^2+s22^2+s33^2-s11*s22-s22*s33-
s33*s11+3*(s12^2+s23^2+s13^2));
727         VMS_vect(EID) = element(EID).VMS;
728     end
729     VMS_mat = reshape(VMS_vect, [nely, nelx, nelz]); % Form VMS matrix
730     shearXY_mat = reshape(shearXY_vect, [nely, nelx, nelz]); % Form SS matrix
731 end
732
733 %% Sensitivity Analysis
734 function [element, VMS_sensitivity_mat, vol, d_vol, pshearXY_mat] =
sensitivity(nelx, nely, nelz, xPhys, element, H, Hs, C, nele, k_penal, s_penal, E, Emin, delta)
735     vol = sum(xPhys(:)); % Objective function
736     d_vol = ones(nely, nelx, nelz); % Objective function sensitivity
737
738     % Constraint function sensitivity
739     pshearXY_vect = zeros(nely, nelx, nelz);
740     VMS_sensitivity_vect = zeros(nele,1);
741     for EID = 1:nele
742         element(EID).pshear = [];
743         xi = element(EID).x;
744         Ei = (Emin+xi'^k_penal*(E-Emin))*C; % SIMP stiffness
745         Eip = (xi'^(k_penal-1)*(E-Emin))*k_penal*C; % Partial derivative of stiffness
(SIMP)
746         ui = element(EID).u; % Unperturbed displacement
747         uid = element(EID).ud; % Perturbed displacement
748         uip = (uid-ui)/delta; % Finite Difference of displacement
749         q = element(EID).q; % Partial of displacement
750         pi = xi^s_penal; % Penalized design variable
751         pip = s_penal*xi^(s_penal-1); % Derivative of penalized design variable
752         Bi = element(EID).BCP; % B matrix at centerpoint
753         VMS = element(EID).VMS;
754         stress_vect = element(EID).stressCP;
755
756         % Calculates Shear Stress Sensitivity
757         if xi >= delta
758             sense_vect = Eip*Bi*ui*pi + Ei*Bi*uip*pi + Ei*Bi*ui*pip;
759             pshearXY_vect(EID) = sense_vect(4);
760         else
761             pshearXY_vect(EID) = 0;
762         end
763     end

```

```

764     % Calculate Derivatives of VMS with respect to its stress componenets
765     VMS_p_wrts = zeros(6,1);
766     VMS_p_wrts(1) = (1/(2*VMS))*(2*stress_vect(1)-stress_vect(2)-stress_vect(3));
767     VMS_p_wrts(2) = (1/(2*VMS))*(2*stress_vect(2)-stress_vect(1)-stress_vect(3));
768     VMS_p_wrts(3) = (1/(2*VMS))*(2*stress_vect(3)-stress_vect(1)-stress_vect(2));
769     VMS_p_wrts(4) = (3/VMS)*stress_vect(4);
770     VMS_p_wrts(5) = (3/VMS)*stress_vect(5);
771     VMS_p_wrts(6) = (3/VMS)*stress_vect(6);
772
773     % Calculate VMS Sensitivity
774     element(EID).VMS_sense = (((VMS^s_penal)^(1/s_penal-1))*(VMS^(s_penal-
1))) * VMS_p_wrts' * (((0.5/sqrt(xi)) * E*C*Bi*ui) + (sqrt(xi) * (E*C*Bi*q)));
775
776     % Prevents VMS sensitivity singularities
777     if isreal(element(EID).VMS_sense)
778         VMS_sensitivity_vect(EID) = element(EID).VMS_sense;
779     end
800     if (isnan(element(EID).VMS_sense) == true) || (isinf(element(EID).VMS_sense)
== true)
801         VMS_sensitivity_vect(EID) = 0;
802     end
803 end
804
805 % Form sensitivity matrices
806 VMS_sensitivity_mat = reshape(VMS_sensitivity_vect,[nely,nelx,nelz]);
807 pshearXY_mat = reshape(pshearXY_vect,[nely,nelx,nelz]);
808
809 % Filtering of Sensitivities
810 pshearXY_mat(:) = H*(pshearXY_mat(:)./Hs);
811 VMS_sensitivity_mat(:) = H*(VMS_sensitivity_mat(:)./Hs);
812 d_vol(:) = H*(d_vol(:)./Hs);
813 end
814
815 % === DISPLAY 3D TOPOLOGY (ISO-VIEW) === (From Top3D)
816 function display_3D(rho)
817 [nely,nelx,nelz] = size(rho);
818 hx = 1; hy = 1; hz = 1; % User-defined unit element size
819 face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
820 set(gcf,'Name','ISO display','NumberTitle','off');
821 for k = 1:nelz
822     z = (k-1)*hz;
823     for i = 1:nelx
824         x = (i-1)*hx;
825         for j = 1:nely
826             y = nely*hy - (j-1)*hy;
827             if (rho(j,i,k) > 0.3) % User-defined display density threshold
828                 vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx; x y-hx z+hx;
829 x+hx y-hx z+hx; x+hx y z+hx];
830                 vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = -vert(:,2,:);
831                 patch('Faces',face,'Vertices',vert,'FaceColor',[0.2+0.8*(1-
832 rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k))]);
833                 hold on;
834             end
835         end
836     end
837 end
838 axis equal; axis tight; axis off; box on; view([30,30]); pause(1e-6);
839 end

```

References

- Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization* 51:565-572.
- ACI Committee 318 (2014) Building Code Requirements for Structural Concrete (ACI 318-14) and Commentary (ACI 318R-14). American Concrete Institute, Farmington Hills.
- Aitcin PC, Flatt RJ (2015) *Science and Technology of Concrete Admixtures*. Elsevier Science & Technology
- Amir O (2013) A topology optimization procedure for reinforced concrete structures. *Computers and Structures* 46–58.
- Amir O, Bogomolny M (2011) Topology Optimization for Conceptual Design of Reinforced Concrete Structures. 9th World Congress on Structural and Multidisciplinary Optimization.
- Amir O, Shakour E (2017) Simultaneous shape and topology optimization of prestressed concrete beams. *Structural and Multidisciplinary Optimization* 57:1831–1843.
- Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2010) Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization* 43:1–16.
- Apis Cor. www.apis-cor.com
- Asprone D, Menna C, Bos FP, Salet TAM, Mata-Falcon J, Kaufmann W (2018) Rethinking reinforcement for digital fabrication with concrete. *Cement and Concrete Research* 112:111-121.
- Banfill PFG (1990) *Rheology of Fresh Cement and Concrete*. Taylor & Francis Group, London, UK.
- Barnes HA (1999) The yield stress—a review or “—everything flows? *Journal of Non-Newtonian Fluid Mechanics* 81:133–178.
- Bauer EI, de Sousa JGG, Guimaraes EA, Silva FGS (2007) Study of the laboratory Vane test on mortars. *Building and Environment* 42:86-92.
- Bendsoe MP (1989) Optimal shape design as a material distribution problem. *Structural Optimization* 1:193-202.
- Bendsoe MP, Kikuchi N (1988) Generating Optimal Topologies in Structural Design using a Homogenization Method. *Computer Methods in Applied Mechanics and Engineering* 71:197-224.
- Bendsoe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. *Archive of Applied Mechanics* 69:635-654.

- Biyikli E, To AC (2015) Proportional Topology Optimization: A new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in Matlab. PLoS ONE 10: e0145041.
- Bogomolny M, Amir O (2012) Conceptual design of reinforced concrete structures using topology optimization with elastoplastic material modeling. International Journal for Numerical Methods in Engineering 90:1578–1597.
- Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press. Cambridge, UK.
- Bruggi M, Duysinx P (2012) Topology optimization for minimum weight with compliance and stress constraints. Structural and Multidisciplinary Optimization 46:369–384.
- Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. Computational Methods in Applied Mechanics and Engineering 190:3443–3459
- Bruyneel M, Duysinx P (2005) Note on topology optimization of continuum structures including self-weight. Structural and Multidisciplinary Optimization 29: 245–256.
- Buswell RA, Leal de Silva WR, Jones SZ, Dirrenberger J (2018) 3D printing using concrete extrusion: A roadmap for research. Cement and Concrete Research 112:37–49.
- CES EduPack software, Granta Design Limited, Cambridge, UK, 2019 (www.grantadesign.com).
- Chandler DL (2017). 3-D printing offers new approach to making buildings.
<https://news.mit.edu/2017/3-d-printing-buildings-0426>
- Challis V (2010) A discrete level-set topology optimization code written in Matlab. Structural and Multidisciplinary Optimization 41:453–464.
- Chu S, Xiao M, Gao L, Li H (2018) A level set-based method for stress-constrained multimaterial topology optimization of minimizing a global measure of stress. International Journal for Numerical Methods in Engineering 117:800–818.
- Crump, S (1992) Apparatus and method for creating three-dimensional objects. US Patent 5121329A.
- Cunha J, Chaves LP (2014) The use of topology optimization in disposing carbon fiber reinforcement for concrete structures. Structural and Multidisciplinary Optimization 49:1009–1023.
- Cybe. R&Drone Laboratory. www.cybe.eu/case/rdrone-lab/
- D-Shape: www.d-shape.com
- Dapogny C, Faure A, Michailidis G, Allaire G, Couvelas A, Estevez R (2017) Computational Mechanics 59:933–965.

- Diaz A, Sigmund O (1995) Checkerboard patterns in layout optimization. *Structural Optimization* 10:40-45.
- Du Z, Zhang Y, Zhang W, Guo X (2016) A new computational framework for materials with different mechanical responses in tension and compression and its applications. *International Journal of Solids and Structures* 100–101:54-73.
- Du Z, Zhang W, Zhang Y, Xue R, Guo X (2018) Structural topology optimization involving bi-modulus materials with asymmetric properties in tension and compression. *Computational Mechanics* 63:335-363.
- Du Z, Zhou X, Picelli R, Kim H (2018) Connecting Microstructures for Multiscale Topology Optimization With Connectivity Index Constraints. *Journal of Mechanical Design* 140: 111417.
- Dumas J, Hergel J, Lefebvre S (2014) Bridging the gap: automated steady scaffoldings for 3D printing. *ACM Transactions on Graphics* 33:98
- Duysinx P (2020) Structural and Multidisciplinary Optimization [pdf]. Retrieved from http://www.ingveh.ulg.ac.be/uploads/education/meca-0027-1/Opti_PT_2019-2020_Ch4.pdf
- Duysinx P, Bendsoe MP (1998) Topology optimization of continuum structures with local stress constraints. *International Journal for Numerical Methods in Engineering* 43:1453-1478.
- Eindhoven University of Technology. Concrete printing with precision. <https://www.cursor.tue.nl/en/news/2015/december/concrete-printing-with-precision/>
- Elaty MAA, Ghazy MF (2012) Flow properties of fresh concrete by using modified geotechnical Vane shear test. *HBRC Journal* 8:159-169.
- Ewoldt R (2009) Yield Stress Fluids. Lecture, Massachusetts Institute of Technology.
- Feng P, Meng X, Chen JF, Ye L (2015) Mechanical properties of structures 3D printed with cementitious powders. *Construction and Building Materials* 93:486-497.
- Gaynor AT, Guest JK (2016) Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization* 54:1157–1172.
- Gaynor AT, Meisel NA, Williams CB, Guest JK (2014) Multiple-Material Topology Optimization of Compliant Mechanisms Created Via PolyJet Three-Dimensional Printing. *Journal of Manufacturing Science and Engineering* 136: 061015.
- Gebremedhen HS, Woldemicael DE, Hashim FM (2019) Three-dimensional stress-based topology optimization using SIMP method. *International Journal for Simulation and Multidisciplinary Design Optimization* 10:A1.

- Gosselin C, Duballet R, Roux Ph, Gaudilliere N, Dirrenberger J, Morel Ph (2016) Large-scale 3D printing of ultra-high performance concrete – a new processing route for architects and builders. *Materials and Design* 100:102-109.
- Guest JK, Moen CD (2010) Reinforced concrete design with topology optimization. 19th Analysis & Computation Specialty Conference.
- Hambach M, Volkmer D (2017) Properties of 3D-printed fiber-reinforced Portland cement paste. *Cement and Concrete Composites* 79:62-70.
- Heikal M, Ibrahim NS (2016) Hydration, microstructure and phase composition of composite cements containing nano-clay. *Construction and Building Materials* 112:19-72.
- Holmberg E, Torstenfelt B, Klarbring A Stress constrained topology optimization. *Structural and Multidisciplinary Optimization* 48:33-47.
- Holt C, Edwards L, Keyte L, Moghaddam F, Townsend B (2019) Construction 3D Printing. In Sanjayan JG, Nazari A, Nematollahi B (ed) *3D Concrete Printing Technology*, Butterworth-Heinemann, Cambridge, MA, 349-370
- Huang X, Xie YM (2010) A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization* 41:671–683.
- Hull, CW (1986) Apparatus for Production of Three-Dimensional Objects by Stereolithography. US Patent 4575330.
- Jeong H, Han SJ, Choi SH, Lee YJ, Yi ST, Kim KS (2019) Rheological Property Criteria for Buildable 3D Printing Concrete. *Materials* 12:657.
- Jipa A, Bernhard M, Meibodi M, Dillenburger B (2016) 3D-Printed Stay-in-Place Formwork for Topologically Optimized Concrete Slabs. 77th Texas Society of Architects Annual Convention and Design Expo.
- Juenger MCG, Winnefeld F, Provis JL, Ideker JH (2011) Advances in alternative cementitious binders. *Cement and Concrete Research* 41:1232-1243.
- Kawashima S, Chaouche M, Corr DJ, Shah SP (2013) Rate of thixotropic rebuilding of cement pastes modified with highly purified attapulgite clays. *Cement and Concrete Research* 53:112-118.
- Kazemian A, Yuan X, Cochran E, Khoshnevis B (2017) Cementitious materials for construction-scale 3D printing: Laboratory testing of fresh printing mixture. *Construction and Building Materials* 145:639-647.
- Khalil N, Aouad G, Cheikh KE, Remond S (2017) Use of calcium sulfoaluminate cements for setting control of 3D-printing mortars. *Construction and Building Materials* 157:382-391.

- Khoshnevis B, Bukkapatnam S, Kwon H, Saito J (2001) Experimental investigation of contour crafting using ceramics materials. *Rapid Prototyping Journal* 7:32-41.
- Khoshnevis B, Hwang D (2004) Concrete Wall Fabrication by Contour Crafting. *International Symposium on Automation and Robotics in Construction*.
- Khoshnevis B, Hwang D, Yao KT, Yeh Z (2006) Mega-scale fabrication by contour crafting. *Int. J. Industrial and Systems Engineering* 1:301-320.
- Kruger J, Cho S, Zeranka S, Viljoen C, van Zijil G (2020) 3D concrete printer parameter optimisation for high rate digital construction avoiding plastic collapse. *Composites Part B* 183:107660.
- Kruger J, Zeranka S, van Zijil G (2019) 3D concrete printing: A lower bound analytical model for buildability performance quantification. *Automation in Construction* 106:102904
- Larsson R (2016) Methodology for topology and shape optimization: Application to a rear lower control arm. Dissertation, Chalmers University of Technology.
- Le TT, Austin SA, Lim S, Buswell RA, Gibb AGF, Thorpe T (2012) Mix design and fresh properties for high-performance printing concrete. *Materials and Structures* 45:1221-1232.
- Leary M, Merli L, Torti F, Mazur M, Brandt M (2014) Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials & Design* 63:678-690.
- Lee E, James KA, Matrins J (2012) Stress-Constrained Topology Optimization with Design-Dependent loading. *Structural and Multidisciplinary Optimization* 46:647-661.
- Li D, Kim IY (2018) Multi-material topology optimization for practical lightweight design. *Structural and Multidisciplinary Optimization* 58:1081–1094.
- Li Q, Chen W, Liu S, Tong L (2016) Structural topology optimization considering connectivity constraint. *Structural and Multidisciplinary Optimization* 54:971–984.
- Lim S, Buswell RA, Le TT, Austin SA, Gibb AGF, Thorpe T (2012) Developments in construction-scale additive manufacturing processes. *Automation in Construction* 21:262-268.
- Lim S, Le T, Webster J, Buswell R, Austom S, Gibb A, Thorpe T (2009) Fabricating construction components using layer manufacturing technology. *Global Innovation in Construction Conference*.
- Liu K, Tovar A (2014) An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 50:1175–1196.
- Livesu M, Ellero S, Martinez J, Lefebvre S, Attene M (2017) From 3D models to 3D prints: an overview of the processing pipeline. *Eurographics* 36.2:537-564.

- Lopes RA, Stump FV, Silva ECN (2005) Topology Optimization of Three Dimensional Structures under Self-weight and Inertial Forces. 6th World Congresses of Structural and Multidisciplinary Optimization Rio de Janeiro, Brazil."
- Luo Y, Kang Z (2012) Topology optimization of continuum structures with Drucker–Prager yield stress constraints. *Computers and Structures* 90-91:65-75.
- Luo Z, Gao W, Song C (2010) Design of Multi-phase Piezoelectric Actuators. *The Journal of Intelligent Materials Systems and Structures* 21:1851-1856.
- Ma G, Li Z, Wang L (2018) Printable properties of cementitious material containing copper tailings for extrusion based 3D printing. *Construction and Building Materials* 162:613-627.
- Ma G, Wang L (2018) A critical review of preparation design and workability measurement of concrete material for largescale 3D printing. *Frontiers of Structural and Civil Engineering* 12:382–400.
- Malaeb Z, Hachem H, Tourbah A, Maalouf T, El Zarwi N, Hamzeh F (2015) 3D Concrete Printing: Machine and Mix Design. *International Journal of Civil Engineering and Technology* 6.6:14-22.
- Marchment T, Xia M, Dodd E, Sanjayan J, Nematollahi B (2017) Effect of Delay Time on the Mechanical Properties of Extrusion-based 3D Printed Concrete. 34th International Symposium on Automation and Robotics in Construction.
- Martens PA, Mathot M, Coenders JL, Bos FP, Rots JG (2017) Optimising 3D printed concrete structures using topology optimisation .*Proceedings of the IASS Annual Symposium*.
- Meli E, Rindi A, Ridolfi A, Furferi R, Buonamici F, Iurisci G, Corbo S, Cangioli F (2019) Design and Production of Innovative Turbomachinery Components via Topology Optimization and Additive Manufacturing. *International Journal of Rotating Machinery*. Vol. 2019, Article ID 9546831, 12 pages.
- Moosavi M, Bawden WF (2003) Shear strength of Portland cement grout. *Cement & Concrete Composites* 25:729-735.
- Nerella V, Mechtcherine V (2019) Studying the Printability of Fresh Concrete for Formwork-Free Concrete Onsite 3D Printing Technology (CONPrint3D). In Sanjayan JG, Nazari A, Nematollahi B (ed) *3D Concrete Printing Technology*, Butterworth-Heinemann, Cambridge, MA, 333-347
- Nomura T, Dede EM, Matsumori T, Kawamoto A (2015) Simultaneous optimization of topology and orientation of anisotropic material using isoparametric projection method. 11th World Congress on Structural and Multidisciplinary Optimization.
- Omran A, Naji S, Khayat K (2011) Portable Vane Test to Assess Structural Buildup at Rest of Self-Consolidating Concrete. *ACI Materials Journal* 108-M67 628:637

- Osher S, Sethian J (1988) Fronts Propagating with Curvature- Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics* 79:12-49.
- Panda B, Lim JH, Tan MJ (2019) Mechanical properties and deformation behaviour of early age concrete in the context of digital construction. *Composites Part B: Engineering* 165:563-571.
- Panda B, Paul SC, Mohamed NAN, Tay YWD, Tan MJ (2018) Measurement of tensile bond strength of 3D printed geopolymers mortar. *Measurement* 113:108-116.
- Panda B, Tan MJ (2018) Experimental study on mix proportion and fresh properties of fly ash based geopolymer for 3D concrete printing. *Ceramics International* 44:10258-10265.
- Paolini A, Kollmannsberger S, Rank E (2019) Additive manufacturing in construction: A review on processes, T applications, and digital planning methods. *Additive Manufacturing* 30:100894.
- Papadarakakis M, Tsompanakis Y (1996). Advanced solution methods in topology optimization and shape sensitivity analysis. *Engineering Computations* 13:57-90.
- Paul SC Tay YWD, Panda B, Tan MJ (2018) Fresh and hardened properties of 3D printable cementitious materials for building and construction. *Archives of Civil and Mechanical Engineering* 18:311-319.
- Paul SC, van Zijil G, Tan MJ, Gibson I (2018) A review of 3D concrete printing systems and materials properties: current status and future research prospects. *Rapid Prototyping Journal* 24.4:784-798.
- Pegna J (1997) Exploratory investigation of solid freeform construction. *Automation in Construction* 5:427-437.
- Perez RE, Jansen PW, Martins JRRA (2012) pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization* 45:101–118.
- Pu Zhang, Jikai Liu, Albert C. To (2017) Role of anisotropic properties on topology optimization of additive manufactured load bearing structures. *Scripta Materialia* 135:148-152.
- Rahul AV, Santhanam M, Meena H, Ghani Z (2019) 3D printable concrete: Mixture design and test methods. *Cement and Concrete Composites* 97:13-23.
- Rao SS (2011) *The Finite Element Method in Engineering*, 4th Edition, Butterworth-Heinemann, Cambridge, MA, Chapter 11
- Rasmussen MH, Stolpe M (2008) Global optimization of discrete truss topology design problems using a parallel cut-and-branch method. *Computers and Structures* 86:1527–1538.
- Robbins J, Owen SJ, Clark BW, Voth TE (2016) An efficient and scalable approach for generating topologically optimized cellular structures for additive manufacturing. *Additive Manufacturing* 12:296-304.

- Roussel N (2006) A thixotropy model for fresh fluid concretes: Theory, validation and applications. *Cement and Concrete Research* 36:1798-1806.
- Roussel N, Lemaitre A, Flatt RJ, Coussot P (2010) Steady state flow of cement suspensions: A micromechanical state of the art. *Cement and Concrete Research* 40:77–84.
- Sethian JA, Wiegmann A (2000) Structural Boundary Design via Level Set and Immersed Interface Methods. *Journal of Computational Physics* 163:489–528.
- Sigmund O (2001) A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 21:120–127.
- Sigmund O, Maute K (2013) Topology optimization approaches. *Structural and Multidisciplinary Optimization* 48:1031–1055.
- Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization* 16:68-75.
- Stojanov D, Falzon BG, Wu X, Yan W (2016) Implementing a structural continuity constraint and a halting method for the topology optimization of energy absorbers. *Structural and Multidisciplinary Optimization* 54:429–448.
- Stolpe M (2003) On Models and Methods for Global Optimization of Structural Topology. Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden.
- Stolpe M, Bendsoe MP (2011) Global optima for the Zhou--Rozvany problem. *Structural and Multidisciplinary Optimization* 43:151–164.
- Suresh K (2010) A 199-line Matlab code for Pareto-optimal tracing in topology optimization. *Structural and Multidisciplinary Optimization* 42:665–679.
- Suresh K, Takaloozadeh M (2013) Stress-constrained topology optimization: a topological level-set approach. *Structural and Multidisciplinary Optimization* 48:295–309.
- Svanberg K (1987) The Method of Moving Asymptotes - A New Method for Structural Optimization. *International Journal for Numerical Methods in Engineering* 24:359-373.
- Svanberg K (2002) A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. *Journal of Optimization* 12:555-573.
- Talisch C, Paulino GH, Pereira A, Menezes IFM (2012) PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Structural and Multidisciplinary Optimization* 45:329–357.

- Tregger NA, Pakula ME, Shah SP (2010) Influence of clays on the rheology of cement pastes. *Cement and Concrete Research* 40:384-391.
- Tucker A, Krog L, Kemp M, Boyd R (2004) Topology Optimization of Aircraft Wing Box Ribs. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.
- van Zijil G, Paul SC, Tan MJ (2016) Properties of 3D printable concrete. 2nd International Conference on Progress in Additive Manufacturing.
- Vanek J, Galicia JAG, Benes B (2014) Clever Support: Efficient Support Structure Generation for Digital Fabrication. *Segmentation and Structure* 33:117-125.
- Vogiatzis P, Chen S, Zhou C (2017) An Open Source Framework for Integrated Additive Manufacturing and Level-Set-Based Topology Optimization. *Journal of Computing and Information Science in Engineering* 17:041012
- Voigt T, Mbele JJ, Wang K, Shah SP (2010) Using Fly Ash, Clay, and Fibers for Simultaneous Improvement of Concrete Green Strength and Consolidability for Slip-Form Pavement. *Journal of Materials in Civil Engineering* Feb.196-206.
- Wang SY, Lim KM, Khoo BC, Wang MY (2007) An extended level set method for shape and topology optimization. *Journal of Computational Physics* 221:395–421.
- Wang W, Wang TY, Yang Z, Liu L, Tong X, Tong W, Deng J, Chen F, Liu X (2013) Cost-effective printing of 3D objects with skin-frame structures. *ACM Transactions on Graphics* 32:177.
- Weng Y, Li M, Tan MJ, Qian S (2018) Design 3D printing cementitious materials via Fuller Thompson theory and Marson-Percy model. *Construction and Building Materials* 163:600-610.
- WinSun. Yingchuang 200 sets of 3D printed donation isolated houses urgently need supporting facilities. www.winsun3d.com/En/News/news_inner/id/544
- World's Advanced Saving Project. <https://www.3dwasp.com/en/>
- Wu P, Wang J, Wang X (2016) A critical review of the use of 3-D printing in the construction industry. *Automation in Construction* 68:21-31.
- Yamada T, Izui K, Nishiwaki S, Takezawa A (2010) A topology optimization method based on the level set method incorporating a fictitious interface energy. *Computer Methods in Applied Mechanics and Engineering* 199:2876–2891.
- Young V, Querin OM, Steven GP, Xie YM (1999) 3D and multiple load case bi-directional evolutionary structural optimization (BESO). *Structural Optimization* 18:183-192.
- Zhang P, Liu J, To AC (2017) Role of anisotropic properties on topology optimization of additive manufactured load bearing structures. *Scripta Materialia* 135:148-152.

Zhang X, Li M, Lim JH, Weng Y, Tay YWD, Pham H, Pham QC (2018) Large-scale 3D printing by a team of mobile robots. *Automation in Construction* 95:98-106.

Zhang Y, Zhang Y, She W, Yang L, Liu G, Yang Y (2019) Rheological and harden properties of the high-thixotropy 3D printing concrete. *Construction and Building Materials*. 201:278-285.

Zollo R (1996) Fiber-reinforced Concrete: an Overview after 30 Years of Development. *Cement and Concrete Composites* 19:107-122.

Zuo W, Saitou K (2017) Multi-material topology optimization using ordered SIMP interpolation. *Structural and Multidisciplinary Optimization* 55:477–491.

Zuo ZH, Xie YM (2015) A simple and compact Python code for complex 3D topology optimization. *Advances in Engineering Software* 85:1-11.