



THE HONG KONG
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library

包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

**DEVELOPMENT AND APPLICATION OF
ISOGOMETRIC ANALYSIS AND TOPOLOGY
OPTIMIZATION FOR FUTURE STRUCTURAL
DESIGN**

ZHANG ZIXIN

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University
Department of Building Environment and Energy
Engineering

**Development and application of isogeometric analysis and
topology optimization for future structural design**

ZHANG Zixin

A thesis submitted in partial fulfilment of the requirements for the degree
of Doctor of Philosophy

August 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

ZHANG Zixin

Abstract

In the context of rapid development in construction technology, civil engineers have greater design freedom for pursuing efficient yet aesthetic structures. The conceptual design phase may often produce a complex structure and prove challenging for structural safety requiring multiple revisions and negotiations between the architect and structural engineer in order to obtain a practical, economical, and safe design. This thesis presents a framework to facilitate better communication between architects and structural engineers by establishing an isogeometric analysis and topology optimization platform that may also explicitly account for constructability constraints.

Isogeometric analysis (IGA) is an emerging numerical approach for simulating the structural response, which aims at integrating Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) into a consistent mathematical expression. Benefiting from its precise boundary description and high-order continuity, IGA is recognized to have great application potential in various industrial scenarios. To explore the application value of IGA-based structural optimization for the construction industry, an isogeometric optimization approach of the post-tensioned concrete beam, a typical construction component, was developed. This approach uses NURBS curves for the geometric description and interpolation, which incorporates the SIMP method to generate the optimized topology and enables simultaneous optimization of prestress tendon shape and concrete beam topology for pursuing optimal structural performance. With a Drucker-Prager criterion, unequal tension and compression stress constraints are adopted for the use of concrete material. The isogeometric description of the concrete beam not only enhances

the accuracy of stress calculation but also gives a simplified and straightforward prestress mapping scheme. To avoid the existence of slim components, a NURBS-based minimum width control approach is adopted in a consistent isogeometric framework.

To standardize and continuously maintain the structural analysis and topology optimization process under the isogeometric framework, an integrated structural simulation tool shall be based on open-source software and facilitated by a specially developed tool to consider various construction demands. This includes an isogeometric analysis module in OpenSees for simulating the structural response, as well as a corresponding topology optimization package. In this thesis, the development of IGA module begins from the IGAQuad and IGABrick elements with NURBS-based shape functions and geometric descriptions in OpenSees, a widely used finite element modelling framework. The reliability and modelling capabilities of these IGA solid elements have been verified from the 2D and 3D benchmark problems. The isogeometric optimization package includes two classical topology optimization models that are the minimum compliance model with volume constraint and the minimum volume model with unequal stress constraint. The base of development and maintenance is the opensource and modular infrastructure of OpenSees and its extensive library of material models and solution methods that provides strong code support for pursuing the futural construction-oriented design.

With the computational platform established, various construction considerations can be integrated. In the context of the still-developing construction printing technology, prefabrication technology is relatively more economical and reliable at the current stage. From the structural design perspective, periodic topology optimization addresses the prefabricability,

regularity aesthetic, and assemblability, which fits the demands of the construction-oriented design. However, due to the strict geometric constraint, full periodic structure over-sacrifices structural performance like compliance. To trade-off the manufacturability of full-periodic structures with the high structural performance of free-form design, it is effective to combine the advantages of both by assembling the free-form and periodic components according to a rational scheme, which is called multi-pattern design. Herein, we proposed a new multi-pattern topology optimization method for allowing user-defined periodic groups between the individual unit-cells, several unit-cells are grouped with the same periodic configurations (type 1 to n) to find the possible designs between full-periodic and free-form structures. To automatically define a more beneficial scheme of the periodic group with a huge number of unit cells in case no preferred scheme, a new clustering-based multi-pattern selection approach is also proposed. It can be found that the multi-pattern design performs the merits of regularity from full-periodic design and efficient structural performance from free-form design, users can simply control the optimization tendency to determine the solution closer to free-form or periodic design.

Publications During Ph.D. Study

First Author Journal Papers of construction-oriented topology optimization:

1. **Zhang, Z.**, Yarlagadda, T., Zheng, Y., Jiang, L.*, & Usmani, A. (2021). Isogeometric analysis-based design of post-tensioned concrete beam towards construction-oriented topology optimization. *Structural and Multidisciplinary Optimization*, 64(6), 4237-4253. (IF=4.542, JCR 2021, Q1)
2. **Zhang, Z.**, Jiang, L.*, Yarlagadda, T., Zheng, Y., & Usmani, A. (2022) OPS-ITO: Development of Isogeometric Analysis and Topology Optimization in OpenSEES for free-form structural design. *Computer-Aided Design* (IF=4.3, JCR 2023, Q1)
3. **Zhang, Z.**, Jiang, L.*, Yarlagadda, T., Zheng, Y., & Usmani, A. (2023) A novel multi-pattern control for automated structure design using topology optimization to balance form and performance. *Engineering Structures* (Major Revision, IF=5.582, JCR 2022, Q1)

Other Journal Papers on construction-oriented topology optimization:

1. Yarlagadda T*., **Zhang Z.**, Jiang L., Bhargava P., Usmani A. (2022). Solid isotropic material with thickness penalization - A 2.5D method for structural topology optimization. *Computers & Structures* (IF=5.372, JCR2022, Q1)
2. Yarlagadda T*., **Zhang Z.**, Jiang L., Bhargava P., Usmani A. (2023). Exploring the 2.5D SIMTP with the adaptive refinement based on the nodal thickness. *Proceedings of the 4th International Conference on 3D Construction Printing*

Other Journal Papers:

1. Jiang, L*., Jiang, Y., **Zhang, Z.**, & Usmani, A. (2021). Thermal analysis infrastructure in OpenSees for fire and its smart application interface towards natural fire modelling. *Fire Technology*. (IF=3.605, JCR2022, Q1)

Conference Proceedings/Abstracts of construction-oriented topology optimization:

2022.10, Shanghai, China. Oral presentation, Best Research Award.

Forum: The 4th Young Scholars Forum of National Civil Structure Prefabrication Research Center, 2022.

2022.6, Shanghai, China. Oral presentation.

Conference: The International Conference on 3D Printing Concrete Material and Structures, 2022.

2023.5, Hong Kong, China. Poster presentation

Conference: PAIR Conference 2023: Research Excellence for Societal Impacts.

2023.7, Singapore. Oral presentation & Proceeding paper

Conference: The 4th International Conference on 3D Construction Printing.

Project Funding Obtained Based on this Thesis

Research Institute for Advanced Manufacturing (RIAM) P0031564

Topic: Digital fabrication in favor of form and function: a vision towards carbon-neutral construction.

Acknowledgment

I can still recall that night at Zhejiang University, my master's supervisor Prof. Zheng shared with me the information from Prof. Usmani that presented with me a brand-new way of exploring the topic of automated architectural design. For the first time, I realized that algorithms could gradually remove useless material to produce efficient and beautiful pieces. Without hesitation, I expressed my intention and fortunately got unreserved support from Prof. Usmani and Prof. Zheng, who brought me into this inspiring research field.

At the beginning of my Ph.D. study at PolyU, the openness and freedom of research were new to me, and I was temporarily lost by the magnitude of the research field. But fortunately, the warm and kind encouragement from Prof. Usmani support me to explore my research interest. Especially with the constructive and candid help from Dr. Liming Jiang and Dr. Tejeswar Yarlagadda, I can finally overcome all the obstacles.

I would like to sincerely thank my supervisor, Prof. Asif Sohail Usmani, for all the opportunities and research freedom he gave me, who made me better than I thought I could be. I also need to express my gratitude and respect to Prof. Yao Zheng, Prof. Jianguo Dai, Prof. Hau-Chung Man, Prof. Tak-Ming Chan, and all group members for their helpful assistance. Appreciate Dr. Xueyan Hu, Zonghao Li, Qicong Li, Zihao Yu, Yizhang Zhu, Daniyar Kurmankhojayev, Jiaming Zhu, Lei Qian, Dr. Hui Yang for keeping accompanying me during this meaningful period.

Sincerely thank my family for all their heart-warming care and unconditional understanding. I would also appreciate my girlfriend, Weichen

Liu, for her presence in my life and gave me unlimited support. Finally, a motto for myself. The road may have no direction, but the innermost being is the beacon of your life journey.

Table of Contents

Abstract	I
Publications During Ph.D. Study	V
Acknowledgment	VII
Table of Contents	IX
List of Tables	XIII
List of Figures	XV
Symbols	XIX
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Trinity principles of architecture.....	1
1.1.2 The environmentally sustainable way to use concrete.....	3
1.1.3 The trend of automated construction	4
1.1.4 Considering the trinity principle in architecture design using topology optimization	6
1.2 Scopes and objectives.....	8
1.3 Thesis overview	9
Chapter 2 Literature review	13
2.1 Prefabrication and 3D printing in construction.....	13
2.1.1 Prefabrication	13
2.1.2 3D construction printing	14
2.2 Topology optimization.....	16
2.2.1 Computational frameworks of topology optimization	17
2.2.2 Common numerical problems and corresponding solutions.....	17
2.2.3 Isogeometric topology optimization	21
2.3 Key considerations of concrete structure design using topology optimization	24
2.3.1 Durability performance	24
2.3.2 Utility performance	27
2.3.3 Aesthetic performance	29
2.4 Concluding remarks	30
Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam	33
3.1 Introduction	33
3.2 Isogeometric Analysis and NURBS curve.....	36
3.2.1 NURBS curve	36
3.2.2 Curve refinement.....	38
3.2.3 Isogeometric structural analysis and IGA-SIMP method	39

3.3	Modeling post-tensioned concrete beams using IGA	41
3.3.1	IGA for prestressed beam	42
3.3.2	Equivalent prestress load along the tendon	43
3.3.3	A simplified load mapping scheme using IGA.....	45
3.4	IGA-SIMP optimization model with unequal stress constraints	47
3.4.1	Drucker-Prager stress constraint.....	47
3.4.2	Refinements of stress constraint	48
3.4.3	Geometry control of optimized configuration	50
3.5	Sensitivity analysis	53
3.5.1	Volume sensitivity	53
3.5.2	Stress constraint sensitivity.....	55
3.6	Demonstrative examples.....	58
3.6.1	Simply supported beams with single span.....	59
3.6.2	Simply supported beams with multiple spans	65
3.7	Concluding remarks.....	67
Chapter 4 An object-oriented framework for the isogeometric topology optimization		69
4.1	Introduction of OpenSees framework.....	72
4.1.1	The object-oriented programming infrastructure.....	72
4.1.2	OpenSees for finite element analysis.....	74
4.1.3	Interpreters of OpenSees.....	75
4.2	Development of isogeometric analysis module in OpenSees.....	77
4.3	Verification examples of using Isogeometric elements in OpenSees.....	80
4.3.1	2D model using IGAQuad: an infinite plate with a circular hole subjected to tension	80
4.3.2	3D model using IGABrick: a cylinder with internal pressure	84
4.4	Development of isogeometric topology optimization package for the OpenSees..	88
4.4.1	Basic optimization models.....	88
4.4.2	Details of the optimization package	93
4.5	Demonstrative examples of using the OPS-ITO package	97
4.5.1	A planar quarter annulus structure design using the minimum compliance model	98
4.5.2	A planar L-shape beam design using the minimum volume model with stress constraint	100
4.5.3	A 3D cantilever beam design using minimum compliance model	102
4.6	Concluding remarks.....	104
Chapter 5 Between free-form and periodic design - a multi-pattern approach .		107
5.1	Introduction.....	107
5.2	Multiple patterns design using topology optimization	110
5.3	Automated grouping of unit cells for multi-pattern topology optimization	113
5.3.1	Clustering-based selection of unit cell scheme for automated grouping	114
5.3.2	Post-processing of automated multi-pattern topology optimization.....	120

5.4	Application of multi-pattern control in manually assigned cells	121
5.4.1	Performance verification of multi-pattern topology optimization	121
5.4.2	Demonstration of multi-pattern control in 3×1 and 4×1 design domains	124
5.4.3	Demonstration of multi-pattern control for a frame subjected to horizontal load	126
5.4.4	Demonstration of multi-pattern topology optimization with automated grouping.....	127
5.4.5	Demonstration of automated grouping with zero-force removal.....	130
5.5	Concluding remarks	132
Chapter 6	Conclusions and future study.....	135
6.1	Conclusions	135
6.2	Future works.....	137
Appendix A	139
Appendix B	143
Appendix C	153
Appendix D	165
Reference	169

List of Tables

Table 3.1 Analogous between the finite element method and isogeometric analysis with NURBS	40
Table 3.2 Illustration of various minimum-width control in topology optimization	62
Table 3.3 Effect of different strength ratio to the topology optimization with low width control (11.6 mm)	64
Table 3.4 Topology optimization results after applying strength ratios and 23mm minimum-width control.....	65
Table 4.1 Example of Tcl script for defining a node in OpenSees	76
Table 4.2 IGA models for the infinite plate with a circular hole	82
Table 4.3 Simulation cases for modeling 3D cylinder with different order and mesh schemes	86
Table 4.4 Input parameters to OPS-ITO for optimization of a quarter annulus structure	99
Table 4.5 Input parameters for optimization of an L-shape beam.....	101
Table 4.6 The input parameters of the 3D cantilever beam case	103

List of Figures

Figure 1.1 Architectures inspired by the load-carrying nature	2
Figure 1.2 Architectures designs using topology optimization	7
Figure 2.1 A few representative projects manufactured by 3D printing. (a) the concrete castle by Rudenko in 2014, Minnesota, US, (b) concrete houses and a 6-storey building by WinSun in 2015, Shanghai, China, (c) Public restroom by WinSun in 2016, (d) Habitat for Humanity, a charitable project for low-income families, by Alquist in 2021, Virginia, US, (e) Shamballa village, the vision of the first community living entirely in 3D printed buildings, by WASP in 2018, Massa Lombarda, Italy, (f) an odd-shaped pedestrian bridge by MX3D in 2021, Amsterdam, the Netherlands, (g) Urban cabin for disaster relief and temporary functional housing, by DUS Architects in 2015, Amsterdam, the Netherlands, (h) Office of the future, the first 3D printed commercial building by Gensler and WinSun in 2016, Dubai, United Arab Emirates.	15
Figure 2.2 General workflow of the topology optimization process	16
Figure 3.1 A brief vision of the futural construction design for the post-tensioned concrete beam	35
Figure 3.2 Example of a 2-order B-spline curve, (a) the basis functions of all control points, (b) control points and the B-spline curve	37
Figure 3.3 Example of prestress tendon: (a) Tendon curve described by NURBS, (b) equivalent prestress load along the tendon	43
Figure 3.4 The process of prestressed load discretion	44
Figure 3.5 The h-refinement process of tendon control point for improving the load discretion accuracy	45
Figure 3.6 Comparison of the physical-parametrical mapping process between FEA and IGA	46
Figure 3.7 The inverse mapping process of an equivalent load from the tendon to the concrete domain	47
Figure 3.8 Designable region of the tendon curve	48
Figure 3.9 Illustration of geometry control	51
Figure 3.10 Comparison of projection schemes with different project threshold, (η_{ero} , η_{dil}): (a) no Heaviside; (b) projection threshold is 0; (c) variable projection threshold e.g.: (0.4, 0.6)	53
Figure 3.11 A half model of the simply supported beam with a uniform load	60

Figure 3.12 Topology optimization result(equal strength limits, low width control (11.6 mm)): (a) optimization process; (b) the distribution of D-P index ($I_{dp} = \phi_{F,e-I}$), negative value of I_{dp} indicates below strength limit.	61
Figure 3.13 Optimized topology with medium width control (23.3 mm) and equal strength limits: (a) optimization process; (b) the distribution of D-P index I_{dp}	62
Figure 3.14 Topology optimization with medium width control and 3:1 strength limit ratio: (a) optimization process; (b) the distribution of D-P index I_{dp}	65
Figure 3.15 Schematic plot of continuous beams.: (a) a two-span beam (2×2.0m); (b) a three-span beam (3×2.0m).....	66
Figure 3.16 Topology optimization process of two-span beam: (a) density distribution of two-span beam; (b) D-P index I_{dp} distribution of two-span beam; (c) density distribution of three-span beam; (d) D-P index I_{dp} distribution of three-span beam.....	66
Figure 4.1 A design tool for future free-form structural design	70
Figure 4.2 Illustration of different class relationships in the class diagram	74
Figure 4.3 Class hierarchy of OpenSees framework	75
Figure 4.4 Class hierarchy of OpenSees framework with our ITO extensions	78
Figure 4.5 Workflow of computing IGA shape function.....	79
Figure 4.6 An infinite plate with a circular hole subjected to tension: (a) schematic of the model; (b) a quarter-plate model for analysis.....	81
Figure 4.7 Various mesh schemes for modelling the quarter plate: (a) case 1-3, control points of case 1; (b) case 4; (c) case 5; (d) Case 6 (fine mesh)	83
Figure 4.8 Contour plots of stress σ_{11} : (a) Case 1; (b) Case2 using elevated element order (c) Case 6 using fine mesh as reference	83
Figure 4.9 Accuracy of various IGA models: (a) elevated element order; (b) mesh refinement.....	84
Figure 4.10 A hollow cylinder with internal pressure: (a) schematic of the model; (b) quarter model for numerical analyses	85
Figure 4.11 Mises stress distribution of case E4 and reference FEM case.....	86
Figure 4.12 Computational performance of IGABrick elements in modeling a hollow cylinder using various modeling schemes	87
Figure 4.13 OPS-ITO framework comprising model setup, IGA analysis, and optimization module	94
Figure 4.14 Graphical user interface of OPS-ITO platform	97

Figure 4.15 Illustration of elemental boundary indexes (boxed): (a) 2D model; (b) 3D model	97
Figure 4.16 Optimization of a plane stress quarter annulus example: (a) model schematic; (b) optimized topology; (c) iteration history	100
Figure 4.17 Optimization case of L-shape beam for light weight with material strength constraints: (a) using Mises stress constraint; (b) using Drucker-Prager stress constraint.....	102
Figure 4.18 Optimization of 3D cantilever beam: (a) a rectangular beam; (b) a curved beam.....	104
Figure 5.1 Illustrations of multi-pattern design that balances preferences of performance and form	109
Figure 5.2 Schematic illustration of RUC and grouping schemes using a 3×2 design domain.....	112
Figure 5.3 Compliance distribution and differences of a 4×1 domain: (a) elemental distribution, (b) compliance difference between unit-cells (N·mm), and two illustrative examples of grouping calculation.....	115
Figure 5.4 The positive correlation between the optimized compliance and accumulative unit-cell difference S_{ij} for (a) the 4×1 beam cases (b) the 6×1 building cases subjected to lateral load	116
Figure 5.5 Workflow of k-means clustering for automated grouping of unit cells	118
Figure 5.6 Topology optimization flowchart for multi-pattern design.....	119
Figure 5.7 Multi-pattern topology optimization applied for free-form and fully periodic models.....	122
Figure 5.8 Multi-pattern topology optimization of a beam using 3×1 unit-cells: (a) NoV=1 (fully periodic design); (c) NoV= 3 (free-form design); (c) NoV=2, (d) Histories of compliance values.....	125
Figure 5.9 Topology optimization results of the beam model using 4×1 unit-cells.....	126
Figure 5.10 Multi-pattern topology load optimization of a plane frame subjected to horizontal load	127
Figure 5.11 Topology optimization-based structural design of a building frame model with different NoV.....	130
Figure 5.12 Compliance values of optimized models with various NoVs	130
Figure 5.13 Optimized configurations of 8×4 simply supported beam with zero-force removal.....	132

Symbols

\mathbf{K}	Stiffness matrix
\mathbf{D}	Elastic matrix
\mathbf{B}	Strain-displacement matrix
\mathbf{U}	Displacement vector
$\boldsymbol{\sigma}$	Stress tensor
$\boldsymbol{\varepsilon}$	Strain tensor
\mathbf{J}	Jacobian matrix
\mathbf{F}^{ext}	External load vector
\mathbf{F}^{sw}	Self-weight load vector
\mathbf{F}^{pre}	Pre-stressed load vector
$\mathbf{F}^{\text{pre}*}$	Anchorage pre-stressed load vector
T_{pre}	Prestress force
N_i^P	B-spline basis function
R_i^P	NURBS basis function
ρ_{min}	Minimum density
$\boldsymbol{\rho}$	Density vector
$\hat{\boldsymbol{\rho}}$	Tendon-concrete filtered density vector
$\tilde{\boldsymbol{\rho}}$	Projected (eroded/dilated) density vector
d_{ei}	Shortest distance between element and tendon
β_{HS}	Sharpness value
$\eta_{\text{ero}} / \eta_{\text{dil}}$	Erode-dilate projection threshold

φ_e	Relaxed stress
h_e	Epsilon-relaxation coefficient
G_{KS}	K-S aggregation function
ϕ	Adjoint function
λ	Adjoint vector
c_{ACS}	ACS factor for aggregation correction
$C_{i,j}$	Pattern scheme mapping matrix
S_{ij}	Compliance difference between unit-cells

Chapter 1 Introduction

firmitas, utilitas, venustas

----- *De Architectura*, by Vitruvius

1.1 Background

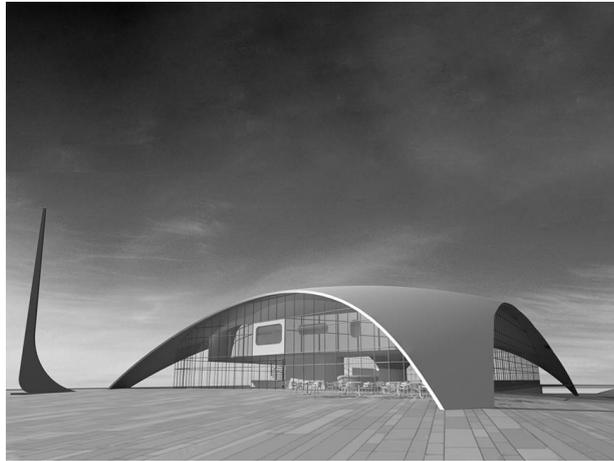
1.1.1 Trinity principles of architecture

The construction industry has been the cornerstone of social development from ancient times to the present. The imaginative landmarks have become national and regional calling cards, and architects go down in history for their masterpieces. Designers' desire to find beautiful and efficient architecture has led to innovative and mechanically sound form-finding approaches, such as Gaudi designing the Sagrada Familia Church [1] with its Gothic spire by observing the tension of chains under gravity, Swiss engineer Heinz Isler designed many thin concrete shell structures by observing inflatable and draped membranes [2]. It shows that the mechanical properties of a structure can guide the configuration design while bringing an organic aesthetic that obeys the laws of nature. However, relying on the episodic inspiration sparkling of architects is not enough to guide the structural design in the increasingly complex construction of modern buildings, thus requiring the evolution of design and construction techniques. As construction

technology advances, from bungalows to high-rise buildings, from one-way bridges to cross-sea bridges, civil construction is gaining more and more freedom and imagination of structural design. Although structural design has become increasingly free, the fundamental considerations of architecture design have not changed. As Vitruvius proposed more than two thousand years ago, the architectural trinity principle, i.e., *firmitas* (durability), *utilitas* (utility), and *venustas* (beauty) are still held in high esteem even in the modern construction industry [3]. In the recent era, the trinity principles are interpreted as Efficiency, Economy, and Elegance[4].



(a) Sagrada Familia Church



(b) Concrete shell structure

Figure 1.1 Architectures inspired by the load-carrying nature

The architectural trinity is the foundation for admiring a design. It is easy to notice the importance of the first two principles, but the architecture would be tedious to just say the foundations or the structural framing. Architectures are not solely admired for their impressive structural framing and extraordinary ventilation performance, but aesthetics are equally important. By the same token, a beautiful building is not admirable if lacking durability for easy collapse or lacking utility like overheating from too much glazing. However, the trinity principle of architecture does not form an efficient and unified framework in the practical

design stage. The architect is responsible for the utility and aesthetic performance, while the durability is ensured by the structural engineer based on experience and structural analysis. Therefore, satisfying the trinity requires time-consuming and laborious iterative design between architects and structural engineers [5]. One way for resolving this dilemma is using structural optimization in the conceptual design stage, which allows the architect to optimize the preliminary draft based on the structural simulation, thus resulting in a material-saving and safe design scheme that can then be checked and refined by the structural engineer for significantly reducing design costs.

1.1.2 The environmentally sustainable way to use concrete

As the population exploded and technology development in recent centuries, the demand for architecture reached unprecedented heights and brings severe pollution issues. Therefore, the demand for environmentally sustainable products is increasing in response to global carbon neutrality strategies. Compared with other manufacturing industries, the traditional construction industry suffers from severe pollution during the production of raw materials such as concrete and the construction process. Concrete is widely favored by the construction industry for its reliable load-carrying capacity, rich combination of raw material options, and low cost, but the same make concrete production and construction the third largest source of carbon emissions in the world, accounting for about 10% of global carbon dioxide emissions [6].

There are two major ways to reduce pollution levels in the concrete industry

via saving the amount of cement usage, which are optimizing the concrete composition and reducing concrete usage. The former is mainly from a material point of view, such as adding recycled aggregate to replace cement [7, 8]. Reducing cement composition tends to deteriorate its mechanical properties such as compressive strength, which may not be suitable for load-bearing components with higher consumables, thus the carbon footprint still has room for reduction. The latter, on the other hand, seeks more material-efficient configurations from a structural design perspective. Traditional structures have a large number of inefficient and even non-load-bearing members for the prescribing loading scenarios, and the redundancy of the structure leads to the overuse of concrete. The adoption of numerical structural optimization can automatically generate lighter structures to save concrete materials, answering the call for carbon neutrality.

Admittedly, structural design by removing inefficient materials often results in free-formed members that are difficult to manufacture, but the fluidity of concrete, before it sets, makes it extremely malleable and allows any possible structural form to be manufactured. The development of automated construction de-emphasizes the reliance on advanced in-site labor skills for the production of free-form concrete components and reduces construction expenses, which provides a construction foundation of numerical structural optimization in practical application.

1.1.3 The trend of automated construction

With the signing of the Paris Agreement [9], carbon neutrality has become a global target to ensure sustainable development. It is achieved by replacing fossil fuels with low-carbon energy, reforestation, energy saving, and emission reduction to offset the carbon dioxide or greenhouse gas emissions produced during

industrial and daily life activities, finally achieving “zero emission”. There are two general approaches to achieving carbon neutrality: the first is an administrative or economic approach that uses carbon offsetting mechanisms that produce carbon emissions equal to the carbon emissions reduced elsewhere, such as planting trees and purchasing renewable energy certificates. The second is technological improvement, using low or zero-carbon technologies such as (a) using renewable energy (e.g. wind and solar energy) to avoid carbon dioxide emissions into the atmosphere from burning fossil fuels, and (b) technological innovations in heavy-emissions industries that contribute to carbon neutrality in a way of reducing emissions.

In the era of carbon neutrality, the construction industry is eager to embrace the evolution of automation to reduce its gigantic level of carbon emission. In addition to the demand for emission reduction, the chronic problems of traditional construction methods such as labour aging and limitations in manual construction skills are also driving the construction industry toward automation to pursue faster, more economic, environmental-friendly, and safer workflow. Prefabrication and additive manufacturing as the representative technologies of automated construction address the demands on the construction side. In the current stage, prefabrication is known as its reliance on the mold precast and hence often focuses on manufacturing structures composed of repeated components to save the cost of producing molds. The advent of additive manufacturing has removed the limitation of mold precasting to allow for freer structural forms, which has shown great promise in futural construction. It should be noted that with the emergence of such advanced construction technologies, the topology optimization design methods find the corresponding manufacturing process and thus bring lightweight, high-

performance structures closer to practical applications, which can follow the trinity principles of architecture.

1.1.4 Considering the trinity principle in architecture design using topology optimization

Typically, architectural designs are based on the standard requirement and the experiences of designers, and the design should undergo a series of time-consuming iterative discussions between architects and structural engineers. As the construction industry embraces automated construction, the numerical optimization architecture design has likewise become a hot research topic that can provide inspiring configurations in the early design stage, thus saving the effort of manual design and enhancing structural performance.

Topology optimization has been recognized as a significant tool for the identification of the best concepts in the numerical optimization design of various industries such as aerospace, automotive, and biomechanics. Despite the topology optimization can provide designs with excellent structural performance and inspiring geometry, the organic and odd-shaped structures are often difficult and costly for manufacturing, hence it is limited to high-tech industries with high manufacturing budgets and hence prevents it from being brought to widespread use. Benefiting from the development of automated construction technologies such as prefabrication and additive manufacturing in recent decades, this disadvantage is gradually being overcome and a series of construction projects have emerged with adopting topology optimization, including Shanghai Zendai Himalaya Art Center [10], Qatar National Convention Center [10], Shenzhen CITIC Financial Center

[11], Concrete Pavilion [12] presented in **Figure 1.2**.



(a) Himalaya Art Center



(b) Qatar National Convention Center



(c) CITIC Financial Center



(d) Concrete Pavilion

Figure 1.2 Architectures designs using topology optimization

As introduced in **Section 1.1.1**, the trinity principle of architecture contributes significant guidance when it comes to construction design. In the context of topology optimization, the trinity principle has many connotations that can be explored, and a series of construction-oriented optimization models can be formed by means of mathematical expressions. Noted that durability is a broad concept and not exclusively the studies related to structural service life, it includes all the connotations of structural safety. Since safety concerns often have associated mechanical indices and systematic calculation methods, abstraction into mathematical models is often possible, so this classification has been extensively studied and is still in progress to approach realistic applications. Utility performance can be interpreted as the serviceability of the architecture, including ventilation, insulation, water supply, lightning protection, manufacturability, etc. Among them, thermal insulation and manufacturability are more related to the

architecture design, and the rest considerations are more likely to be expressed as pre-venting apertures and fixtures. The last is beauty, which is a subjective performance that is less likely to be quantified, but the aesthetic-oriented optimization model is still a significant direction to explore to enhance the aesthetic performance of the optimized structure, the valid ways might be to translate the concept of aesthetics into quantifiable metrics or provide a variety of reliable designs for users to choose from.

1.2 Scopes and objectives

To date, many researchers have carefully addressed their goal of bringing the topology optimization approach into the architectural design for responding to the call for carbon neutrality and tackling the bottlenecks of the current construction industry, which requires abundant effort in equipping the tool with the various requirements of the architecture trinity principle, such as construction material behavior, structural safety considerations, and aesthetics. In addition, the development of structural simulation methods in recent decades such as isogeometric analysis gives more precise solutions, which have great application potential. In this regard, it is meaningful to conduct this project to establish a platform, which adopts a new simulation method and the necessary design constraints on architectural design.

To sum up, this project capitalizes on:

(a) Proposing an isogeometric topology optimization approach of the typical construction components (post-tensioned concrete beam in this thesis) from the

perspective of material saving and safety to explore the capabilities of the application potential of topology optimization;

(b) Creating a new architectural design platform that facilitates concurrent engagement of architects and structural engineers on the building design stage. The aim of this platform is to generate highly customized designs using isogeometric topology optimization and it is open to including various considerations of structural design;

(c) Developing a methodology that provides a range of optimized designs for design diversity, which also combines structural performance with manufacturability;

The overall narrating flow follows a sequential order from the practical component design (*i.e.*, post-tensioned concrete beam) for demonstrating the safety-oriented application potential of the isogeometric topology optimization method to the establishment of an IGA-based simulation and optimization platform for enabling the existing and futural extension on various optimization models (*i.e.*, minimum compliance/volume model), and take a new multi-pattern optimization model as the first construction-oriented model extension.

1.3 Thesis overview

This thesis contains six chapters in total. Following **Chapter 1** as the introduction, **Chapter 2** is the literature review that summarizes the conception of construction-oriented design by addressing the key considerations of civil structure design using topology optimization, and the evolution of advanced construction

technologies such as prefabrication and concrete additive manufacturing to demonstrate the development trend of the futural construction industry. It also contains a review of existing topology optimization methods and platforms with common numerical problems and extensions based on isogeometric analysis.

Chapter 3 describes the design methodology of the post-tensioned concrete beam, including a simplified expression of prestress load effect with corresponding IGA-based load mapping approach, the Drucker-Prager stress constraint to address the unequal tensile-compressive strength property of concrete, the solution of numerical problems of stress-based topology optimization such as the stress-singularity phenomenon, the computational burden of abundant local stress constraint, and the instability of optimization process. In addition, a series of geometry control methods are also introduced to ensure an extinct, robust design. Several groups of examples demonstrate the effectiveness, and we explored the effect on different levels of component size and tensile-compressive strength ratio.

As a starting point for the OPS-ITO platform, **Chapter 4** illustrates the basic hierarchy of the OpenSEES with our newly developed Isogeometric elements for enabling 2D and 3D solid structure simulation. For construction-oriented topology optimization usage, an isogeometric topology optimization package is correspondingly developed with a user-friendly graphical interface.

Chapter 5 takes a step forward in that construction-oriented topology optimization requires a series of specialization constraints to approach real architectural design scenarios. Thus, we innovatively introduce multi-pattern control in this part as the first construction-oriented module, which can balance the manufacturability and structural performance of the final design.

Chapter 6 concludes the key contributions in this thesis and suggests the

next-stage research of the construction-oriented topology optimization to make this tool approach the practical construction industry.

Chapter 2 Literature review

2.1 Prefabrication and 3D printing in construction

With the overwhelming progress of global industrialization, construction activities are continuously catering to the enormous demands of the ever-growing human population from various aspects. Accordingly, undesired on-site construction dilemmas such as material waste, pollution, safety concern, labor aging shortage, and low construction efficiency force construction technologies to update and move to off-site construction plants to pursue more efficient and safer workflow[13]. The most significant advantage of off-site manufacturing is the high-level automation, which enables producing construction components in a precise and standard environment by customizing molds (prefabrication) and additive manufacturing. There are 7 major advantages of adopting automated manufacturing technologies, including the collaborating capability with advanced management tools such as Building Information Modeling (BIM), safer work environment, lower cost, quality control, time-saving, lower emission, and construction flexibility, which gives sufficient motivation of engineers for adoption.

2.1.1 Prefabrication

Prefabrication is a typical and promising off-site manufacturing technology,

which represents a new round of technological revolution in the construction industry and the direction of industrial upgrading. Benefiting from global policy support and the improvement of relevant standards, it quickly entered practical construction projects and continually expanded its market share, now it has become an indispensable part of the modern construction industry. Essentially, prefabrication is not a new building method, its history can be traced back to two hundred years ago (in 1830) by a London carpenter H. John Manning [14]. The core idea is to carry out standardized production through molds and cutting in factories with a better processing environment, rather than being limited by on-site manufacturing skills and equipment unavailability, which results in the compromise of low production quality and unnecessary waste.

2.1.2 3D construction printing

Unlike prefabrication has taken over a certain proportion of the construction market, 3d concrete printing is an emerging technology widely considered a significant trend in the next generation of construction [15-17]. It overcomes the inherent reliance on molds of prefabrication technology and thus provides a new solution to the manufacturing of odd-shaped structures. The 3D printing attempt appeared in the construction industry in the mid-1990s using contour crafting [18]. Later on, the deposition strategies were proposed in the first decade of the 21st century [19] with laboratory-grade demonstrative tests. The practical 3D printing construction starts in mid-2014, a concrete castle using Total Kustom's technology[20], and WinSun Decoration Design Engineering Co. demonstrated efficiency by printing ten houses in a day for under \$5,000 each [21] and a larger-scale 6-storey building with 60% material, 70% time cost, and 80% labor saved to

present the potential [22]. Not only for building printing but a pre-stress concrete bridge was also segmentally printed and installed in the Netherlands in 2015 [23] which can carry up to 2 tons load. Massive construction organizations are attracted by the bright potential of 3D printing and a series of fascinating 3D printing projects are emerging in recent decades, as illustrated in **Figure 2.1**.



Figure 2.1 A few representative projects manufactured by 3D printing. (a) the concrete castle by Rudenko in 2014, Minnesota, US, (b) concrete houses and a 6-storey building by WinSun in 2015, Shanghai, China, (c) Public restroom by WinSun in 2016, (d) Habitat for Humanity, a charitable project for low-income families, by Alquist in 2021, Virginia, US, (e) Shamballa village, the vision of the first community living entirely in 3D printed buildings, by WASP in 2018, Massa Lombarda, Italy, (f) an odd-shaped pedestrian bridge by MX3D in 2021, Amsterdam, the Netherlands, (g) Urban cabin for disaster relief and temporary functional housing, by DUS Architects in 2015, Amsterdam, the Netherlands, (h) Office of the future, the first 3D printed commercial building by Gensler and WinSun in 2016, Dubai, United Arab Emirates.

2.2 Topology optimization

Topology optimization is a mathematical structural design methodology via optimizing the material distribution of the designable domain, aiming at finding the solution with the optimal objective under prescribed constraints. Since Bendsøe and Kikuchi proposed the first approach called homogenization method, several design methods have been proposed one after another, which can be generally classified as two types of material description methods (e.g., basic homogenization method, Solid Isotropic Material with Penalty (SIMP) [24], and Bi-directional Evolutionary Structural Optimization (BESO) [25]) and boundary description methods (such as the Level-Set Method (LSM) [26]), and geometric description methods (e.g., Moving Morphable Units method (MMU) [27]). The topology optimization process involves iterative simulation, optimization, and state update, the typical workflow is generally illustrated in **Figure 2.2** and the general optimization model example can be written as below to pursue the optimal objective f (e.g. compliance) under the prescribed volume limit \bar{V} in the design domain Ω :

$$\begin{aligned} \min f \\ \text{s.t. } V \leq \bar{V} \end{aligned} \quad (2.1)$$

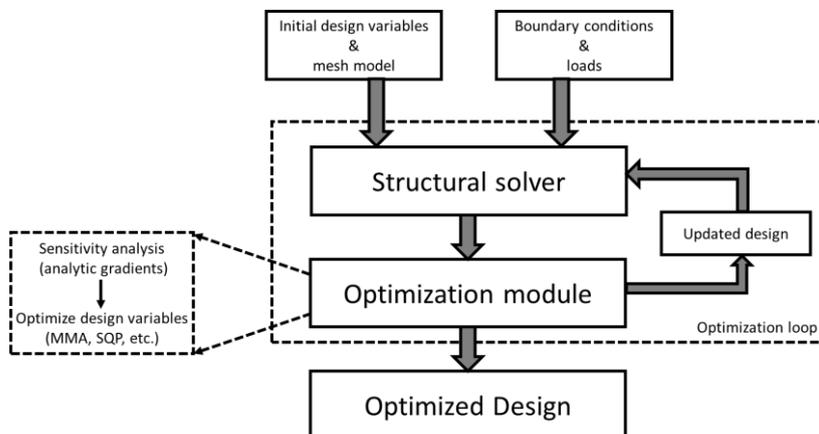


Figure 2.2 General workflow of the topology optimization process

2.2.1 Computational frameworks of topology optimization

As an advanced numerical structural design approach, topology optimization has undergone plenty of research [5, 28-30] and industrial applications [11, 31] in multiple fields. Establishing a standard and ease-to-extension framework is essential for delivering the topology optimization approach in both research and industrial use. In the recent decade, several open-source topology optimization frameworks have been established for engineering use such as OpenMDAO [32-34], PolyTop [35, 36], and TopOpt [37]. In addition, the commercial software Altair OptiStruct [38], Ameba [39], Abaqus Tosca[40], Ansys Discovery[41], Comsol [42], and Fusion 360 [43] are also extending topology optimization methods to more structural design scenarios. These growing platforms ensure various demands of structural optimization, which focus on the multi-disciplinary design based on various stand-alone programs. However, there are abundant types of construction and service requirements in civil engineering that require the adoption of corresponding material models, element classes, and especially corresponding considerations for architectural design. Thus, it is beneficial and worthwhile to build up a topology optimization module under professional civil engineering simulation software with equipping various construction-oriented design constraints.

2.2.2 Common numerical problems and corresponding solutions

Among all topology optimization approaches, the SIMP method is widely

adopted as it is intuitive and easy-to-extension, therefore it is the key optimization method in this thesis. As a density-based method, the SIMP model of the Eq.(2.2) is discrete as:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & f(\boldsymbol{\rho}) \\ \text{s.t.} \quad & V = \sum_{i=1}^N \rho_i V_i \leq \bar{V} \\ & 0 \leq \rho_i \leq 1 \end{aligned} \quad (2.2)$$

where $\boldsymbol{\rho}$ is the design variable vector of elemental density. As a numerical design approach, the SIMP method faces several inherent numerical problems such as the checkerboard phenomenon, mesh dependency, and so on. After decades of development, the numerical problems of topology optimization have been alleviated or avoided with the help of a series of new alternations, which are reviewed in this section.

2.2.2.1 Intermediate density variables

As presented in Eq.(2.2), the density variables are continuous between zero to one. However, only integer variables of 0 and 1 have the physical meaning of voids and solid material, the intermediate density should be avoided since it is not manufacturable. Penalization is the most widely accepted approach, and the general way is adding a penalty exponential coefficient on the density variables to punish the elemental stiffness of those with intermediate density [44]. In addition, several projection filtering methods are also proposed to further enhance the 0-1 solution of density variables [45], including dilation-erosion [46], threshold projection [47], and multi-phase projection [48].

2.2.2.2 Checkerboard and mesh dependency

It is well-known that the introduction of more holes without changing the

structural volume will computationally increase the structural performance index [49], which causes the optimization process converges to a configuration in which solid material and voids alternative, called the checkerboard phenomenon. Accordingly, the number of holes is changed when adopting different mesh sizes, which can be explained as the non-existence of the general optimal solution and thus results in a mesh-dependent configuration, which is termed mesh dependency.

The checkerboard-style configuration represents the “optimal topology” in the numerical perspective but is not manufacturable, it is believed that the erroneous FE-modeling is the key reason[50], several schemes are proposed to avoid the appearance of the checkerboard pattern such as smoothing, high-order elements, and filter. Smoothing is a preliminary image-processing way that ignores the underlying problems and is not widely applied. Adopting high-order elements is a relatively valid way, the checkerboard phenomenon is mostly avoided by using 8 or 9-node finite elements if the penalization is weak [51] and the computational cost is correspondingly increased due to the adoption of high-order elements. The last way is introducing filtering techniques by efficiently weighting the density variables of neighboring elements, it is widely adopted and the mesh-independent filter can also extensively avoid the mesh dependency [52].

2.2.2.3 Local minima

As topology optimization is usually a non-convex problem, is clear that many local minima solutions appeared in literature. Mathematically, the pursuit of the optimal solution to the non-convex problem is still a big challenge, thus the value of convexifying the problems and generating reproducible designs are recognized and investigated. Nevertheless, the final design can be drastically affected by even small variations in initial optimization parameters such as move limits of

optimization algorithm, geometry and mesh scheme of design domains, and filter parameters. The undesirable nature of the objective function, such as flatness, is an important factor contributing to this problem, but the limitations of the optimization algorithm are more responsible. As a remediation, the algorithm convergence check of the non-convex problem only ensures the solution to a nearby stationary point (not necessarily be similar to the global optimum).

2.2.2.4 Numerical problems of stress-constrained topology optimization

Most of the research on topology optimization focus on the minimization of compliance with other global responses such as frequency, while stress is another significant issue to ensure structural safety. Nevertheless, stress is relatively less considered because of three significant challenges [53] termed the stress singularity phenomenon [54-56], the computational burden of the abundant local stress constraints, and the convergence problem from the strong non-linear property of stress response.

The issue of stress singularity is first noticed when optimizing the truss system, the n-dimensional design space decreases into a subspace with a lower dimension, and the feasible region becomes discontinued [57]. The programming algorithms are unable at searching singular regions the global optimum is located and hence converge to the local optimal solution. As a remediation, relaxation methods, such as ϵ -relaxation [55] and smooth envelope functions [58], are proposed to eliminate those discontinued and lower-dimensional regions and later on applied to the stress-constrained problems [59-61].

Stress constraints are a huge group of local constraints that are considered at all material points, and the computational cost is unaffordable for practical applications. The widely adopted resolution is finding a single representative stress

constraint that is close to the maximum stress value to replace local stress constraints, such as p-norm [62] and the Kreisselmeier-Steinhauser (K-S) [63] aggregation function. By aggregating the huge number of local stress constraints into a global aggregated stress measurement, the computational cost is effectively controlled, but the approximation of the maximum stress has two drawbacks. The first is the dilemma between the approximation error and convergence problem, the error will reduce when the aggregation degree is strengthened and correspondingly worsen the smoothness of the aggregation function, hence resulting in a worse convergence performance, and vice versa. The aggregation degree should be carefully selected. The second is the approximation of the maximum stress value, which is unable to adequately control the local stress behavior.

The highly non-linear stress behavior brings dramatic reliance on the change of structural configuration, which is exasperated in the regions with high-stress gradients. Hence, the optimization formulations and corresponding solution algorithms should be numerically consistent to alleviate the convergence problem.

2.2.3 Isogeometric topology optimization

Among all iterative processes in **Figure 2.2**, solving the structural response is one of the most essential steps throughout the entire iteration, the accuracy and efficiency of simulation have a profound impact on the quality of the optimization. As it is common that accuracy and efficiency are typically contradictory. In a traditional Finite Element workflow, the shape function is described using the polynomial function. There is an inherent error between the spline-based geometric model and the polynomial-based mesh model thus cannot capture the exact structural geometry. And accordingly, the mesh model cannot be simply discrete

from the geometric model to build an analysis-suitable model, a great extent of time is consumed in the pre-processing step. Commonly, only 20% of the overall time is consumed for the analysis model solving, the rest 20% and 60% are sacrificed in the mesh generation and the generation of simulation-specific geometry model [64], which is desirable to cut down by integrating CAD and CAE. In addition, the piece-wise polynomial shape function suffers from the lower continuity (C^0) on the boundary of the element. These two drawbacks should be alleviated by achieving a high quality of the finite element mesh, which is less efficient. Adopting a new simulation method is a valid way to improve the simulation quality of both efficiency and accuracy.

2.2.3.1 Isogeometric analysis

Isogeometric analysis [64] is inspired by Computer-Aided Design (CAD) for employing spline curves as the shape function of mesh models to conduct Computer-Aided Engineering (CAE) [65]. It can be viewed as an extension and generalization of the traditional finite element method for solving the structural response. The core concept of IGA is adopting the various types of spline basis functions, the most widely used forms of geometric description, to replace the conventional polynomial shape function in the finite element framework. The most straightforward merit is the consistent geometry-mesh description and thus eliminates the inevitable geometric error, which is so-called the integrated CAD-CAE approach. In IGA, the B-spline function is the basic description form [64, 65]. Despite the B-spline function being flexible in describing many geometries, it is still unable to exactly express some specific and important shapes like circles and ellipsoids. To enhance the generalizability, the Non-Uniform Rational B-Spline (NURBS) is introduced by adding a positive weight to each B-spline basis function

[66] and becomes the most widely adopted form. Afterward, other spline functions have been employed in IGA, like the T-Spline curve [67], to give more beneficial mathematical properties such as the variation diminishing to tackle the local oscillation and the convex hull issues [68].

2.2.3.2 Isogeometric topology optimization

There are two branches in the development of Isogeometric Topology Optimization (ITO), focusing on the new curve description of IGA and the isogeometric cooperation with the existing topology description for optimization. From the curve description perspective, NURBS is the most widely-used curve form for ITO development [69-71]. In addition to this, a series of earlier works [72] considered trimmed spline surfaces to use trimming information in solving the trimmed surface responses and deriving the corresponding sensitivity in topology optimization by adopting the physical coordinates of control points as the design variables.

Correspondingly, developing the ITO design methods on the classic topology optimization framework is the major research field, which can be further classified into two main categories of material description model and boundary description method, which covers the major approaches of typical topology optimization. Compared to the boundary description method, the formulation of the material description model is more intuitive through directly expressing the existence of elemental material, and hence independent design variables accordingly bring several numerical problems during the topology transformation, like the checkerboard phenomenon, the zig-zag or wavy boundary, and the mesh-dependency, which should be alleviated or removed using several modificative methods [73]. With cooperating with IGA, the checkerboard phenomenon, and

mesh dependency are alleviated from the local supportive property of the spline curve, which is similar to the effect of the classical density filter. While the boundary description methods describe implicit iso-contour (e.g., LSM) or explicit virtual morphable units (e.g., MMU) to define the topology.

2.3 Key considerations of concrete structure design using topology optimization

The topology optimization method has not received sufficient attention in the civil engineering field until recent decades, especially for structure optimization with concrete-type material, which is a valid lever within the cementitious construction supply chain to cut off carbon emissions. The adoption of topology optimization is also helpful for the civil designer to explore the limit of modern architectures such as high-rise building and long-span bridge, which is currently blank in conventional construction codes. Unlike the topology optimization approaches are recently been adopted, the potential of concrete structure optimization has been recognized over two centuries, which starts in 1866, Culmann considered minimizing cement usage via graphical statics. Afterward, the concept of mathematical optimum finding was considered since Michell investigated the economic limit of frame structures, which were described by an orthogonal system composed of a group of curves to pursue minimum compliance when carrying maximum tension and compression loads.

2.3.1 Durability performance

It should be noted that the “Durability” in the architecture trinity principle is a generalized concept, which includes all the structural performances related to structural safety. The safety concern is the most important issue that has undergone a vast number of investigations in topology optimization, including concrete stress constraints [59, 74-89], concrete damage modeling [90-92], multi-material modeling [93-99], strut and tie modeling [28, 100-111], truss-continuum topology optimization [96, 97, 112], multi-load problems [113-117], prestressed structures [29, 118-120] and thermal-structural problems [121, 122]. Among all, the research on stress constraints and prestressed structures are highly related to the scope of this thesis, hence will be detailed in the following paragraphs.

2.3.1.1 Stress constraint

As concrete material has different strength levels on tension and compression, the conventional Von-Mises stress criterion is improper to be adopted, hence Drucker-Prager criterion [74] is mostly consulted in this scenario. Luo and Kang [79] first investigated the D-P constrained topology optimization model to minimize the material volume under local stress constraints, it demonstrates that the unequal strength criterion has a significant effect on the material distribution. In the same year, they also proposed a three-phase model [80] to enable the optimization of steel-concrete composites. Furthermore, the shrinkage effect and strength failure are considered [83] by introducing an additional design-dependent load vector. Bruggi and Duysinx approximated the unilateral stress field to replace the symmetric material in a typical minimum compliance model [84] in 2013, Jewett and Carstensen then optimized and conducted the test of plain concrete structure in 2019 [86]. From the material model perspective, the elastoplastic model is adopted with the Drucker-Prager criterion by Bogomolny and Amir [87]

and highlighted that the mesh refinement is important for concrete-steel structures to ensure the simulation accuracy, as the volume of steel is often only a very small part of the total. Later on, Amir adopted a rate-independent elastoplastic model and proposed a single global plastic strain constraint to enable the control of the violated local stress [88]. Pastore introduced risk factors to quantify the distance between the current principal stresses and the prescribed stress limits [89].

2.3.1.2 Prestressed structures

To date, the optimization of prestressed structure gradually attracts research interests but has still not been investigated in full depth from the topology optimization perspective. The intention of prestressed structure optimization starts in the 1980s, such as the optimal tendon shape of the multi-span prestressed bridge [123] by Kirsch, Quiroga, and Arroyo [124] optimized the size and position of the tendon of a fixed bridge deck to balance the transverse loading, Lounis and Cohn [125] developed a single objective model from the multi-objective model by transforming other objectives into a series of constraints and further implemented the optimization of prestressed structures with a series of considerations such as the ultimate and serviceability limit states [126]. Han et al. [127, 128] developed a DCOC-based optimization model to minimize the cost expense of the multi-span beams and T-beams which are partially prestressed. The first research on the topology optimization of prestressed structure appeared in 2018, Amir and Shakour [118] proposed a simultaneous method to design the concrete topology and tendon shape, the effect of prestress tendon is simplified as a group of equivalent prestress load vector on the linear tendon segments. To ensure the tendon-covered configuration, a tendon-concrete filter is correspondingly introduced, and hence the prestress load can be correctly transferred to the body of the concrete beam. The

optimized structure is later converted into a 3D model using Fusion 360 and manufactured [29] using 3D printing and conducted corresponding experiments to test the structural performance. Zhang et al [129] used isogeometric analysis to extend the above work into a more continued formulation, updated the linear segmental tendon form with the continued NURBS curve to enhance the accuracy of equivalent load, and considered the unequal stress limit using Drucker-Prager criterion.

2.3.2 Utility performance

From the traditional perspective of architecture, utility performance means the functionality of usage, such as the industrial building, domestic building, etc., and the building services performance, such as heating, ventilation, air conditioning, water supply, lightning prevention, manufacturability, expense during construction and maintenance, and so on. As a structural design approach, topology optimization mainly focuses on the performance of the manufacturability from the utility perspective at the current stage, which can be interpreted as the considerations of construction techniques [130-141].

Before the emergence of 3D printing technology, manufacturability has two major connotations of simplicity and regularity. The simplicity indicates the structure is made of few components to reduce the difficulty of installation, and lower resolution of components is preferred to avoid relatively expensive high-end manufacturing technologies. Regularity means the components appear multiple times, which reduces the expense of mold making limiting the number of molds. The most intuitive way to ensure structural simplicity is by increasing the radius of the density filter. On the one hand, it avoids optimized structures from tiny

components that require higher resolution of manufacturing. On the other hand, thicker components provide greater load-support capacity, hence reducing the demand for load-bearing members and further simplifying the final configuration. As for regularity, it refers to the generation of periodic structures using topology optimization. Past researchers have explored various periodic styles using affine transformation, such as translational offset [142], pattern gradation [5], and rotation-reflection combination [143].

The introduction of 3D construction printing technology provides a new dimension to manufacturability, that is printability of the optimized structure. The major challenge of 3D printing is the overhang component has no load-bearing capacity before solidification and collapses under gravity thus leading to the failure of the printing process. The first solution is to set up removable supports, Vantghem [29] used the sand hill as the temporary support during the 3D printing of the bridge. The second solution is simply to consider the collapse issue during the topology optimization stage, which is termed a self-support structural design [144]. Morgan et al. [145] proposed an approach to consider the optimum printing orientation to minimize the volume of extra supports. Later on, Hu et al. developed an orientation-based shape optimization method to obtain the self-support structure by introducing the global rigid energy as the objective to further reduce the additional supports. Johnson and Gaynor [146] applied the density projection method to achieve a self-supported structure by only preserving those that should be projected to solid and satisfy overhang constraints, it avoids additional explicit geometric constraints but brings difficulty to convergence. Qian [147] proposed the density gradient-based integral method to form overhang constraints with geometrical meaning, but cannot avoid V-shape on the boundary. Mass and Amir

[148] considered preserving the overhang features of the final design and hence developed the computationally efficient Virtual Skeleton Method. Allaire et al [139] avoided overhang features by combining the geometric and mechanical constraints, but computationally inefficient.

2.3.3 Aesthetic performance

According to the aforementioned trinity principle of architecture, durability, and utility have been considered by structural engineers in various service scenarios to ensure the safety and functionality of the structure. Likewise, aesthetic performance is equally significant for designing a successful architecture, which should carefully explore its underlying principles for truly achieving automated architectural design. From the structural optimization perspective, unlike quantifiable durability and utility performance, aesthetics is a rather subjective consideration that is hard to directly express in the optimization model straightforwardly [5, 149-151]. Regularity is a typical form of aesthetics in architectural design, which is mathematically practical to be considered in the optimization model by adopting periodic geometrical constraint, which has been introduced in **Section 2.3.2**. Accordingly, it is more realistic to use alternative ways to transfer the task of assessing aesthetic performance to the users. Xie et, al. [152] proposed diverse designs by penalizing precedent configurations, which are capable of providing various designs under the prescribed conditions if the current design is unsatisfactory. Shannon et, al. [153] explored the possibility of converting the un-quantifiable aesthetic performance to a qualitative prediction and explaining the aesthetic responses of users when adopting the concept of Unity-in-Variety, which balances the unity of the entire design and interest-stimulating variety. In

conclusion, the consideration of structural aesthetics in automated design remains open and such persuasion is helpful for integrating the duty of structural engineers and architects.

2.4 Concluding remarks

This chapter of the literature review, many issues and the evolution history of advanced construction technology, topology optimization process, and some key points for structural engineering applications are discussed. The major remarks are concluded in this section, which established the research basis of the construction-oriented topology optimization and forms this thesis.

1. Prefabrication and 3D printing technology enables advanced construction by adopting off-site manufacturing, which has significant potential for collaborating with topology optimization to construct efficient, lightweight, and green structures;
2. So far, topology optimization has established several FE-based computational platforms for multi-disciplinary usage, and corresponding numerical problems of topology optimization methods are gradually alleviated or eliminated by relentless research. From the simulation perspective, isogeometric analysis as an extension of FEM is widely recognized for its ideal computational accuracy and continuity, hence has great application prospects;
3. In recent decades, topology optimization techniques are gradually entering the field of structural engineering design, the corresponding applications can be generally classified by the trinity principle of architecture, that is durability, utility, and beauty;
4. “Durability” is implied as a generalized concept of structural safety, which is

the major research field for adopting topology optimization. Among all, the prestress structure and stress constraint are considered in this thesis. “Utility” is interpreted as a building services-oriented performance, manufacturability is the major consideration of topology optimization in this category. “Beauty” is the only subjective performance among the trinity principle, the development can help users to generate diverse and aesthetic structures.

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam

3.1 Introduction

The state of art construction industry faces a series of dilemmas, such as labor ages, low efficiency, pollution, safety issue, and material waste. The development of the automation industry brings more efficient manufacturing technology, which requires less labor that working in a cleaner and safer workplace. In the construction industry, new construction technologies such as prefabrication and additive manufacturing are sprouting from the development of the automation industry. With such new technologies keep developing, the main issues are how to ensure the strength of the structure and how to build a design platform that addresses construction requirements. Recall that the question of why iconic architecture is so rare lies primarily in the new challenges they pose to aesthetics, structural reliability, and economic property. Therefore, the futural architectural design approach requires designers to propose new configurations to explore the unknown beyond traditional forms. Topology optimization methods are thus

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam uniquely valuable in architectural design for seeking optimal configurations under given constraints.

In this section, the major purpose is to demonstrate and explore the potential of isogeometric topology optimization in the construction industry and approaching futural construction design. To achieve this goal, a typical construction component, the post-tensioned concrete beam, is picked to develop an optimization method under various structural reliability considerations.

Unlike the finite element-based method (FEM) widely adopted in analyzing the stress distribution [154, 155], the Isogeometric Analysis (IGA) [156] is an emerging approach. As pointed out in the milestone work of Hughes et al.(2005), IGA was an extension of FEM and was inspired by the computer-aided design (CAD) methodology, featuring an exact geometric description from CAD by adopting Non-uniform Rational B-Splines (NURBS) as its basis functions. The IGA method enables the identical description of the analysis model to the geometry model during optimization, as they both use the same NURBS parameters and the adaptive mesh for FEM could be avoided. The typical advantage of IGA-based structural topology optimization has been discussed by [70]. Along with IGA, the aforementioned density-based SIMP method can be incorporated into the framework, in which a vision of construction-oriented topology optimization is proposed corresponding to the above-manifested needs. To implement topology optimization in a wider scope of realistic construction, the generated design of optimal topology should also be construction-friendly. Using NURBS based IGA model, the structural shape can be naturally compatible with the design input, which enables potential industrialized construction approaches such as factory prefabrication.

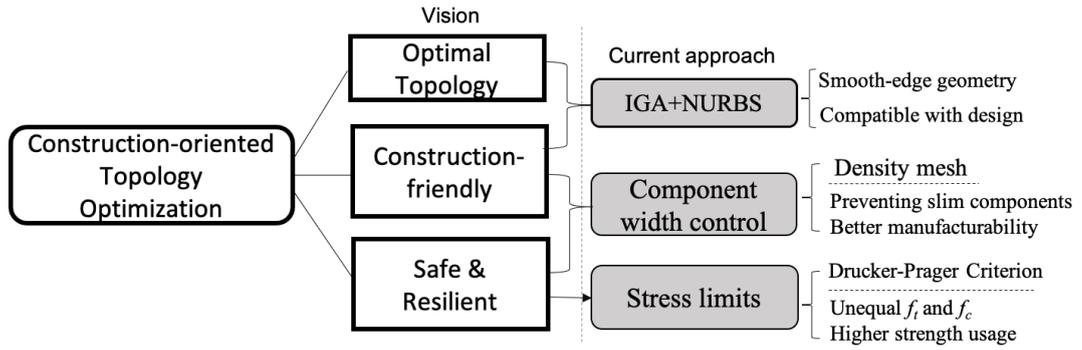


Figure 3.1 A brief vision of the futural construction design for the post-tensioned concrete beam

Moreover, the structural components of the optimal design should fulfil the safety and resilience requirements in a performance-based design domain. Ideally, the stresses of structural components should be maintained underneath specified limits in various loading scenarios [157]. It is also equally important that these components shall be of limited variation and easy-to-manufacture sections. Thus, geometry control, such as minimum width and tendon cover (discussed in this chapter), should be considered in the construction-oriented topology optimization framework.

In this chapter, a prestressed concrete beam with unequal tensile and compressive strengths and a tendon is used to demonstrate the vision of the construction-oriented topology optimization. The Drucker-Prager criterion is adopted in the optimization procedure to constrain the material stress within the realistic concrete strengths of low tensile strength and relatively high compressive strength. This is implemented with an IGA-SIMP topology optimization framework, where a NURBS tendon curve representing equivalent prestress load is achieved using a simplified load mapping process from the tendon to the concrete (**Section 3.3**). A NURBS-based minimum-width control is applied to ensure manufacturability and prevent slim components for better resilience performance. The formulation and sensitivity analysis of the outlined optimization problem is

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam discussed, followed by an investigation of various effects on the optimized topology, including the tensile-compressive strength ratio and the controlled component widths.

3.2 Isogeometric Analysis and NURBS curve

Isogeometric analysis (IGA) is a new simulation approach, which unifies Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) by adopting various types of spline functions (NURBS [65], T-spline [67], etc.) for describing the geometric models and as the shape function of mesh models to build up a seamless CAD-CAE system. Among all types of spline functions, NURBS excels in the generalizability of curve representation and thus becomes the mainstream modeling description method.

3.2.1 NURBS curve

NURBS is a generalized format of B-spline, basic B-spline curve is determined by a group of control points \mathbf{P}_i , as below:

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{P}_i \quad (3.1)$$

where $N_{i,p}(\xi)$ is the i -th of a total of n p -th order B-spline basis function at the knot location ξ , the basis function could be defined in Cox-de Boor recursive format [66]:

$$N_{i,0}(\xi) = \begin{cases} 1, & \xi_i \leq \xi \leq \xi_{i+1} \\ 0, & \text{else} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (3.2)$$

$$(i = 1, 2, \dots, n + p + 1)$$

where the given knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ is a monotonically increasing scalar array that defined parametric space to describe the B-spline curve. Noted that if the knot vector repeats $p+1$ times at the beginning and end, the described curve is not connected at both ends, which is also called an open knot vector. Take an example of a simple 2-order curve with the open knot vector $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$, 4 control points (right) have their basis functions (left) to contribute weight value on arbitrary parametrical coordinates for describing the curve.

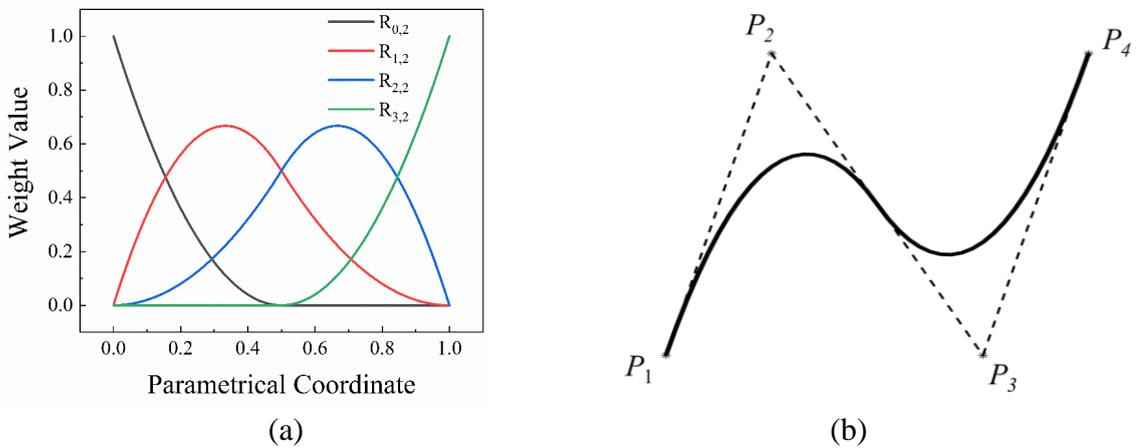


Figure 3.2 Example of a 2-order B-spline curve, (a) the basis functions of all control points, (b) control points and the B-spline curve

NURBS is the generalized form of B-spline, with non-uniform and rational properties. It can accurately describe various types of curves such as rectangular, circle, and elliptical by adding a weighted index w_i of control points \mathbf{P}_i , which can be described as:

$$C(\xi) = \frac{\sum_{i=1}^n N_{i,p}(\xi) w_i \mathbf{P}_i}{\sum_{j=1}^n N_{j,p}(\xi) w_j} = \sum_{i=1}^n R_{i,p}(\xi) \mathbf{P}_i \quad (3.3)$$

to simplify the expression, the i -th and p -order NURBS basis function is denoted as $R_{i,p}(\xi)$. NURBS inherits 3 important properties from the B-spline curve: (a) non-negativity $R_{i,p}(\xi) \geq 0$, (b) local supportive $R_{i,p}(\xi) = 0$ if $\xi \notin [\xi_i, \xi_{i+p+1}]$ and (c) smoothness, the interval $\xi \in [\xi_i, \xi_{i+p+1}]$ is continuously derivable of order $p-1$ and order $p-k$ at the knots, where k is the repeat times of the knot.

3.2.2 Curve refinement

CAE system requires a well-refined mesh model to ensure a continuous and accurate solution. NURBS as the shape function of isogeometric has a set of embedded curve refinement methods for knot insertion and order elevation [158] that can preserve the original geometric model.

Firstly, the knot insertion method of B-spline is called h-refinement. The insertion of m knots extends the former knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ into $\bar{\Xi} = \{\bar{\xi}_1 = \xi_1, \bar{\xi}_2, \dots, \bar{\xi}_{m+n+p+1} = \xi_{n+p+1}\}$, the new basis function still follows Eq. (3.2), and the control points $\bar{\mathbf{P}} = \{\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2, \dots, \bar{\mathbf{P}}_{m+n}\}^T$ are updated as below:

$$\bar{\mathbf{P}}_i = \alpha_i \mathbf{P}_i + (1 - \alpha_i) \mathbf{P}_{i-1}$$

$$\alpha_i = \begin{cases} 1, & i \leq k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & i \geq k + 1. \end{cases} \quad (3.4)$$

The knot insertion process is analogous to the mesh regeneration process of the classical finite element method, which has the advantage of not requiring a revisit

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam to the geometry model and maintaining the original curve shape for promising geometric accuracy. Noted that the insertion of the existing knot will decrease the curve continuity at the corresponding location and changes the curve shape, which has no analogous function in the finite element method.

Secondly, the mechanism of order elevation is p-refinement. During order elevation, the polynomial order of spline basis function s is thus raised without changing the curve shape. To preserve original C^{p-k} continuity across the curve boundaries, the repeat knot number k at the existing knots should raise with p increasing. The next step is removing excess knots to merge the segments and achieve order elevation of the B-spline curve. There are various efficient algorithms for saving the computational costs of order elevation, which is referred from [66] and detail omitted here since out of the scope of this thesis.

To rationalize the B-spline curve into NURBS, the basis functions are firstly derived as:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{j=1}^n N_{j,p}(\xi)w_j} \quad (3.5)$$

$$C(\xi) = \sum_{i=1}^n R_{i,p}(\xi) \frac{\mathbf{P}_i}{w_i}$$

3.2.3 Isogeometric structural analysis and IGA-SIMP method

Compared to the classical finite element method, isogeometric analysis has a series of analogous, which list as below:

Table 3.1 Analogous between the finite element method and isogeometric analysis with NURBS

	Isogeometric analysis	Finite element method
Shape function	NURBS basis	Polynomial basis
Element Freedom	Control points	Node points
Mesh refinement	Knot insertion & order elevation	re-mesh
Element-wise continuity	Exact geometry C^{p-1} (local supportive)	revisit geometry (approximate) C^0
Domain type	Patches	Sub-domains
Isoparametric element		Support
Patch tests		Pass

Under the isogeometric analysis concept, the isoparametric NURBS patches can express exact rigid motion and deformation[65]. For structural analysis, the discrete formulation of governing equation remains in the classical form:

$$\mathbf{K}\mathbf{U} = \mathbf{F} \quad (3.6)$$

where \mathbf{K} is the global stiffness matrix, and \mathbf{U} and \mathbf{F} are the global displacement and load vectors. The key point is that the shape function changed as the NURBS basis function from the polynomial basis function of the classical finite element method.

When adopting the IGA concept, the design variables can logically extend as the pseudo density of IGA control points. For the planar problem, the shape function using NURBS is employed as:

$$\mathbf{x}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q}(\xi, \eta) \mathbf{x}_{i,j} \quad (3.7)$$

where $\mathbf{x}_{i,j}$ denotes the variable vectors such as coordinate, displacement, or density value of the control point in the SIMP method.

The Solid Isotropic Material with Penalty (SIMP) method is a widely used topology optimization approach, which introduces pseudo-elemental density as the design variable to describe any possible material distribution. Accordingly, the

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam
 elemental stiffness matrix of planar problem \mathbf{K}_e and elastic matrix \mathbf{D}_e can be
 described as:

$$\begin{aligned}\mathbf{K}_e &= \int_{\Omega_e} \mathbf{B}_e^T \mathbf{D}_e \mathbf{B}_e d\Omega_e \\ \mathbf{D}_e &= (\rho_e^p + \rho_{\min}) \mathbf{D}_0\end{aligned}\quad (3.8)$$

where \mathbf{B}_e is the strain-displacement matrix, \mathbf{D}_0 is the solid elastic matrix, and p is
 the penalty coefficient to penalize the intermediate density that is generally set as 3
 [159]. ρ_{\min} is the minimum density, generally set as 0.001 for avoiding
 computational instability when the elemental density ρ_e close to zero. Under the
 IGA concept, the elemental density is approximately set as the density value in the
 element centroid as below:

$$\rho_e = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q} (\xi_e^c, \eta_e^c) \rho_{i,j} \quad (3.9)$$

ξ_e^c, η_e^c are the parametric coordinates of the e -th element center. For the plane
 stress problem, the stress vector of point a is expressed as below:

$$\boldsymbol{\sigma}_a = \rho_e^s \mathbf{D}_0 \mathbf{B}_a \mathbf{U}_e \quad (3.10)$$

where \mathbf{B}_a is the strain-displacement matrix of point a ; \mathbf{U}_e is the displacement of
 the element control point; s is generally set as 0.5 [160] to penalize the
 intermediate density value.

3.3 Modeling post-tensioned concrete beams using IGA

A post-tensioned concrete beam is optimized using the proposed construction-
 oriented topology optimization framework as a typical structure comprising
 separate components of different materials. Based on the equivalent load method

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam [161], a set of equivalent loads can replace the prestressing effect. A new load transfer formulation of the continuous equivalent prestress load is developed for the IGA-SIMP framework, enabling the prestressed concrete beam's optimization procedure.

3.3.1 IGA for prestressed beam

For the concrete part of a simply supported prestressed concrete beam, it is subjected to various types of loads, which include the external loads such as vehicles and structural self-weight as well as the equivalent loads representing the action of the prestressed tendon. The displacement caused by the equivalent prestresses \mathbf{U}_{pre} will balance the displacement caused by external load and self-weight load \mathbf{U}_{ext} , \mathbf{U}_{sw} . Similar to the finite element method, the equilibrium of the prestressed beam can be formulated as below:

$$\mathbf{K}\mathbf{U} = \mathbf{F}^{\text{ext}} + \mathbf{F}^{\text{sw}} + \mathbf{F}^{\text{pre}} + \mathbf{F}^{\text{pre}^*} \quad (3.11)$$

where \mathbf{F}^{ext} and \mathbf{F}^{sw} represent the external force vector and the self-weight force vector, respectively. The tendon geometry is described using a NURBS curve with equidistant x coordinates and the varying y coordinates of control points. A smooth curve across the concrete beam can be thereafter assigned to the tendon, as shown in **Figure 3.3(a)**. The prestressed load due to the tendon comprises two parts: equivalent distributed load along the tendon \mathbf{F}^{pre} and concentrated anchor load $\mathbf{F}^{\text{pre}^*}$, as shown in **Figure 3.3(b)**. It should be noted that the prestress loss during construction and tendon friction is ignored in this model.

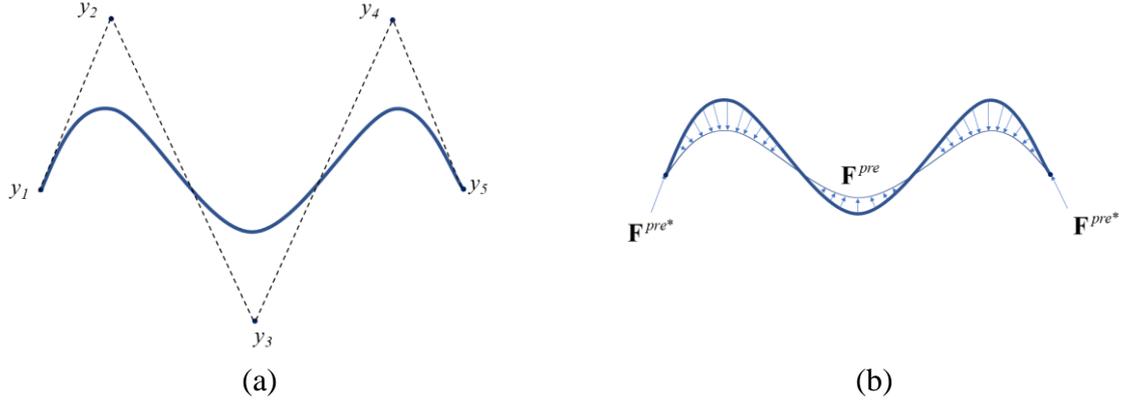


Figure 3.3 Example of prestress tendon: (a) Tendon curve described by NURBS, (b) equivalent prestress load along the tendon

3.3.2 Equivalent prestress load along the tendon

The equivalent distributive prestressed load is calculated according to the curvature of the tendon, irrespective of the shape of the member and support condition. The distributed load along the tendon is expressed as:

$$\mathbf{q}(\xi) = q(\xi) \mathbf{n}(\xi) \quad (3.12)$$

where $q(\xi) = \kappa(\xi) \cdot T_{\text{pre}}$ describes the curvature-based load distribution along the tendon, and $\mathbf{n}(\xi)$ is the corresponding normal direction. T_{pre} indicates the prestressed force and $\kappa(\xi)$ is the curvature. According to the principle of virtual work, the distributed load could be equivalented as a load vector on control points:

$$\mathbf{F}^{\text{ten}} \mathbf{U}^{\text{ten}} = \int \mathbf{q}(\xi) \cdot \mathbf{U}(\xi) ds \quad (3.13)$$

where \mathbf{F}^{ten} and \mathbf{U}^{ten} are the load vector and the displacement vector at the tendon control point, respectively. The equivalent load using Gaussian integration is expressed as follows:

$$\mathbf{F}^{\text{ten}} = \sum_e \sum_{i=1}^{N_{\text{cn}}^{\text{ten}}} w_i \mathbf{q}_i R_i^{\text{ten}} \left| \mathbf{J}_1^{\text{ten}} \right| \left| \mathbf{J}_2^{\text{ten}} \right| \quad (3.14)$$

where N_{GP}^{ten} is the number of tendon elemental Gaussian points. \mathbf{J}_1^{ten} is the tendon elemental Jacobian matrix from the parametric space to the physical space, while \mathbf{J}_2^{ten} is the parent space of the Gaussian point to the parametric space. w_i represents the weight value of i -th Gaussian point in e -th tendon element.

It should be noted that the initial number of control points of tendon shape is limited, which would lead to an inaccurate estimation of the equivalent prestressed in the mapping process, as shown in **Figure 3.4**. By introducing the h-refinement process in Section 3.2.2, more knots and control points can be automatically added to the curve to improve the accuracy with consistent geometry at the model formulation stage.

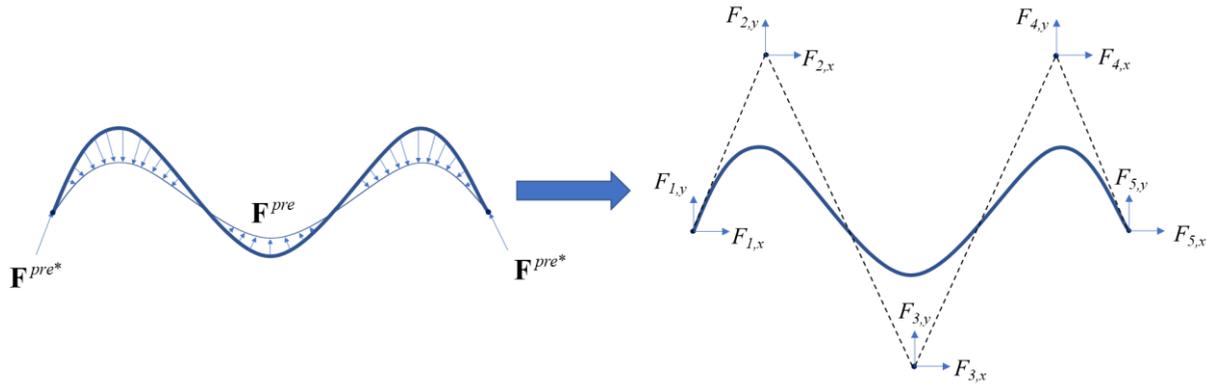


Figure 3.4 The process of prestressed load discretization

The refined set of control points set can be thereafter attained, as shown in **Figure 3.5**. To simplify the expression of h-refinement sensitivity analysis, the relationship between the coordinates of refined and original control points in Eq. (3.15) could be explicitly rewritten as follows:

$$\mathbf{P}_i^{ref} = \sum_{j=1}^{N_{init}} W_{i,j}^{ref} \mathbf{P}_j^{init} \quad (3.15)$$

where $W_{i,j}^{ref}$ indicates the weight value from j -th original control point to i -th refined control point. It should be noted that the h-refinement process of the tendon control point affects the sensitivity analysis of tendon shape variables.

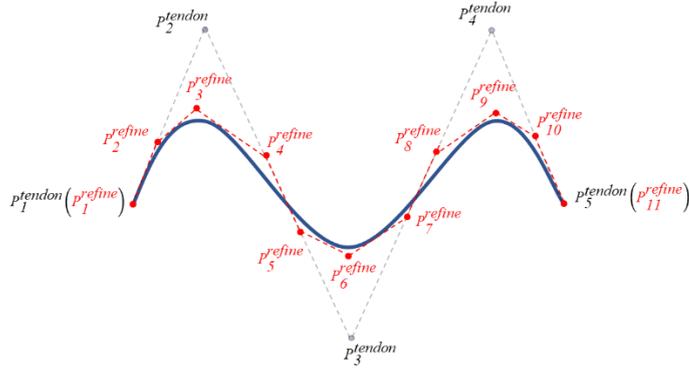


Figure 3.5 The h-refinement process of tendon control point for improving the load discretion accuracy

For the axial load at both ends of the prestressed tendon, it is also applied along the tangential direction at the end of the tendon curve, as shown below:

$$\mathbf{F}^{\text{ten}*} = T_{\text{pre}} \cdot \mathbf{n}^* \quad (3.16)$$

where \mathbf{n}^* indicates the tangential direction of the tendon curve end; T_{pre} is the prestressed load value.

3.3.3 A simplified load mapping scheme using IGA

The load mapping scheme of the prestressed concrete beam optimization can take advantage of using IGA. To address the prestress load on the concrete section, a mapping scheme is applied to transform physical coordinates into parametrical coordinates of the concrete domain. Traditionally in FEA, the process is achieved by calculating the inverse Jacobian matrix for each finite element. While in the IGA framework, an direct and explicit relationship [162] between physical coordinates and parametrical coordinates can be applied. The differences between these two approaches are illustrated in **Figure 3.6**. The transformation from the physical coordinate of tendon control points to the concrete domain can be expressed as follows:

$$\xi_i^{\text{con}} = \frac{P_i^x}{L}, \eta_i^{\text{con}} = \frac{P_i^y}{D} \quad (3.17)$$

where P_i^x, P_i^y are the physical coordinates of the i -th tendon control point, and $\xi_i^{\text{con}}, \eta_i^{\text{con}}$ are the corresponding parametrical location in the concrete domain. The load mapping operation based on this geometric transformation is illustrated in **Figure 3.7**. According to the principle of virtual work, the equivalent prestressed load can be written as follows:

$$\begin{aligned} \mathbf{F}_j^{\text{pre}} &= \sum_{i=1}^{N_{\text{con}}} \mathbf{F}_i^{\text{ten}} R_{(\xi_i^{\text{con}}, \eta_i^{\text{con}})}^{\text{con}}(\xi_j, \eta_j) = \sum_{i=1}^{N_{\text{con}}} \mathbf{F}_i^{\text{ten}} R_i^{\text{con}}(j) \\ \mathbf{F}_j^{\text{pre}*} &= \sum_{i=1}^n \mathbf{F}_i^{\text{ten}*} R_i^{\text{con}}(j) \end{aligned} \quad (3.18)$$

where $R_{(\xi_i^{\text{con}}, \eta_i^{\text{con}})}^{\text{con}}(\xi_j, \eta_j)$ (alternatively denoted as $R_i^{\text{con}}(j)$ for simplicity) represents the NURBS basis function, which maps the parametric location of the tendon control point $(\xi_i^{\text{con}}, \eta_i^{\text{con}})$ to the concrete control point (ξ_j, η_j) .

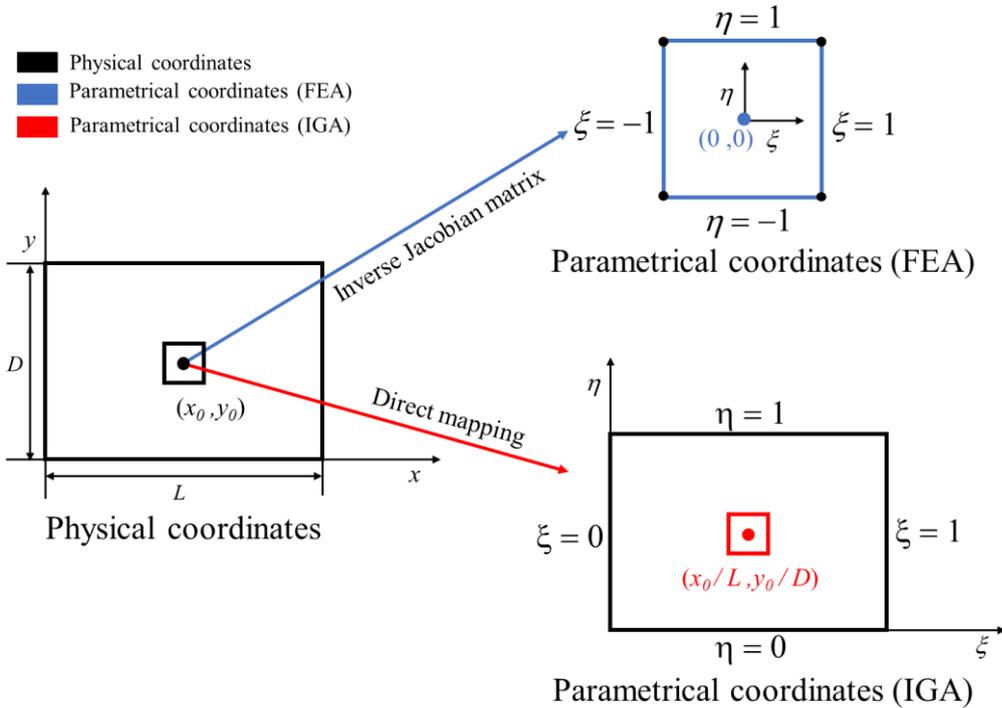


Figure 3.6 Comparison of the physical-parametrical mapping process between FEA and IGA

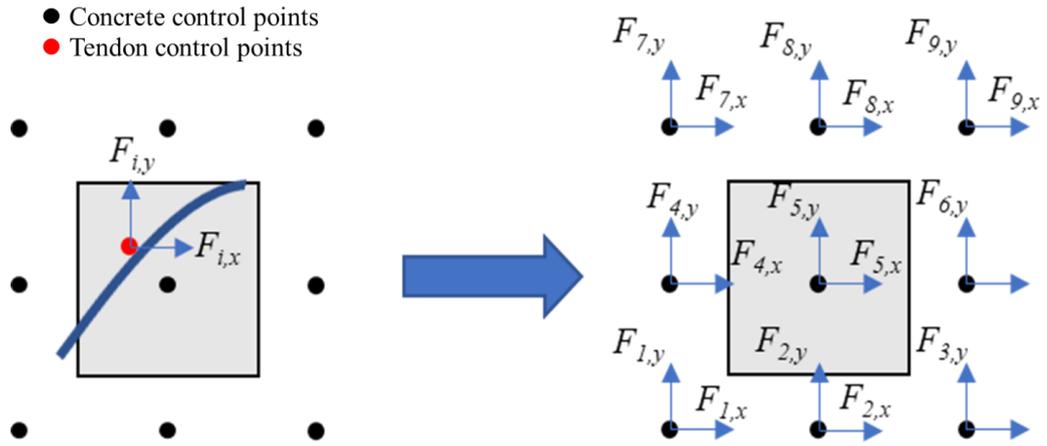


Figure 3.7 The inverse mapping process of an equivalent load from the tendon to the concrete domain

3.4 IGA-SIMP optimization model with unequal stress constraints

3.4.1 Drucker-Prager stress constraint

As previously explained, stress constraints are implemented in the IGA-SIMP optimization framework. For structural materials like concrete, the Drucker-Prager stress criterion is more suitable to exploit the unequal material strengths in tension and compression than the Von-Mises criterion of identical stress limits [163]. The Drucker-Prager criterion is usually written as below:

$$F(\boldsymbol{\sigma}) = \alpha I_1(\boldsymbol{\sigma}) + \sqrt{J_2(\boldsymbol{\sigma})} - H \quad (3.19)$$

where I_1, J_2 indicate the first and second invariant of the stress tensor, respectively. α and H are variables and can be calculated with compressive strength σ_c and tensile strength σ_t :

$$\alpha = \frac{\sigma_c - \sigma_t}{\sqrt{3}(\sigma_c + \sigma_t)}$$

$$H = \frac{2\sigma_c\sigma_t}{\sqrt{3}(\sigma_c + \sigma_t)}$$
(3.20)

Along with the Drucker-Prager criterion, the IGA-SIMP topology optimization can be formulated as:

$$\min_{[\boldsymbol{\rho}, \mathbf{y}]} V = \sum_{e=1}^N \rho_e V_e$$

$$\text{s.t. } \mathbf{K}\mathbf{U} = \mathbf{F}^{\text{ext}} + \mathbf{F}^{\text{sw}} + \mathbf{F}^{\text{pre}} + \mathbf{F}^{\text{pre}^*}$$

$$\alpha I_1(\boldsymbol{\sigma}_{e,i}) + \sqrt{J_2(\boldsymbol{\sigma}_{e,i})} - H \leq 0, \quad 0 \leq i \leq N_{\text{GP}}$$

$$0 \leq \rho_i \leq 1$$

$$t \leq y_j \leq h - t, \quad j = 1, \dots, N_{\text{ten}}$$
(3.21)

The optimization is herein performed regarding the concrete topology and tendon shape. $[\boldsymbol{\rho}, \mathbf{y}]$ as the design variables of the optimization model represent the density of concrete ($\boldsymbol{\rho}$) at control points and the \mathbf{y} coordinate of the tendon curve. $\boldsymbol{\sigma}_{e,i}$ is the stress vector on Gaussian point. $N_{\text{GP}}, N_e, N_{\text{ten}}$ are the number of elemental Gaussian point, concrete elements and initial tendon control points, respectively. h is the total beam height, and t is the interval of the unreachable region of the tendon, as illustrated in **Figure 3.8**.

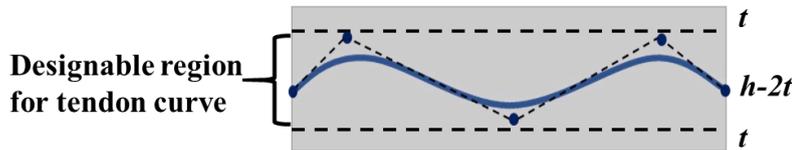


Figure 3.8 Designable region of the tendon curve

3.4.2 Refinements of stress constraint

For stress constraint problem, the following numerical issues have to be carefully resolved: (1) the stress singularity problem; (2) the unaffordable

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam computational cost of numerous local stress constraints; and (3) the optimization stabilization. Various approaches are used to alleviate the above technical barriers, which have been briefly explained in this section.

3.4.2.1 Stress relaxation

For issue (1): when the density of the element tends to be zero, it will result in extremely large stress and violate the yield criterion. Consequently, the yield constraints in the low-density region will prevent materials from being completely removed, leading to numerical difficulty in obtaining the actual optimal material distribution. This phenomenon is usually termed stress singularity, which is alleviated using an epsilon-relaxation method [164] in this paper:

$$\begin{aligned} \varphi_{F,e}(\boldsymbol{\sigma}_e, \rho_e) &= \frac{1}{h_e} \left(\alpha I_1(\boldsymbol{\sigma}_e) + \sqrt{J_2(\boldsymbol{\sigma}_e)} \right) / H \leq 1 \\ h_e &= 1 + \frac{\varepsilon}{\rho_e} - \varepsilon, \quad (e = 1, 2, \dots, N_e) \end{aligned} \quad (3.22)$$

where h_e is the relaxation coefficient representing the relaxation degree of the corresponding constraint and ε is a prescribed small positive real number. A greater value means a higher relaxation degree.

3.4.2.2 Local stress aggregation

For issue (2): The computational cost of implementing stress constraints for the elements of the model is a technical challenge during the optimization process. This issue is addressed by wrapping the local constraints as a single global constraint using the Kreisselmeier-Steinhauser (K-S) and p-norm stress measure [165-167] to reduce the computational complexity. The major difference between these two global constraint functions is that the P-norm requires the non-negative local function value, while K-S has no such requirements [160]. Due to the

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam
 negative values of the Drucker-Prager criterion, the K-S function is chosen as the
 aggregation function:

$$G_{\text{KS}}(\boldsymbol{\rho}, \mathbf{y}) = \frac{1}{\eta} \ln \sum_e^{N_e} \exp(\eta \varphi_{F,e}(\boldsymbol{\sigma}_e, \rho_e)) \leq 1 \quad (3.23)$$

where $\varphi_{F,e}(\boldsymbol{\sigma}_e, \rho_e)$ is the relaxation function in Eq.(3.22), and $\eta > 0$ is the
 aggregation parameter, which is used to penalize the violation of the local
 constraints. By increasing the value of the aggregation parameter η , the
 approximation error could be reduced. However, the larger η also leads to higher
 nonlinearity, which means more difficulty in convergence.

3.4.2.3 Stabilization of optimization

For issue (3), the STM-based stress correction scheme [160] is used to correct
 the global stress measures and to stabilize the optimization process with the
 oscillation control, as shown in Eq. (3.24).

$$\tilde{G}_{\text{KS}} = cp \cdot G_{\text{KS}} \quad (3.24)$$

cp is the correction parameter, as:

$$cp_n = \begin{cases} \frac{\max(\varphi_{F,e})_n}{(G_{\text{KS}})_n}, & n = 1 \\ q_n \frac{\max(\varphi_{F,e})_n}{(G_{\text{KS}})_n} + (1 - q_n) cp_{n-1}, & n > 1 \end{cases} \quad (3.25)$$

where the control factor ($0 < q_n < 1$) is to alleviate the discontinue change of cp_n
 and to avoid iterative oscillation leading to unstable convergence.

3.4.3 Geometry control of optimized configuration

Aiming for construction-oriented topology optimization, the design variables,

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam including the minimum width (length scale) control and the tendon-concrete filter, are necessary to be included in order to prevent segments of low manufacturability, as illustrated in **Figure 3.9**. For instance, very slim components are challenging to manufacture and of little resilience during lifetime service. This optimal design effect is avoided by a desirable width control method to set up a minimum width.

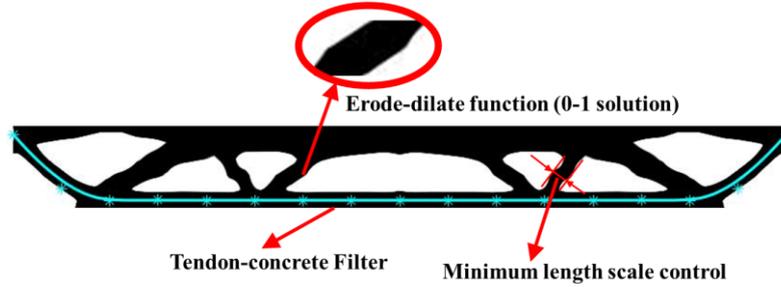


Figure 3.9 Illustration of geometry control

3.4.3.1 Tendon-concrete filter

a tendon-concrete filter is used to ensure the minimum cover of the tendon by the concrete. The filter proposed by [168] can be expressed as follows:

$$\hat{\rho}_e = \rho_e + (1 - \rho_e) e^{-\frac{1}{2} \left(\frac{d_{ei}}{\beta_{fil}} \right)^{\mu_{pre}}} \quad (3.26)$$

where $\hat{\rho}_e$ means the e -th elemental density and d_{ei} indicates the shortest distance between the center of e -th element and the tendon control points. β_{fil} is the function width, while μ_{pre} is the sharpness of the Super-Gaussian function.

3.4.3.2 Component size control

Moreover, a NURBS-based minimum length scale control approach [162] is adopted in this paper and further developed to co-work with the IGA-SIMP optimization framework. Besides the geometric mesh, an independent density mesh is used to control the minimum length (width) scale. The density filter can be thereafter similarly formulated as Eq. (3.9) and can be written as below:

$$\rho_e = \sum_{j=1}^{N_R} R_j^{\text{den}}(e) \rho_j \quad (3.27)$$

where $R_j^{\text{den}}(e)$ is the NURBS rational basis function of density variables, N_R is the number of supporting control points.

3.4.3.3 Density projection

Finally, the erode-dilate function [169] is applied to obtain a clear 0-1 solution. the formulation of the erode-dilate method is as follows:

$$\begin{aligned} \tilde{\rho}_e^{\text{ero}} &= \frac{\tanh(\beta_{\text{HS}} \eta_{\text{ero}}) + \tanh(\beta_{\text{HS}} (\hat{\rho}_e - \eta_{\text{ero}}))}{\tanh(\beta_{\text{HS}} \eta_{\text{ero}}) + \tanh(\beta_{\text{HS}} (1 - \eta_{\text{ero}}))} \\ \tilde{\rho}_e^{\text{dil}} &= \frac{\tanh(\beta_{\text{HS}} \eta_{\text{dil}}) + \tanh(\beta_{\text{HS}} (\hat{\rho}_e - \eta_{\text{dil}}))}{\tanh(\beta_{\text{HS}} \eta_{\text{dil}}) + \tanh(\beta_{\text{HS}} (1 - \eta_{\text{dil}}))} \end{aligned} \quad (3.28)$$

As the sharpness value β_{HS} increases, the 0-1 solution is better, but it reduces the stability of the optimization iteration. η_{ero} and η_{dil} are the projection threshold for eroded and dilated design [169], respectively. It should be noted that too high or too low projection thresholds can produce Zig-Zag boundaries since the NURBS-based length scale method reduces the topological design space. As shown in **Figure 3.10**, a zoomed 3×3 nodes region using different projection schemes is presented. The boundary of density distribution is not sharp enough if no Heaviside is adopted (**Figure 3.10** (a)), whereas the Heaviside method of a zero projection leads to a Zig-Zag boundary. When the projection threshold is defined as a variable similar to the erode-dilate method [169], a relatively smooth boundary can be obtained.

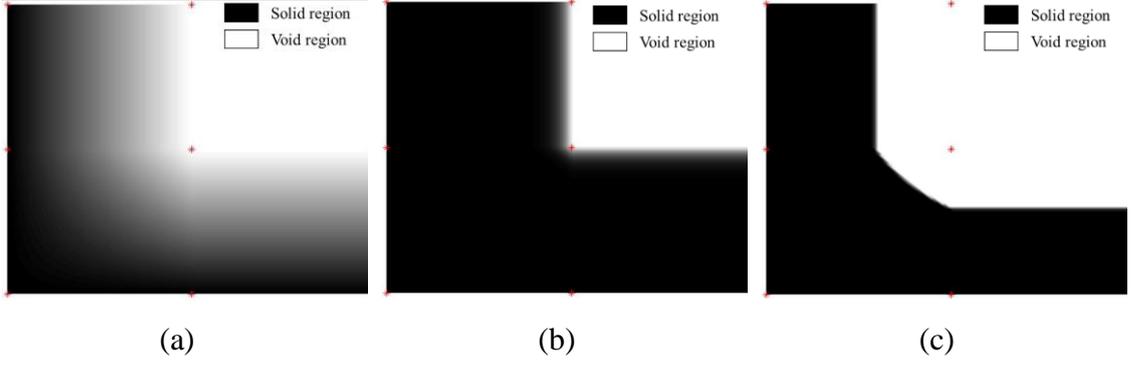


Figure 3.10 Comparison of projection schemes with different project threshold, (η_{ero}, η_{dil}) : (a) no Heaviside; (b) projection threshold is 0; (c) variable projection threshold e.g.: (0.4, 0.6).

After aforementioned modifications including local Drucker-Prager stress constraint, stress relaxation, local constraint aggregation, stabilization, and geometric control, the optimization model can be updated from Eq.(3.21) as:

$$\begin{aligned}
 \min_{[\boldsymbol{\rho}, \mathbf{y}]} \quad & V = \sum_{e=1}^{N_e} \hat{\rho}_e V_e \\
 \text{s.t.} \quad & \mathbf{K}\mathbf{U} = \mathbf{F}^{\text{ext}} + \mathbf{F}^{\text{sw}} + \mathbf{F}^{\text{pre}} + \mathbf{F}^{\text{pre}^*} \\
 & \tilde{G}_{\text{KS}} \leq 1 \\
 & 0 \leq \rho_i \leq 1 \\
 & t \leq y_i \leq h - t, \quad i = 1, \dots, N_{\text{ten}}
 \end{aligned} \tag{3.29}$$

3.5 Sensitivity analysis

3.5.1 Volume sensitivity

The material distribution of concrete is described with the topological variables of concrete and shape variables of the tendon. The volume objective and the stress constraint are coupled during the topology optimization, which eventually produces the design variables $\mathbf{x} = [\boldsymbol{\rho}, \mathbf{y}]$. For the derivation of volume

objective function Eq. (3.19), the partial derivative for the design variable x_i can be given using a direct form:

$$\frac{\partial V}{\partial x_i} = \sum_e \frac{\partial \tilde{\rho}_e}{\partial x_i} V_e \quad (3.30)$$

The explicit gradient of the term $\frac{\partial \tilde{\rho}_e}{\partial x_i}$ is calculated from the partial derivatives of the topological variables ρ_i :

$$\frac{\partial \tilde{\rho}_e}{\partial \rho_i} = \frac{\partial \tilde{\rho}_e}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial \rho_e} \frac{\partial \rho_e}{\partial \rho_i} \quad (3.31)$$

In Eq. (3.31), the derivatives of variables can be explicitly computed as follows: (i) $\frac{\partial \tilde{\rho}_e}{\partial \hat{\rho}_e}$ is derived from the Heaviside function in Eq. (3.28); (ii) $\frac{\partial \hat{\rho}_e}{\partial \rho_e}$ is a differential form of the tendon-concrete filter in Eq. (3.26); (iii) $\frac{\partial \rho_e}{\partial \rho_i}$ is derived from Eq. (3.27), where the derivatives of the NURBS filter from density design variables to elemental density. Particularly the partial derivative of $\tilde{\rho}_e$ with respect to the shape variable y_i is given as below:

$$\frac{\partial \tilde{\rho}_e}{\partial y_i} = \frac{\partial \tilde{\rho}_e}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial d_{ei}} \frac{\partial d_{ei}}{\partial y_i} \quad (3.32)$$

where $\frac{\partial \hat{\rho}_e}{\partial d_{ei}}$ is derived from the Heaviside function in Eq. (3.28) and $\frac{\partial d_{ei}}{\partial y_i}$ is

based on the Euclidean distance. According to the h-refinement operation in Eq. (3.4), the distance is computed from the element center to the refined tendon control points and then projected to shape variables y_i , as shown in the following form:

$$\frac{\partial d_{ei}}{\partial y_i} = \frac{\partial d_{ei}}{\partial y_j^{\text{ref}}} \frac{\partial y_j^{\text{ref}}}{\partial y_i} \quad (3.33)$$

where the first component $\frac{\partial d_{ei}}{\partial y_j^{\text{ref}}}$ is derived from the formula of distance, it should be noted that the derivatives of the non-closest control points are zero. The second component $\frac{\partial y_j^{\text{ref}}}{\partial y_i}$ is given by Eq. (3.15).

3.5.2 Stress constraint sensitivity

For the stress constraint function, the adjoint method is used for sensitivity analysis, which is expressed as follows:

$$\phi(\mathbf{p}, \mathbf{y}) = \tilde{G}_{KS} + \boldsymbol{\lambda}^T (\mathbf{K}\mathbf{U} - \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{sw}} - \mathbf{F}^{\text{pre}} - \mathbf{F}^{\text{pre}^*}) \quad (3.34)$$

where $\boldsymbol{\lambda}$ is the adjoint vector. The partial derivative of Eq. (3.34) to a certain variable x_i is given by:

$$\begin{aligned} \frac{\partial \phi}{\partial x_i} = & \frac{\partial \tilde{G}_{KS}}{\partial \varphi_{F,e}} \left(\frac{\partial \varphi_{F,e}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \rho_e} \frac{\partial \rho_e}{\partial x_i} + \frac{\partial \varphi_{F,e}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x_i} + \frac{\partial \varphi_{F,e}}{\partial h_e} \frac{\partial h_e}{\partial \rho_e} \frac{\partial \rho_e}{\partial x_i} \right) \\ & + \boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{K}}{\partial x_i} \mathbf{U} + \mathbf{K} \frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial \mathbf{F}^{\text{sw}}}{\partial x_i} - \frac{\partial \mathbf{F}^{\text{pre}}}{\partial x_i} - \frac{\partial \mathbf{F}^{\text{pre}^*}}{\partial x_i} \right) \end{aligned} \quad (3.35)$$

To eliminate the implicit term $\frac{\partial \mathbf{U}}{\partial x_i}$, by collecting all the related terms in Eq.

(3.35), the following adjoint equation is to be solved:

$$\frac{\partial \tilde{G}_{KS}}{\partial \varphi_{F,e}} \frac{\partial \varphi_{F,e}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{U}} + \boldsymbol{\lambda}^T \mathbf{K} = 0 \quad (3.36)$$

Where $\frac{\partial \tilde{G}_{KS}}{\partial \varphi_{F,e}}$ is calculated by the K-S aggregation function in Eq. (3.24), and

$\frac{\partial \varphi_{F,e}}{\partial \boldsymbol{\sigma}}$ can be obtained from Eq. (3.22), which is related to the stress relaxation

procedure. The last term $\frac{\partial \boldsymbol{\sigma}_e}{\partial \mathbf{U}_e}$ is derived from Eq. (3.10), for measuring the

elemental stresses. Then, by solving Eq. (3.36), the adjoint vector $\boldsymbol{\lambda}$ is obtained.

After plugging it into Eq. (3.35), the expression can be rewritten as:

$$\begin{aligned} \frac{\partial \phi}{\partial x_i} = & \frac{\partial \tilde{\mathbf{G}}_{KS}}{\partial \varphi_{F,e}} \left(\frac{\partial \varphi_{F,e}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}_e}{\partial \rho_e} \frac{\partial \rho_e}{\partial x_i} + \frac{\partial \varphi_{F,e}}{\partial h_e} \frac{\partial h_e}{\partial \rho_e} \frac{\partial \rho_e}{\partial x_i} \right) \\ & + \boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{K}}{\partial x_i} \mathbf{U} - \frac{\partial \mathbf{F}_{sw}}{\partial x_i} - \frac{\partial \mathbf{F}_{pre}}{\partial x_i} - \frac{\partial \mathbf{F}_{pre}^*}{\partial x_i} \right) \end{aligned} \quad (3.37)$$

The unknown derivatives in Eq. (3.37) can be explicitly obtained as follows:

(i) $\frac{\partial \boldsymbol{\sigma}_e}{\partial \rho_e}$ is obtained from the stress expression in Eq. (3.10); (ii) $\frac{\partial \varphi_{F,e}}{\partial h_e}$ and $\frac{\partial h_e}{\partial \rho_e}$

are calculated from the stress relaxation function in Eq. (3.22); (iii) $\frac{\partial \mathbf{K}}{\partial x_i}$ is based on

the interpolation rule of the stiffness matrix in Eq. (3.8); (iv) For $\frac{\partial \mathbf{F}_{sw}}{\partial x_i}$, the self-

weight load is expressed as $\mathbf{F}_{sw} = \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{E}} w_i R_i^{\text{con}} \rho_e g |\mathbf{J}_1| |\mathbf{J}_2|$, which suggests the

derivative can be given as:

$$\frac{\partial \mathbf{F}_{sw}}{\partial x_i} = \sum_e \sum_{i=1}^{N_{\text{pr}}} w_i R_i^{\text{con}} \frac{\partial \rho_e}{\partial x_i} g |\mathbf{J}_1| |\mathbf{J}_2| \quad (3.38)$$

(v) The derivatives of the prestress force vectors $\frac{\partial \mathbf{F}_{pre}}{\partial x_i}$, $\frac{\partial \mathbf{F}_{pre}^*}{\partial x_i}$ are only related to

shape variables (i.e., the geometry of the tendon). The derivatives of distributive prestress load can be given below:

$$\frac{\partial \mathbf{F}_{pre}}{\partial y_i} = \left[\sum_e \sum_{j=1}^{N_{GP}^{ten}} \left(w_j \frac{\partial \mathbf{q}_j}{\partial y_i} R_j^{ten} |\mathbf{J}_1^{ten}| |\mathbf{J}_2^{ten}| + w_j \mathbf{q}_j R_j^{ten} \frac{\partial |\mathbf{J}_1^{ten}|}{\partial y_i} |\mathbf{J}_2^{ten}| \right) \right] \mathbf{R}^{con} + \left(\sum_e \sum_{j=1}^{N_{GP}^{ten}} w_j \mathbf{q}_j R_j^{ten} |\mathbf{J}_1^{ten}| |\mathbf{J}_2^{ten}| \right) \frac{\partial \mathbf{R}^{con}}{\partial y_i} \quad (3.39)$$

- (i) From Eq. (3.12), $\frac{\partial \mathbf{q}_j}{\partial y_i} = \frac{\partial q_j}{\partial y_i} \mathbf{n}_j + q_j \frac{\partial \mathbf{n}_j}{\partial y_i}$ and $\frac{\partial q_j}{\partial y_i} = \frac{\partial \kappa_j}{\partial y_i} T_{pre}$ can be calculated.

For the derivatives of curvature $\frac{\partial \kappa_j}{\partial y_i}$ and normal direction $\frac{\partial \mathbf{n}_j}{\partial y_i}$, please refer

to the detailed explanation in Appendix A;

- (ii) $\frac{\partial |\mathbf{J}_1^{ten}|}{\partial y_i}$ is also available in the process of derivation of the normal direction in

Appendix A;

- (iii) The last term $\frac{\partial \mathbf{R}_j^{con}(k)}{\partial y_i} = \frac{\partial \mathbf{R}_j^{con}(k)}{\partial \eta_j^{con}} \frac{\partial \eta_j^{con}}{\partial y_j^{refine}} \frac{\partial y_j^{refine}}{\partial y_i}$, $\frac{\partial \mathbf{R}_j^{con}(k)}{\partial \eta_j^{con}}$ is the first-order

derivatives of the NURBS surface in Eq. (3.18), while $\frac{\partial \eta_j^{con}}{\partial y_j^{ref}}$ is from the

physical and parametrical coordinates relation in Eq. (3.17) and $\frac{\partial y_j^{ref}}{\partial y_i}$ is

available in Eq. (3.33).

The partial derivate of the concentrated prestressed load $\frac{\partial \mathbf{F}_{pre}^*}{\partial y_i}$ is given as:

$$\frac{\partial \mathbf{F}_{pre}^*}{\partial y_i} = \frac{\partial \mathbf{F}_{ten}^*}{\partial y_i} \mathbf{R}^{con} + \mathbf{F}_{ten}^* \frac{\partial \mathbf{R}^{con}}{\partial y_i} \quad (3.40)$$

where $\frac{\partial \mathbf{R}^{con}}{\partial y_i}$ is already available and $\frac{\partial \mathbf{F}_{ten}^*(\xi)}{\partial y_i}$ can be obtained from Eq.

(3.16). The corresponding tangential derivation term $\frac{\partial \mathbf{n}^*(\xi)}{\partial y_i}$ is given in Eq. (A.6)

in Appendix A.

3.6 Demonstrative examples

The proposed topology optimization framework is demonstrated using a simply-supported post-tensioned concrete beam, as shown in **Figure 3.11**, where a half model is presented considering symmetry. An open-source IGA package programmed in MATLAB [170] has been adapted in the proposed framework to conduct optimization. The effects of various strength ratios (compressive strength f_c to tensile strength f_t) and minimum width are discussed within this section to demonstrate the construction-oriented concerns. It should be clarified that these attempts are somehow still preliminary regarding the real application of civil engineering structures. The proposed method is conceptually demonstrated using a 2D problem in this paper, which can be extended for 3D formulation later and will be discussed in the follow-up work.

When performing the IGA-SIMP-based optimization, the MMA [171] moving limit of the topology and shape variables are set as 0.1 and 0.01, respectively. The SIMP penalty value is used to accelerate the formation of the configuration of the structure. The penalty is increased from 1.0 to 3.0, with a step size of 0.25 per 25 iterations. A high sharpness value of the erode-and-dilate function β_{HS} strengthens the 0-1 sharpness however, unfavorably reduces optimization stability. Hence, the sharpness value is increased by 1 per 25 iterations from 1 to 5. The tendon-concrete

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam filter width β_{fil} and the sharpness μ_{pre} are set as 0.015 and 5.0. A relatively high epsilon-relaxation coefficient ε is set as 0.4 to ensure the well-relaxed low-density region, while the aggregation coefficient η is taken as 8.0. The value of q_n in the STM-based stress correction scheme is fixed at 0.5 to improve the continuity of the aggregated stress evaluation.

Regarding the material properties of the beams, Young's modulus of the concrete (E) is 28 GPa, and Poisson ratio $\nu=0.2$. The density of concrete is taken as 2450 kg/m³. The load balancing method[172] for curved tendons has been adopted to estimate the tendon force corresponding to the given beam geometry. This analytical estimation assumes the tendon of a parabola curve, and the prestress force for the beam of length L is given as:

$$T_{pre} = \frac{q_{sv} L^2}{8a} \quad (3.41)$$

where q_{sv} is the service load (such as vehicles) on the top surface, a indicates the tendon's drupe, set as 0.8 of beam height to ensure a safe distance from tendon to both top and bottom chords.

3.6.1 Simply supported beams with single span

In this beam example, the beam length L is 2m, and the height D is 0.2m. When the service load q_{sv} equals 200 kN/m, the analytical T_{pre} is calculated as 625 kN. According to the IGA result of this rectangular beam, the maximum tensile and compressive stress is 0.28 MPa and 1.33 MPa, respectively, which are both far below the strength limit, indicating low utilization of material capacity in a traditional design. Through utilizing the theoretical prestress force in the proposed

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam optimization method, the restriction of parabolic tendon shape and prismatic concrete topology are eliminated to achieve a more efficient tendon shape and lighter concrete beam, which enhances the practical value of such traditional prestress beam design methods. It should be noted that the concentrated prestress load on the edge is equivalented to a localized pressure to simulate the effect of the anchor and to avoid stress concentration on the loading region.

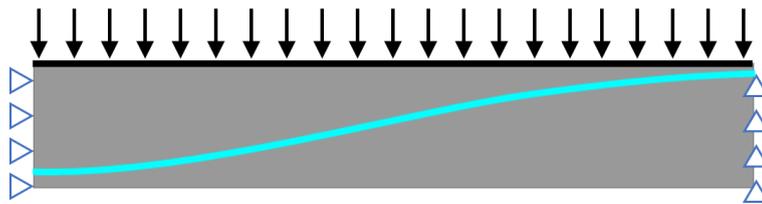


Figure 3.11 A half model of the simply supported beam with a uniform load

The right half of the beam is modelled considering the axial symmetry and discretized as 200×40 elements under the IGA framework. For convenient visualization, the half model is duplicated to show a full model in the following discussion. Nine tendon control points are used to generate the curve shape for the prestressed tendon, while 200 knots are inserted to improve the accuracy of load equivalence, as discussed in Section 3.3.2. The elemental density value of the first two upper surface layers is fixed as 1.0 to form a rigid load-carrying surface, and the edge tendon control point is fixed on the upper right corner to predetermine the location of the prestressing anchor, as shown in **Figure 3.11**.

Figure 3.12 (a) shows the typical iteration of topology optimization, where the density distribution has been generated after several steps. During the iteration, the structural volume ratio decreases from 1.0 to a converged 0.47, while the D-P criterion value approaches zero, suggesting that the stresses are constrained underneath the yielding surface to remain elastic. The presented optimization is performed with equal strength limits and low minimum-width control (the reference length scale bar can be found in **Table 3.2**). Thus, the equal-strength D-P

value reflects the Von-Mises stress criterion, and the distribution is shown in **Figure 3.12(b)**, where the relatively higher level of stress is in the upper chord. This is due to (1) the bottom chord assigned with a fixed filter thickness to ensure a sufficient tendon cover; (2) the bottom chord only co-working the tendon to resist the tendon forces, and the main resisting action is taken by the prestressed tendon. **Figure 3.12** presents a visualized topology optimization process without the construction-oriented concerns, whereas the effect of minimum width and unequal strengths is discussed in the following subsections.

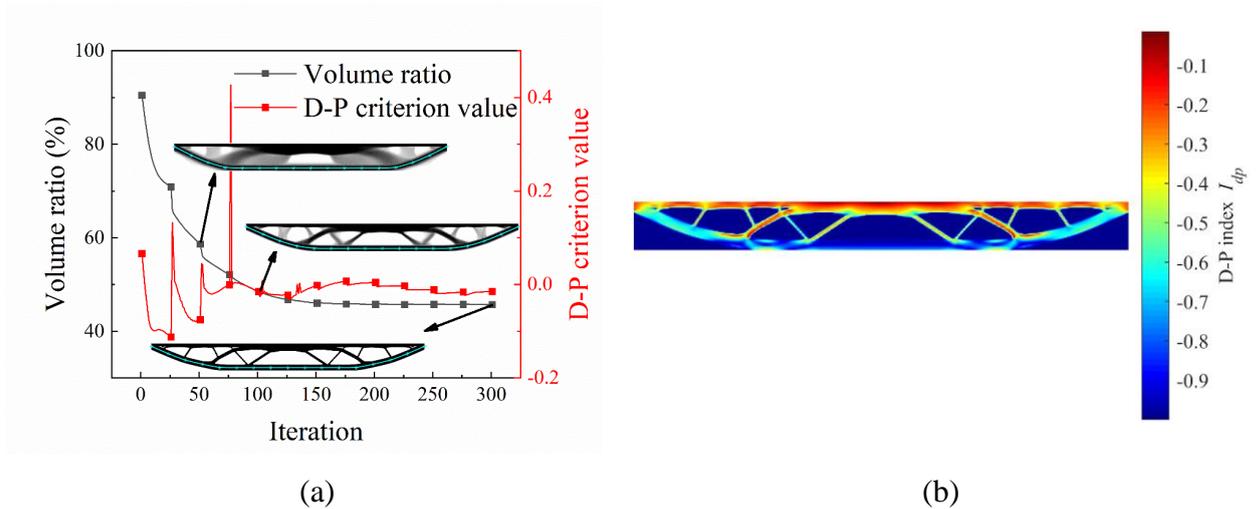


Figure 3.12 Topology optimization result(equal strength limits, low width control (11.6 mm)): (a) optimization process; (b) the distribution of D-P index ($I_{dp} = \varphi_{F,e} - 1$), negative value of I_{dp} indicates below strength limit.

3.6.1.1 Effect of different component size control

A NURBS-based minimum length scale control [162] is applied to prevent slim components in an optimized structure. As shown in, the optimized models of various minimum-width control are presented, where the width is selected as 11.6mm, 23.3 mm, and 46.6 mm corresponding to the different density mesh grids: 100×20, 50×10, 25×5. In these examples, the compressive strength is taken as 10 MPa, equal to the tensile strength. The iteration of topology correlated to the

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam material distribution during the optimization is shown in **Figure 3.13**, in which the variation of volume ratio and D-P value are presented as well. In the optimized models, a density control mesh of 50×10 is adopted as the minimum width is limited to 23.3mm.

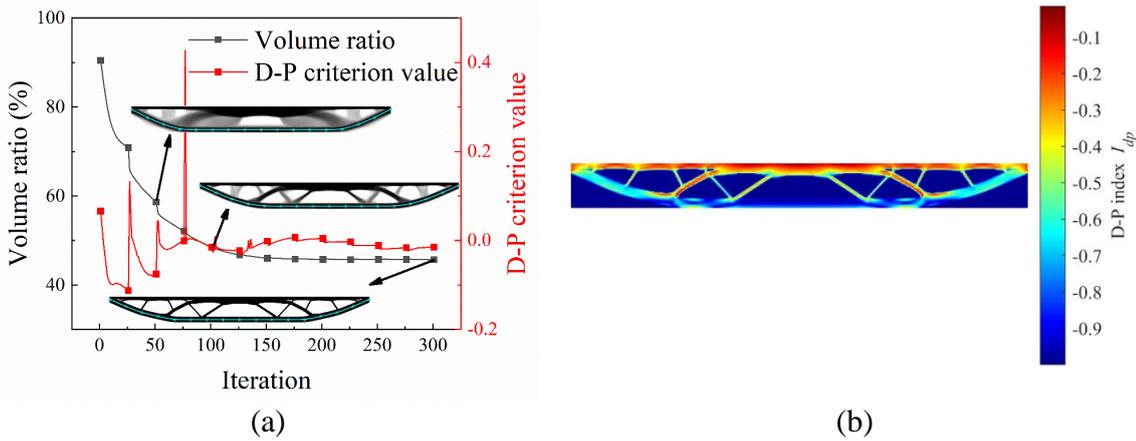


Figure 3.13 Optimized topology with medium width control (23.3 mm) and equal strength limits: (a) optimization process; (b) the distribution of D-P index I_{dp} .

Meanwhile, the geometric mesh independent from the density mesh remains at 200×40 . When the minimum width limit varies, it can be observed that the slim components and tiny branches are gradually prevented, as shown in **Table 3.2**. The structure of a reasonably controlled minimum width thereafter represents a better manufacturability performance for construction.

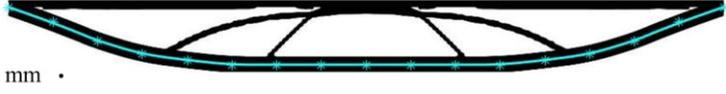
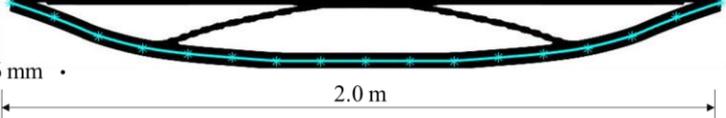
Table 3.2 Illustration of various minimum-width control in topology optimization

Volume	Configuration
47.39%	11.6 mm •
48.91%	23.3 mm •
58.52%	46.6 mm •

3.6.1.2 Effect of different ratio of unequal compressive and tensile strength

Concrete is a common structural material used with steel rebars as reinforcement, which can be either strengthened by adding steel fibers or other types of fibers. The critical feature of concrete is its different strengths in tension and compression. When the tensile strength (f_t) and the compressive strength (f_c) are known, the topology optimization should maximize the use of the material strength while the safety performance is ensured. As shown in **Table 3.3**, different topology optimization results can be obtained after applying the Drucker-Prager criterion to consider unequal f_t and f_c , while a minimum-width control of 11.6 mm is applied in these models. To observe the effect of strength ratios, the compressive and tensile strength are selected as three sets: $f_c = 10\text{MPa}$, $f_t = 10\text{MPa}$ (Von-Mises Criterion); $f_c = 13.3\text{MPa}$, $f_t = 6.7\text{MPa}$; $f_c = 15\text{MPa}$, $f_t = 5\text{MPa}$. Here the scale of summed f_c and f_t has been kept the same to represent the similar material capacity combining tension and compression. The optimized structural configuration varies when the ratio of f_c / f_t is increased from one to three times the tensile strength. As the f_c / f_t ratio increases, the number of components between the upper and lower chords is reduced because the major components of higher compressive strength can solely bear the load transferred from the chords. Meanwhile, the volume ratio declines from 47.39% to 37.94% as higher compressive strength and f_c / f_t ratio are employed. It should be noticed that further improvement may be necessary for the optimized structure since the structural performance of the model has to be examined for various loading scenarios considering the service loading and extreme loading, which can lead to additional variation to the optimized topology.

Table 3.3 Effect of different strength ratio to the topology optimization with low width control (11.6 mm)

Compressive/Tensile Strength	Volume	Configuration
10/10 MPa (1:1)	47.39%	
13.3/6.7 MPa (2:1)	40.89%	
15/5 MPa (3:1)	39.77%	

When applying the D-P stress constraints, and the minimum-width control simultaneously, the iterative process of topology optimization is illustrated in **Figure 3.14** (a). As the strength ratio, f_c / f_t is set to 3, and the minimum-width control is 23.3mm, the volume ratio and the D-P ratio converge to the optimized values after 300 iterations. Upon the completion of optimization, the obtained shape comprises fewer components and the middle components yield to an arch. Such a regularized shape could significantly simplify the construction procedure, indicating a high level of constructional ability. **Figure 3.14** (b) shows the D-P value distribution after the completion of the optimization. When using the same minimum-width control (23.3mm), the variations of optimal topology corresponding to various strength ratios have been shown in **Table 3.4**. Compared to the results in **Table 3.3**, the optimized models with a minimal width limit of 23.3mm comprise less complexity in terms of the middle component distribution. As a compromising result of the width constraint, higher volume ratios are seen in these optimized models. In real structural design, a balanced performance between the volume ratio (material cost) and the width control (manufacturing cost) should be considered.

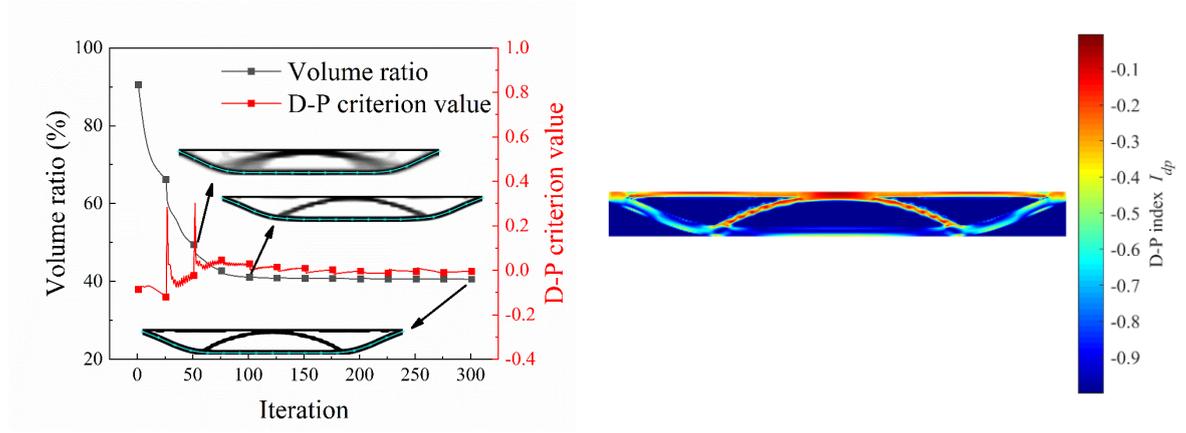


Figure 3.14 Topology optimization with medium width control and 3:1 strength limit ratio: (a) optimization process; (b) the distribution of D-P index I_{dp} .

Table 3.4 Topology optimization results after applying strength ratios and 23mm minimum-width control

Compressive/Tensile Strength	Volume	Configuration
10/10 MPa (1:1)	48.91%	
13.3/6.7 MPa (2:1)	41.04%	
15/5 MPa (3:1)	40.54%	

3.6.2 Simply supported beams with multiple spans

To further demonstrate the proposed optimization approach, two continuous beams are briefly presented here, which are of two spans and three spans, respectively. Each span is 2.0 m long and similarly configured as the previous single-span beam model. f_c, f_t is adopted as 13.3 MPa and 6.7 MPa. The schematic plots of the continuous beams are shown in **Figure 3.15** (a) and (b). The upper surface pressure is assigned as 250 kN/m for the continuous beams, and the prestress force is fixed to 625 kN. The optimization process and the distribution of

the D-P index of optimized models are presented in **Figure 3.15**.

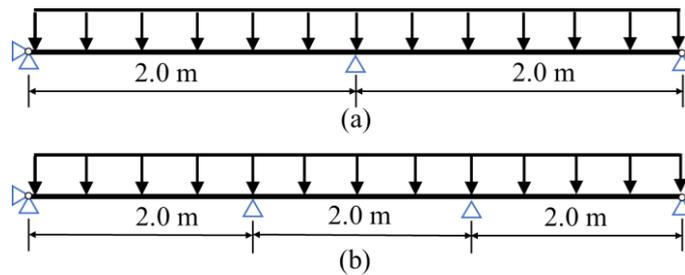


Figure 3.15 Schematic plot of continuous beams.: (a) a two-span beam ($2 \times 2.0\text{m}$); (b) a three-span beam ($3 \times 2.0\text{m}$).

The optimized models are obtained under the same width control limit as 23.3 mm, while the volume ratios of the two beam cases eventually converge at 0.49 and 0.47. As shown in the figures, the tendon profiles of both beams are of smooth curved shapes after the automatic optimization. The tendon-covering region of the stress distribution contours is well below the strength limit, indicating that the prestress tendons effectively alleviate the tensile stress in the concrete beams. As shown in **Figure 3.16**, the optimized topology of continuous beams is of expected performance, which presents the well-controlled complexity of component distribution and the well-achieved efficiency in pursuing a lightweight design.

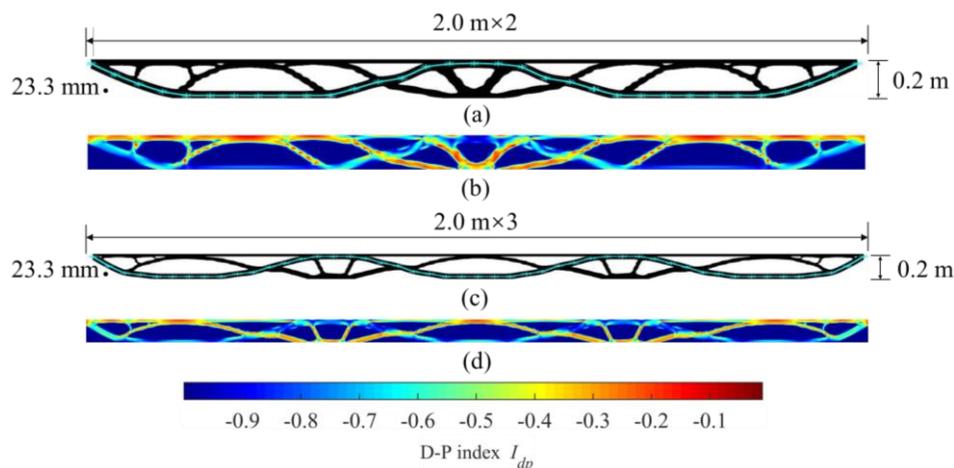


Figure 3.16 Topology optimization of multi-span beam: (a) density distribution of two-span beam; (b) D-P index I_{dp} distribution of two-span beam; (c) density distribution of three-span beam; (d) D-P index I_{dp} distribution of three-span beam

3.7 Concluding remarks

In this chapter, an IGA-SIMP topology optimization approach has been developed along with a vision of a construction-oriented topology optimization framework. The constraint functions for material stresses and geometric configuration have been implemented to address safety and manufacturability in real construction practice.

A prestressed concrete beam comprising a steel tendon and a concrete beam is modelled and optimized using the proposed topology optimization framework. The NURBS-IGA-based mapping from the tendon to the concrete entity can achieve a simultaneous optimization over the concrete topology and the tendon curve shape. The Drucker-Prager criterion has been applied to consider the unequal strength limits for tension and compression as commonly encountered for concrete-type materials in real construction. A minimum-width control is implemented during the topology optimization to prevent very slim components in the optimum model.

The effect of strength ratio f_c / f_t and minimum width has been demonstrated using the prestressed concrete beam. The increasing strength ratio can lead to a different distribution of concrete in the optimized beam and a lower volume ratio. When applying minimum-width control, the branch components are significantly reduced, and the pattern of internal components tends to be more regularized, which could essentially improve the manufacturability of the optimized structures.

As discussed in this chapter, the construction-oriented constraint functions are not limited to the stress criterion and width control. To implement topology optimization in structural engineering, the performance of the optimized structure subjected to various loading scenarios during its lifetime service should be

Chapter 3 Isogeometric structural optimization of the post-tensioned concrete beam considered. The manufacturing cost and the maintenance cost (i.e., resilience) of the structure should also be considered. These concerns should be plugged into the construction-oriented topology optimization framework, and the future work of the authors will continue to explore it.

Chapter 4 An object-oriented framework for the isogeometric topology optimization

The soaring growth of computational power has enabled the unprecedented evolution of structural design practice [111, 173], which was represented by the milestone work of the Finite Element Method (FEM) [174]. FEM employed in structural analyses enables analysis models at various scales, which commonly comprise beam-column elements for frame members and shell elements for slabs or brick (solid) elements for a detailed model of structural components. In the conventional design framework, typical building structures, e.g., steel structures or reinforced concrete structures, usually consist of rectangular or circular sections limited by the current construction approaches. Dating back to the early 20th century, Gaudi has shown off his legendary attempts in architecture design pursuing optimal form [175] as shown in **Figure 4.1**. In recent years, modern architectural designs have re-entered this regime as represented by Zaha Hadid's masterpiece designs (also shown in **Figure 4.1**). Moreover, the evolution of construction technologies, such as prefabricated construction and 3D-printing construction [17, 176] has fundamentally freed the traditional design constraints, which potentially enables structures designed with the least material cost and

Chapter 4 An object-oriented framework for the isogeometric topology optimization optimal safety performance [106]. In this context, an adequate design tool using advanced computational modeling remains missing, while it is devoted to finding optimized structural design in the context of modern architecture design and sophisticated construction technologies.

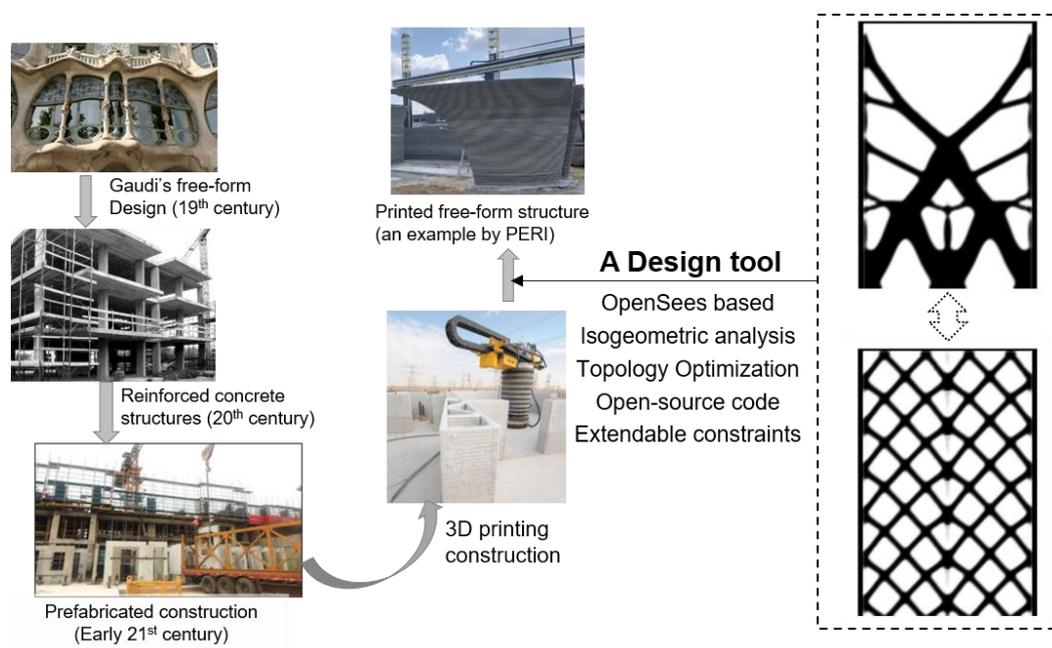


Figure 4.1 A design tool for future free-form structural design

Instead of manually conducting iterative designs with implicit attempts, executing automatic iteration is a key feature of topology optimization. Finite element models were usually employed in various software packages to obtain indices such as structural deformation and material usage. The FE analysis models during the iterations of the TO procedure are likely to experience complex geometry, which inevitably requires locally refined mesh to sustain FE analyses. While the number of optimization iterations increases, simultaneously generating mesh refinement according to the optimized model becomes nearly impossible. Isogeometric Analyses (IGA) as the latest development [177] exhibit an outstanding capability in tackling the compatibility challenge between mesh scheme and model geometry [70]. The Isogeometric Analysis concept was

Chapter 4 An object-oriented framework for the isogeometric topology optimization proposed by Hughes et al [65] as an extension of conventional FEM, which uses Non-Uniform Rational B-Spline (NURBS) for shape functions instead of the polynomial shape functions in FEM. Moreover, IGA adopts the same NURBS shape functions for geometry description [178], which can be seamlessly ported to computer-aided design (CAD) for the shared NURBS basis [179, 180]. Another benefit of using IGA is the increased computational accuracy achieved by elevating the order of shape continuity (p-refinement or k-refinement) without refining the mesh (h-refinement) [24, 181].

To implement IGA for topology optimization of structures [182, 183], an open-source and object-oriented FEM framework has been chosen for code development considering the sound experience of authors in this framework [184, 185], which was initially established as the Open System for Earthquake Engineering Simulation (OpenSees) by McKenna (1997) at UC Berkely. The code architecture of OpenSees is constructed for finite element analyses, consisting of material models, sections, elements, loads, solvers, etc. [157, 186-188]. OpenSees is primarily programmed in C++ and interpreted by Tool command language (Tcl) [189] and Python, which has attracted a variety of development for different purposes [184, 190-192]. The authors of this paper have been deeply engaged in developing OpenSees [157, 186-188], which has formulated a fire edition: OpenSees for fire, enabling fire modeling, heat transfer, thermo-mechanical analyses, and integrated analyses with GUIs. With these experiences, OpenSees has been extended in this paper with newly developed IGA elements to enable isogeometric analyses for the topology optimization process. An IGAQuad and an IGABrick element have been created for 2D analyses and 3D analyses in the FEM-based OpenSees framework, respectively. The NURBS-based geometry definition has been enabled along with the solved nodal displacements taking advantage of

Chapter 4 An object-oriented framework for the isogeometric topology optimization the existing components of OpenSees. The computational performance of these IGA elements has been examined and demonstrated, showing the outstanding capability of dealing with irregular-shape models compared to FEM. The topology optimization platform OPS-ITO is thereafter established, which is programmed in Python and developed to integrate the OpenSees IGA-based iterative analyses into the SIMP-based topology optimization. The developed codes have been encapsulated as an executable program with graphic user interface, which offers various constraint functions for modern structural design including stress limit, minimum material consumption, and pattern compliance.

4.1 Introduction of OpenSees framework

OpenSees is an open-source simulation framework initially developed for finite element analyses of structures subjected to earthquake loading [184], which was later extended by the authors for modeling structures in fires (OpenSees for fire) [157, 186-188]. Analyses in OpenSees follow the conventional routine of the finite element method, and the simulation process is disseminated into the following common steps: (1) mesh discretization; (2) formulation of stiffness matrices and residual forces in each finite element; (3) assembly of elemental stiffness matrices and force vectors; (4) solving the system of the equation; (5) check of convergence and updating elemental state.

4.1.1 The object-oriented programming infrastructure

As an object-oriented software framework for computing structural response

Chapter 4 An object-oriented framework for the isogeometric topology optimization using finite element analysis, OpenSees is implemented with C++ language and interpreted by Tool Command Language (Tcl) [189] and Python application programming interface (API) [157]. Benefiting from the open-source platform, allows researchers to further develop the source code in different research areas.

The main characteristic of object is its independence, which can be considered as an independent entity with corresponding data and functions for describing its attributes and properties. The functions of object are responsible for operating the contained data and can respond to the calling requests from other functions. Accordingly, object-oriented programming language is also called class-based language. The object is created by instantiating the class, and its internal structure is encapsulated to operate only through the interface, and the related computational process is not visible. Encapsulation makes the program architecture modular, thus adapting it for developing large software systems. Another important property is inheritance, which defines the subordination between classes. A subclass inherits data and functions from its base class and can produce new private components for saving redundant and repeated coding effort. The property of polymorphism means that the inherited methods can also be redefined in subclass for allowing different forms of input and outputs. A virtual function is an iconic form of dynamic polymorphism, which ensures the inherited function calls the implementation of the fundamental subclass for bypassing the originally declared function as the base class.

There are three types of relationships between classes, namely association, generalization, and aggregation that can be visualized using Unified Modelling Language (UML [193]), as illustrated in **Figure 4.2**, each class contains its attributes and operations. Association means the objects of different classes have interactions, which is illustrated with a line linking the corresponding classes.

Generalization illustrates the inheritance relationship between subclass and base class, and aggregation means the object of the assembly class is composed by the object of other classes

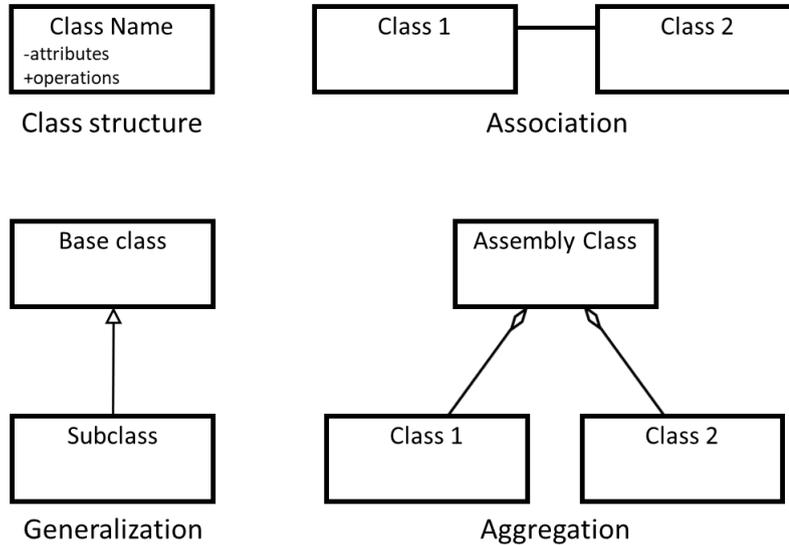


Figure 4.2 Illustration of different class relationships in the class diagram

4.1.2 OpenSees for finite element analysis

Taking an object-oriented architecture of source code development, these tasks are conducted in OpenSees within four basic modules: *ModelBuilder*, *Domain*, *Analysis*, and *Recorder* (blue boxes in **Figure 4.4**) along with a variety of supported packages. The *ModelBuilder* is responsible for building FE models after the mesh discretization, which adds the components of the FE model to the *Domain*. The *Domain* stores the information of an FE model, enabling updates during the FE analysis. As illustrated in **Figure 4.3**, the objects of the FE model (green boxes) stored in *Domain* include *Node*, *Material*, *Element*, *Constraint* (single point *SP_Constraint* and multi-point *MP_Constraint*, for boundary conditions), and *LoadPattern* (for imposing various load types). These objects are defined in various projects (modules) as inherited classes taking advantage of the

Chapter 4 An object-oriented framework for the isogeometric topology optimization object-oriented nature of C++. When the model is built and stored in *Domain*, finite element analyses can be carried out with the *Analysis module*, which formulates the global system of equations and solves the equation to obtain the converged solution of nodal displacements. To record the analysis results as output, the *recorder* includes various recorder objects for nodes and elements deployed under the request of the user, which retrieves the data from the corresponding model objects.

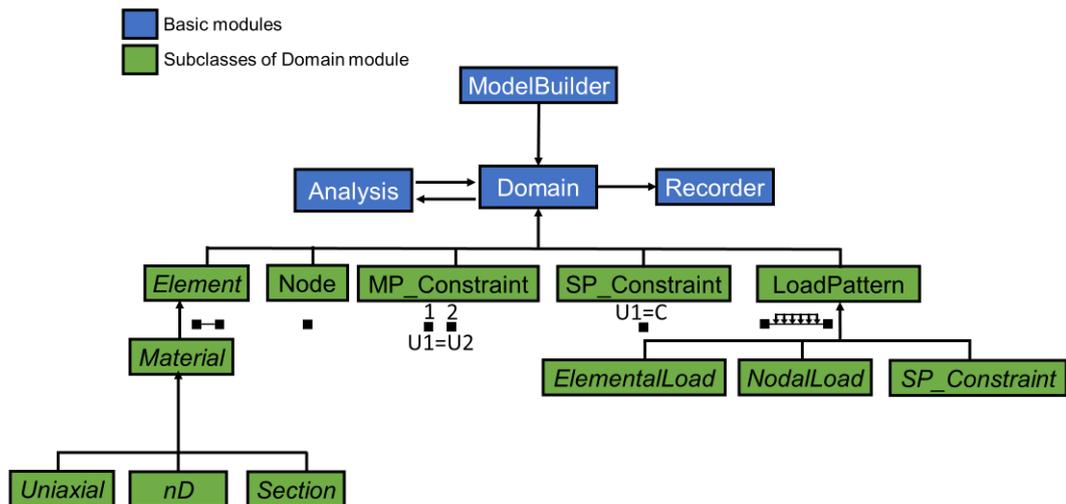


Figure 4.3 Class hierarchy of OpenSees framework

4.1.3 Interpreters of OpenSees

4.1.3.1 Tcl interpreter

As the most fundamental command interpreter of OpenSees, Tcl is a scripting language with a simple syntax and standard programming structure. Besides, Tcl embeds a user-friendly interface to extend the command library that allows OpenSees to give a standard entry, thus users can control OpenSees through Tcl script to complete all processes such as model defining and analysis solving instead of writing in complex C++ format. As exemplified in **Table 4.1** [194], Tcl commands are typically written in following format with hashtag (#) commented and the

variables are substituted by a dollar sign (\$).

Table 4.1 Example of Tcl script for defining a node in OpenSees

```
# "set" is a built-in command of Tcl, which assigns a value to the variable
set nodex 0;

# "expr" is an evaluation command, which computes the following expression
set nodey [expr $nodex+2];

# extended command from OpenSees
Node 1 $nodex $nodey;
```

4.1.3.2 Python API

Despite Tcl has shown outstanding flexibility and strength in string process, it is still not sufficient for increasingly complex scientific computing. String is the only native data type of Tcl, which results in cumbersome mathematical expressions and raises the cost of learning OpenSees. To remove the constraint of scripting-only expression and extend the usage possibilities of more scripting languages, it is necessary to adopt a new and versatile interpreter.

Python is a high-level combination of interpreted compiled, interactive, and object-oriented scripting language. Benefiting from its powerful extensibility, it has rich interfaces and is therefore known as a glue language. Glue language is a language that integrates well with modules developed in other languages, combining them organically to form new and more straightforward programs. Likewise, OpenSees also provides a Python interface that supports all features [195]. With its extension of Python interpreter, the users of OpenSees are also capable to use the extensive libraries of Python such as using Matplotlib for visualization, Scipy for scientific computing, and PyQt for establishing a graphical user interface. In addition, with the rich interfaces, OpenSees can be used as a

Chapter 4 An object-oriented framework for the isogeometric topology optimization structural simulation tool to achieve more complex computational tasks in collaboration with other software and scripts to accomplish hybrid programming.

4.2 Development of isogeometric analysis module in OpenSees

The major challenge of developing IGA capability in a FE framework is the different scopes and processes for element definition and state determination. IGA employs NURBS as shape functions with the nodes serving as control points of the splines. The model geometry and shape functions are both described using these splines. As depicted in **Figure 4.4**, the newly developed IGA element classes are denoted as IGAQuad and IGABrick for 2D and 3D isogeometric analyses, respectively. These elements are to be duplicated as objects and stored in *Domain*, which is now connected to the *ITO tool* package for running topology optimization processes with the IGA approach (discussed later). When developing IGA elements in OpenSees, the interface functions for the computation of characteristic components (e.g., stiffness matrix, load vector, etc) are inherited from the base class of *Element*. The fundamental functions of these interface functions are: (1) to perform in-element interpolation from nodal displacements (NURBS-based interpolation in IGA) to section deformation at each integration point; (2) to formulate resisting forces at nodes from the integration of section forces; (3) to formulate element stiffness matrix from the section stiffness matrices. Due to the use of NURBS, these in-element processes (also termed element state determination) are rewritten with NURBS-based interpolation and integration.

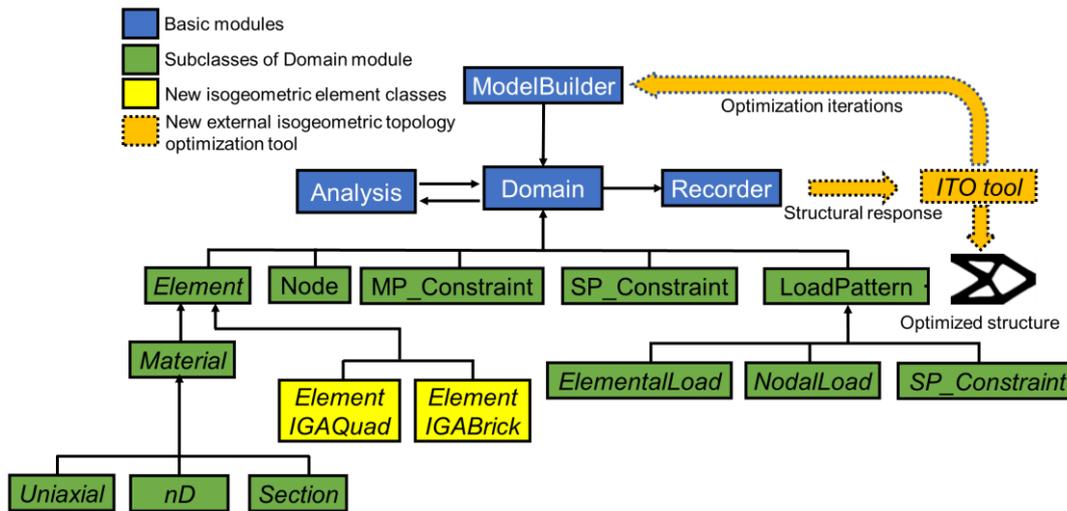


Figure 4.4 Class hierarchy of OpenSees framework with our ITO extensions

Taking a 2D model as an example, the computation of shape functions for each integration point is conducted within the procedure of *shapeFunction* in the *IGAQuad* element class, which serves as a key step to transform the Isogeometric formulation to the standard FE interface of OpenSees. The workflow of computing the IGA shape function is presented in **Figure 4.5**. The knot vector in an IGA element is used to find the belonging knot span of Gaussian points for determining the NURBS shape functions (represented by the blue shade) and the Jacobian matrix J_2 for mapping the parent space into parametric space (yellow shade). For each element, the belonging knot span index (*KntSpan*) of each gauss point (X_g & W_g) is computed from the function *findSpan* [66], which is followed by the calculation of basis function (N & $dN/d\xi$) for B-Spline (*DerBasisFuns*) and the rationalized R & $dR/d\xi$. As shown in **Figure 4.5**, two steps of space mapping are carried out to compute the corresponding Jacobian matrixes J_1 and J_2 to map the parent space into physical space, which is completed in the functions *SpaceMap#1* and *SpaceMap#2*. The stiffness determination is of a similar interface as general FE elements in OpenSees, whereas the NURBS-based shape functions are embedded in calculating the stiffness components and the elemental volume.

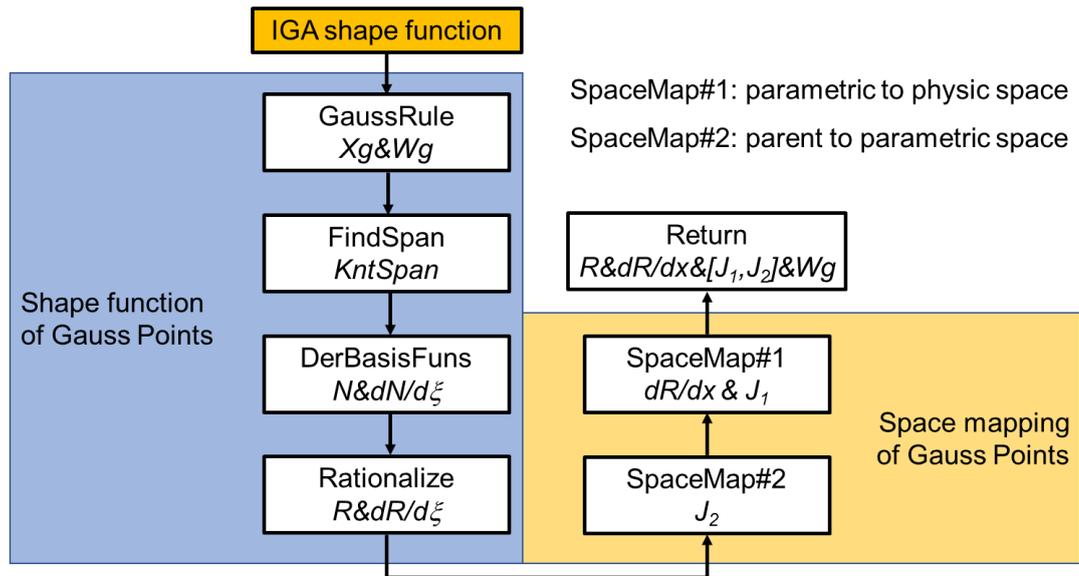


Figure 4.5 Workflow of computing IGA shape function

With the development of IGAQuad and IGABrick elements, isogeometric analyses can be now performed using OpenSees. The material definition should be specified such as an elastic 2D or 3D material, which is consistent with the original OpenSees user commands. When using the Python interpreter in OpenSees, the input command for defining the IGAQuad element can be written as follows:

```
element IGAQuad ex, nex, ey, ney, OrderX, OrderY, CPIs, nKntVectX, KntVectX, nKntVectY, KntVectY, rho_e, WCPs, thk, matTag;
```

Similarly, the definition script of the IGABrick element is in the following form:

```
element IGABrick ex, nex, ey, ney, ez, nez, OrderX, OrderY, OrderZ, CPIs, nKntVectX, KntVectX, nKntVectY, KntVectY, nKntVectZ, KntVectZ, rho_e, WCPs, matTag
```

In these command lines, the integers *ex*, *nex*, *OrderX* represent the element label, element amount, and element order along the x-direction, respectively, the format of y- and z-direction are in the same manner. *CPIs* is an integer array of the control point label of an element, while the integer *nKntVectX* represents the knot vector length along the x-direction (*KntVectX*), likewise in y- and z-direction.

The float number ρ_e is the elemental density that is specially added for the application of topology optimization, which is of a default value of 1. The float number $WCPs$ indicate the weight values of each elemental control point, whereas the integer $matTag$ is the label of the applied material. When generating the model using IGA elements in OpenSees, a more practical approach is to use Python-based functions to automatically define the model from the NURBS-based geometry. The pre-processing is enabled by taking the components from the open-source IGA tool SIMOPackage[196], which has been integrated into the pre-processing tool of the topology optimization framework in this paper.

4.3 Verification examples of using Isogeometric elements in OpenSees

4.3.1 2D model using IGAQuad: an infinite plate with a circular hole subjected to tension

The first benchmark case of the IGA element is to model an infinite plate with a circular hole, which is subjected to constant tension along the x-axis on both edges as shown in **Figure 4.6**. Considering the symmetry, this infinite plate can be simplified as a quarter plate with translational constraints along the x-axis at the right edge and translational constraints along the y-axis at the bottom edge. The analytical solution to the stresses of the quarter plate can be given as:

$$\begin{aligned}\sigma_{xx}(r, \theta) &= \frac{\sigma_0}{2} \left(2 + \frac{3R^4}{r^4} \cos 4\theta - \frac{R^2}{r^2} (3 \cos 2\theta + 2 \cos 4\theta) \right), \\ \sigma_{yy}(r, \theta) &= \frac{\sigma_0}{2} \left(-\frac{3R^4}{r^4} \cos 4\theta - \frac{R^2}{r^2} (\cos 2\theta - 2 \cos 4\theta) \right), \\ \sigma_{xy}(r, \theta) &= \frac{\sigma_0}{2} \left(\frac{3R^4}{r^4} \sin 4\theta - \frac{R^2}{r^2} (\sin 2\theta + 2 \sin 4\theta) \right)\end{aligned}\quad (4.1)$$

where the magnitude of constant tension (on the infinite boundary) is denoted as σ_0 . As shown in **Figure 4.6**, the exact center is the origin of polar coordinates. R and L are the hole radius and finite quarter plate length, respectively. Regarding material properties, the Young's modulus E , Poisson ratio ν , and thickness t of the planar plate is defined using the values presented in **Figure 4.6**.

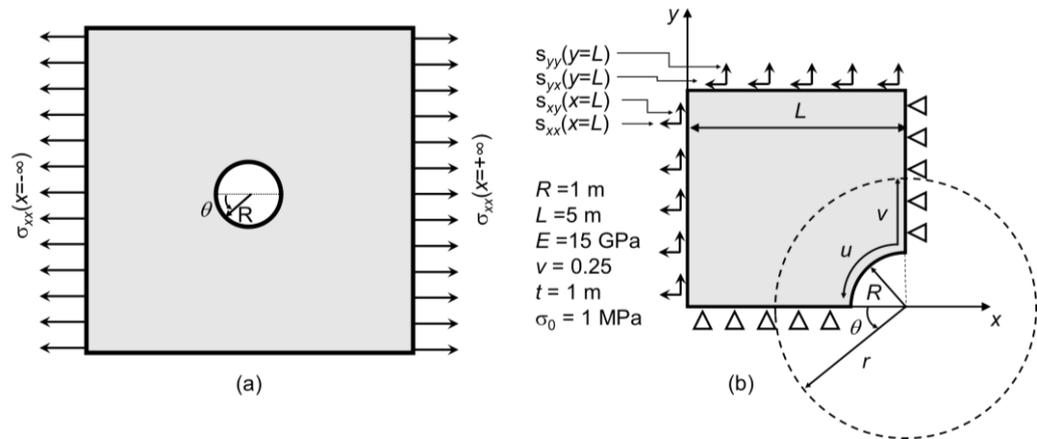


Figure 4.6 An infinite plate with a circular hole subjected to tension: (a) schematic of the model; (b) a quarter-plate model for analysis

This problem is analyzed using the newly developed IGA element (*IGAQuad*) in OpenSees to evaluate the element performance. A k-refinement process of IGA is used to enable convenient mesh refinement for benchmark tests, which simply elevates the order of the basis function by inserting additional knots to refine the IGA continuity without significantly adding control points (nodes) [66]. The corresponding models of Case 1-6 using order elevation and mesh refinement are illustrated in **Figure 4.7** to show the mesh schemes. For models of Case 1 to Case 3, the basis function order is gradually elevated from 2 to 6, while adopting the

Chapter 4 An object-oriented framework for the isogeometric topology optimization same mesh scheme (12×8 elements). The elevation of the NURBS order leads to the increasing number of control points from 140 to 308. For Cases 4 and 5, the mesh of elements is refined with doubled elements along each axis, while the NURBS order is kept as 2. Moreover, Case 6 adopts an extremely fine mesh, which should provide results sufficiently close to the analytical solution given in Eq. (4.1).

Table 4.2 IGA models for the infinite plate with a circular hole

Case	Order (u/v)	Control points (u/v)	Elements (u/v)
1	2	140 (14/10)	96 (12/8)
2	4	216 (18/12)	96 (12/8)
3	6	308 (22/14)	96 (12/8)
4	2 (doubled elements)	468 (26/18)	384 (24/16)
5	2 (four times elements)	1700 (50/34)	1536 (48/32)
6	2 (Extremely fine mesh)	10004 (122/82)	9600 (120/80)

The distributions of stress σ_{11} (normal stress along the x-axis) analyzed using the IGA models are illustrated in **Figure 4.8**, which include the contour plots from Case 1, Case 2 using elevated order, and Case 6 using a fine mesh scheme. Except for Case 1 of minor differences, Case 2 and Case 6 present nearly identical stress distributions. While the stress distribution of Case 6 is considered as the reference solution, the modeling results of Case 1-5 have shown that good computational performance can be achieved by either elevating the NURBS order or simply adding elements. Furthermore, the σ_{11} at point A and σ_{22} at point B are retrieved from the models and compared in **Figure 4.8**, where the analytical solutions of stresses σ_{11} at A and σ_{22} at B are given as straight lines. In **Figure 4.9**, the competitiveness of the IGA element is clearly shown in terms of order elevation. While the element order is elevated from 2 to 4 and 6 improving the continuity from C^1 to C^5 , the error percentage is down to less than 0.1% (**Figure 4.9(a)**).

Alternatively, mesh refinement (cases 4, 5) can achieve similar accuracy improvement, with the price being the significantly increased number of elements (Figure 4.9(b)).

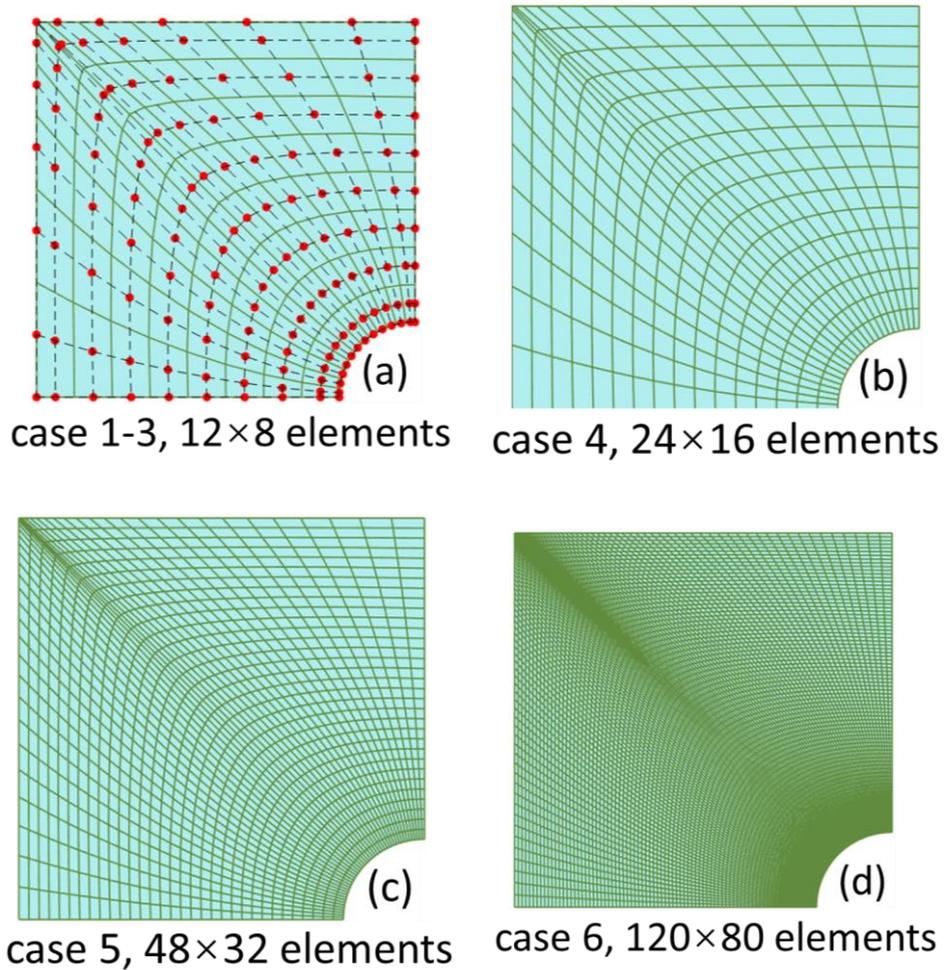


Figure 4.7 Various mesh schemes for modelling the quarter plate: (a) case 1-3, control points of case 1; (b) case 4; (c) case 5; (d) Case 6 (fine mesh)

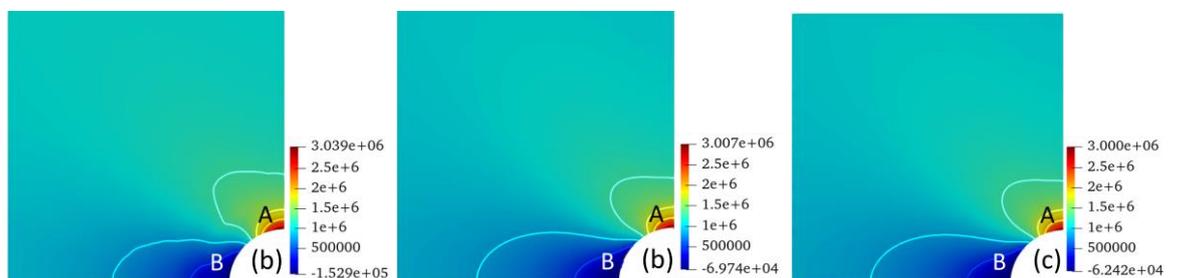


Figure 4.8 Contour plots of stress σ_{11} : (a) Case 1; (b) Case 2 using elevated element order (c) Case 6 using fine mesh as reference

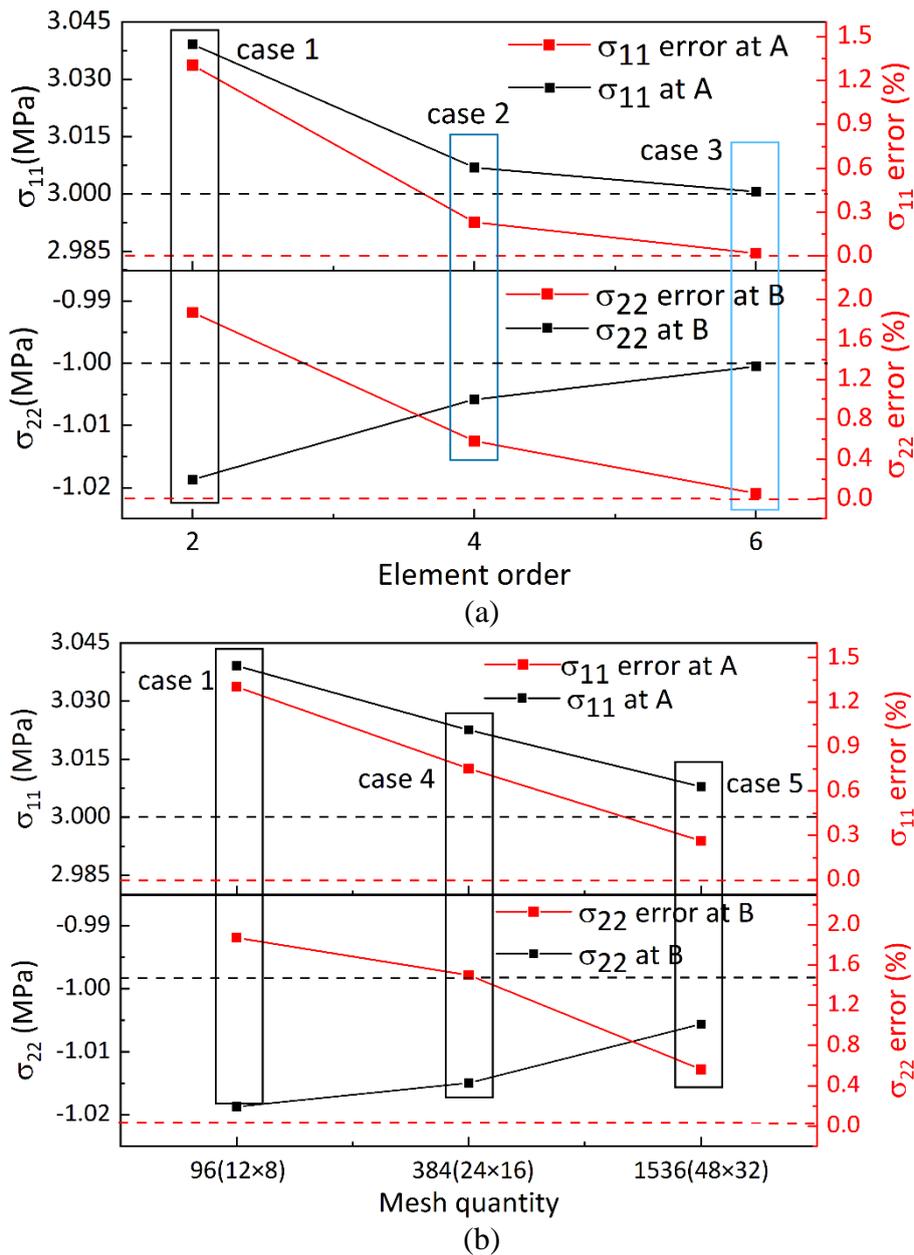


Figure 4.9 Accuracy of various IGA models: (a) elevated element order; (b) mesh refinement

4.3.2 3D model using IGABrick: a cylinder with internal pressure

The second benchmark case is to demonstrate the performance of the *IGABrick* element using a 3D case, which models a cylinder with uniform pressure on its internal surface, as shown in **Figure 4.10**. A quarter model is built for

Chapter 4 An object-oriented framework for the isogeometric topology optimization analyses using IGA elements and FE elements after considering symmetry. Here an FE model of a fine mesh analyzed using Abaqus [197] serves as the reference solution for performance comparison. For this 3D model, its geometric parameters, boundary conditions, and loading conditions are illustrated in **Figure 4.10**, whereas its material parameters are identically defined as the material model in the 2D plate cases.

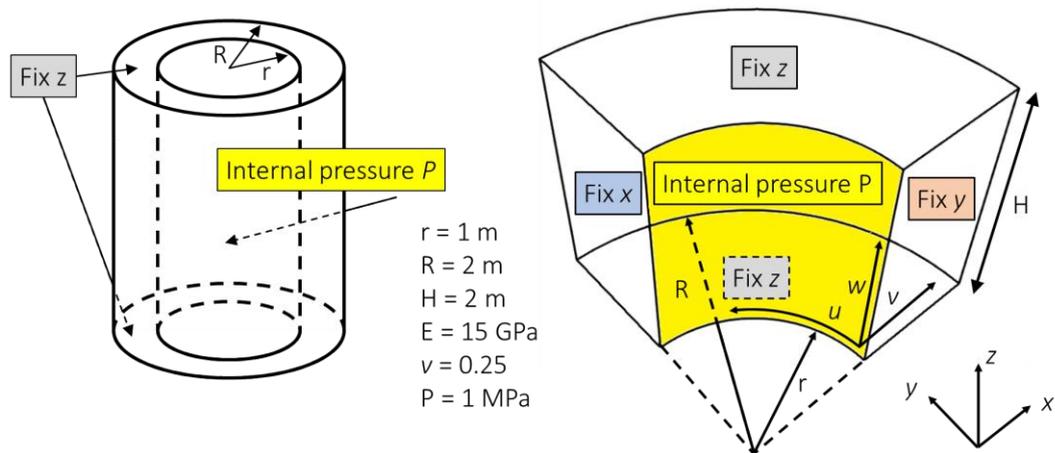


Figure 4.10 A hollow cylinder with internal pressure: (a) schematic of the model; (b) quarter model for numerical analyses

To examine the computational performance of the *IGABrick* element, the models of this 3D cylinder are varied from a relatively coarse mesh to the models of elevated NURBS order and refined mesh using *k-refinement*. Similarly, the various simulation schemes using the *IGABrick* elements are listed in **Table 4.3**. The modeled cases are numbered in four groups: A1~A4, B1~B4, C1~C4, and D1~D4. For these groups, the NURBS order has been changed in different directions of u , v , and w (as shown in **Table 4.3**) from 2 to 4. In each group, the mesh schemes are changed from a relatively coarse mesh to a fine mesh, which is of two additional elements in all directions as the case number increases. Correspondingly, the number of control points changes as a result of elevated order and refined mesh scheme, as shown in **Table 4.3**. Considering that the resolution

Chapter 4 An object-oriented framework for the isogeometric topology optimization over the thickness of the cylinder (v -direction) is crucial to the accuracy of analyses, relatively more knots are used than the directions of u and w .

Table 4.3 Simulation cases for modeling 3D cylinder with different order and mesh schemes

Case	Order (u/v/w)	Control points (u/v/w)	Elements (u/v/w)
A1		96 (4×6×4)	16 (2×4×2)
A2	2×2×2	288(6×8×6)	96 (4×6×4)
A3		640 (8×10×8)	288 (6×8×6)
A4		1200 (10×12×10)	640 (8×10×8)
B1~B4	2×3×2	112~1300 (4~10/7~13/4~10)	16~640 (2~8/4~10/2~8)
C1~C4	2×4×2	128~1400 (4~10/8~14/4~10)	16~640 (2~8/4~10/2~8)
D1~D4	3×3×3	175~1573 (5~11/7~13/5~11)	16~640 (2~8/4~10/2~8)
E1~E4	4×4×4	288~2016 (6~12/8~14/6~12)	16~640 (2~8/4~10/2~8)

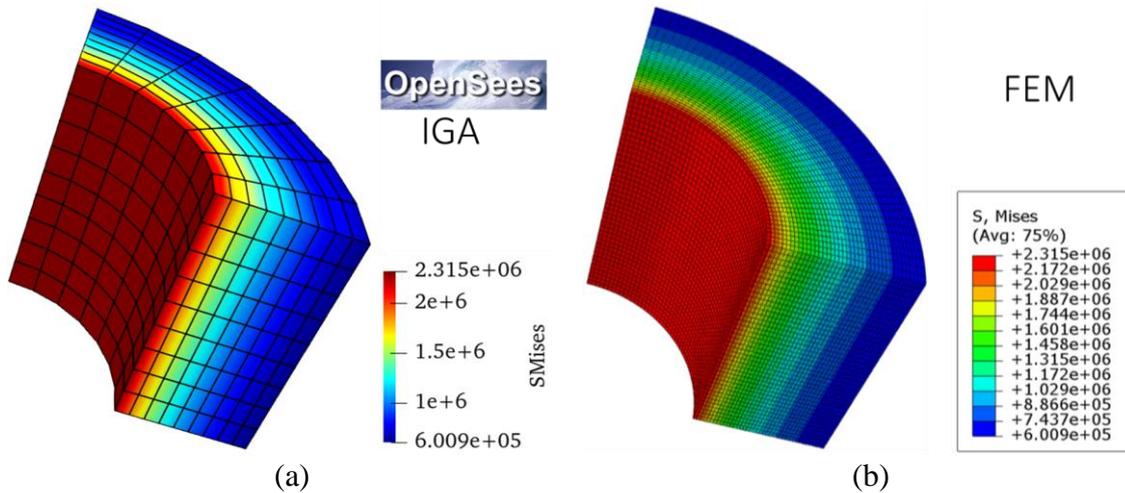


Figure 4.11 Mises stress distribution of case E4 and reference FEM case

The reference solution using a FEM model is built with $50 \times 40 \times 40$ isoparametric elements in ABAQUS. As presented in **Figure 4.11**, the Mises stress distribution of Case E4 and the reference FEM solution are illustrated as contour plots, which are very close in terms of stress distribution. After retrieving the maximum Mises stresses from the models of all the above-mentioned cases, the performance of the IGA element in OpenSees can be observed in **Figure 4.12**.

With the mesh refinement in A1~A4, the peak stresses quickly approach the reference solution, which is 2.315 MPa indicated by the dashed line. Moreover, the elevation of NURBS order in case groups B, C, D, and E is significantly more efficient. Even with the least elements ($2 \times 4 \times 2$ elements), the results have become very close to the reference solution. If comparing the case groups of B and C with the groups of D and E, a lower order over the directions of u and w causes a minor effect on the analysis accuracy. However, increasing the order over thickness direction (v) in the models as shown among the case groups of A, B, and C poses high sensitivity. Nevertheless, these analyses successfully demonstrate that the IGA model of much fewer elements provides nearly identical results. Like the IGAQuad element, the simulation accuracy can be ensured by solely increasing the NURBS order while keeping a relatively coarse mesh scheme. This unique feature ultimately facilitates the simulation in OpenSees-based topology optimization as it enables a fixed mesh scheme with an adaptively refined NURBS order during the iterations of model topology.

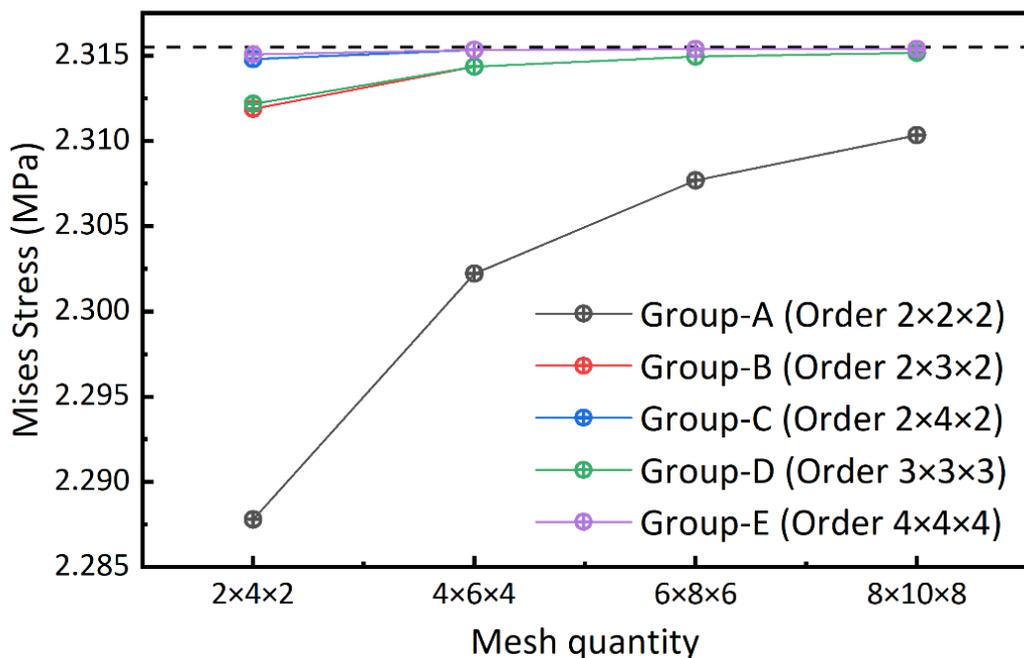


Figure 4.12 Computational performance of IGABrick elements in modeling a hollow cylinder using various modeling schemes

4.4 Development of isogeometric topology optimization package for the OpenSees

4.4.1 Basic optimization models

The developed OpenSees-ITO tool package currently supports two typical topology optimization models, a minimum compliance model and a minimum volume (with stress constraints) model [198]. These models are employed to define the objectives of topology optimization, which are briefly explained below:

4.4.1.1 Minimum compliance model

The minimum compliance model aims to find the optimal material distribution that minimizes structural compliance under the prescribed volume ratio. The optimization model is preliminarily defined as:

$$\begin{aligned}
 \min_{\boldsymbol{\rho}} \quad & c = \mathbf{F}^T \mathbf{U} \\
 \text{s.t.} \quad & V = \sum_{e=1}^{N_e} \rho_e V_e \leq \bar{V} \\
 & \mathbf{K}\mathbf{u} = \mathbf{F} \\
 & 0 \leq \boldsymbol{\rho} \leq 1
 \end{aligned} \tag{4.2}$$

The NURBS shape function is based on multiple input variables according to their order and mesh density, which is unlike the shape function of FE analysis limited to the domain of every single element. Since it is a natural density filter, the filter size relies on the order and size of the mesh, which leads to the unfavorable dependency of the filter size on the element sizes (model resolution). There are two major approaches to facilitate independent filter under the ITO framework: (1) to apply an additional distance-based filter[46]; or (2) to apply a multi-resolution approach [162], which uses the different sizes of NURBS shape functions in

Chapter 4 An object-oriented framework for the isogeometric topology optimization structural analysis (Eq. (3.7)) and in addressing the elemental density (Eq. (3.9)). The second approach utilizes the NURBS shape function as a filter, and the first approach is simpler when predicting the minimum length scale of the optimal structure in the context of construction-oriented topology optimization [119]. The distance-based filter is adopted in the present formulation, and the filter is expressed as:

$$\tilde{\rho}_e = \frac{\sum_{j \in N_i} w_{ij} \rho_j}{\sum_{j \in N_i} w_{ij}} \quad (4.3)$$

where w_{ij} is a linear distance weighting function given as:

$$w_{ij} = r_{\min} - \|\mathbf{x}_j - \mathbf{x}_i\| \quad (4.4)$$

In the above equation, r_{\min} is the filter radius, representing the cartesian distance between the centroid of i -th and j -th element. ρ_j is the density variable of the j -th element.

The projection is based on the filtered elemental density $\tilde{\rho}_e$ in Eq. (4.2). The formulation is expressed as:

$$\hat{\rho}_e = \frac{\tanh(\beta_{HS} \eta_{th}) + \tanh(\beta_{HS} (\tilde{\rho}_e - \eta_{th}))}{\tanh(\beta_{HS} \eta_{th}) + \tanh(\beta_{HS} (1 - \eta_{th}))} \quad (4.5)$$

Where $\hat{\rho}_e$ is the projected elemental density by using the tanh function [199] to enhance the binarization degree of density variables. As the sharpness value β_{HS} increases, the density binarization is strengthened with the threshold η_{th} .

After aforementioned modifications, the optimization model can be updated from Eq. (4.10) as:

$$\begin{aligned}
 \min_{\boldsymbol{\rho}} \quad & c = \mathbf{F}^T \mathbf{U} \\
 \text{s.t.} \quad & V = \sum_{e=1}^{N_e} \hat{\rho}_e V_e \leq \bar{V} \\
 & \mathbf{K}\mathbf{u} = \mathbf{F} \\
 & 0 \leq \boldsymbol{\rho} \leq 1
 \end{aligned} \tag{4.6}$$

The sensitivity of the volume constraint $V(\boldsymbol{\rho})$ with respect to the design variable control point density can be given as:

$$\frac{\partial V}{\partial \boldsymbol{\rho}} = \sum_{e=1}^{N_e} \frac{\partial V}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial \rho_i} \tag{4.7}$$

where the term $\frac{\partial \hat{\rho}_e}{\partial \rho_i}$ follows the chain rule, which is sequentially derived

from the density filter in Eq. (4.3) and the projection in Eq. (4.5) given as below:

$$\frac{\partial \hat{\rho}_e}{\partial \rho_i} = \frac{\partial \hat{\rho}_e}{\partial \tilde{\rho}_e} \frac{\partial \tilde{\rho}_e}{\partial \rho_e} \frac{\partial \rho_e}{\partial \rho_i} \tag{4.8}$$

The sensitivity of compliance objective $c(\boldsymbol{\rho})$ is:

$$\frac{\partial c}{\partial \rho_i} = - \sum_e \mathbf{U}_e^T \frac{\partial \mathbf{K}_e}{\partial \rho_i} \mathbf{U}_e \tag{4.9}$$

where the term $\frac{\partial \mathbf{K}_e}{\partial \rho_i} = \frac{\partial \mathbf{K}_e}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial \rho_i}$ is derived by Eq. (3.9) and Eq. (4.8).

4.4.1.2 Minimum volume model with stress constraint

The minimum volume model is used to find the lightweight structure under prescribed Mises stress constraints to ensure structural safety and resilience. This optimization model can be preliminarily described as:

$$\begin{aligned}
 \min_{\boldsymbol{\rho}} \quad & V = \sum_{e=1}^{N_e} \rho_e V_e \\
 \text{s.t.} \quad & \begin{cases} \mathbf{K}\mathbf{U} = \mathbf{F} \\ \sigma_{e,i}^M \leq \bar{\sigma}, \quad 0 \leq i \leq N_{GP} \\ 0 \leq \boldsymbol{\rho} \leq 1 \end{cases}
 \end{aligned} \tag{4.10}$$

where the Mises stress at each Gaussian point $\sigma_{e,i}^M$ is constrained below the stress limit $\bar{\sigma}$. For the sake of formulation simplicity, the density at the elemental center is taken as the density of all corresponding Gaussian points. To interpolate the stress value of the intermediate density variable and tackle the stress singularity issue, the q-p stress relaxation scheme as suggested in [59] is applied. Therefore, the penalized Mises stress is expressed as:

$$\begin{aligned}\sigma_{e,i}^M &= \left(\boldsymbol{\sigma}_{e,i}^T \mathbf{V} \boldsymbol{\sigma}_{e,i} \right)^{\frac{1}{2}} \\ \mathbf{V} &= \begin{bmatrix} 1 & -1/2 & 0 \\ -1/2 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \\ \boldsymbol{\sigma}_{e,i} &= \hat{\rho}_e^q \mathbf{D}_0 \mathbf{B}_{e,i} \mathbf{u}_e\end{aligned}\tag{4.11}$$

where $\mathbf{B}_{e,i}$ is the strain-displacement matrix at the i -th gauss point of the element. Likewise, the Drucker-Prager stress is also supported to enable unequal tension-compression stress limit, the formulation is available from our former paper[119].

Another challenge is the high computational cost induced by a large number of local stress constraints, which can be reduced using Kreisselmeier-Steinhauser(K-S) or p-norm [61, 75, 200] function to aggregate all local constraints as a single global constraint. The K-S function is here applied considering the convenient development of future optimization models:

$$G_{KS}(\boldsymbol{\rho}) = \frac{1}{\eta} \ln \sum_e \sum_i^{N_{gp}} \exp \left(\eta \frac{\sigma_{e,i}^M}{\bar{\sigma}} \right) \leq 1\tag{4.12}$$

where the aggregation factor η is a positive value, which is to penalize the violated local constraints. A higher η helps to reduce the approximation error of stress measurement but brings higher nonlinearity and convergence difficulty. To scale the approximation to a reasonable level, an adaptive constraint scaling (ACS)

Chapter 4 An object-oriented framework for the isogeometric topology optimization scheme [77] is adopted. It offers more accurate stress measurement with a relatively low factor and it also stabilizes the iteration process. The ACS-corrected stress evaluation is described as:

$$\tilde{G}_{KS}^{(I)} \approx c_{ACS}^{(I)} G_{KS}^{(I)}$$

$$c_{ACS}^{(I)} = \begin{cases} \frac{\sigma_{\max}^{(I)}}{G_{KS}^{(I)}}, & I \leq 2 \\ \frac{\sigma_{\max}^{(I-1)}}{\alpha_{ACS}^{(I)} G_{KS}^{(I-1)}} + (1 - \alpha_{ACS}^{(I)}) c_{ACS}^{(I-1)}, & I > 2 \end{cases} \quad (4.13)$$

where the ACS factor $c_{ACS}^{(I)}$ at I -th iteration is weighted by the stress measurement of past iterations, and the weight factor $\alpha_{ACS}^{(I)}$ is suggested as constant 0.5 after 2 initial iterations. As pointed out by Le et al. [77], this weighted ACS factor converges at a constant throughout the iteration, and the influence is gradually reduced to the optimum.

After aforementioned modifications, the optimization model can be updated from Eq. (4.10) as:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & V = \sum_{e=1}^N \hat{\rho}_e V_e \\ \text{s.t.} \quad & \mathbf{KU} = \mathbf{F} \\ & \tilde{G}_{KS} \leq 1 \\ & 0 \leq \boldsymbol{\rho} \leq 1 \end{aligned} \quad (4.14)$$

The sensitivity of volume constraint presented in Eq. (4.7), which is along with the adjoint method for the stress constraint sensitivity analysis:

$$\phi = G_{KS} + \boldsymbol{\lambda}^T (\mathbf{KU} - \mathbf{F}) \quad (4.15)$$

where the adjoint vector $\boldsymbol{\lambda}$ is arbitrary, and the adjoint formulation sensitivity is given by the chain rule:

$$\frac{\partial \phi}{\partial \rho_i} = \sum_e \sum_i^{N_{\text{els}}} \frac{\partial G_{\text{KS}}}{\partial \sigma_{e,i}^M} \begin{pmatrix} \frac{\partial \sigma_{e,i}^M}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial \rho_i} \\ \frac{\partial \sigma_{e,i}^M}{\partial \mathbf{U}_e} \frac{\partial \mathbf{U}_e}{\partial \rho_i} \end{pmatrix} + \boldsymbol{\lambda}^T \begin{pmatrix} \frac{\partial \mathbf{K}}{\partial \rho_i} \mathbf{U} + \mathbf{K} \frac{\partial \mathbf{U}}{\partial \rho_i} \end{pmatrix} \quad (4.16)$$

To eliminate the component with the implicit term $\frac{\partial \mathbf{U}_e}{\partial \rho_i}$, a specific adjoint vector

$\boldsymbol{\lambda}$ is solved by:

$$\frac{\partial G_{\text{KS}}}{\partial \sigma_{e,i}^M} \frac{\partial \sigma_{e,i}^M}{\partial \mathbf{U}_e} + \boldsymbol{\lambda}^T \mathbf{K} = 0 \quad (4.17)$$

where the term $\frac{\partial G_{\text{KS}}}{\partial \sigma_{e,i}^M}$ is the sensitivity of K-S aggregation in Eq. (4.12). $\frac{\partial \sigma_{e,i}^M}{\partial \hat{\rho}_e}$ and

$\frac{\partial \sigma_{e,i}^M}{\partial \mathbf{U}_e}$ are from Eq. (4.11).

4.4.2 Details of the optimization package

Based on OpenSees-IGA and SIMP formulation, an ITO tool with a graphical user interface has been developed to perform topology optimization with the developed IGA in OpenSees. As depicted in **Figure 4.13**, the OPS-ITO tool package comprises three modules: (1) ITO tool: A toolset with a graphical user interface (GUI) embedding the middleware and interface ported to the analysis module and TO module. The GUI provides an entry for defining the model geometry, mesh, and boundary conditions to be used in OPS-IGA and the optimization parameters for TO iterations. For visualization after the optimization procedure, a middleware is established to export jpg/vtk files for 2D and 3D models to support the display of displacements, strains, and stresses. (2) OPS-IGA Module: When receiving an OpenSees input file (*Tcl* or *Python* script), the

Chapter 4 An object-oriented framework for the isogeometric topology optimization simulation model will be created in OpenSees using the corresponding IGA elements (*IGAQuad* or *IGABrick*). In each iteration, the results from the completed analysis will be recorded using nodal and elemental recorders and transferred to the TO module and GUI. (3) TO module: It is responsible for performing topology optimization with various sensitivity functions for different optimization goals, such as minimum compliance, minimum volume with material strength, pattern compliance to control the number of variations, and other constraint functions of potential interest. The density of control points is generated from the TO procedures and transferred to the IGA models for iterative analyses.

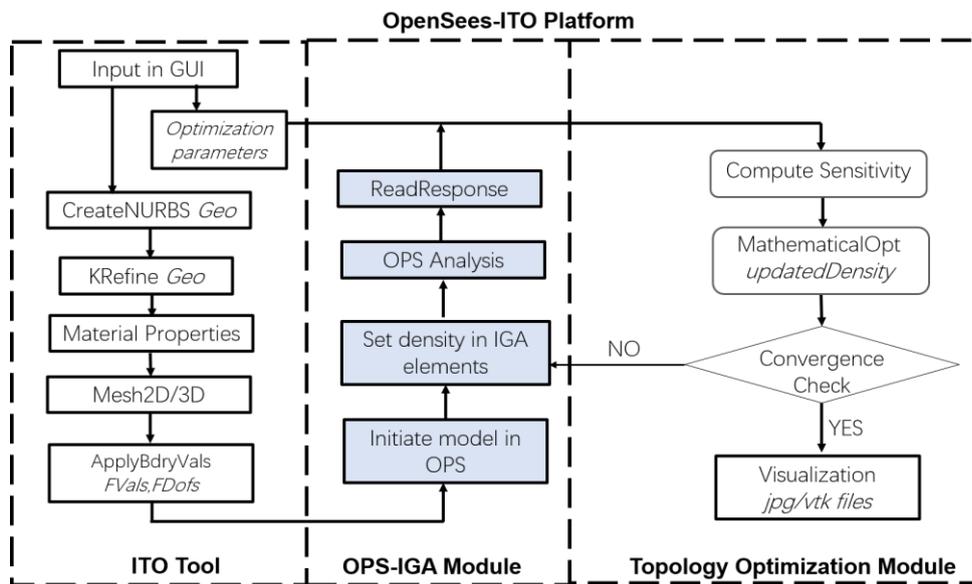


Figure 4.13 OPS-ITO framework comprising model setup, IGA analysis, and optimization module

The OPS-ITO framework has been packed as a tool package and can be executed using an executable program with GUI, which is illustrated in **Figure 4.14**. The currently available input parameters are listed in **Figure 4.14** (a~c), while the model visualizations are set up in **Figure 4.14** (d~e). It presents the real-time topology configuration and the iterative histories of objective indices and constraints. 2D and 3D problems have been enabled, which allows optimization using the typical optimization models as *minCompliance* (compliance

Chapter 4 An object-oriented framework for the isogeometric topology optimization (volume minimization with volume constraint) and *minVolume* (volume minimization with Mises stress constraint).

To launch the optimization analysis, the input parameters are required in three categories: (a) model dimension, optimization type, and geometric model, (b) material, boundary, and load conditions, and (c) optimization parameters. After setting the 2D/3D option and optimization type option, the model geometry can be defined for two types of models: (1) Simple models: Frequently used rectangular or brick types of initial design space can be activated ‘*By setting*’, which allows users to define the size of design space, element number and orders of each direction. (2) Complex models: The option ‘*By input file*’ can be chosen for models of irregular design space using the pre-process of SIMOPackage [196] as introduced before. The information of generated geometric model is stored as the *Surf/Volu* variables saved as a Matlab file (.mat), which can be imported by the OPS-ITO executable.

The material definition of the model takes advantage of the extensive collection of materials in OpenSees as an open-source simulation platform. Currently, the isotropic linear elastic material is used as a default material, which can be further extended by co-working with the OpenSees isogeometric analyses. Moreover, the definition of loading and boundary conditions can be conducted using the pre-processing tool of the SIMOPackage. Additional definitions of loading or boundary conditions are allowed on the user interface using a script line comprising a boundary index, a type label, and a distribution expression. The typical face indices of 2D and 3D models are presented in **Figure 4.15**. The type label allows *UX*, *UY*, *UZ* for boundary conditions and *FX*, *FY*, *FZ*, *PRESS* for load conditions, and the 2D/3D function handle expression begins with *@(x,y)/@(x,y,z)*.

As shown in **Figure 4.14**, the category of (c) OptParameters defines various

Chapter 4 An object-oriented framework for the isogeometric topology optimization parameters for the topology optimization process, which are corresponding to the previously mentioned target constraint functions. The *ConstValue* (volume limit \bar{V} in Eq.10 or Mises stress limit $\bar{\sigma}$ in Eq.15) indicates the maximum value of constraint, while the maximum number of iterations is given as *MaxIter*. *MMAMoveLimits* sets the move limit of the MMA optimization algorithm, and *Penalty* indicates the density penalty order of the SIMP method. *Projection* (β_{HS}) is the sharpness value of the erode/dilate Heaviside function, which determines the 0-1 solution degree of optimized topology configuration, and the projection threshold is set by *ProjThreshold* (η_{th}). To accelerate the convergence and optimal topology generation, the stepped penalty and projection can be respectively activated by *StpPenalty* and *StpProj*, these two parameters default raise 1 per 10 iterations, and the function of the minimum component size scale is controlled by *r_min* (r_{min}) to avoid possible over-thin components and improve the constructability.

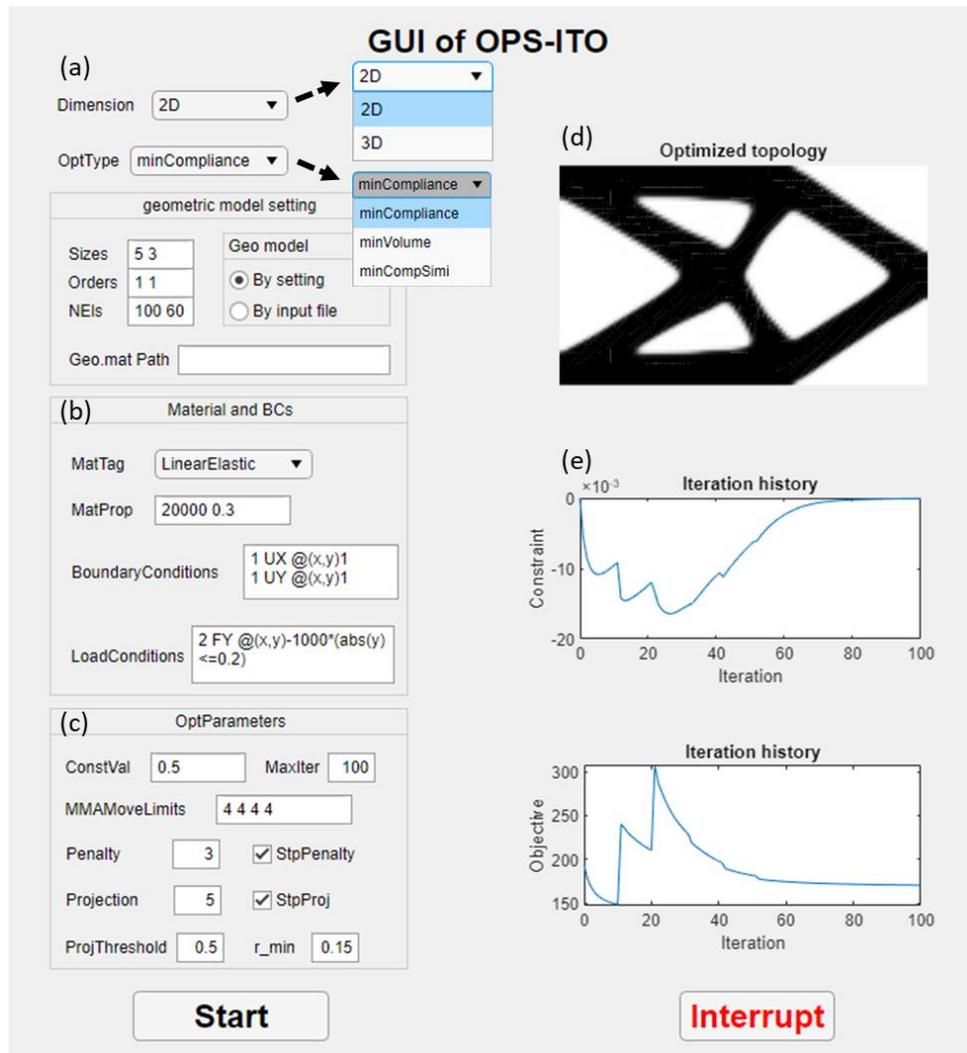


Figure 4.14 Graphical user interface of OPS-ITO platform

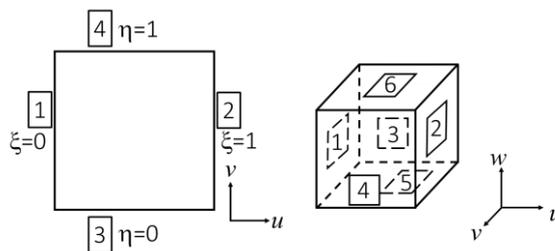


Figure 4.15 Illustration of elemental boundary indexes (boxed): (a) 2D model; (b) 3D model

4.5 Demonstrative examples of using the OPS-ITO package

To demonstrate the use of the OPS-ITO framework for structural design optimization, four cases have been conducted and briefly described regarding the different optimization schemes. A 2D cantilever beam is optimized towards *minimum compliance*, i.e., the stiffest beam with a prescribed volume ratio. An L-shape beam is optimized for minimum volume ratio under various stress constraints of structural materials, followed by a 3D cantilever beam case to showcase the 3D optimization under minimum compliance. The last case is to show the optimization of a structural system reflecting a long-term vision, which considers pattern compliance regarding manufacturability and aesthetic performance.

4.5.1 A planar quarter annulus structure design using the minimum compliance model

A 2D quarter annulus is initially assumed 2m outer radius and 1m inner radius, subjected to a downward load at the top right free end. As shown in **Figure 4.16**, the load is distributed from 1.95m to 2.0m along the x-axis with the load density of 5 kN/m. The geometric model meshes as a 2-order grid of 240×48 IGAQuad elements of plane stress material. Using the linear elastic material, its Young's modulus is defined as 2×10^5 MPa and the Poisson ratio is 0.3. In this optimization case, it is aimed to find a beam design of maximum stiffness (minimum compliance) while the volume ratio is assigned as 50% of the initial domain. All the relevant input parameters required by the OPS-ITO platform are listed in **Table 4.4**. To enable the user-defined design domain, a geometric file is allowed as the input to replace the default rectangular domain. Noted that the boundary indexes are labeled as given in **Figure 4.15** when defining the Boundary

Conditions and Load Conditions. In this case, the density filter size is assigned as 0.15 m to avoid slim components.

Table 4.4 Input parameters to OPS-ITO for optimization of a quarter annulus structure

Parameter	Value	Parameter	Value
Dimension	2D	OptType	minCompliance
Geo model	User input	Geo.mat Path	quartAnnulus.mat
MatTag	LinearElastic	MatProp	2e11 0.3
BoundaryConditions	1 UX @(x,y)1 1 UY @(x,y)1	LoadConditions	4 FY @(x,y)-5e3*(x>=1.95)
ConstValue	0.5	MaxIter	300
MMAMoveLimits	2 1 1 2	r_min	0.15
Penalty	3	StpPenalty	√
Projection	5	StpProj	√
ProjThreshold	0.5		

The initial density variables are uniformly initialized as 0.5 to start the optimization iterations. When the stepped penalty (*StpPenalty*) and projection (*StpProj*) are activated, the penalty value (*Penalty*) starts at 1 and increases with 1 per 10 iterations until it reaches 3 to ensure a clear topology. Likewise, the projection sharpness value (*Projection*) is also initialized as 0.1 to approximate the non-projected state, which is gradually increased by 1 per 10 iterations with a threshold (*ProjThreshold*) of 0.5. The optimization history is presented in **Figure 4.16** (b) with respect to the iterations, including the indexes of compliance and volume ratios at each iteration. It can be seen the stepwise fluctuation from the stepped penalty and projection disappears after 30 iterations as the optimized topology is formulated, and the optimization converged to an eventual compliance of 18.11N•mm after 300 iterations. Meanwhile, the volume ratio of the optimized topology reaches 0.5, and high sharpness of density distribution is achieved as shown in **Figure 4.16** (b-c). Observing the optimized density distribution, the

obtained topology exhibits decent smoothness, continuity, and distinct boundaries between solid and void segments.

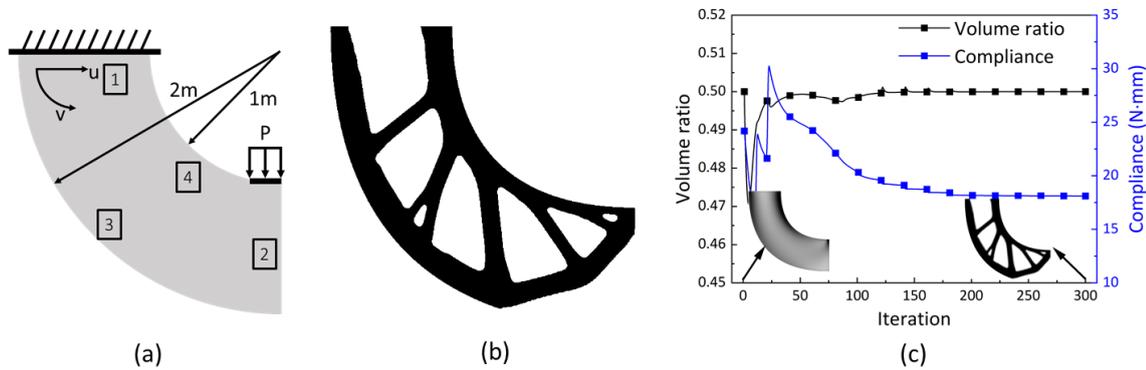


Figure 4.16 Optimization of a plane stress quarter annulus example: (a) model schematic; (b) optimized topology; (c) iteration history

4.5.2 A planar L-shape beam design using the minimum volume model with stress constraint

The strength limits of construction materials are a practical concern for real structural design, which shall be included in the OPS-ITO package. An L-shape beam is modeled for demonstration of this purpose, which is fixed at the top edge with a downward load of 250 kN at the free end ($5e6 \text{ N/m}$, $x \geq 1.95$) as shown in **Figure 4.17**. The initial design domain of this L-shape is defined by using the *SIMOPackage* by giving a readable file through the ‘*input file*’ option in the GUI, which contains the *Surf* variables of the model. For the optimization parameters, lower MMA move limits and a higher maximum of iteration number 400 are adopted since the stress constraint of structural material introduces difficulties to the convergence. The filter size is reduced to 0.01 to allow for slightly slimmer components with high-stress levels. The input parameters for GUI are listed in **Table 4.5**.

Regarding the stress constraint, the Mises and Drucker-Prager strength limits

are applied separately to this L-shape beam, which is usually suitable for steel sections and concrete sections (including fiber-reinforced concrete), respectively. The K-S aggregation factor is default set as 8, and the Mises stress state is corrected by the ACS method in Eq. (4.13). The stress contour plots are shown in **Figure 4.17**, where the effect of imposing the Mises strength limit is shown in **Figure 4.17** (a) and the effect of imposing the Drucker-Prager strength limit is given in **Figure 4.17** (b). The contour plots represent the ratios of corresponding stress terms over the assigned strengths, which are iteratively changing as a result of the model. For a concrete type of material of different strengths for tension and compression, the eventual topology of the L-shape beam requires stronger components in the tension region. This strength limit enabled in topology optimization can be later applied for designs of real 3D printable structures, only requiring the corresponding material properties [28] prior to the design.

Table 4.5 Input parameters for optimization of an L-shape beam

Parameter name	Value	Parameter name	Value
Geo model	User input	Geo.mat Path	stressCase.mat
MatTag	ElasticIsotropic	MatProp	4e10 0.3
BCs	1 UX @(x,y)1	Loads	4 FY @(x,y)-5e6*(x>=1.95)
	1 UY @(x,y)1		
ConstVal	10e6	MaxIter	400
MMAParams	1 0.5 0.5 1	Penalty	3
StpPenalty	×	Projection	0.1
StpPrj	×	ProjThreshold	0.5
rmin	0.01		

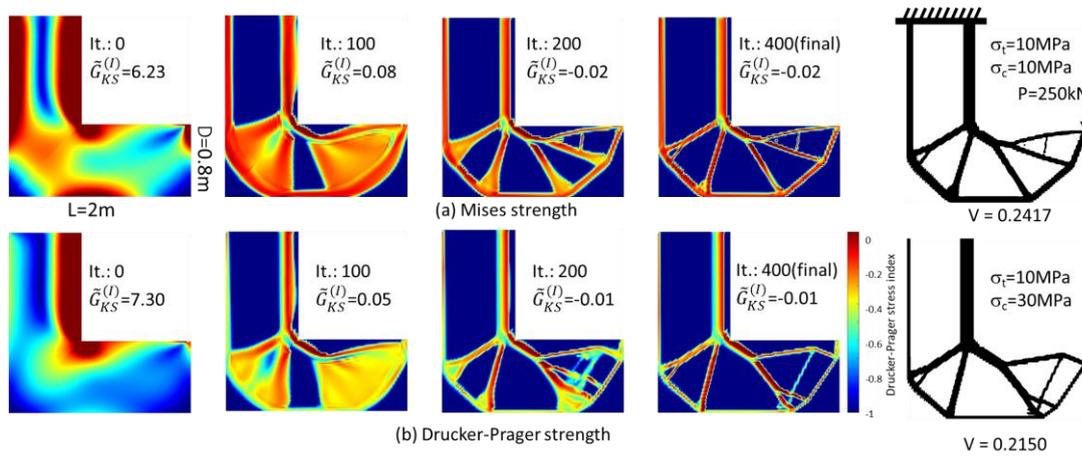


Figure 4.17 Optimization case of L-shape beam for light weight with material strength constraints: (a) using Mises stress constraint; (b) using Drucker-Prager stress constraint

4.5.3 A 3D cantilever beam design using minimum compliance model

In addition to the 2D planar cantilever beam optimized in Section 5.1, it can be either optimized regarding a 3D domain using IGABrick elements and the OPS-ITO platform. This further alleviates the constraint along the thickness direction during the optimization iterations. The *minCompliance* module for a 3D domain is employed using the boundary conditions and material properties identical to the 2D beam except for the thickness of 1 m. The input parameters of this 3D cantilever beam are listed in **Table 4.6**.

Table 4.6 The input parameters of the 3D cantilever beam case

Parameter name	Value	Parameter name	Value
Geo model	Default	Sizes	5 3 1
Orders	1 1 1	NEls	30 18 6
MatTag	ElasticIsotropic	MatProp	2e11 0.3
	1 UX @(x,y,z)1		2 FY @(x,y,z)-
BCs	1 UY @(x,y,z)1	Loads	1e5*(abs(y)<=0.05)*(abs(z-0.5)<=0.05)
	1 UZ @(x,y,z)1		
ConstVal	0.3	MaxIter	100
MMAParams	4 2 2 4	Penalty	5
StpPenalty	√	Projection	5
ProjThreshold	0.5	ProjThreshold	0.5
rmin	0.3		

In **Figure 4.18**, the variations of beam topology are shown alongside the iteration tags. When the optimization iteration proceeds, the rendering of beam topology gradually becomes crisp from the initially gloomy topology. The volume ratio converges to 0.3 after the first 20 iterations, which is accompanied by the compliance index reducing to 352.76N*m upon the completion of 100 iterations as illustrated in **Figure 4.18** (d). Compared to the 2D topology, hollow sections, and voids are seen inside the beam, and its behavior may be similar to a space truss. 3D printing construction is more suitable for this type of beam design after optimization. This enables the most efficient use of construction materials and it would potentially reduce the carbon footprint at the construction stage, especially when this technique is combined with the use of other sustainable structural materials in the era of pursuing carbon neutrality.

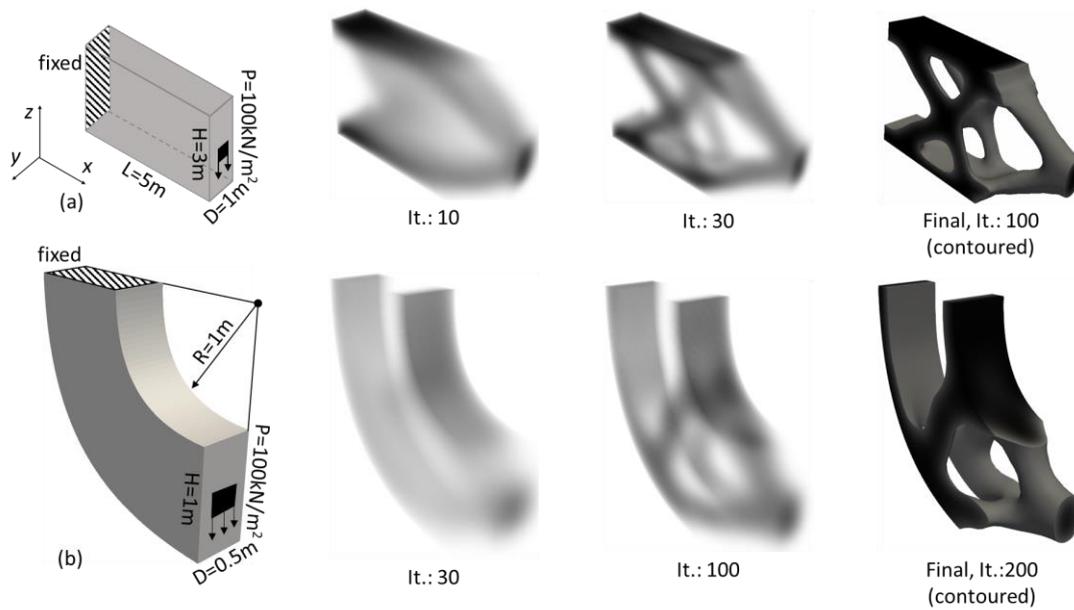


Figure 4.18 Optimization of 3D cantilever beam: (a) a rectangular beam; (b) a curved beam

4.6 Concluding remarks

In this chapter, an OPS-ITO platform has been established, which employs IGA elements to perform analyses of structural members or structural systems for the iteration of topology optimization. The SIMP-based optimization is developed and conducted in this OPS-ITO framework, which can find the optimal distribution of materials in a design domain aiming for minimum volume or minimum compliance with the consideration of material strength limits. The OPS-ITO tool package reflects the authors' vision of future building design, which first ever attempts to provide a design tool to resolve the need as the structure is unnecessarily designed with conventionally regular shapes. The work presented in this paper can be summarized as follows:

- (1) The IGA elements for 2D and 3D structural analyses have been developed in the open-source simulation platform OpenSees, which uses NURBS for

Chapter 4 An object-oriented framework for the isogeometric topology optimization geometric description and shape functions in element state determination. NURBS as the basis of computer-aided design provides seamless solutions from model geometry to the numerical description.

(2) The outstanding performance of IGA elements has been demonstrated using 2D and 3D benchmark studies. IGA models can achieve accurate results by simply elevating NURBS order rather than the traditional mesh refinement.

(3) An OPS-ITO tool is developed to integrate the graphic user interface, pre-processing, OPS-IGA modeling, and optimization process. Standard 2D or 3D minimum compliance models and minimum volume models with stress constraints have been developed as optimization objectives and constraint functions. The optimized geometric model can be exported for further post-processing or 3D printing.

(4) Three case studies using the OPS-ITO package are discussed, including the optimization of beams in 2D and 3D domains, and an L-shape beam with Mises and Drucker-Prager strength limits.

(5) The OPS-ITO package is a platform for structural design, which is motivated by the need for automated construction and integrated design. It is expected to promote 3D printing in building construction and to bridge the architectural design with structural engineering.

Chapter 5 Between free-form and periodic design - a multi-pattern approach

5.1 Introduction

Modern building construction has entered a new era that adopts prefabrication and modular design, the structural design paradigm ought to accordingly adapt to this evolution and potentially integrates with architectural design and building service design. Topology optimization is favored for industrial applications because of its effectiveness as a numerical form-finding approach that pursues optimal structural performance via determining material distribution [144]. However, the topology optimization method generally yields free-form designs that are structurally efficient but costly for manufacturing [201]. Benefiting from the development of prefabrication technology [13], periodic structures can be fabricated in the factory and assembled on-site, so periodic topology optimization has gained interest due to its manufacturability. The development of the periodic design method is based on microscopic [202] and macroscopic [143] perspectives, respectively. The former treats the unit cells as infinitely small and optimizing

microscale materials while the latter is the scope of this study, which optimizes them for finite sizes of large macrostructures.

The periodic design usually divides the structural design domain into multiple sub-domains, and each sub-domain is of an identical topology after the optimization, which is termed a representative unit cell (RUC). Researchers have proposed several RUC arrangement methods to achieve various structural forms of single RUC, such as the translational offset method [142], pattern gradation method [5], rotation-reflection combination [143], and joint-based assemblies of multi-component [203]. Currently, a key premise of this optimization-generated structural design is a structural pattern of a single type of RUC. While from the manufacturing perspective, the full-periodic design is not always necessary since it is acceptable to prefabricate several molds for multiple types of RUC. To simultaneously allow multiple RUCs, this study proposed multi-pattern topology optimization. From the structural performance perspective, the multi-pattern design has a lower requirement for material distribution of the entire design domain, it is beneficial for deriving more efficient macrostructure. In addition from an aesthetic point of view, the organic of free-form and the regularity of periodic designs each have their own unique aesthetics, and the multi-pattern design still harbours many aesthetic design possibilities. As an intermediate area between full-periodic and free-form design, multi-pattern design has not been investigated adequately in the literatures so far.

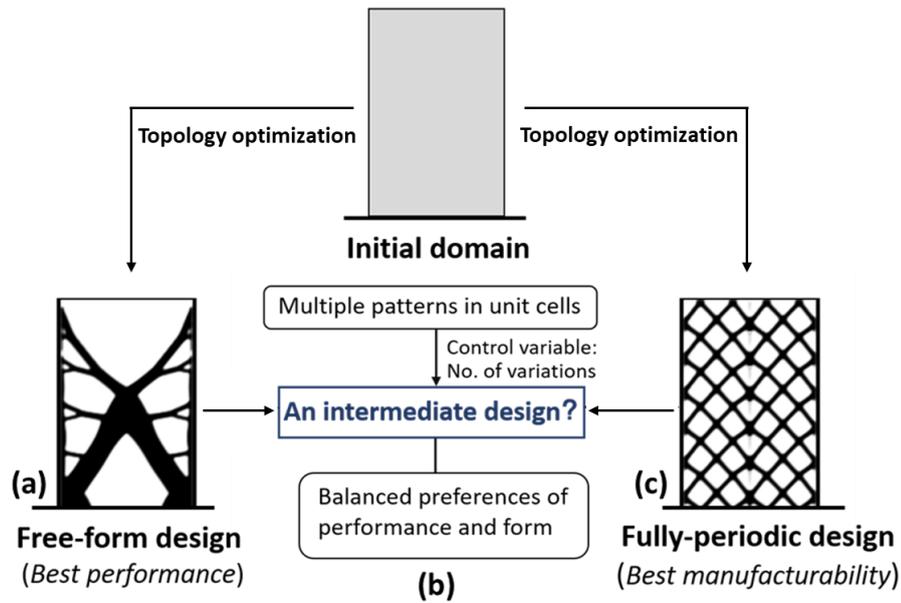


Figure 5.1 Illustrations of multi-pattern design that balances preferences of performance and form

As briefly reviewed, the use of topology optimization for structural design currently mainly falls into two categories: (1) free-form optimization in a global domain, and (2) optimization in the single unit cell for a fully periodic design. Inevitably, the former compromises manufacturability and aesthetics, whereas the latter eliminates the variations between unit cells and sacrifices the structural performance. To answer the need for balanced performance, manufacturability, and form aesthetics (form and performance, in general, as indicated in **Figure 5.1**). In this paper, a novel methodology is presented to meet this genuine need in real structural designs, which introduces a multi-pattern control into topology optimization-based methodology. This pattern control can be varied by an index (Number of variations: NoV) to quantitatively express the design freeform of structural form, it ranges from 1 for a fully periodic pattern to the number of unit cells, i.e., a free-form design. On the other hand, the structural performance is quantitatively interpreted as the conventional compliance value. This paper begins with the mathematical formulation of this methodology and demonstrates the

Chapter 5 Between free-form and periodic design – a multi-pattern approach inspiring structural shapes while imposing pattern control in manually assigned groups of unit cells. This is followed by the discussion of a clustering method that enables automatic grouping for multi-pattern topology optimization. Furthermore, post-processing for the optimized topology is developed, which helps remove the segments of low load-bearing contribution but is imposed for the sake of prevailing pattern recurrence. Case studies of pinned beams subjected to a center load and plane frames subjected to a horizontal load are carried out. Eventually, a series of automatically generated design patterns have been presented in this paper, which shows how inspiring a multi-pattern TO design tool is to structural engineers and underlines the fascinating potential of computer-generated designs using topology optimization between a free-form design and a fully periodic design.

5.2 Multiple patterns design using topology optimization

In this paper, the topology optimization process is formulated with the density-based SIMP method, empowered by the Isogeometric analysis (IGA) taking advantage of the foundation work of OPS-ITO developed by the authors. Unlike the existing free-form optimization in a global domain or single-pattern optimization in the basic unit cells. This paper, for the first time, establishes an automatically generated structural design with a multi-pattern formulation in topology optimization, which allows the number of pattern variations (denoted as NoV) in unit cells to be varied from 1 (i.e., fully periodic design) to the number of unit cells (i.e., free form) and enables a multi-pattern design that quantitatively

balances the preferences on performance and form (modular manufacturability). This section will briefly describe the theoretical formulation of this innovative TO scheme.

To enable the generation of multiple patterns in various groups of unit cells, the topology optimization process should be able to impose an identical density distribution of materials in each specific group. The grouping of unit cells can be manually assigned or automatically selected, which are both discussed in this paper. To map the optimized density distribution of a unit cell to the global structure of the multi-pattern design, the density distribution is described as follows:

$$\begin{aligned} \rho_e^i &= \sum_u^{N_V} C_{i,u} \rho_e^u, \quad e = 1, 2, 3, \dots, R_u, \\ C_{i,u} &= \begin{cases} 1, & u = T_i \\ 0, & \text{else} \end{cases}, \quad i = 1, 2, 3, \dots, N_u \end{aligned} \quad (5.1)$$

where \mathbf{C} is the mapping matrix between the optimized topology of an RUC and the unit cell in the design domain. Hereby, $C_{i,u} = 1$ is valid when the RUC u is equal to the i -th index of the RUC index set \mathbf{T} . The e -th elemental density of the i -th unit cell ρ_e^i is mapped from the e -th elemental density of the u -th RUC ρ_e^u . R_u is the resolution (number of elements) of each unit cell, N_u is the number of unit cells, and N_V is the number of pattern variations (NoV), which is equal to the number of RUC.

When considering a user-defined grouping of unit cells, all the unit cells are manually divided and labeled by the RUC index set \mathbf{T} . For instance, in a fully periodic pattern scheme, the index set of all the unit cells is simply given as $T_i = N_V$, hereby the N_V yields 1. On the opposite side, a free-form scheme could be also

gained by defining the index set as $T_i = i$ ($N_v = N_u$), which implies that the topology pattern of each unit cell is different.

For a multi-pattern design, the manual setting of \mathbf{T} should be assigned prior to executing topology optimization. For example, the grouping of unit cells ($u_1 \sim u_6$) as shown in **Figure 5.2** is given by $\mathbf{T} = [1, 2, 1, 3, 2, 3]$, indicating that the unit cell 1 and unit cell 3 should be of the same pattern or using the same RUC (tagged as RUC1), and the rest of unit cells are so forth. It should be noted that the actual topology has not been optimized yet, and the material distribution in the RUC remains unknown. It can be expected that a fully periodic design consisting of a single RUC will certainly impose much higher optimization constraints compared with the free-form design. In the case of a multi-pattern design (i.e., NoV is between 1 and N_u), the constraints would be gradually released and thereby the structural performance is better than the full-periodic scheme and approaches to the performance of a free-form scheme.

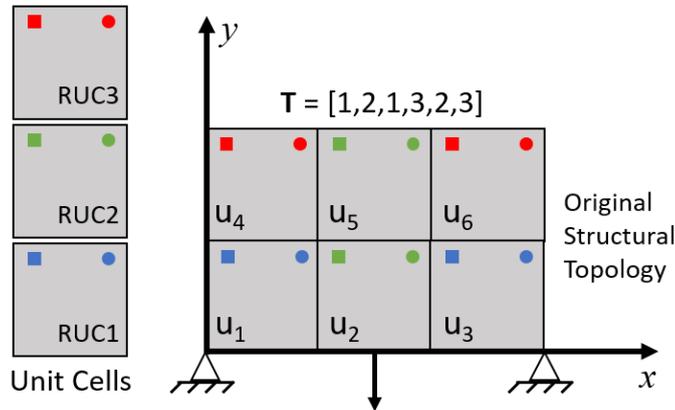


Figure 5.2 Schematic illustration of RUC and grouping schemes using a 3×2 design domain. (u_1, u_3 are mapped by RUC1; u_2, u_5 are mapped by RUC2; u_4, u_6 are mapped by RUC3)

When the mapping scheme is determined, the SIMP method of topology optimization can be operated to pursue an optimal design of NoV-controlled multiple patterns. If using the objective function of minimum compliance (aiming

for the highest stiffness), the mathematical form is given as follows:

$$\begin{aligned}
 & \min_{\boldsymbol{\rho}^u} \quad c = \mathbf{F}^T \mathbf{U} \\
 & \text{s.t.} \quad V(\boldsymbol{\rho}^u) = \sum_{e=1}^{N_e} \rho_e V_e \leq \bar{V} \\
 & \quad \mathbf{K} \mathbf{U} = \mathbf{F} \\
 & \quad 0 \leq \rho_e^u \leq 1 \\
 & \quad N_e = 1, 2, \dots, N_v \cdot R_u
 \end{aligned} \tag{5.2}$$

In the above formulation, the elemental density of RUCs $\boldsymbol{\rho}^u$ are the design variables during automated iterative analyses. Particularly, the sensitivity function of the compliance objective c in a multi-pattern optimization is given as:

$$\frac{\partial c}{\partial \rho_i} = - \sum_e \mathbf{U}_e^T \frac{\partial \mathbf{K}_e}{\partial \rho_e^u} \mathbf{U}_e \tag{5.3}$$

where the term $\frac{\partial \mathbf{K}_e}{\partial \rho_e^u} = \frac{\partial \mathbf{K}_e}{\partial \hat{\rho}_e} \frac{\partial \hat{\rho}_e}{\partial \rho_e^u}$ and the term $\frac{\partial \hat{\rho}_e}{\partial \rho_e^u}$ follows the chain rule from the periodic mapping in Eq. (5.1), filter in Eq. (4.3) but for elemental density, and projection in Eq. (4.5), which is given as below:

$$\frac{\partial \hat{\rho}_e}{\partial \rho_e^u} = \frac{\partial \hat{\rho}_e}{\partial \tilde{\rho}_e} \frac{\partial \tilde{\rho}_e}{\partial \rho_e^i} \frac{\partial \rho_e^i}{\partial \rho_e^u} \tag{5.4}$$

5.3 Automated grouping of unit cells for multi-pattern topology optimization

The above-presented method has shown how a multi-pattern control enables balanced design preferences on aesthetic form in patterns and structural performance in compliance. It should be clarified that by now the grouping of unit cells for the pattern control remains manually done, which is based on a

prescriptive vector of \mathbf{T} for unit cells and feasible for a limited number of unit cells. However, if a design domain involves many unit cells to be grouped, it is unlikely to manually enumerate all the possible schemes to find the optimal arrangement of unit cells and to achieve the best structural performance. To further enhance the degree of automation, it is necessary to develop an automated grouping technique for assigning unit cells to different patterns while the variable of NoV becomes the only index for pattern control.

5.3.1 Clustering-based selection of unit cell scheme for automated grouping

To enable the automated selection of unit cells for multi-pattern control, the key is to identify a characteristic index to divide the unit cells into the given number of groups. As an optimal structure tends to allocate materials in the areas formulating the most effective loading paths, it may be feasible to group them according to their contribution to the load resistance. In the BESO methodology, the non-effective elements are identified according to their elemental compliance [204, 205], which could be extended for the automatic grouping of unit cells in the multi-pattern control of topology optimization. The elemental compliance distribution of a structural domain has been illustrated in **Figure 5.3(a)**, where the elements of high-strain energy (high compliance) are printed in red. Based on this elemental distribution, the similarity between unit cells can be identified by calculating the summation of compliance differences of the corresponding elements in these two unit cells. Hence, the compliance difference between i -th and j -th unit-cell is expressed as:

$$S_{ij} = \sum_k |c_k^i - c_k^j| \quad (5.5)$$

where c_k^i and c_k^j are the compliances of the k -th element in the i -th unit cell and the j -th unit cell. The element tag k in each unit cell is also indicated in **Figure 5.3 (a)**.

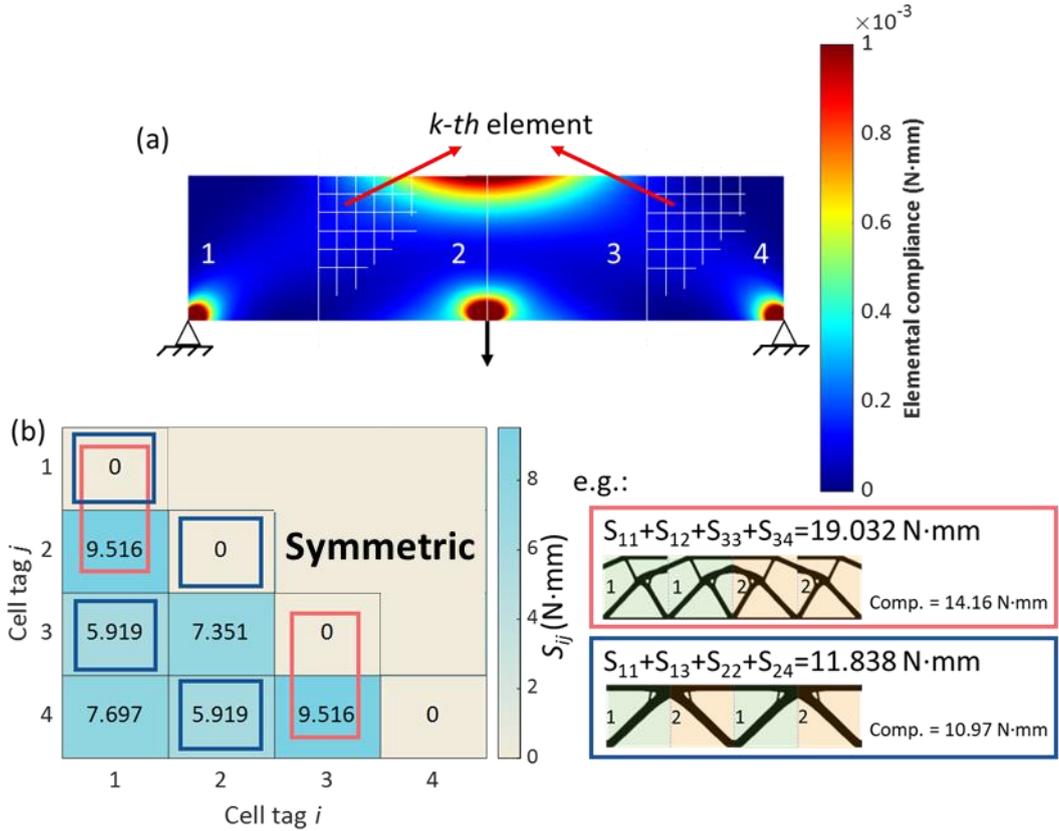


Figure 5.3 Compliance distribution and differences of a 4×1 domain: (a) elemental distribution, (b) compliance difference between unit-cells (N·mm), and two illustrative examples of grouping calculation.

Using Eq. (5.5), a matrix of S_{ij} quantifying the average compliance differences of unit cells is calculated for the beam of a 4×1 domain, which has been displayed in **Figure 5.3 (b)**. For instance, the scheme $\mathbf{T} = [1, 1, 2, 2]$ groups the unit cells as (1, 2) and (3, 4) in two groups. Hence, for the 1st group of (1,2), the compliance difference of Unit cell 1 and 2 are 0 (S_{11}) and 9.516 N·mm (S_{12}), whereas the 2nd groups of (3,4) are also 0 (S_{33}) and 9.516 N·mm (S_{34}). For this

grouping scheme, the cumulative difference is 19.032 N·mm. Similarly, the cumulative compliance difference of the Grouping Scheme with $\mathbf{T} = [1, 2, 1, 2]$ is $(S_{11}+S_{13}+S_{22}+S_{24}) = 11.838$ N·mm. Using this approach, the cumulative compliance differences of different pattern controls can be obtained. In **Figure 5.4**, the scatters show the compliance of the optimized structure with respect to the cumulative compliance difference of unit cells. It can be observed that a grouping scheme of a smaller cumulative difference leads to lower compliance of the macrostructure, indicating a model of the best structural performance. Therefore, the key action is to search for the minimum cumulative difference from all the possible grouping schemes as shown in **Figure 5.3** that grouping the unit cells with similar compliance distribution, and this is technically a clustering problem for data processing.

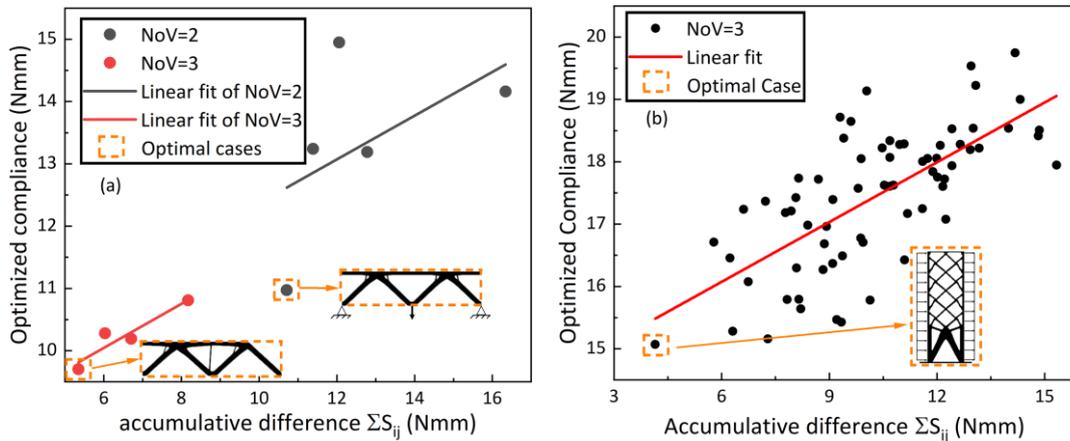


Figure 5.4 The positive correlation between the optimized compliance and accumulative unit-cell difference S_{ij} for (a) the 4×1 beam cases (b) the 6×1 building cases subjected to lateral load

Clustering as a typical approach for classifying a series of samples into several clusters, has been adopted in various topology optimization models to group similar multi-scale structures [206-208]. K-means is a widely-used clustering method [209], which well suits the grouping need in the developed multi-pattern control. The method is initiated with random cluster centers and followed by

iterations by assigning the closest samples (lowest S_{ij}) to the center and averaging grouped samples to update the location of centers to achieve the minimum clustering error. K-means is efficient in classifying the data, and it is sensitive to the initial position of centers. To enhance the stability, it is usually recommended to relocate the position for several trials and to record the best solution. While implementing an auto-grouping for multi-pattern control, the compliance distributions of all the unit cells become the samples, and the number of patterns (NoV) is the number of clusters. Therefore, a compliance difference S_{ij} between a sample unit c_k^i and the clustering center μ_k^j is quantified as the generalized distance:

$$S_{ij} = \sum_k |c_k^i - \mu_k^j| \quad (5.6)$$

To apply the K-means clustering for automated grouping of unit cells in a multi-pattern scheme, a few challenges need to be resolved. Firstly, the density variation in unit cells during optimization varies, and its average compliance changes correspondingly throughout the iterations, which may compromise the adequacy of the grouping scheme chosen at the initial stage. Hence, it would be beneficial to update the grouping vector \mathbf{T} after each iteration if any change is incurred. As shown in **Figure 5.5**, the grouping scheme is evaluated regarding the clustering distances at each iterative analysis, and the suitable cluster centers as well as the grouping schemes are kept for the next iteration. Eventually, the clustering-based method will automatically generate multiple groups of unit cells (clusters) to fit the number of patterns and to achieve the minimum average compliance.

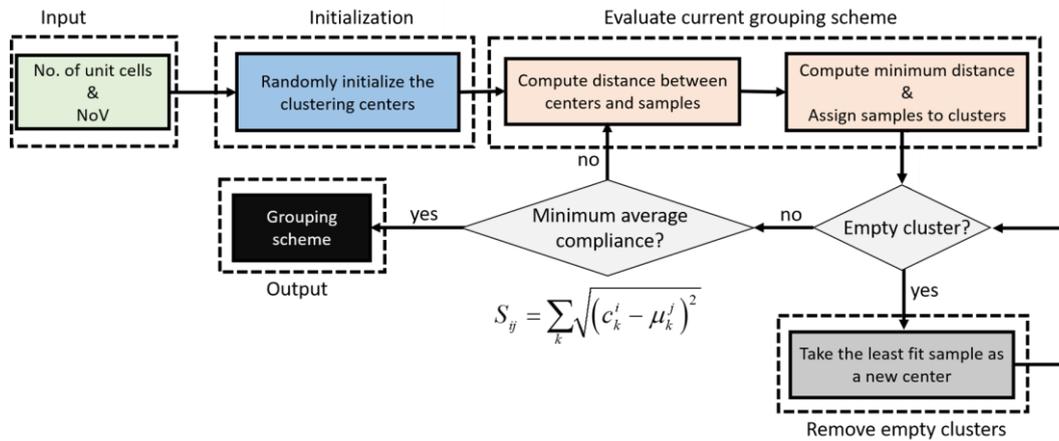


Figure 5.5 Workflow of k-means clustering for automated grouping of unit cells

Secondly, since the initial cluster centers are randomly located, an over-far center is of lower priority for assigning unit cells, which could potentially result in an empty cluster. Because no unit cells are associated with this empty cluster, the initial number of variations defined by the NoV cannot be maintained (reduced by the number of empty clusters). It is noted that computationally a higher NoV would provide more freedom to the topology optimization (as seen in the previous demonstration cases), resulting in better structural performance. Although the prevention of empty clusters remains a challenge to the research of K-means clustering [210], a simple solution is to remove the farthest center among all the clusters and add a new center. As shown in **Figure 5.5**, the occurrence of the empty cluster is checked after assigning samples (unit cells) to the clusters. If identified, the empty cluster will be removed and the unit cell with the least fit compliance distribution will become a new center, and the evaluation process loops until the empty clusters are all removed.

Lastly, the typical k-means clustering does not preserve the sequential relationship between the clusters. In multi-pattern optimization, the clusters as groups of unit cells not only store the attributed RUCs to the unit cells but also map the density distribution of corresponding RUCs. While the optimization proceeds,

the clusters formed at a new step would lose the projection of RUCs even though the grouping scheme is given (e.g., $\mathbf{T} = [1,2,1,2]$ or $\mathbf{T} = [2,1,2,1]$). This shuffled projection between grouped unit cells and RUCs is likely to cause convergence oscillation during the optimization. To resolve this issue, the clustering centers from the last optimization step are recorded as the starting centers for the next step to avoid oscillation.

The workflow to establish multi-pattern topology optimization is illustrated in **Figure 5.6**, and this new capability along with corresponding codes has been computationally developed and implemented in the open-source structural simulation platform OPS-ITO [211]. In the figure below, the functions for multi-pattern with manual grouping and the later introduced auto-grouping have been labeled in different colors.

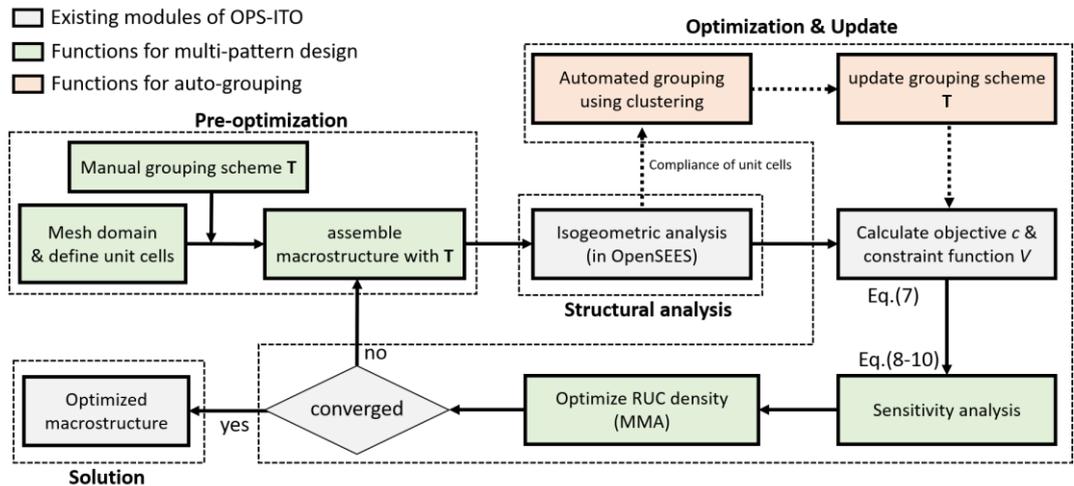


Figure 5.6 Topology optimization flowchart for multi-pattern design

Phase 1 Pre-optimization. Firstly defining the mesh domain, boundary condition, loads, and dividing unit cells. Secondly, assembling the macrostructure using the prescribed grouping scheme \mathbf{T} . If the grouping scheme is not given, the auto-grouping process is activated at the beginning of **Phase 3**, the macrostructure is temporarily free-form until the group scheme is auto-selected.

Phase 2 Structural analysis. The structural response is computed in OpenSEES using the newly developed isogeometric analysis module [211].

Phase 3 Optimization & Update. When the auto-grouping process is activated, the grouping scheme is obtained for the macrostructure assembling of the next iteration. Then compute the objective compliance c and constraint volume V in Eq. (5.2) and corresponding sensitivity analysis, the design variables RUC density is optimized using MMA [212]. The macrostructure is re-assembled for the next iteration until converged (the change ratio of structural compliance less than 0.1% or reach the maximum iteration limit, e.g., 200).

5.3.2 Post-processing of automated multi-pattern topology optimization

One inevitable side effect of enforcing pattern control in structural design is the generation of some structural segments for pattern recurrence but of minor contribution to load-bearing capacity, which are so-called zero-force (or nearly zero-force) components. The major reason is that this method defines the continuous material distribution of macrostructure using a group of mesoscale unit cells, the weak/discontinuous segments may therefore appear between the boundary of unit cells.

To remove the zero-force components in an optimized structural design using multi-pattern control, the post-processing can be conducted using the following criteria:

$$\rho_i = \begin{cases} \rho_{\min}, & \text{if } c_i \leq \eta_c c_{\max} \\ \rho_i, & \text{else} \end{cases} \quad (5.7)$$

where ρ_i is the density variable of the element of tag i , ρ_{\min} is the minimum density to enable computational removal. Herby the identification of zero-force elements is based on elemental compliance c_i , i.e., the elemental strain energy induced by external loading. In Eq. (5.7), η_c is the penalty parameter for determining zero-force components (10^{-7} is often used) with respect to c_{\max} , while c_{\max} is the maximum elemental compliance in the design domain. As long as c_i is significantly lower than the maximum compliance, the element i identified as of nearly zero contribution to load-bearing could be removed.

5.4 Application of multi-pattern control in manually assigned cells

Based on the work described in the above section and the code implementation in OpenSees, an optimal form of a given structural domain with multi-pattern control can be obtained. In this section, the performance of the multi-pattern scheme is compared to the conventional free-form optimization and fully periodic topology optimization. A case study on a vertical structure subjected to horizontal load is thereafter demonstrated, which illustrates the inspiring structural forms automatically generated by the developed multi-pattern topology optimization framework.

5.4.1 Performance verification of multi-pattern topology optimization

Taking the classic beam model as shown in **Figure 5.7** to benchmark the performance. The multi-pattern topology optimization framework is deployed for the two currently available cases of topology optimization, i.e., a free-form design and a fully periodic design. The reference results were given by Thomas et al. [143], which include the cases of a volume ratio of 0.5 presented in the paper and the cases of a volume ratio (Vol) of 0.3 using the code shared by Thomas et al. [143]. In this paper, all of the topology optimization attempts were under the premise of Vol = 0.3 for the expectation of more light structures rather than occupying a large portion of the initial design domain. In these benchmark cases, steel material is assumed with Young’s modulus E of 200 GPa and a Poisson’s ratio ν of 0.3. A downward force of 10 kN is imposed at the center of the bottom edge, while both ends of a beam are pinned supports. The 2×1 m initial domain is divided into two unit cells and each unit cell of a 1×1 m size is meshed as 80 × 80 IGA elements. For a model of 2 unit cells only, the grouping vector of multi-pattern optimization yields $\mathbf{T} = [1,1]$ for a fully periodic case (NoV = 1) or $\mathbf{T} = [1,2]$ for a free-form case (NoV = 2). The optimization targeted minimum structural compliance (or highest stiffness) with a prescribed volume ratio of 30% and the density variables are uniformly initialized as 0.5 at the beginning of optimization iterations along with a filter radius r_{\min} set as 0.03 m.

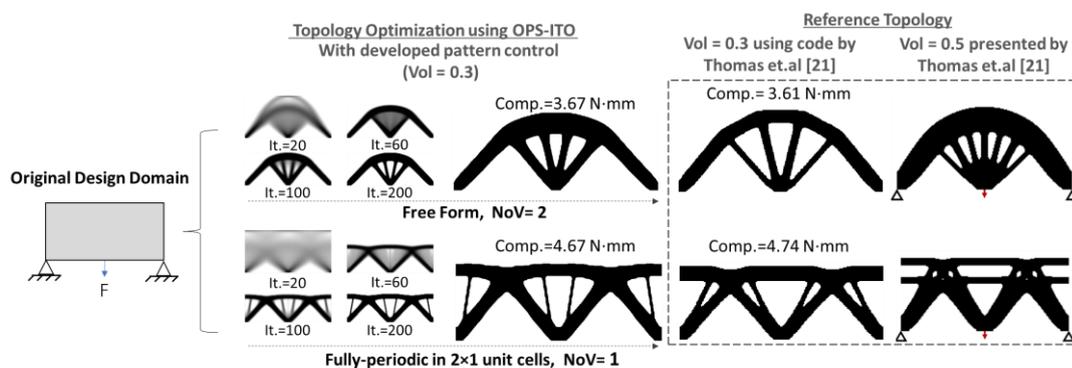


Figure 5.7 Multi-pattern topology optimization applied for free-form and fully periodic models

The optimization results for free-form and fully periodic designs are shown in **Figure 5.7**, where the optimized topologies obtained from the developed open-source tool are compared to very similar to the optimized models given by Thomas et.al [143]. Due to the different optimization algorithms and software environments (OC criterion and FEM were adopted in [143], while we adopts MMA [212] and IGA), slight differences exist between the reported models and the present models. However, the distributions of materials after topology optimization are very similar, especially in the case of free-form design, which both contains an arch and a hanging region. For the fully periodic designs, the model obtained in the present framework is close to the reference model when the volume ratios are identical, whereas some differences can be observed between the originally presented topologies and the obtained topologies of different Vol. Additionally, a high degree of agreement is also found in the case of 3×1 unit cells and 4×1 unit cells between the models shown in **Figure 5.8(a)** and **Figure 5.9** (the case of $NoV=1$) and the reference models in [143].

The iterative models during the optimization process extracted from the OPS-ITO tool have been also illustrated, which clearly shows how the multi-pattern topology optimization evolves from the initial domain to the eventual designs. The blurry components gradually disappear, whereas the load-bearing components become crisp. Regarding the structural performance, it is verified that a softer model of higher compliance ($C = 4.67 \text{ N}\cdot\text{m}$) results from the imposed single-pattern control compared to the free-form optimization model ($C = 3.67 \text{ N}\cdot\text{m}$) with no pattern control. Through this comparison, the capability of multi-pattern topology optimization yielded for two classic cases of optimization is verified.

5.4.2 Demonstration of multi-pattern control in 3×1 and 4×1 design domains

It is certainly of more interest to explore a multi-pattern design in a domain of more unit cells. This could potentially lead to a variety of grouping schemes and different pattern designs to fulfill the needs of architects and engineers. A simple example is drawn with a design domain of 3×1 unit cells for the above-optimized beam model (3m×1m), which is now prescribed as a fully periodic pattern, a double pattern, or a fully-free form pattern separately, as shown in **Figure 5.8**. Hereby a double-pattern design can be realized in different grouping choices, either as a symmetrical one ($\mathbf{T} = [1, 2, 1]$) or a non-symmetrical one ($\mathbf{T} = [1, 2, 2]$), as shown in **Figure 5.8(c)**. Noted that the other grouping schemes for a double pattern design are not presented here, due to the consideration of symmetry and equivalency. While observing the eventual topologies of various NoVs and grouping schemes, the aesthetic performance becomes different and various patterns lead to different forms and performances. Moreover, the compliance values (C) of these optimization-designed structures are shown in **Figure 5.8(d)**. The minimum compliance ($C=8.35 \text{ N}\cdot\text{m}$) is achieved in the fully-free form case as no pattern control is imposed as the additional constraints. The maximum compliance ($C=14.39 \text{ N}\cdot\text{m}$) indicating the lowest stiffness under the identical volume ratio appears in the fully periodical case ($\text{NoV}=1$). On the other hand, this design is of the highest degree of modular prefabrication, as it comprises three identical units. While only allowing two patterns exhibited in unit cells, the compliance values are between the minimum and the maximum ones, which read $C=7.52 \text{ N}\cdot\text{mm}$ for a symmetrical design and $C=8.25 \text{ N}\cdot\text{mm}$ for a non-symmetrical

case. In **Figure 5.8 (d)**, the variations of compliance values during the iterations of topology optimization have been plotted regarding the four cases of different NoVs. The rapid drops of compliance values are achieved relatively faster in the cases of NoV=2 and free-form design as the constraint function is looser compared with the fully periodic model.

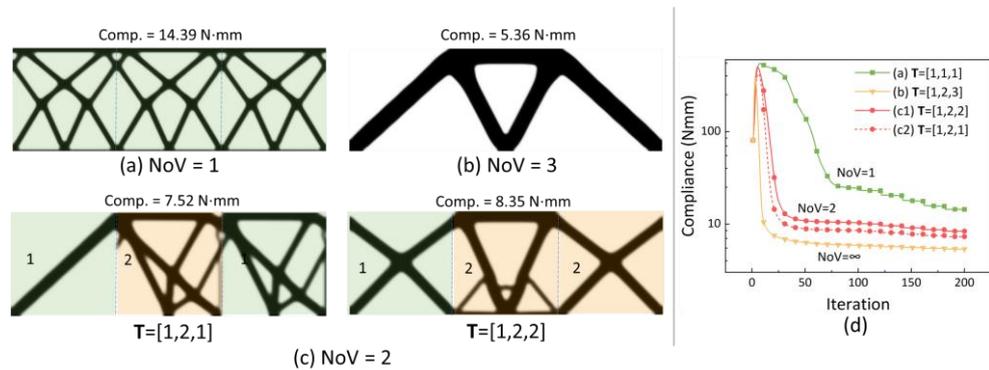


Figure 5.8 Multi-pattern topology optimization of a beam using 3×1 unit-cells: (a) NoV=1 (fully periodic design); (b) NoV=3 (free-form design); (c) NoV=2, (d) Histories of compliance values.

If further extending the design domain to a model of 4×1 unit cells, the number of design variations is correspondingly increased, which thereby offers more candidate designs to the decision-makers. When NoV is set to 2 for these 4 unit cells, it is interesting to see that all three variations exhibit regularity and symmetry, as the program tends to find the optimal load-bearing path. Due to the forced recurrence of patterns in unit cells, some components of the optimized structure should be of zero force, with no contribution to the load-bearing performance. If comparing the model of C equal to $10.97 \text{ N}\cdot\text{m}$ ($\mathbf{T} = [1, 2, 1, 2]$) to the free-form design (NoV= 4), the structural topologies are very similar but differ at the upper chords resulting from the enforced pattern singularity. For the models allowing three patterns in unit cells, the two models on the right-hand side are similar to the free-form design result, whereas the other two are between the fully periodic design and the free-form design. Generally, the structural topologies given

by the optimization process closer to the free-form topology are of higher load-bearing efficiency, as indicated by their lower compliance values (flexibility) compared to the other models of the same NoV.

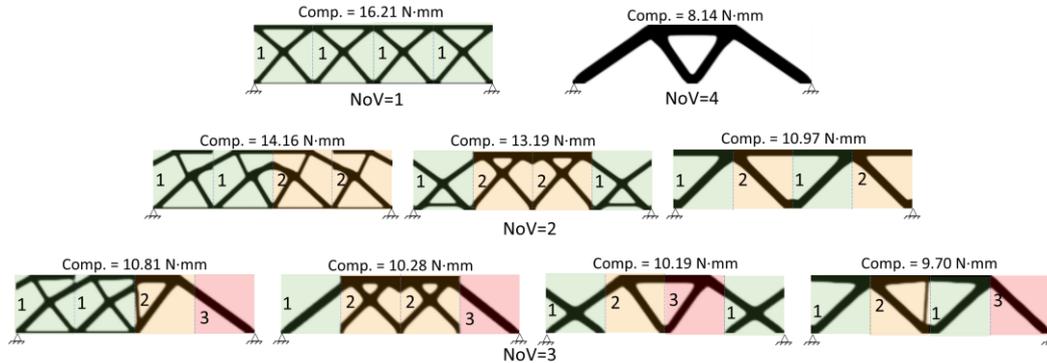


Figure 5.9 Topology optimization results of the beam model using 4×1 unit-cells.

5.4.3 Demonstration of multi-pattern control for a frame subjected to horizontal load

A multi-pattern control of topology optimization is not only used for the design of single structural members but also is expected to inspire the designs of structural systems for optimal form and performance. As shown in **Figure 5.10**, a plane frame is 2m wide and 6m high, which is subjected to a uniformly distributed load (1kN/m) on both two vertical edges to represent the horizontal load. The bottom surface is assumed to be fixed to the ground. The material remains steel, while Young's modulus $E = 200$ GPa and Poisson's ratio $\nu=0.3$. The volume ratio maintained is set as 40%. Prior to running topology optimization, the model meshes with 90×242 IGA elements, and the frame boundary layers are prescribed as a fixed elemental density of 1 (initially filled) for 5 layers on the left- and right-hand sides and 2 layers for the top surface. In this problem, the initial domain covers 80×240 elements and is divided into 6 unit cells vertically, and the dividing of the unit cell can be implemented for standard storeys of a building design. Noted

that the horizontal loading scenario is considered here, which could be extended to multiple loading scenarios for structural performance examination and is currently being developed by the authors.

The multi-pattern designs of this plane frame generated by the topology optimization have been shown in **Figure 5.10**, where the control index NoV is given as 1, 2, 3, and 4 (free form), separately. The grouping schemes for pattern control follow the convention of ‘standard storey’ along with the load accumulation. When the NoV is increased from 1 (fully periodic) to 2, slender components appear in the upper group as it is subjected to lower shear force. As a result of loosed pattern control, the structural lateral stiffness is increased from 19.74 N·mm to 16.44 N·mm. While further increasing the number of variations, the lower zone is prone to the bottom of the free-form model after topology optimization without pattern control. Meanwhile, the unit cells in the upper zone of the NoV=3 case remain close to the optimized design of the NoV=2 model.

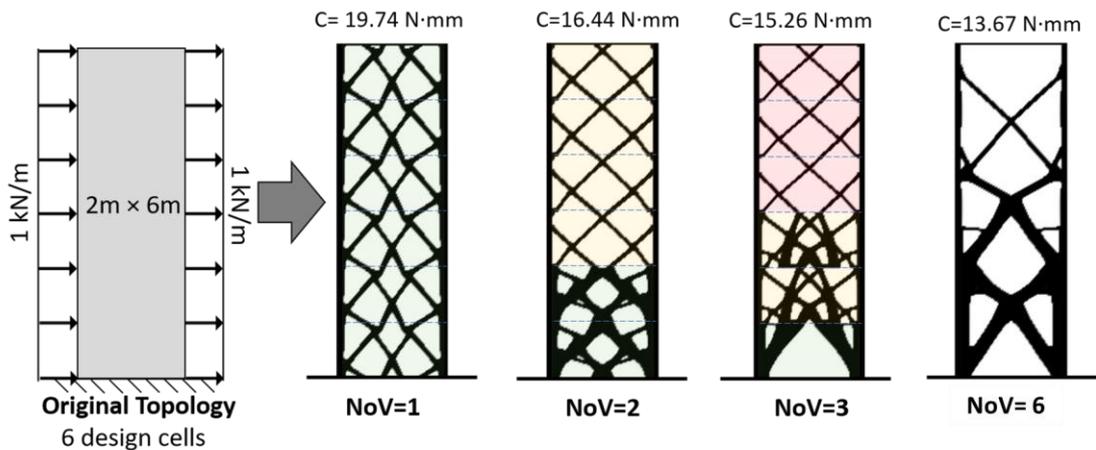


Figure 5.10 Multi-pattern topology optimization of a plane frame subjected to horizontal load

5.4.4 Demonstration of multi-pattern topology optimization with automated grouping

The automated grouping of unit cells applied along with multi-pattern topology optimization is demonstrated using a plane frame, which is similar to the above-discussed case using manual grouping but the size is further enlarged to 4 m×6 m. Correspondingly, the number of unit cells has been increased to 2×6 for the design domain, while each cell comprises 30×60 IGA elements. On the boundaries of this plane frame, solid layers are assumed with a prescribed density index of 1 for the outer profile of the frame. Noted this plane frame model remains a scaled one to real structures, which can be certainly extended to a real scale. This along with 3D application and optimization for multiple loading scenarios will be studied as the follow-up work.

Using multi-pattern topology optimization with auto-grouping, the optimization results are illustrated in **Figure 5.11**. Now the only control index becomes the NoV (1~11, and 12 for free form), while the distribution of patterns into different unit cells is automatically found by the program. In the illustrations of the optimized models, the unit cells mapped to the same RUC (i.e., the same group of unit cells) are labeled by the same color shade, whereas the unit cells of the no-repeating pattern (i.e., the RUC only appears in a single unit-cell) are not colored as the group number is above 6. Generally, the structural forms of different NoVs exhibit different pattern formulations and different cost-effectiveness in terms of modular fabrication. When the NoV is set to 2 or 3, it is clearly observed that the auto-grouping program divides the unit cells into two or three zones differentiated by height. This agrees with the common design logic as the horizontal shear force decreases from the lower storeys to the upper ones. While further increasing the NoV to allow more patterns in unit cells, the structural design varies and gradually approaches the free-form design (NoV=12).

From all of the optimized models of different NoVs, the structural compliance varies while the multi-pattern control is differently applied. The corresponding compliance values have been given in **Figure 5.12**, where an obviously decreasing trend is found along with the increased NoV. When the volume ratio is fixed to 30%, it verifies that applying a higher degree of pattern control (from free-form to single-pattern periodicity) would compromise the structural performance (higher compliance) but improves the degree of modular prefabrication. When allowing only one pattern in all the unit cells, the structural compliance reaches the maximum (7.34 N·mm). An intermediate value (5.75 N·mm) of compliance can be obtained while increasing NoV to 4 for this 2×6 design domain. It can be further reduced to 4.81 N·mm in the case of NoV= 7 as the topology of the upper zone becomes very similar to the free-form optimized model, which is of the lowest compliance (3.66 N·mm) indicating the best structural performance. In real design practice, these design control variables and performance indices could support quantifiable parameters to evaluate the structural design, which can thereafter fulfill the different design needs in consideration of pattern aesthetics and construction cost.

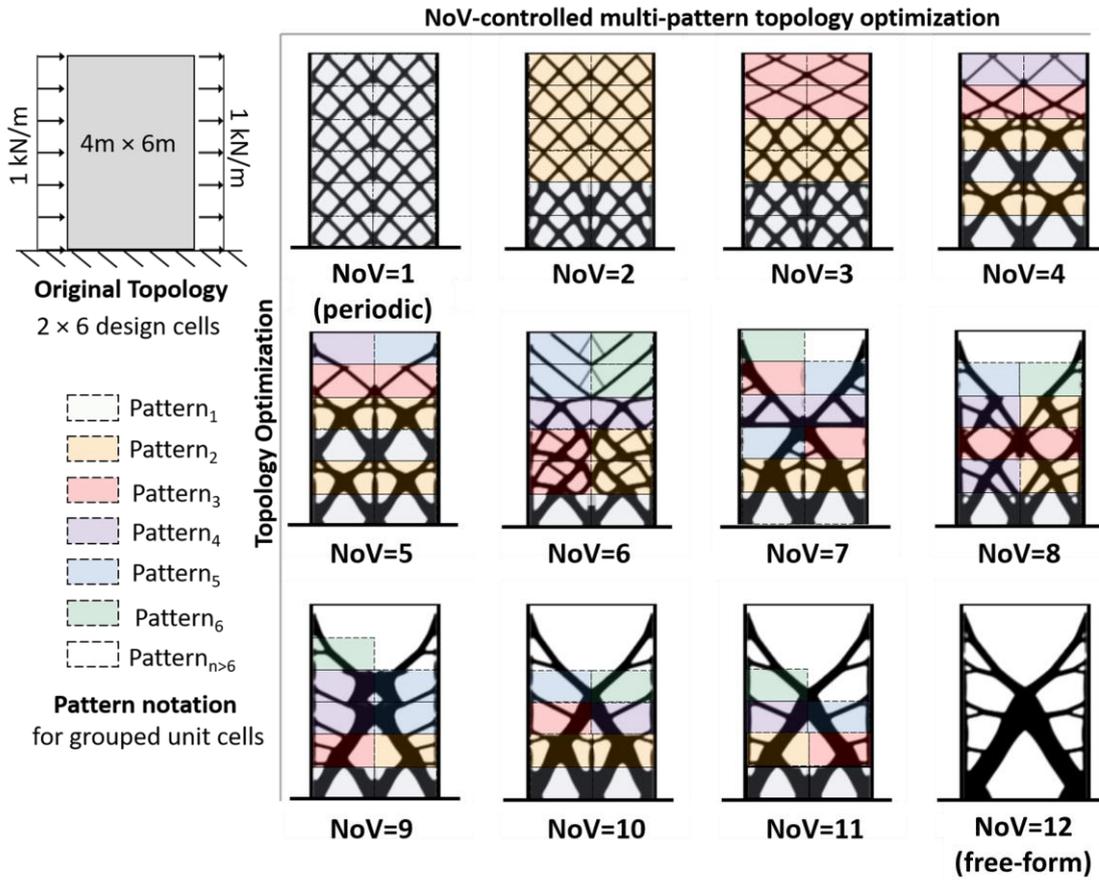


Figure 5.11 Topology optimization-based structural design of a building frame model with different NoV

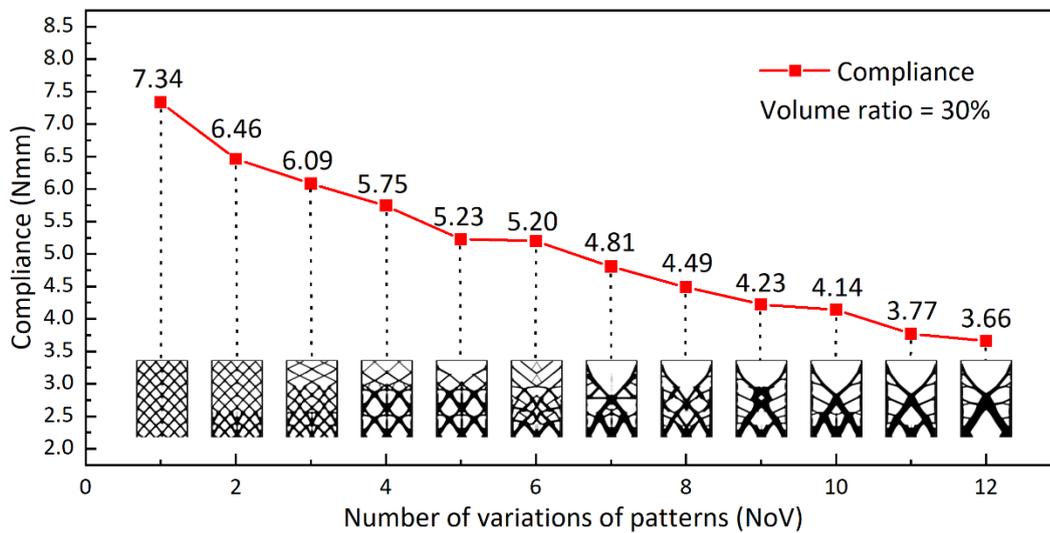


Figure 5.12 Compliance values of optimized models with various NoVs

5.4.5 Demonstration of automated grouping with zero-force removal

Such a post-optimization process can be demonstrated with the aforementioned $2\text{m}\times 1\text{m}$ beam model, which has been optimized using 8×4 unit cells and 30×30 IGA 1-order elements for each unit cell, as shown in **Figure 5.13**. The beam model optimized using the multi-pattern control with auto-grouping has led to various forms of different NoVs (such as NoV=1, 3, 5, 4, 6, 8), where a handful of branching components do not form effective load transferring paths and, therefore, could be removed. Certainly, the removal of zero-force (or nearly zero-force) components from a multi-pattern design would potentially generate new patterns and increase the NoV. To pursue an optimal design, it may be more practical to begin the topology optimization with a lower NoV and to execute the zero-force removal as post-processing.

As a result of pattern control, some components of very low-stress levels should be ideally removed to avoid material waste. For example, in the optimized model of NoV=6, the outer branches undoubtedly bear nearly zero force (stress). After using the removal process defined in Eq. (5.7), these components are identified and removed from the optimized model and visualized as a light grey shade. The post-process model presents a more practical loading path and the pattern aesthetics have not been compromised. On the contrary, the aesthetic performance may be even improved, which in practice surely depends on the architects' preference. The post-processing using zero-force removal has been also implemented for the models generated using NoV = 4, 8, 12, 24. After removal, the benefit similar to the case of NoV = 6 is gained.

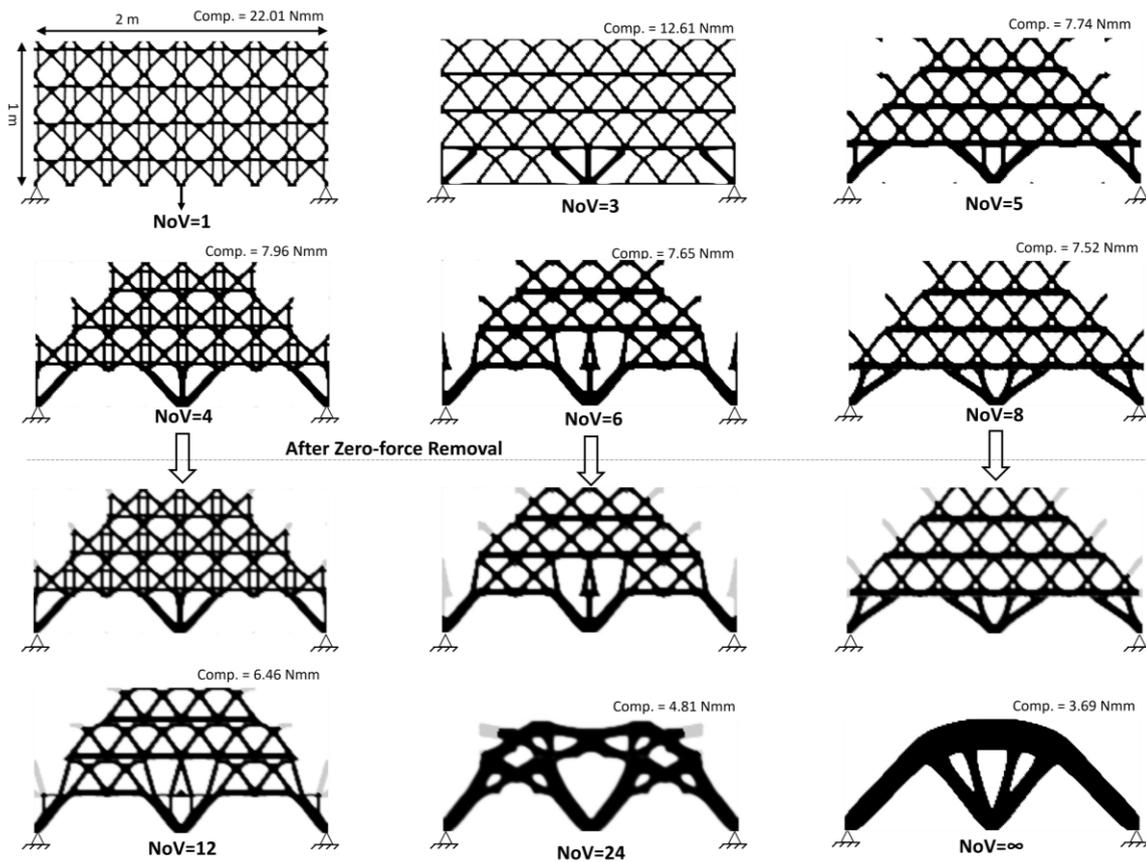


Figure 5.13 Optimized configurations of 8×4 simply supported beam with zero-force removal

5.5 Concluding remarks

In this chapter, a new topology optimization strategy for prefabrication called multi-pattern design is proposed. Since the practical prefabricated structure may be composed of several types of the component instead of full-periodic, multi-pattern design can trade-off the structural performance and prefabricability by allowing multiple RUCs and user-defined unit-cell arrangement scheme (defining the mapping relationship from RUC to unit-cell). Although the determination of an optimal arrangement scheme may require traversal search, it is computationally infeasible for cases with a large number of unit cells because of the plenty of

possible combinations.

Considering the unbearable computational cost of selecting an optimal arrangement scheme, this study also proposed a k-means-based method for computationally efficiently finding a more beneficial scheme if no preferred given. This scheme selection approach is inspired by the topology evolution strategy of the BESO method (e.g., refer to the distribution of elemental compliance).

Several demonstrative cases are presented using the multi-pattern design method, and this approach coincident with the building design rule of standard layer, thus it is capable to optimize the building under the guise of the prescribed standard layer as the arrangement scheme. In the cases with a prescribed arrangement scheme, it can be observed that the optimized structural compliance decreases with the NoV increasing (deterioration of prefabricability), which can be traded off by users. And in the cases with an automated scheme selection method, more beneficial schemes can be found and adopted to ensure a reliable design.

It should be noted that efficiently finding the optimal scheme for multi-pattern design is still an open issue, and the proposed k-means-based method has the noteworthy ability for giving relatively reliable schemes when only giving NoV.

Chapter 6 Conclusions and future study

6.1 Conclusions

This thesis is an early-stage project of establishing the construction-oriented topology optimization platform, which presents our vision for the automatic construction of the next generation. This platform considers a workflow of converting the most manual design and construction tasks into automatic ones to shape a sustainable future by reducing carbon emission, material wastage, workload pressure, safety risks, duration, and expense costs. In addition from the structural design perspective, it guarantees the optimized architectures with higher structural and aesthetic performances based on the trinity principle of architecture design, and the simulation quality can be enhanced by adopting new methods (e.g. isogeometric analysis in this thesis).

In this thesis, the structural optimization framework is established based on OpenSEES by developing the isogeometric solid elements and the corresponding topology optimization tool package. In addition, we proposed two optimization models under the connotation of the trinity principle of architecture design as the preliminary exploration. These two models respectively investigate (a) the explorative application of the isogeometric topology optimization method on designing typical engineering structures (e.g. post-tensioned concrete beam), introducing corresponding structural analysis model, and proposing suitable

optimization model. (b) simultaneously considering the manufacturability and aesthetic performance of optimized structures via combining the advantages of modular prefabricable periodic design and the efficient free-form design to explore a collection of intermediate solutions. The main contributions of the thesis are concluded as follows:

- By employing the IGA-SIMP method, a topology-shape optimization model is developed to design the post-tensioned concrete beam comprising a prestressing tendon. It is capable of simultaneously optimizing the concrete topology and NURBS-based tendon curve shape. (Chapter 3)
- From the structural simulation perspective, the tendon is described by a NURBS curve to enhance the continuity and accuracy of the expression of equivalent prestress load, and an IGA-based mapping process is proposed to simplify the expression of prestressed load, which avoids the calculation of inverting jacobian matrix from the tendon to the concrete domain. (Chapter 3)
- From the optimization perspective, considering the typical unequal strength limits for tension and compression of concrete material, the Drucker-Prager stress constraint is considered and firstly extended into the isogeometric framework. In addition, a series of numerical problems of stress-based topology optimization (e.g., stress singularity, local stress constraint, and stabilization of iteration) and geometric control of the optimized structure (e.g., minimum width control, density projection, tendon-concrete filter) are considered to ensure a reliable design. (Chapter 3)
- OPS-ITO platform is established as the foundation for our futural building design, which employs 2D and 3D IGA elements for structural simulation and the IGA-SIMP method for topology optimization, it includes classical

minimum compliance model and minimum volume model with Von-Mises and Drucker-Prager stress constraints. To ease the usage, a graphical user interface integrates all the available functions of OPS-ITO and it is continually developing to consider more construction-oriented considerations. The optimized 3D model can also be exported for further post-processing in a widely used *vtk* format. (Chapter 4)

- A multi-pattern control method has been developed to allow intermediate designs between the free-form design and the fully periodic design. Noted that this approach requires users to provide a prescribed grouping scheme to determine the layout of the macrostructure, the optimal grouping is often computationally infeasible using the traversal method when lacking a preferred scheme. Hence, an automatic grouping selection method is proposed based on the K-means clustering approach to provide relative ideal solutions, and the post-processing of zero-force removal is also developed to reduce the possible material waste of the multi-pattern design. (Chapter 5)

6.2 Future works

For the direction of our future research, calculation, and manufacturing are both considered. From the calculation perspective, multi-load design, fail-safe design, and thermal insulation design are preferable to further enhance the durability and utility performance of the optimized structures. Generally, architectures are often subjected to different complicated loads during the course of their service, while current construction-oriented topology optimization research focuses on finding the optimal structure subjected to a single type of load, it is

essential to consider the multi-load scenarios to enhance the structural reliability. In addition, the traditional optimized structures often pursue the ultimate utilization of materials without considering the structural redundancy hence posing a great structural safety threat when some of the components fail unexpectedly. On the utility concern, the thermal insulation performance of the architecture is our first concern, which will be deeply investigated.

From the manufacturing perspective, we are adopting 3D concrete printing technology to produce some demonstrative examples and extend the manufacturing work to a real scale manufacturing with the collaboration of prefabrication technology. So far, the ductility and strength properties of 3D printing material are still a major challenge, improving the material performance will be our main interest such as using high-performance composite concrete with fiber and prestressing process to enhance the tensile strength of the material.

Appendix A

A1 The curvature sensitivity analysis of the tendon curve

The formulation of curvature can be written as:

$$\kappa(\xi) = \frac{|\mathbf{C}'(\xi) \times \mathbf{C}''(\xi)|}{|\mathbf{C}'(\xi)|^3} = \frac{|\mathbf{A}|}{|\mathbf{B}|^3} \quad (\text{A.1})$$

where $\mathbf{C}'(\xi), \mathbf{C}''(\xi)$ are the first and second derivative values of the tendon curve, respectively. For the sake of formulation simplicity, we define terms $\mathbf{A} = \mathbf{C}'(\xi) \times \mathbf{C}''(\xi)$ and $\mathbf{B} = \mathbf{C}'(\xi)$. Therefore, the expression of Eq. (A.1) could be rewritten as:

$$\frac{\partial \kappa(\xi)}{\partial y_i} = \frac{\mathbf{A}}{|\mathbf{A}||\mathbf{B}|^3} \frac{\partial \mathbf{A}}{\partial y_i} - \frac{3\mathbf{B}|\mathbf{A}|}{|\mathbf{B}|^5} \frac{\partial \mathbf{B}}{\partial y_i} \quad (\text{A.2})$$

where the term $\frac{\partial \mathbf{A}}{\partial y_i}$ is resulted in $\frac{\partial \mathbf{A}}{\partial y_i} = \frac{\partial \mathbf{C}'(\xi)}{\partial y_i} \times \mathbf{C}''(\xi) + \mathbf{C}'(\xi) \times \frac{\partial \mathbf{C}''(\xi)}{\partial y_i}$; and the term

$\frac{\partial \mathbf{B}}{\partial y_i}$ is straightforward: $\frac{\partial \mathbf{B}}{\partial y_i} = \frac{\partial \mathbf{C}'(\xi)}{\partial y_i}$.

According to the formulation of the first and second derivative values of the tendon curve, the corresponding derivatives are given as follows:

$$\begin{aligned} \frac{\partial \mathbf{C}'(\xi)}{\partial y_i} &= \sum_{j=1}^{N_{\text{ten}}} \left(\frac{\partial R_{j,p}^{\text{ten}}(\xi)}{\partial \xi} \frac{\partial \mathbf{P}_j^{\text{ten}}}{\partial y_i} \right) \\ \frac{\partial \mathbf{C}''(\xi)}{\partial y_i} &= \sum_{j=1}^{N_{\text{ten}}} \left(\frac{\partial^2 R_{j,p}^{\text{ten}}(\xi)}{\partial \xi^2} \frac{\partial \mathbf{P}_j^{\text{ten}}}{\partial y_i} \right) \end{aligned} \quad (\text{A.3})$$

A2 Normal direction sensitivity analysis of the tendon curve

The formulation of normal direction can be written as:

$$\mathbf{n}(\xi) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} \cos \theta_\xi \\ \sin \theta_\xi \end{bmatrix} \quad (\text{A.4})$$

where θ_ξ represents the angle at the location ξ , and the expression could further take the

form as $\begin{bmatrix} \cos \theta_\xi \\ \sin \theta_\xi \end{bmatrix} = \frac{d\mathbf{x}(\xi)}{ds} = \frac{\mathbf{J}_1^{\text{ten}}}{|\mathbf{J}_1^{\text{ten}}|}$. Therefore, the derivative of normal $\frac{\partial \mathbf{n}(\xi)}{\partial y_i}$ and is as

below:

$$\frac{\partial \mathbf{n}(\xi)}{\partial y_i} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * \left(\frac{1}{|\mathbf{J}_1^{\text{ten}}|} - \frac{(\mathbf{J}_1^{\text{ten}})^2}{|\mathbf{J}_1^{\text{ten}}|^3} \right) \frac{\partial \mathbf{J}_1^{\text{ten}}}{\partial y_i} \quad (\text{A.5})$$

Likewise, the tangential direction $\frac{\partial \mathbf{n}^*}{\partial y_i}$ can be written as:

$$\frac{\partial \mathbf{n}^*}{\partial y_i} = \left(\frac{1}{|\mathbf{J}_1^{\text{ten}}|} \frac{\partial \mathbf{J}_1^{\text{ten}}}{\partial y_i} - \frac{\mathbf{J}_1^{\text{ten}}}{|\mathbf{J}_1^{\text{ten}}|^2} \frac{\partial |\mathbf{J}_1^{\text{ten}}|}{\partial y_i} \right) \quad (\text{A.6})$$

where $\mathbf{J}_1^{\text{ten}}$ is the tendon elemental Jacobian matrix from physical space to parametric space is calculated by:

$$\mathbf{J}_1^{\text{ten}} = \sum_{j \in e} \frac{\partial R_j^{\text{ten}}(e)}{\partial \xi} \mathbf{P}_j^{\text{ref}} \quad (\text{A.7})$$

and the corresponding derivatives term $\frac{\partial \mathbf{J}_1^{\text{ten}}}{\partial y_i}$ is:

$$\frac{\partial \mathbf{J}_1^{\text{ten}}}{\partial y_i} = \sum_{j \in e} \frac{\partial R_{j,p}^{\text{ten}}(e)}{\partial \xi} \frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_i} \quad (\text{A.8})$$

according to Eq. (3.15), the term $\frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_i}$ in Eq. (A.8) can be given as:

$$\frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_i} = \frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_j^{\text{ref}}} \frac{\partial y_j^{\text{ref}}}{\partial y_i} = W_{j,i}^{\text{ref}} \frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_j^{\text{ref}}} \quad (\text{A.9})$$

where the corresponding derivative term is as $\frac{\partial \mathbf{P}_j^{\text{ref}}}{\partial y_j^{\text{ref}}} = [0, 1, 0]$. Further, the derivatives term

$\frac{\partial |\mathbf{J}_1^{\text{ten}}|}{\partial y_i}$ is calculated by:

$$\frac{\partial |\mathbf{J}_1^{\text{ten}}|_e}{\partial y_i} = \frac{(\mathbf{J}_1^{\text{ten}})_e}{|\mathbf{J}_1^{\text{ten}}|_e} \frac{\partial (\mathbf{J}_1^{\text{ten}})_e}{\partial y_i} \quad (\text{A.10})$$

Appendix B

B1 MATLAB code of designing post-tensioned concrete beam

B1.1 A driver for iterative running the main function

```
clear;clc;
SavePathList = "\13.3&6.7MPa_50x10\";
CompThegLimList = 13.3e6;
TensThegLimList = 6.7e6;
MeshSizeList = [50,10];
for i = 1:length(CompThegLimList)
    SavePathDetail = char(SavePathList(i));
    SavePath = [pwd, '\myResult2\', SavePathDetail];
    mkdir(SavePath)
    CompThegLim = CompThegLimList(i);
    TensThegLim = TensThegLimList(i);

    MeshSize = MeshSizeList(i,:);
    CaseLabel = 'Single';
    disp(['CurrentPath: ' char(SavePathDetail) ' CompTheg: '
char(num2str(CompThegLim)) ' MeshSize: '
char(num2str(MeshSize(1)))...
' ' char(num2str(MeshSize(2))))]);

    MainFuncPC(SavePath,CompThegLim,TensThegLim,MeshSize,CaseLabel);
end
```

B1.2 Main function (based on SIMOPackage [196])

```
function
MainFuncPC(SavePath,CompThegLim,TensThegLim,MeshSize,CaseLabel)
if CaseLabel == 'Single'
    L = 1;
    SpanSupRatio = 0;% no support region
elseif CaseLabel == 'Double'
    L = 2;
    SpanSupRatio = 0.05; % half mid support region
elseif CaseLabel == 'Triple'
    L = 3;
    SpanSupRatio = 0.1; % entire support region
end
D = 0.2;
eleL = 0.005; eleD = 0.005;
FixedTopRatio = 0.1;%the thickness of fixed top layer
%location
CtrlPts = zeros(4, 2, 2);
CtrlPts(1 : 3, 1, 1) = [0; 0; 0];
CtrlPts(1 : 3, 2, 1) = [L; 0; 0];

CtrlPts(1 : 3, 1, 2) = [0; D; 0];
CtrlPts(1 : 3, 2, 2) = [L; D; 0];
%weight
CtrlPts(4, :, :) = 1;

KntVect[181] = [0 0 1 1];
KntVect[181] = [0 0 1 1];
% create NURBS surface in CAD
Surf = CreateNURBS(KntVect, CtrlPts);
% degree of basis function
p=1;q=1;
% repeated knot value inside knot vector
kx=1;ky=1;
% number of elements in each direction
nelx_Den=MeshSize(1); nely_Den=MeshSize(2);

nelx = L/eleL; nely = D/eleD;
% h,p,k-refinements
Surf_Den = KRefine(Surf, [nelx_Den, nely_Den], [p, q], [p-kx, q-ky]);
Surf_Anal = KRefine(Surf, [nelx, nely], [p, q], [p-kx, q-ky]);
% calculate Rij for all element centroids
check_x = linspace(0,1,p*nelx*(length(KntVect[181])-p-1)+1);
check_y = linspace(0,1,q*nely*(length(KntVect[181])-q-1)+1);
node_x = check_x(2:2:end);
node_y = check_y(2:2:end);
%R(kxi,eta,i,j)-->CtrlPts -- ParaPts (centroid of elements)
R_Center = FindInterpFunc(Surf_Den,{node_x, node_y});
node_xx = linspace(0,1,nelx+1); node_yy = linspace(0,1,nely+1);
RR_node = FindInterpFunc(Surf_Den,{node_xx, node_yy});
x_CtrlPts = permute(Surf_Den.CtrlPts3D(1, :, :), [2,3,1]);
y_CtrlPts = permute(Surf_Den.CtrlPts3D(2, :, :), [2,3,1]);
loc_CtrlPts = [x_CtrlPts(:), y_CtrlPts(:)];
CurvStore_center = (R_Center*loc_CtrlPts)';
CurvStore_Xi = (RR_node*loc_CtrlPts)';
% penalty for SIMP
penalty = 3;
% rho for CtrlPts
rho_CtrlPts = 0.8*ones(Surf_Den.NCtrlPts);
NCP_Tendon = 8*L + 1;
tendon_x=linspace(0,L,NCP_Tendon);
tendon_y=0.1*D*ones(1,NCP_Tendon);
tendon_y(end) = 0.9*D;
nelx_tendon = 200;
```

```

[Tendon, Tendon_Refine] =
BuildTendon(nelx_tendon, tendon_x, tendon_y);
RefWeight = RefineWeight(Tendon, Tendon_Refine);
rho_e = reshape(R_Center*rho_CtrlPts(:), nelx, nely);
miu_pre = 1; beta_fil = FixedTopRatio*D;
[rho_ehat, pdist_pyi, dist] =
TendonModiDensity(CurvStore_center, Tendon_Refine, rho_e, RefWei
ght, miu_pre, beta_fil);
[prhoeHat_prhoi, prhoeHat_pyi] =
calc_drhoedvar(R_Center, dist, beta_fil, miu_pre, rho_e, pdist_pyi);
beta_HS = 1; eta_ero = 0.4; eta_dil = 0.6;
[rho_estar, prhoeStar_prhoi, prhoeStar_pyi] =
DilHeaviside_PC(rho_ehat, beta_HS, eta_ero, eta_dil, prhoeHat_prhoi, p
r hoeHat_pyi);
NFixLayer = floor(FixedTopRatio*nely);
rho_estar(:, end-NFixLayer:end) = 1;
Mesh = Mesh2D(Surf_Anal, 'VectorField');
E = 28e9; nu = 0.2;
theg_c = CompThegLim;
theg_t = TensThegLim;
%initialize MMA
t1nn = size(rho_CtrlPts(:, 1), 1);
t2nn = size(tendon_y(:, 1), 1);
tnn = t1nn + t2nn;
xy00 = [rho_CtrlPts(:); tendon_y(:)];
numVar = length(xy00);
rho_min = 1e-3;
xval = xy00;
xold1 = xy00;
xold2 = xy00;
x1min = 0*ones(t1nn, 1);
x1max = ones(t1nn, 1);
x2min = 0.1*D*ones(t2nn, 1);
x2max = 0.9*D*ones(t2nn, 1);
xmin = [x1min; x2min];
xmax = [x1max; x2max];
low = xmin;
upp = xmax;
numDV = length(low);
m = 1;
Var_num = tnn;
c = 1000*ones(m, 1);
dd = ones(m, 1);
a0 = 1;
a = zeros(m, 1);
loop = 0;
maxiter = 300;
looppp = [];
volp = []; stress = [];
initial_Ve = CalcVe(Mesh, CurvStore_Xi, ones(size(rho_estar)));
initial_Ve = sum(initial_Ve(:));
if CaseLabel == 'Single'
    Pressure = @(x, y) -200e3;
elseif CaseLabel == 'Double'
    Pressure = @(x, y) -250e3;
elseif CaseLabel == 'Triple'
    Pressure = @(x, y) -250e3;
end
Tpre = 6e5;
while loop < maxiter
    loop = loop + 1;
    Ve = CalcVe(Mesh, CurvStore_Xi, rho_estar); %element volume (area)
    [KVals, KE] = calcLocalStiffnessMatrices2D(Mesh, Surf_Anal, E, nu,
'PlaneStress', rho_estar, penalty, rho_min);
    KEe = reshape(KE, size(KE, 1), size(KE, 2), []);
    [Rows, Cols, Vals] = convertToTripletStorage(Mesh, KVals);
    % Convert triplet data to sparse matrix
    K = sparse(Rows, Cols, Vals);
    clear Rows Cols Vals
    f = zeros(Mesh.NDof, 1);
    [Fy, DofsFy] = applyNewmannBdryVals(Surf_Anal, Mesh, Pressure,
4, 'FY');
    f(DofsFy) = f(DofsFy) + Fy;
[F_pre, TendonLoc, dFCondy] =
Prestress_Force(Surf_Anal, L, D, Tendon_Refine, RefWeight, Tpre, Mesh,
beta_fil); %seems solved
f = f + F_pre;
[grav, dfswdrhoi, dfswdyi] =
applyGravity_PC(Surf_Anal, Mesh, 9.8*2450, rho_estar, prhoeStar_prh
oi, prhoeStar_pyi);
f = f + grav;
h = @(x, y) 0;
[UX, DofsX] = projDrchtBdryVals(Surf_Anal, Mesh, h, 1, 'UX');
[UY, DofsY] = projDrchtBdryVals(Surf_Anal, Mesh, h, 2, 'UY');
if CaseLabel == 'Single'
    DofsY1 = [];
    UY1 = [];
elseif CaseLabel == 'Double'
    DofsY1 =
(1:floor(SpanSupRatio*nelx))' + Mesh.NDof/2; %double_span beam
    UY1 = zeros(size(DofsY1));
elseif CaseLabel == 'Triple'
    DofsY1 = (1:floor(SpanSupRatio*nelx))' + Mesh.NDof/2 +
floor((nelx+1)/3); %triple_span beam
    UY1 = zeros(size(DofsY1));
end
BdryIdcs = [DofsY; DofsX; DofsY1];
BdryVals = [UY; UX; UY1];
FreeIdcs = setdiff(1 : Mesh.NDof, BdryIdcs);
d = zeros(Mesh.NDof, 1);
d(BdryIdcs) = BdryVals;
f(FreeIdcs) = f(FreeIdcs) - K(FreeIdcs, BdryIdcs) * BdryVals;
% Solve the system
d(FreeIdcs) = K(FreeIdcs, FreeIdcs) \ f(FreeIdcs); %d-->uij
f0val = sum(Ve(:))/initial_Ve;
df0dx =
calc_df0dx_ConTen(Ve, prhoeStar_prhoi, prhoeStar_pyi, rho_estar);
ParaPts = {node_x, node_y};
edof =
[Mesh.El, Mesh.El+size(d, 1)/2]; %d(u1x, u2x, ..., unx, u1y, u2y, ..., uny)
[D_Mat, Be, ue] = calc_B_ue(Surf_Anal, ParaPts, d, Mesh, E, nu,
'PlaneStress');
epsi = 0.4;
[ele_dp1, ele_s, alpha, H, J2, he] =
Drucker_Prager_paper(rho_estar(:), Be, D_Mat, ue, theg_c, theg_t, epsi,
ho_min);
ele_dp = ele_dp1./he;
center_x = reshape(CurvStore_center(1,:), nelx, nely);
center_y = reshape(CurvStore_center(2,:), nelx, nely);
if rem(loop, 5) == 0
    f1 = figure(1);
    clf(f1)
    set(f1, 'visible', 'off');
    hold on
    daspect([1, 1, 1])
    PlotTopo(Surf_Den, Mesh, rho_CtrlPts, Tendon_Refine, miu_pre, ...
beta_fil, beta_HS, eta_ero, eta_dil, TendonLoc, tendon_x, tendon_y, Fixe
dTopRatio);
    pos = get(gcf, 'Position');
    pos(4) = pos(3)/7;
    set(gcf, 'Position', pos)
    set(gca, 'LooseInset', [0, 0, 0, 0]);
    saveas(gcf, [SavePath, num2str(loop), '.jpg'])
    f2 = figure(2);
    clf(f2)
    set(f2, 'visible', 'off');
    hold on
    axis equal
    daspect([1, 1, 1])
    center_dp = reshape(ele_dp-1, nelx, nely);
    center_x_plt = [flipud(center_x); -center_x];
    center_y_plt = [flipud(center_y); center_y];
    center_dp_plt = [flipud(center_dp); center_dp];

```

```

surf(center_x_plt,center_y_plt,zeros(size(center_dp_plt)),center_dp_
plt);
    caxis([max(-1.5,min(center_dp(:))) max(center_dp(:))]); axis off;
view(0,90);shading interp;drawnow;colorbar;colormap(jet)
    shading interp;
    saveas(gcf,[SavePath,'CurrentDP.jpg'])
end
miu = 8;
DP_KS = 1/miu*log(sum(exp(miu*ele_dp)));
qn = 0.5;
cp_old = max(ele_dp)/DP_KS;
if loop == 1
    cp = max(ele_dp)/DP_KS;
else
    cp = qn*max(ele_dp)/DP_KS+(1-qn)*cp_old;
end
cp_old = cp;
DP_KS1 = cp*DP_KS;
fval = DP_KS1-1;
dfdx = calc_dfdx_ConTen_DP(ele_dp,ele_s,cp,he,miu,rho_estar,...

```

```

Be,D_Mat,ue,d,K,edof,FreeIdcs,BdryIdcs,BdryVals,Mesh,KE,alpha,H,J2
,epsi,...

```

```

prhoeStar_prhoi,prhoeStar_pyi,dfswdrhoi,dfswdyi,dFCondy,penalty,r
ho_min);

```

```

% use MMA to optimize the system
[xmma,ymma,zmma,lam,xsi,eta,mu,zet,ss,low,upp] = ...
mmasub_PC(m,numDV,loop,xval,xmin,xmax,xold1,xold2, ...
f0val,df0dx,fval,dfdx,low,upp,a0,a,c,dd,NCP_Tendon);
xold2 = xold1;
xold1 = xval;
xval = xmma;
rho_CtrlPts = reshape(xval(1:end-NCP_Tendon),Surf_Den.NCtrlPts);
tendon_y = xval(end+1-NCP_Tendon:end);
tendon_y(end) = 0.9*D;
[Tendon,Tendon_Refine] =
BuildTendon(nelx_tendon,tendon_x,tendon_y);
rho_e = reshape(R_Center*rho_CtrlPts(:),nelx,nely);%rho_e
[rho_ehat,pdist_pyi,dist] =
TendonModiDensity(CurvStore_center,Tendon_Refine,rho_e,RefWei
ght,miu_pre,beta_fil);
rho_ehat(:,end-3:end) = 1;%lock upper bound of the beam
[prhoeHat_prhoi, prhoeHat_pyi] =
calc_drhoedvar(R_Center,dist,beta_fil,miu_pre,rho_e,pdist_pyi);
[rho_estar,prhoeStar_prhoi,prhoeStar_pyi] =
DilHeaviside_PC(rho_ehat,beta_HS,eta_ero,eta_dil,prhoeHat_prhoi,p
rhoeHat_pyi);
rho_estar(:,end-NFixLayer:end) = 1;
disp([' It.: ' sprintf('%4\t',loop) ' Obj(Vol): ' sprintf('%7.5f\t',f0val) '
Const(D-P): ' ...
sprintf('%7.5f\t',fval)]);
loopp = [loopp,loop];
volp = [volp, f0val];
stress = [stress, fval];
if rem(loop,25) == 0
    if penalty < 3
        penalty = penalty+0.5;
    end
end
end

```

B1.3 Finding the interpped location of nodes

```

function [R] = FindInterpFunc(Surf, ParaPts)
KntVects=Surf.KntVect;
CtrlPts=Surf.CtrlPts4D;
weight = permute(Surf.Weights(1,:,:),[2,3,1]);
dim = numel(KntVects);
NCtrlPts = size(CtrlPts);

```

```

if rem(loop,25) == 0
    if beta_HS < 5
        beta_HS = beta_HS + 0.25;
    end
    if miu_pre < 4
        miu_pre = miu_pre + 1;
    end
end
if loop == maxiter-1
    beta_HS = 20;
end
f1 = figure(1);
clf(f1)
set(f1,'visible','off');
hold on
daspect([1, 1, 1])
PlotTopo(Surf_Den,Mesh,rho_CtrlPts,Tendon_Refine,miu_pre,...
beta_fil,beta_HS,eta_ero,eta_dil,TendonLoc,tendon_x,tendon_y,Fixe
dTopRatio);
pos=get(gcf,'Position');
pos(4)=pos(3)/7;
set(gcf,'Position',pos)
set(gca, 'LooseInset', [0,0,0,0]);
saveas(gcf,[SavePath,'finalTopo.jpg'])
f2 = figure(2);
clf(f2)
set(f2,'visible','off');
hold on
axis equal
daspect([1, 1, 1])
center_dp = reshape(ele_dp-1,nelx,nely);
center_x_plt = [flipud(center_x):-center_x];
center_y_plt = [flipud(center_y);center_y];
center_dp_plt = [flipud(center_dp);center_dp];
surf(center_x_plt,center_y_plt,zeros(size(center_dp_plt)),center_dp_
plt);
caxis([max(-1.5,min(center_dp(:))) max(center_dp(:))]); axis off;
view(0,90);shading interp;drawnow;colorbar;colormap(jet)
shading interp;
saveas(gcf,[SavePath,'finalDP.jpg'])

Ve = CalcVe(Mesh,CurvStore_Xi,rho_estar);%element volume(area)
f0val = sum(Ve(:))/initial_Ve;
volp = [volp, f0val];
stress = [stress, fval];
disp([' It.: ' sprintf('%4\t',loop) ' Obj(Vol): ' sprintf('%7.5f\t',f0val) '
Const(D-P): ' ...
sprintf('%7.5f\t',fval)]);
writematrix([volp,stress],[SavePath,'finalRec.csv'])
f3 = figure(3);
set(f3,'visible','off');
subplot(2,1,1)
plot(volp*100);title('Volume(%)')
subplot(2,1,2)
plot(stress);title('D-P')
saveas(gcf,[SavePath,'finalHistory.jpg'])
save([SavePath,'Workspace.mat'])
end

```

```

p = zeros(1, dim);
idx = cell(1, dim);
N0 = cell(1, dim);
CtrlPts_id = cell(1, dim);
for i = 1 : dim
    p(i) = numel(KntVects{i}) - NCtrlPts(i + 1) - 1;%p=k-n-1
    idx{i} = FindSpan(NCtrlPts(i + 1), p(i), ParaPts{i}, KntVects{i});

```

```

NO{i} = BasisFuns(idx{i}, ParaPts{i}, p(i), KntVects{i});
CtrlPts_id{i} = repmat(idx{i}', 1, p(i)+1)-
p(i)+repmat(1:p(i)+1, size(ParaPts{i}, 2), 1)-1;
end
kxi = 1:size(ParaPts[181], 2);
eta = 1:size(ParaPts[181], 2);
[KXI, ETA] = meshgrid(kxi, eta);
RR = zeros(numel(KXI), Surf.Order(1)+1, Surf.Order(2)+1);
RR1 = zeros(numel(KXI), size(CtrlPts, 2), size(CtrlPts, 3));
parfor it = 1:numel(KXI)
    x_list(it,:) = CtrlPts_id[181](KXI(it,:));
    Nx_list = NO[181](KXI(it,:));
    y_list(it,:) = CtrlPts_id[181](ETA(it,:));
    Ny_list = NO[181](ETA(it,:));
    x_temp_weight = weight(x_list(it,:), y_list(it,:));

```

```

temp_w_xi = Nx_list*Ny_list.*x_temp_weight;
w_xi = sum(temp_w_xi(:));
RR(it, :, :) = temp_w_xi;
RR(it, :, :) = RR(it, :, :)/w_xi;
end
for i = 1:size(x_list, 1)
    RR1(i, x_list(i,:), y_list(i,:)) = RR(i, :, :);
end
R =
reshape(RR1, size(ParaPts[181], 2), size(ParaPts[181], 2), size(CtrlPts, 2),
size(CtrlPts, 3));
R = permute(R, [2, 1, 3, 4]);
shape_R = size(R);
R = reshape(R, [shape_R(1)*shape_R(2), shape_R(3)*shape_R(4)]);
end

```

B1.4 Generating the tendon curve

```

function [Tendon, Tendon_Refine] =
BuildTendon(nelx_tendon, tendon_x, tendon_y)
p=2;
kx=1;
NCtrlPts = length(tendon_x);
CtrlPts = zeros(4, NCtrlPts, 1);
CtrlPts(1, :, 1) = tendon_x;
CtrlPts(2, :, 1) = tendon_y;

```

```

%weight
CtrlPts(4, :, :) = 1;
KntVect = {[zeros(1, p+1), (p+1:NCtrlPts-1)/(NCtrlPts+p), ones(1, p+1)]};
% create NURBS Curve in CAD
Tendon = CreateNURBS(KntVect, CtrlPts);
InsrtKnt = setdiff(linspace(0, 1, nelx_tendon), KntVect[181]);
Tendon_Refine = HRefine(Tendon, 1, InsrtKnt);
end

```

B1.5 Refining the weight vector of the tendon

```

function Weight = RefineWeight(Tendon, Tendon_Refine)
DiffKnt = setdiff(Tendon_Refine.KntVect[181], Tendon.KntVect[181]);
NCtrlPts = size(Tendon.CtrlPts3D, 2);
Weight = eye(NCtrlPts);
IKntVect = Tendon.KntVect[181];
p = Tendon.Order;
for i = 1:length(DiffKnt)
    k = FindSpan(NCtrlPts, p, DiffKnt(i), IKntVect);
    size_Weight_i = [NCtrlPts+1, NCtrlPts];
    Weight_i = zeros(size_Weight_i);
    Weight_i(1:k-p, 1:k-p) = eye(k-p);

```

```

Weight_i(k+1:end, k:end) = eye(NCtrlPts-k+1);
for l = 1:p
    j = k-p+l;
    alpha = (DiffKnt(i)-IKntVect(j))/(IKntVect(j+p)-IKntVect(j));
    Weight_i(k-p+l, k-p+l-1:k-p+l) = [1-alpha, alpha];
end
Weight = Weight_i*Weight;
NCtrlPts = NCtrlPts + 1;
IKntVect = [IKntVect(1:k), DiffKnt(i), IKntVect(k+1:end)];
end

```

B1.6 Tendon filter

```

function [rho_ehat, pdist_pyi, dist] =
TendonModiDensity(CurvStore_center, Tendon_Refine, rho_e, RefWeight,
miu_pre, beta_fil)
TendonLoc = Tendon_Refine.CtrlPts3D;
ConcreteLoc = [CurvStore_center; zeros(1, size(CurvStore_center, 2))];
distMat = pdist2(TendonLoc, ConcreteLoc);
[dist, dist_id] = min(distMat); %dist_id is the closest tendon CP for i-th
Con-CP
%Super Gaussian
rho_i = rho_e(:);

```

```

rho_modi = exp(-0.5*(dist/beta_fil).^miu_pre);
rho_ehat = rho_i+(1-rho_i).*rho_modi;
rho_ehat = reshape(rho_ehat, size(rho_e));
pdist_pyi = zeros(length(rho_ehat(:)), Tendon_Refine.NCtrlPts);
for i = 1:length(rho_ehat(:))
    pdist_pyii(i, dist_id(i)) = (TendonLoc(2, dist_id(i))-
ConcreteLoc(2, i))/dist(i);
end
pdist_pyi = pdist_pyii*RefWeight;
end

```

B1.7 Computing $prho_e/px$

```

function [prhoehat_prhoi, prhoehat_pyi] = calc_drhoedvar(R_Center, dist, beta_fil, miu_pre, rho_e, pdist_pyi)
rho_Ten = exp(-0.5*(dist/beta_fil).^miu_pre);
prhoehat_prhoe = diag(1-rho_Ten);
prhoe_prhoi = R_Center;
prhoehat_prhoi = prhoehat_prhoe*prhoe_prhoi;
prhoehat_pyi = (1-rho_e(:)).*(-miu_pre/2/beta_fil).*rho_Ten(:).*(dist(:)/beta_fil).^(miu_pre-1)...
.*pdist_pyi;

```

end

B1.9 Density projection

```
function [rho_estar,prhoeStar_prhoi,prhoeStar_pyi] =
DilHeaviside_PC(rho_ehat,beta_HS,eta_ero,eta_dil,prhoeHat_prhoi,p
r hoeHat_pyi)
rho_min = 1e-3;
rho_ero = (tanh(beta_HS*eta_ero)+tanh(beta_HS*(rho_ehat-
eta_ero)))/...
(tanh(beta_HS*eta_ero)+tanh(beta_HS*(1-eta_ero)));
rho_estar = (tanh(beta_HS*eta_dil)+tanh(beta_HS*(rho_ero-
eta_dil)))/...
(tanh(beta_HS*eta_dil)+tanh(beta_HS*(1-eta_dil)));
rho_estar = rho_min + (1-rho_min)*rho_estar;
```

```
prhoeero_prhoehat = beta_HS*(1-tanh(beta_HS*(rho_ehat-
eta_ero)).^2)/...
(tanh(beta_HS*eta_ero)+tanh(beta_HS*(1-eta_ero)));
prhoeStar_prhoeero = beta_HS*(1-tanh(beta_HS*(rho_ero-
eta_dil)).^2)/...
(tanh(beta_HS*eta_dil)+tanh(beta_HS*(1-eta_dil)));
prhoeStar_prhoehat =
diag(prhoeStar_prhoeero(:).*prhoeero_prhoehat(:));
prhoeStar_prhoi = prhoeStar_prhoehat*prhoeHat_prhoi;
prhoeStar_pyi = prhoeStar_prhoehat*prhoeHat_pyi;
end
```

B1.10 Computing the elemental volume

```
function Ve = CalcVe(Mesh,CurvStore_Xi,rho_ParaPts)
Ve = zeros(Mesh.NEIDir);
for i = 1:Mesh.NEIDir(1)
for j = 1:Mesh.NEIDir(2)
temp_node_corner1 = (i-1)+(j-1)*(Mesh.NEIDir(1)+1)+1;
temp_node_corner2 = (i-1)+j*(Mesh.NEIDir(1)+1)+1;
temp_node_id = [temp_node_corner1,temp_node_corner1+1,...
temp_node_corner2,temp_node_corner2+1];
coor_node = CurvStore_Xi(1:2,temp_node_id);
tri_1 = coor_node(:,[1 2 3]);
tri_2 = coor_node(:,[2 3 4]);
```

```
S1 = calc_tri_area(tri_1);
S2 = calc_tri_area(tri_2);
Ve(i,j) = (S1+S2)*rho_ParaPts(i,j);
end
end
end
function area = calc_tri_area(tri)
x1 = tri(1,1); x2 = tri(1,2); x3 = tri(1,3);
y1 = tri(2,1); y2 = tri(2,2); y3 = tri(2,3);
area = abs(x1*y2+x2*y3+x3*y1-x1*y3-x2*y1-x3*y2)/2;
end
```

B1.11 Computing the prestressed load

```
function [F_Con,ConfigTendon,dFCondy] =
Prestress_Force(Surf_Anal,L,D,Tendon_Refine,RefWeight,Tpre,Mesh,
beta_fil)
ConfigTendon = Tendon_Refine.CtrlPts3D;
Mesh_Tendon = Mesh1D(Tendon_Refine);
NGPs = Tendon_Refine.Order + 1;
[Jx, Wx, ~, Nx] = calcDersBasisFunsAtGPs(Tendon_Refine.Order,
Tendon_Refine.NCtrlPts, Tendon_Refine.KntVect[181], 2, NGPs,
Mesh_Tendon.NEI);
Weights = reshape(Tendon_Refine.Weights, 1, []);
CtrlPts = reshape(Tendon_Refine.CtrlPts3D, 3, []);
FTenVals = zeros(Mesh_Tendon.NDof*2, 1);
dFTendy = zeros(Mesh_Tendon.NDof*2, size(RefWeight,2));
loc = [Mesh_Tendon.El, Mesh_Tendon.El + Tendon_Refine.NNP];
% set concentrated load of tendon
[EvalPts,N0n] = DerBsplineEval(Tendon_Refine.KntVect,
Tendon_Refine.CtrlPts4D, 1, {[0,1]});
Bdry_Grad = EvalPts[181](1:3,:);
%normalize the gradient dx/dxi
J01 = norm(Bdry_Grad(:,1));
J02 = norm(Bdry_Grad(:,2));
Grad_1 = Bdry_Grad(:,1)/norm(Bdry_Grad(:,1));
Grad_2 = Bdry_Grad(:,2)/norm(Bdry_Grad(:,2));
Corre_Coor =
[1,Tendon_Refine.NNP,1+Tendon_Refine.NNP,2*Tendon_Refine.NNP
];
N00 = N0n[181](:,:);
N01 = N0n[181](:,:);
[~,R01] =
Rationalize(Weights(Mesh_Tendon.El(1,:)),N00(1,:),N01(1,:));%for
initial
```

```
[~,R02] =
Rationalize(Weights(Mesh_Tendon.El(end,:)),N00(2,:),N01(2,:));%for
end
pj01py = (R01*RefWeight(Mesh_Tendon.El(1,:)))*[0,1];
pj01normpy = Grad_1(1:2)*pj01py;
pj02py = (R02*RefWeight(Mesh_Tendon.El(end,:)))*[0,1];
pj02normpy = Grad_2(1:2)*pj02py;
pn1py = 1/J01*pj01py'-Grad_1(1:2)/J01*pj01normpy;
pn2py = 1/J02*pj02py'-Grad_2(1:2)/J02*pj02normpy;
Fx_preCon = -Grad_2(1)*Tpre*0.75;
Fy_preCon = Grad_1(2)*Tpre;
dFTendy(Corre_Coor,:) = Tpre*[pn1py(1,:);-pn2py(1,:);pn1py(2,:);-
pn2py(2,:)];
xi = Tendon_Refine.CtrlPts3D(1,:)/L;
eta = Tendon_Refine.CtrlPts3D(2,:)/D;
[R_ConTen,dR_ConTen] = FindInterpFunc_ConTen(Surf_Anal,{xi, eta});
shape_R = size(R_ConTen);
R_ConTen =
reshape(R_ConTen,[shape_R(1)*shape_R(2),shape_R(3)*shape_R(4)
]);% for center of element
shape_dR = size(dR_ConTen);
dR_ConTen =
reshape(dR_ConTen,[shape_dR(1)*shape_dR(2),shape_dR(3)*shape_
dR(4)]);% for center of element
dRConTendy = zeros([size(dR_ConTen),size(RefWeight,2)]);
for i = 1:size(dRConTendy,3)
dRConTendy(:,i) =
dR_ConTen.*repmat(RefWeight(:,i),1,size(dR_ConTen,2))/D;
end
for e = 1:Mesh_Tendon.NEI
for q = 1:NGPs
NO = Nx(e, q, :, 1);
```

```

N1 = Nx(e, q, :, 2);
N2 = Nx(e, q, :, 3);
[R0, R1, R2] = Rationalize(Weights(Mesh_Tendon.El(e, :)), N0(:)',
N1(:)', N2(:)');
dxdxi = R1 * CtrlPts(Mesh_Tendon.El(e, :), :);
J1 = norm(dxdxi);
C1 = R1 * CtrlPts(Mesh_Tendon.El(e, :), :);
C2 = R2 * CtrlPts(Mesh_Tendon.El(e, :), :);
cross_result = cross(C2,C1);
curv = cross_result(3)/norm(C1)^3;
p_GP = curv*Tpre;
n0 = dxdxi(1:2)/norm(dxdxi);
n = [0 1; -1 0]*n0; % dxdxi: tangential vector
[pQpy,pnormJ1py] =
calc_pqpy(J1,R1,R2,RefWeight,n0,n,e,C1,C2,Mesh_Tendon,Tpre,p_GP
);
pressure = repmat(p_GP, 2, 1) .* n;
ROMat = zeros(2, Mesh_Tendon.NEN * 2); % matrix of tendon
basis functions
ROMat(1, 1 : Mesh_Tendon.NEN) = R0;
ROMat(2, Mesh_Tendon.NEN + 1 : 2 * Mesh_Tendon.NEN) = R0;
FTenVals(loc(:,e)) = FTenVals(loc(:,e)) + Wx(q) * ROMat' *
pressure * J1 * Jx(e);
dFTendy(loc(:,e),:) = dFTendy(loc(:,e),:) + Wx(q) * ROMat' * pQpy'
* J1 * Jx(e) + ...
Wx(q) * ROMat' * pressure * pnormJ1py' * Jx(e);
end
end
R_ConTenMat = [R_ConTen zeros(size(R_ConTen));
zeros(size(R_ConTen)) R_ConTen];
temp1 = cat(1,dRConTendy,zeros(size(dRConTendy)));
temp2 = cat(1,zeros(size(dRConTendy)),dRConTendy);
dRConTenMat = cat(2,temp1,temp2);
F_Con = R_ConTenMat*FTenVals;
[Fprex_Con,DofsFx_precon] =
calc_concentratedLoadx(Tendon_Refine,Surf_Anal,Mesh,D,Fx_preCon
n,beta_fil);
F_Con(DofsFx_precon) = F_Con(DofsFx_precon) + Fprex_Con;
[Fprey_Con,DofsFy_precon] =
calc_concentratedLoady(Tendon_Refine,Surf_Anal,Mesh,D,Fy_preCon
n,beta_fil);

```

```

F_Con(DofsFy_precon) = F_Con(DofsFy_precon) + Fprey_Con;
size_dRConTenMat = size(dRConTenMat);
size_dRConTenMat = size_dRConTenMat(2:3);
temp = zeros(size_dRConTenMat);
for i = 1:size(dRConTenMat,3)
temp(:,i) = dRConTenMat(:,i)*FTenVals;
end
dFCondy = R_ConTenMat'*dFTendy + temp;
end
function [pQpy,pnormJ1py] =
calc_pqpy(J1,R1,R2,RefWeight,n0,n,e,C1,C2,Mesh_Tendon,Tpre,p_GP)
pJ1py = (R1*RefWeight(Mesh_Tendon.El(e,:)))*[0,1];
pnormJ1py = pJ1py*n0;
pnpvy = [0,1;-1,0]*(1/J1*pJ1py'-n0/J1*pnormJ1py');
x1 = C1(1);y1 = C1(2);
x2 = C2(1);y2 = C2(2);
pkpy = ((x2*R1-x1*R2)/norm(C1)^3-3*(x2*y1-
x1*y2)*y1/norm(C1)^5*R1)*RefWeight(Mesh_Tendon.El(e,:));
pqpy = pkpy*Tpre;
pQpy = (n*pqpy+p_GP*pnpvy');
end
function [Fprex_Con,DofsFx_precon] =
calc_concentratedLoadx(Tendon_Refine,Surf,Mesh,D,Fx_preCon,beta
_fil)
para_y = Tendon_Refine.CtrlPts3D(2,end)/D;
load_area = 2*beta_fil;
press_pre = @(x,y) Fx_preCon/load_area*exp(-0.5*(abs(y-
para_y*D)/beta_fil).^100);
[Fprex_Con, DofsFx_precon] = applyNewmannBdryVals(Surf, Mesh,
press_pre, 2, 'FX');
end
function [Fprey_Con,DofsFy_precon] =
calc_concentratedLoady(Tendon_Refine,Surf,Mesh,D,Fy_preCon,beta
_fil)
para_y = Tendon_Refine.CtrlPts3D(2,1)/D;
load_area = 2*beta_fil;
press_pre = @(x,y) Fy_preCon/load_area*exp(-0.5*(abs(y-
para_y*D)/beta_fil).^100);
[Fprey_Con, DofsFy_precon] = applyNewmannBdryVals(Surf, Mesh,
press_pre, 1, 'Fy');
end

```

B1.12 Applying the gravity load

```

function [FVals, dfswdrhoi, dfswdyi] = applyGravity_PC(Surf_Anal,
Mesh, g, rho_estar, prhoe_prhoi,prhoe_pyi)
%R1_Ref: drhoedrhoi
fi = rho_estar*g;
NGPs = Surf_Anal.Order + 1;
[Jx, Wx, ~, Nx] = calcDersBasisFunsAtGPs(Surf_Anal.Order(1),
Surf_Anal.NCtrlPts(1), Surf_Anal.KntVect[181], 1, NGPs(1),
Mesh.NEIDir(1));
[Jy, Wy, ~, Ny] = calcDersBasisFunsAtGPs(Surf_Anal.Order(2),
Surf_Anal.NCtrlPts(2), Surf_Anal.KntVect[181], 1, NGPs(2),
Mesh.NEIDir(2));
Weights = reshape(Surf_Anal.Weights, 1, []);
CtrlPts = reshape(Surf_Anal.CtrlPts3D, 3, []);
FVals = zeros(Mesh.NDof, 1);
dFdrhoeVals = zeros(Mesh.NDof, Mesh.NEI);
loc = [Mesh.El, Mesh.El + Surf_Anal.NNP];
for ey = 1 : Mesh.NEIDir(2)
for ex = 1 : Mesh.NEIDir(1)
e = sub2ind(Mesh.NEIDir, ex, ey);
for qy = 1 : NGPs(2)
for qx = 1 : NGPs(1)
N0x = Nx(ex, qx, :, 1);
N1x = Nx(ex, qx, :, 2);
N0y = Ny(ey, qy, :, 1);
N1y = Ny(ey, qy, :, 2);
N0 = bsxfun(@times, N0x(:), N0y(:)');

```

```

N11 = bsxfun(@times, N1x(:), N0y(:)');
N12 = bsxfun(@times, N0x(:), N1y(:)');
[R0, R1] = Rationalize(Weights(Mesh.El(e, :)), N0(:)', [N11(:)';
N12(:)']);
J2 = Jx(ex) * Jy(ey);
W = Wx(qx) * Wy(qy);
% gradient of mapping from parameter space to physical
space
dxdxi = R1 * CtrlPts(Mesh.El(e, :), 1:2);
t1 = [dxdxi(1, :),0];
t2 = [dxdxi(2, :),0];
J1 = norm(cross(t1,t2));
p = [0,-fi(ex,ey)];%p=rho*g
ROMat = zeros(2, 2 * Mesh.NEN);
ROMat(1, 1 : Mesh.NEN) = R0;
ROMat(2, Mesh.NEN + 1 : 2 * Mesh.NEN) = R0;
FVals(loc(:,e)) = FVals(loc(:,e)) + W * ROMat' * p * J1 * J2;
dFdrhoeVals(loc(:,e),e) = dFdrhoeVals(loc(:,e),e) +
W*ROMat' * p/rho_estar(ex,ey) * J1 * J2;
end
end
end
dfswdrhoi = dFdrhoeVals*prhoe_prhoi;
dfswdyi = dFdrhoeVals*prhoe_pyi;
end

```

B1.13 Computing pV/px

```
function df0dx =
calc_df0dx_ConTen(Ve,prhoeStar_prhoi,prhoeStar_pyi,rho_estar)
Ve = Ve./rho_estar;
Ve1 = Ve(:);
dVdrhoi = prhoeStar_prhoi*Ve1;
```

```
dVdyi = prhoeStar_pyi*Ve1;
dVdrhoi = dVdrhoi/max(abs(dVdrhoi));
dVdyi = dVdyi/max(abs(dVdyi));
df0dx = [dVdrhoi;dVdyi];
```

B1.14 Computing elastic matrix, strain matrix, and elemental displacement

```
function [D,Be,ue] = calc_B_ue(NURBS, ParaPts, d, Mesh, E, nu, lab)
%2d only
KntVects = NURBS.KntVect;
CtrlPts = NURBS.CtrlPts4D;
D = getElastMat(E, nu, lab);
F1 = reshape(d, NURBS.NNP, []);
F2 = reshape(reshape(d, NURBS.NNP, []), [size(F1, 1),
NURBS.NCtrlPts]);
d = F2;
assert(iscell(KntVects), 'Knot vector(s) must be in cell format');
assert(iscell(ParaPts), 'Parameter points must be in cell format');
dim = numel(KntVects);
NCtrlPts = size(CtrlPts);
p = zeros(1, dim);
Idx = cell(1, dim);
N01 = cell(1, dim);
for i = 1 : dim
    p(i) = numel(KntVects{i}) - NCtrlPts(i + 1) - 1;
    Idx{i} = FindSpan(NCtrlPts(i + 1), p(i), ParaPts{i}, KntVects{i});
    N01{i} = DersBasisFuns(Idx{i}, ParaPts{i}, p(i), 1, KntVects{i});
end
Npts = cellfun(@numel, ParaPts);
nen = prod(p + 1);
g = zeros([dim, size(d, 1), Npts]);
Be = zeros([3, 2*(p(1)+1)*(p(2)+1), Npts]);
ue = zeros([2*(p(1)+1)*(p(2)+1), 1, Npts]);
N0 = zeros(1, nen);
N1 = zeros(dim, nen);
j_max = Npts(2);
parfor i = 1 : Npts(1)
    [ue(:,i,:),Be(:,i,:)] =
calc_dRdx(i,j_max,d,Idx,N0,N01,N1,p,CtrlPts,Mesh,nen);
end
Be = reshape(Be,size(Be,1),size(Be,2),[]);
ue = reshape(ue,size(ue,1),size(ue,2),[]);
end
```

```
function [ue,Be] =
calc_dRdx(i,j_max,d,Idx,N0,N01,N1,p,CtrlPts,Mesh,nen)
for j = 1 : j_max
    de = d(:, Idx[181](i) - p(1) : Idx[181](i), Idx[181](j) - p(2) :
Idx[181](j));
    Weights = CtrlPts(4, Idx[181](i) - p(1) : Idx[181](i), Idx[181](j) -
p(2) : Idx[181](j));
    xye = bsxfun(@rdivide, CtrlPts(1 : 2, Idx[181](i) - p(1) : Idx[181](i),
Idx[181](j) - p(2) : Idx[181](j)), Weights);
    k = 1;
    for jk = 1 : p(2) + 1
        for ik = 1 : p(1) + 1
            N0(k) = N01[181](i, ik, 1) * N01[181](j, jk, 1);
            N1(1, k) = N01[181](i, ik, 2) * N01[181](j, jk, 1);
            N1(2, k) = N01[181](i, ik, 1) * N01[181](j, jk, 2);
            k = k + 1;
        end
    end
    [~, R1] = Rationalize(reshape(Weights, 1, []), N0, N1);
    ue(:,j) = reshape(reshape(de, 2, []), 1, []);
    dxdxi = R1 * reshape(xye, [], nen);
    %parallel calculation, solve the issue of multiple small-scale inv
    dRdx = dxdxi^(-1) * R1;
    % B matrix
    %
    %   | N_1,x N_2,x ... 0 0 ... |
    % B = | 0 0 ... N_1,y N_2,y |
    %   | N_1,y N_2,y ... N_1,x N_2,x |
    %   - - - - -
    B = zeros(3, 2 * Mesh.NEN);
    B(1, 1 : Mesh.NEN) = dRdx(1, :);
    B(2, Mesh.NEN + 1 : end) = dRdx(2, :);
    B(3, 1 : Mesh.NEN) = dRdx(2, :);
    B(3, Mesh.NEN + 1 : end) = dRdx(1, :);
    Be(:,j) = B;
end
end
```

B1.15 Computing elemental D-P stress

```
function [ele_dp,ele_s,alpha,H,J2,he] =
Drucker_Prager_paper(rho_ParaPts,Be,D_Mat,ue,the_g_c,the_g_t,epsi,
rho_min)
V = [1,-0.5,0;-0.5,1,0;0,0,3];
ele_dp = zeros(size(rho_ParaPts,1),1);
ele_s = zeros(size(rho_ParaPts,1),3);
I1 = zeros(size(rho_ParaPts,1),1);
J2 = zeros(size(rho_ParaPts,1),1);
w = [1,1,0];
alpha = (the_g_c-the_g_t)/(the_g_c+the_g_t)/sqrt(3);
H = 2*the_g_c*the_g_t/(the_g_c+the_g_t)/sqrt(3);
```

```
he = zeros(size(rho_ParaPts,1),1);
penalty_s = 0.5;
for i = 1:length(rho_ParaPts)
    ele_s(i,:) =
(D_Mat*Be(:,i)*ue(:,i))'*(rho_ParaPts(i).^penalty_s+rho_min);
    I1(i) = w*ele_s(i,:);
    J2(i) = ele_s(i,:)*V*ele_s(i,:)/3;
    he(i) = 1-epsi+epsi/rho_ParaPts(i);
    ele_dp(i) = (alpha*I1(i)+sqrt(J2(i)))/H;
end
```

B1.16 Plotting current design

```
function
rho_ehat=PlotTopo(Surf_Den,Mesh,rho_CtrlPts,Tendon_Refine,...

miu_pre,beta_fil,beta_HS,eta_ero,eta_dil,TendonLoc,tendon_x,tend
on_y,FixedTopRatio)
refine_level = 10;
xi = linspace(0, 1, refine_level*Mesh.NEIDir(1)); % parametric points
eta = linspace(0, 1, refine_level*Mesh.NEIDir(2)); % parametric points
% evaluate the parametric points to plot the surface
CtrlPts =
[Surf_Den.CtrlPts4D;reshape(rho_CtrlPts,[1,size(rho_CtrlPts)])];
c = gray;
c = flipud(c);
colormap(c)
xi_e = (xi(1:end-1)+xi(2:end))/2;
eta_e = (eta(1:end-1)+eta(2:end))/2;
Sw_e = BsplineEval(Surf_Den.KntVect, CtrlPts, {xi_e, eta_e});
[~, m_e, n_e] = size(Sw_e);
w_e = Sw_e(4, :, :);
S_e = bsxfun(@rdivide, Sw_e, w_e);
x_e = reshape(S_e(1, :, :), m_e, n_e);
y_e = reshape(S_e(2, :, :), m_e, n_e);
z_e = reshape(S_e(3, :, :), m_e, n_e);
rho_e = reshape(S_e(5, :, :), m_e, n_e);%permute(S_e(5, :, :),[2,3,1]);
rho_ehat =
TendonModiDensity_ForPlot([x_e(:),y_e(:),z_e(:)]',Tendon_Refine,rho
_e,miu_pre,beta_fil);
rho_ero = (tanh(beta_HS*eta_ero)+tanh(beta_HS*(rho_ehat-
eta_ero)))/...
(tanh(beta_HS*eta_ero)+tanh(beta_HS*(1-eta_ero)));
rho_estar = (tanh(beta_HS*eta_dil)+tanh(beta_HS*(rho_ero-
eta_dil)))/...
```

```
(tanh(beta_HS*eta_dil)+tanh(beta_HS*(1-eta_dil)));
NFixLayer = floor(FixedTopRatio*length(eta_e));
rho_estar(:,end-NFixLayer:end) = 1;
x_e_plt = [flipud(x_e);-x_e];
y_e_plt = [flipud(y_e);y_e];
z_e_plt = [flipud(z_e);z_e];
rho_estar_plt = [flipud(rho_estar);rho_estar];
surf(x_e_plt, y_e_plt, z_e_plt, rho_estar_plt);
axis normal;
caxis([0 1]); axis off; view(0,90);shading
interp;drawnow;colormap(flipud(gray))
TendonLoc_x_plt = [fliplr(-1*TendonLoc(1,2:end)),TendonLoc(1,:)];
TendonLoc_y_plt = [fliplr(TendonLoc(2,2:end)),TendonLoc(2,:)];
tendon_x_plt = [fliplr(-1*tendon_x(2:end)),tendon_x];
tendon_y_plt = [fliplr(tendon_y(2:end)),tendon_y];
plot(TendonLoc_x_plt,TendonLoc_y_plt,'LineWidth',1.5,'Color','c')
plot(tendon_x_plt,tendon_y_plt,'*c','MarkerSize',6)
end
```

```
function rho_ehat =
TendonModiDensity_ForPlot(ConcreteLoc,Tendon_Refine,rho_e,miu
_pre,beta_fil)
TendonLoc = Tendon_Refine.CtrlPts3D;
distMat = pdist2(TendonLoc',ConcreteLoc');
dist = min(distMat);%dist_id is the closest tendon CP for i-th Con-
CP
%Super Gaussian
rho_i = rho_e(:);
rho_modi = exp(-0.5*(dist/beta_fil).^miu_pre);
rho_ehat = rho_i+(1-rho_i).*rho_modi;
rho_ehat = reshape(rho_ehat,size(rho_e));
end
```

B1.17 Computing pS_{dp}/px

```
function dfdx =
calc_dfdx_ConTen_DP(ele_dp,ele_s,cp,he,miu,rho_ParaPts,...

Be,D_Mat,ue,d,K,edof,FreeIdcs,BdryIdcs,BdryVals,Mesh,KE,alpha,H,J2
,epsi,...

prhoe_prhoi,prhoe_pyi,dfswdrhoi,dfswdyi,dFCondy,penalty,rho_min)
KE = reshape(KE,size(KE,1),size(KE,2),[]);
rho_ParaPts = rho_ParaPts(:);
drhoe_dx = [prhoe_prhoi,prhoe_pyi];
dfsw_dx = [dfswdrhoi,dfswdyi];
dfpre_dx = [zeros(size(dfswdrhoi)),dFCondy];
pGks_pphife = exp(miu*ele_dp)/sum(exp(miu*ele_dp));
w0 = [1,1,0];V = [1,-0.5,0;-0.5,1,0;0,0,3];
pphife_pthege =
(repmat(alpha*w0,length(ele_s),1)+ele_s*V./(3*repmat(J2,1,3).^0.5))
/H./he;
comp1 = zeros(1,size(drhoe_dx,2));
pphife_phe = -ele_dp./he;
dhe_drhoe = -epsi./rho_ParaPts.^2;
penalty_s = 0.5;
for i = 1:length(rho_ParaPts)
pthege_prhoe = penalty_s*rho_ParaPts(i)^(penalty_s-
1)*D_Mat*Be(:,i)*ue(:,i);
comp1 = comp1+cp*pGks_pphife(i)*(...
pphife_pthege(i,:)*pthege_prhoe*drhoe_dx(i,:)+...
pphife_phe(i)*dhe_drhoe(i)*drhoe_dx(i,:));
```

```
end
%adjoint method
A = zeros(1,size(d,1));
for i = 1:length(rho_ParaPts)
pthege_pu = (rho_ParaPts(i).^penalty_s+rho_min)*D_Mat*Be(:,i);
A(edof(i,:)) = A(edof(i,:))-cp*pGks_pphife(i)*...
pphife_pthege(i,:)*pthege_pu;
end
lambda_KS = zeros(Mesh.NDof, 1);
lambda_KS(BdryIdcs) = BdryVals;
lambda_KS(FreeIdcs) = K(FreeIdcs, FreeIdcs) \ A(FreeIdcs);%???
lambda_KS_e = extract(edof,lambda_KS);
dfdx = comp1;
for i = 1:length(rho_ParaPts)
adjEleComp = lambda_KS_e(i,:)*KE(:,i)*ue(:,i);
trm1 = rho_ParaPts(i).^penalty_s+rho_min; trm2 =
penalty_s*rho_ParaPts(i).^penalty_s-1;
dfdx = dfdx+adjEleComp*trm2/trm1*drhoe_dx(i,:);
end
dfdx = (dfdx - lambda_KS*(dfsw_dx+dfpre_dx));
dfdx(1:size(dfswdrhoi,2)) =
dfdx(1:size(dfswdrhoi,2))./max(abs(dfdx(1:size(dfswdrhoi,2))));
dfdx(end+1-size(dfswdyi,2):end) = dfdx(end+1-
size(dfswdyi,2):end)./max(abs(dfdx(end+1-size(dfswdyi,2):end)));
dfdx = dfdx./max(abs(dfdx));
end
```

B1.18 Updated MMA function to enable different move limits of concrete density and

tendon shape variables

```

function [xmma,ymma,zmma,lam,xsi,eta,mu,zet,s,low,upp] = ...
mmasub_PC(m,n,iter,xval,xmin,xmax,xold1,xold2, ...
f0val,df0dx,fval,dfdx,low,upp,a0,a,c,d,NCP_Tendon)
epsimin = 10^(-10);
raa0 = 0.01;
albefa = 0.4;
asyinit = 0.1;
asyincr = 0.8;
asydecr = 0.6;
eeen = ones(n,1);
eeem = ones(m,1);
zeron = zeros(n,1);
% Calculation of the asymptotes low and upp :
if iter < 2.5 %Eq 3.11
    low = xval - asyinit*(xmax-xmin);
    upp = xval + asyinit*(xmax-xmin);
else
    zzz = (xval-xold1).*(xold1-xold2); %Eq 3.13
    factor = eeem;
    factor(find(zzz > 0)) = asyincr;
    factor(find(zzz < 0)) = asydecr;
    low = xval - factor.*(xold1 - low);
    upp = xval + factor.*(upp - xold1); %Eq 3.12
    xval1 = xval(1:end-NCP_Tendon);xmin1 = xmin(1:end-
NCP_Tendon);xmax1 = xmax(1:end-NCP_Tendon);
    xval2 = xval(end-NCP_Tendon+1:end);xmin2 = xmin(end-
NCP_Tendon+1:end);xmax2 = xmax(end-NCP_Tendon+1:end);
% ff1 = 2e-2;
if iter < 50
    ff1 = 2e-1;
    ff2 = 2e-1;
elseif iter<150
    ff1 = 1e-1;
    ff2 = 5e-2;
else
    ff1 = 2e-1;
    ff2 = 5e-3;
end
low1min = xval1 - ff1*(xmax1-xmin1);low2min = xval2 - ff2*(xmax2-
xmin2);
low1max = xval1 - ff1*(xmax1-xmin1);low2max = xval2 - ff2*(xmax2-
xmin2);
upp1min = xval1 + ff1*(xmax1-xmin1);upp2min = xval2 +
ff2*(xmax2-xmin2);
upp1max = xval1 + ff1*(xmax1-xmin1);upp2max = xval2 +
ff2*(xmax2-xmin2);
lowmin = [low1min;low2min];
lowmax = [low1max;low2max];
upppmin = [upp1min;upp2min];
upppmax = [upp1max;upp2max];
low = max(low,lowmin);
low = min(low,lowmax);
upp = min(upp,upppmax);
upp = max(upp,upppmin);
end
% Calculation of the bounds alfa and beta :
zzz = low + albefa*(xval-low); %Eq 3.6
alfa = max(zzz,xmin);
zzz = upp - albefa*(upp-xval); %Eq 3.7
beta = min(zzz,xmax);

% Calculations of p0, q0, P, Q and b.
xmami = xmax-xmin;
xmamieps = 0.00001*eeen;
xmami = max(xmami,xmamieps);
xmamiinv = eeem./xmami;
ux1 = upp-xval;
ux2 = ux1.*ux1;
xl1 = xval-low;
xl2 = xl1.*xl1;
uxinv = eeem./ux1;
xlinv = eeem./xl1;
%
p0 = zeron;
q0 = zeron;
% p0 = max(df0dx,0);
% q0 = max(-df0dx,0);
p0(find(df0dx > 0)) = df0dx(find(df0dx > 0)); %Eq3.3 for objective
function
q0(find(df0dx < 0)) = -df0dx(find(df0dx < 0));
pq0 = 0.001*(p0 + q0) + raa0*xmamiinv;
p0 = p0 + pq0;
q0 = q0 + pq0;
p0 = p0.*ux2;
q0 = q0.*xl2;
%
% P = sparse(m,n);
% Q = sparse(m,n);
P = zeros(m,n);
Q = zeros(m,n);
% P = max(dfdx,0);
% Q = max(-dfdx,0);
P(find(dfdx > 0)) = dfdx(find(dfdx > 0));
Q(find(dfdx < 0)) = -dfdx(find(dfdx < 0));
PQ = 0.001*(P + Q) + raa0*eeem*xmamiinv';
P = P + PQ;
Q = Q + PQ;
P = P * spdiags(ux2,0,n,n);
Q = Q * spdiags(xl2,0,n,n);
b = P*uxinv + Q*xlinv - fval;
%-Eq 3.5
%
%% Solving the subproblem by a primal-dual Newton method
[xmma,ymma,zmma,lam,xsi,eta,mu,zet,s] = ...
subsolv(m,n,epsimin,low,upp,alfa,beta,p0,q0,P,Q,a0,a,b,c,d);

```


Appendix C

C1 Main script in Python

(library dependency: numpy, scipy, PyQt5, matplotlib, pyevtk, pyvista)

(API dependency: matlab, opensees)

The version of OpenSees with IGA elements can be found in

<https://openseesforfire.github.io>

```

import matlab.engine
import numpy as np
import scipy.io as scio
import os
os.add_dll_directory('C:\\program files\\tcl\\bin')
import sys
from opensees import *
from PyQt5 import QtWidgets, QtCore, sip
from OPS_ITO_GUI import Ui_Form
import time
import matplotlib
matplotlib.use('Qt5Agg')
from matplotlib.backends.backend_qt5agg import
FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure
from pyevtk.hl import gridToVTK
import pyvista as pv
from pyvistaqt import QtInteractor
import time
class MyMplCanvasTopo2D(FigureCanvas):
    def __init__(self, parent=None):
        self.fig =
Figure(figsize=(1.8,1.1),tight_layout=True,facecolor='0.95')
        self.axes = self.fig.add_subplot(111)
        self.axes.set_aspect("equal")
        self.axes.set_xticks([])
        self.axes.set_yticks([])
        FigureCanvas.__init__(self, self.fig)
        self.setParent(parent)
    def updateTopo2D(self,loc_x,loc_y,rhoi_plt):
        self.axes.cla()
        self.axes.set_xticks([])
        self.axes.set_yticks([])
        self.axes.axis('off')
        self.Cont =
self.axes.contourf(loc_x,loc_y,rhoi_plt,100,vmin=0,vmax=1,cmap=ma
tplotlib.cm.Greys)
        self.fig.canvas.draw()# redraw the canvas
        self.fig.canvas.flush_events()# update the canvas
class MyVtkTopo3D():
    def __init__(self, widget):
        self.widget = widget
        self.vtk_widget = QtInteractor(self.widget)
        self.vtk_widget.set_background('white')
        def updateTopo3D(self,loop,loc_x,loc_y,loc_z,rhoi_plt):
            if loop == 1:
                self.grid =
pv.StructuredGrid(loc_x,loc_y,loc_z)
                self.grid.point_data["rho"] =
rhoi_plt.flatten(order="F")
                self.vtk_widget.add_mesh(self.grid,clim=[0,1],show_scalar_ba
r=False, opacity='linear',cmap='Greys')
                self.vtk_widget.show()
            else:
                self.grid.point_data["rho"] =
rhoi_plt.flatten(order="F")
                if np remainder(loop,5)==0:
                    fileName = "/Topology"+str(loop)
                    gridToVTK(fileName,loc_x,loc_y,loc_z,pointData={'rho':rhoi_pl
t})
class MyMplCanvasLine(FigureCanvas):
    def __init__(self, parent=None):
        self.fig =
Figure(figsize=(1.8,1.1),linewidth=0.5,tight_layout=True,facecolor='0.
95',edgecolor='0.95')
        self.axes = self.fig.add_subplot(111)
        self.axes.tick_params(labelsize=5)
        self.line, = self.axes.plot([],[],color="black")
        FigureCanvas.__init__(self, self.fig)
        self.setParent(parent)
class OPSIThread(QtCore.QThread):
    signal = QtCore.pyqtSignal(str)
    finished = QtCore.pyqtSignal()
    return_2Dfig = QtCore.pyqtSignal(object, object, object, object,
object, object)
    return_3Dfig = QtCore.pyqtSignal(object, object, object, object,
object, object, object)
    def __init__(self, Inputs):
        super(OPSIThread, self).__init__()
        self.flag = True
        self.Inputs = Inputs
    def __del__(self):
        print(">>> __del__")
    def Sim2DInit(self,MatTag,OptType):
        wipe()
        model('basic', '-ndm', 2, '-ndf', 2)

```

```

PreOutput =
scio.loadmat("Output.mat")['PreOutput'][0,0]
CtrlPts = PreOutput['CtrlPts']
NdId = 0
for i in CtrlPts.T:
    NdId += 1
    node(NdId,i[0],i[1])
ConPts = PreOutput['ConDofList'].astype(int).tolist()
for i in ConPts:
    fix(i[0],i[1],i[2])
MatProp = PreOutput['MatProp'][0].tolist()
nDMaterial(MatTag, 1, MatProp[0], MatProp[1])
NEIDir = PreOutput['NEIDir'][0].astype(int).tolist()
Order = PreOutput['Order'][0].astype(int).tolist()
EICPs = PreOutput['EI']
KntVectX = PreOutput['KntVectX']
nKntVectX = np.size(KntVectX)
KntVectY = PreOutput['KntVectY']
nKntVectY = np.size(KntVectY)
Weight = PreOutput['Weight']
rho_eHat = PreOutput['rhoeHat']
cnt = 0
for i in range(1,NEIDir[1]+1):
    for j in range(1,NEIDir[0]+1):
        cnt += 1

    element('IGAQuad',j,NEIDir[0],i,NEIDir[1],Order[0],Order[1],EI
CPs[cnt-
1].tolist(),nKntVectX,KntVectX[0].tolist(),nKntVectY,KntVectY[0].tolist(
),float(rho_eHat[cnt-1]),Weight[cnt-1].tolist(),1.0,'PlaneStress',1)
    timeSeries("Linear", 1)
    pattern("Plain", 1, 1)
    LoadPts = PreOutput['LoadDofList']
    for i in LoadPts:
        load(int(i[0]),float(i[1]),float(i[2]))
    system("SuperLU")
    numberer("RCM")
    constraints("Plain")
    integrator("LoadControl", 1.0)
    algorithm("Linear")
    analysis("Static")
    analyze(1)
    NEI = int(PreOutput['NEI'])
    NCPs = int(PreOutput['NCPs'])
    u = np.zeros([NCPs,2],dtype = float)
    Ve = np.zeros([NEI,1],dtype = float)
    NEN = int(PreOutput['NEN'])
    Ke = np.zeros([NEN*2*NEN*2,NEI],dtype = float)
    SimOutput = {}
    for i in range(1,NCPs+1):# read the disp response of
all dofs
        u[i-1,0] = nodeResponse(i,1,1)
        u[i-1,1] = nodeResponse(i,2,1)
    for i in range(1,NEI+1):
        Ve[i-1] = eleResponse(i,'Ve')
        Ke[:,i-1] =
eleResponse(i,'elementalStiffness')
    if OptType == 'minComp' or OptType ==
'minCompSimi':

        scio.savemat('SimOutput.mat',{'u':u,'Ve':Ve,'Ke':Ke})
        elif OptType == 'minVol':# need shp & Stress
            shp = np.zeros([NEN*NEN*2,NEI],dtype
= float)
            stress = np.zeros([3*NEN,NEI],dtype =
float)
            for i in range(1,NEI+1):
                shp[:,i-1] =
eleResponse(i,'elementalBe')
                stress[:,i-1] =
eleResponse(i,'stress')

        scio.savemat('SimOutput.mat',{'u':u,'Ve':Ve,'Ke':Ke,'shp':shp,'s
tress':stress})
        reset()#revert the domain to the original state
        def AdjSim2D(self):
            LoadDofList =
scio.loadmat("AdjInfos.mat")['AdjInfos'][0,0]['LoadDofList']
            PreOutput =
scio.loadmat("Output.mat")['PreOutput'][0,0]
            LoadDofList = PreOutput['LoadDofList']
            NCPs = int(PreOutput['NCPs'])
            remove('loadPattern',1)
            pattern("Plain", 1, 1)
            # define the adjoint vector
            for i in LoadDofList:
                load(int(i[0]),float(i[1]),float(i[2]))
            analyze(1)
            u = np.zeros([NCPs,2],dtype = float)
            for i in range(1,NCPs+1):# read the disp response of
all dofs
                u[i-1,0] = nodeResponse(i,1,1)
                u[i-1,1] = nodeResponse(i,2,1)
            scio.savemat('SimAdjOutput.mat',{'u':u})
            # redefine the load
            remove('loadPattern',1)
            pattern("Plain", 1, 1)
            for i in LoadDofList:
                load(int(i[0]),float(i[1]),float(i[2]))
            reset()
            def replot2D(self):
                VisDofList =
scio.loadmat("Output.mat")['VisDofList'][0,0]
                # plot the topology distribution
                loc_x = VisDofList['loc_x']

```

```

loc_y = VisDats['loc_y']
rhoi_plt = VisDats['rhoi_plt']
loopHist = VisDats['loopHist'][0]
ObjHist = VisDats['f0valHist'][0]
ConsHist = VisDats['fvalHist'][0]
self.return_2Dfig.emit(loc_x, loc_y, rhoi_plt,
loopHist, ObjHist, ConsHist)
def Sim3DInit(self, MatTag, OptType):
wipe()
model('basic', '-ndm', 3, '-ndf', 3)
PreOutput =
scio.loadmat("Output.mat")['PreOutput'][0,0]
CtrlPts = PreOutput['CtrlPts']
NdId = 0
for i in CtrlPts.T:
NdId += 1
node(NdId, i[0], i[1], i[2])
ConPts = PreOutput['ConDofList'].astype(int).tolist()
for i in ConPts:
fix(i[0], i[1], i[2], i[3])
MatProp = PreOutput['MatProp'][0].tolist()
nDMaterial(MatTag, 1, MatProp[0], MatProp[1])
NEIDir = PreOutput['NEIDir'][0].astype(int).tolist()
Order = PreOutput['Order'][0].astype(int).tolist()
EICPs = PreOutput['EI']
KntVectX = PreOutput['KntVectX']
nKntVectX = np.size(KntVectX)
KntVectY = PreOutput['KntVectY']
nKntVectY = np.size(KntVectY)
KntVectZ = PreOutput['KntVectZ']
nKntVectZ = np.size(KntVectZ)
Weight = PreOutput['Weight']
rho_eHat = PreOutput['rhoeHat']
cnt = 0
for k in range(1, NEIDir[2]+1):
for j in range(1, NEIDir[1]+1):
for i in range(1, NEIDir[0]+1):
cnt += 1

element('IGABrick', i, NEIDir[0], j, NEIDir[1], k, NEIDir[2], Order[0],
Order[1], Order[2], EICPs[cnt-1].tolist(),

nKntVectX, KntVectX[0].tolist(), nKntVectY, KntVectY[0].tolist(), n
KntVectZ, KntVectZ[0].tolist(), float(rho_eHat[cnt-1]), Weight[cnt-
1].tolist(), 1)

timeSeries("Linear", 1)
pattern("Plain", 1, 1)
LoadPts = PreOutput['LoadDofList']
for i in LoadPts:

load(int(i[0]), float(i[1]), float(i[2]), float(i[3]))
system("SuperLU")
numberer("RCM")
constraints("Plain")
integrator("LoadControl", 1.0)
algorithm("Linear")
analysis("Static")
analyze(1)
NEI = int(PreOutput['NEI'])
NCPs = int(PreOutput['NCPs'])
u = np.zeros([NCPs, 3], dtype = float)
Ve = np.zeros([NEI, 1], dtype = float)
NEN = int(PreOutput['NEN'])
Ke = np.zeros([NEN*3*NEN*3, NEI], dtype = float)
SimOutput = {}
for i in range(1, NCPs+1):# read the disp response of

all dofs
u[i-1,0] = nodeResponse(i, 1, 1)
u[i-1,1] = nodeResponse(i, 2, 1)
u[i-1,2] = nodeResponse(i, 3, 1)
for i in range(1, NEI+1):
Ve[i-1] = eleResponse(i, 'Ve')
Ke[:,i-1] =
eleResponse(i, 'elementalStiffness')
scio.savemat('SimOutput.mat', {'u':u, 'Ve':Ve, 'Ke':Ke})
reset()#revert the domain to the original state
def replot3D(self):
VisDats =
scio.loadmat("Output.mat")['VisDats'][0,0]
loc_x = VisDats['loc_x']
loc_y = VisDats['loc_y']
loc_z = VisDats['loc_z']
rhoi_plt = VisDats['rhoi_plt']
loopHist = VisDats['loopHist'][0]
ObjHist = VisDats['f0valHist'][0]
ConsHist = VisDats['fvalHist'][0]
self.return_3Dfig.emit(loc_x, loc_y, loc_z, rhoi_plt,
loopHist, ObjHist, ConsHist)
def run(self):
if self.Inputs['Dimension'] == '2D':
MaxIter = int(self.Inputs['MaxIter'])
OptType = self.Inputs['OptType']
MatTag = self.Inputs['MatTag']
if OptType == 'minComp' or OptType ==
'minCompSimi':
time_start = time.time()
loop = eng.Pre2D(self.Inputs)

self.Sim2DInit(MatTag, OptType)
loop =
eng.SenOpt2DMinComp(self.Inputs)
time_end = time.time()
print("Elapsed time: ",
time_end-time_start, "s")

self.replot2D()
while int(loop) < MaxIter and
self.flag:
time_start =
time.time()
self.Sim2DIter(OptType)
loop =
eng.SenOpt2DMinComp(self.Inputs)

self.signal.emit(str(int(loop/MaxIter*100)))
time_end =
time.time()
print("Elapsed
time: ", time_end-time_start, "s")
self.replot2D()
elif OptType == 'minVol':

```

```

        time_start = time.time()
        loop = eng.Pre2D(self.Inputs)

    self.Sim2DInit(MatTag, OptType)

    eng.SenOpt2DMinVolAdj(nargout=0)# build adjoint model
    self.AdjSim2D()# calculate
the adjoint vector for dSdx
    loop =
eng.SenOpt2DMinVol(self.Inputs)
        time_end = time.time()
        print("Elapsed time: ",
time_end-time_start, "s")
    self.flag:
        time_start =
time.time()

    self.Sim2DIter(OptType)

    eng.SenOpt2DMinVolAdj(nargout=0)# build adjoint model

    self.AdjSim2D()#calculate the adjoint vector for dSdx
        loop =
eng.SenOpt2DMinVol(self.Inputs)

    self.signal.emit(str(int(loop/MaxIter*100)))
        time_end =
time.time()
        print("Elapsed
time: ", time_end-time_start, "s")
        self.replot2D()
        print('>>> optimization end: ')
        self.finished.emit()
        elif self.Inputs['Dimension'] == '3D':
            if self.Inputs['OptType'] == 'minVol':
                print("Error: current 3D
minVol model is unsupported!")
            return
            time_start = time.time()
            loop = eng.Pre3D(self.Inputs)
            MaxIter = int(self.Inputs['MaxIter'])
            OptType = self.Inputs['OptType']
            MatTag = self.Inputs['MatTag']
            self.Sim3DInit(MatTag, OptType)
            loop =
eng.SenOpt3DMinComp(self.Inputs)
            self.replot3D()
            time_end = time.time()
            print("Elapsed time: ", time_end-
time_start, "s")
            while int(loop) < MaxIter and self.flag:
                time_start = time.time()
                self.Sim3DIter(OptType)
                loop =
eng.SenOpt3DMinComp(self.Inputs)

            self.signal.emit(str(int(loop/MaxIter*100)))
            self.replot3D()
            time_end = time.time()
            print("Elapsed time: ",
time_end-time_start, "s")
            print('>>> optimization end: ')
            self.finished.emit()
class MyOPSITO_GUI(QtWidgets.QWidget, Ui_Form):
    initFlag = True
    def __init__(self, parent=None):
        super(MyOPSITO_GUI, self).__init__(parent)
        self.setupUi(self)
        self.StartButton.clicked.connect(self.runMainOpt)

        self.InterruptButton.clicked.connect(self.interruptMainOpt)

        self.ObjPlot = MyMplCanvasLine(self.ObjWidget)
        self.ConsPlot = MyMplCanvasLine(self.ConsWidget)
    def runMainOpt(self):
        self.Inputs = {}
        if self.buttonGroup.checkedId() == -2: # Default geo
model input is activated
            self.Inputs['Sizes'] = self.SizesEdit.text()
            self.Inputs['Orders'] =
self.OrdersEdit.text()
            self.Inputs['NEls'] = self.NElsEdit.text()
            self.Inputs['GeoInputType'] = 'Default'
            elif self.buttonGroup.checkedId() == -3: # User
input geo model is activated
                self.Inputs['GeoModelPath'] =
self.PathOfGeoModelEdit.text()
                self.Inputs['GeoInputType'] = 'User
input'
            self.Inputs['MatTag'] = self.MatTagBox.currentText()
            self.Inputs['MatProp'] = self.MatPropEdit.text()
            self.Inputs['BCs'] = self.BCsEdit.toPlainText()
            self.Inputs['Loads'] = self.LoadsEdit.toPlainText()
            self.Inputs['ConstVal'] = self.ConsValEdit.text()
            self.Inputs['MaxIter'] = self.MaxIterEdit.text()
            self.Inputs['MMAPars'] = self.MMAParsEdit.text()
            self.Inputs['Penalty'] = self.PenaltyEdit.text()
            self.Inputs['StpPenalty'] =
self.StpPntCheckBox.isChecked()
            self.Inputs['Projection'] = self.PrjEdit.text()
            self.Inputs['StpPrj'] =
self.StpPrjCheckBox.isChecked()
            self.Inputs['PrjThreshold'] = self.PrjThrEdit.text()
            self.Inputs['rmin'] = self.rminEdit.text()
            self.Inputs['OptType'] =
self.OptTypeBox.currentText()
            self.Inputs['Dimension'] =
self.DimensionBox.currentText()
            if self.initFlag == False:
                item = self.vertical_layout.itemAt(0)
                self.vertical_layout.removeItem(item)
                item.widget().deleteLater()
            if self.Inputs['Dimension'] == '2D':
                if self.initFlag == True:
                    self.vertical_layout =
QtWidgets.QVBoxLayout()
                    self.TopoWidget.setLayout(self.vertical_layout)
                    self.TopoPlot =
MyMplCanvasTopo2D(self.TopoWidget)
                    self.vertical_layout.addWidget(self.TopoPlot)
                    elif self.Inputs['Dimension'] == '3D':
                        if self.initFlag == True:
                            self.vertical_layout =
QtWidgets.QVBoxLayout()
                            self.TopoWidget.setLayout(self.vertical_layout)
                            self.TopoPlot =
MyVtkTopo3D(self.TopoWidget)
                            self.vertical_layout.addWidget(self.TopoPlot.vtk_widget)
                            self.OPSITOThread = OPSITOThread(self.Inputs)
                            self.OPSITOThread.signal.connect(self.call_backlog)
                            self.OPSITOThread.finished.connect(self.finishedMainOpt)
                            self.OPSITOThread.return_2Dfig.connect(self.plot2D)
                            self.OPSITOThread.return_3Dfig.connect(self.plot3D)
                            self.OPSITOThread.flag = True
                            self.StartButton.setEnabled(False)

```

```

        self.OPSITOThread.start()
def interruptMainOpt(self):
    if not self.OPSITOThread.isRunning():
        return
    self.StartButton.setEnabled(True)
    self.OPSITOThread.flag = False
    self.initFlag = False
    print(">>> stop thread...")
    self.OPSITOThread.quit()
    self.OPSITOThread.wait()
    print(">>> stop thread end...")
def finishedMainOpt(self):
    self.StartButton.setEnabled(True)
    self.initFlag = False
    self.OPSITOThread.flag = False
    self.OPSITOThread.quit()
    self.OPSITOThread.wait()
def call_backlog(self, msg):
    self.progressBar.setValue(int(msg))# update the
parameter to progress bar
def plot2D(self, loc_x, loc_y, rhoi_plt, loopHist, ObjHist,
ConsHist):
    self.TopoPlot.updateTopo2D(loc_x, loc_y, rhoi_plt)
    self.ObjPlot.line.set_data([loopHist, ObjHist])
    self.ObjPlot.axes.set_xlim([loopHist.min()-
0.1*abs(loopHist.max()),loopHist.max()+0.1*abs(loopHist.max())])
    self.ObjPlot.axes.set_ylim([ObjHist.min()-
0.1*abs(ObjHist.max()), ObjHist.max()+0.1*abs(ObjHist.max())])
    self.ObjPlot.draw()
    self.ConsPlot.line.set_data([loopHist, ConsHist])
    self.ConsPlot.axes.set_xlim([loopHist.min()-
0.1*abs(loopHist.max()),loopHist.max()+0.1*abs(loopHist.max())])
    self.ConsPlot.axes.set_ylim([ConsHist.min()-
0.1*abs(ConsHist.max()), ConsHist.max()+0.1*abs(ConsHist.max())])
    self.ConsPlot.draw()

        self.ConsPlot.axes.set_xlim([loopHist.min()-
0.1*abs(loopHist.max()),loopHist.max()+0.1*abs(loopHist.max())])
        self.ConsPlot.axes.set_ylim([ConsHist.min()-
0.1*abs(ConsHist.max()), ConsHist.max()+0.1*abs(ConsHist.max())])
        self.ConsPlot.draw()
        def plot3D(self, loc_x, loc_y, loc_z, rhoi_plt, loopHist, ObjHist,
ConsHist):
            #print('come to plot3D, is the gui frozen?')
            self.TopoPlot.updateTopo3D(loopHist[-1],loc_x,
loc_y, loc_z, rhoi_plt)
            self.ObjPlot.line.set_data([loopHist, ObjHist])
            self.ObjPlot.axes.set_xlim([loopHist.min()-
0.1*abs(loopHist.max()),loopHist.max()+0.1*abs(loopHist.max())])
            self.ObjPlot.axes.set_ylim([ObjHist.min()-
0.1*abs(ObjHist.max()), ObjHist.max()+0.1*abs(ObjHist.max())])
            self.ObjPlot.draw()
            self.ConsPlot.line.set_data([loopHist, ConsHist])
            self.ConsPlot.axes.set_xlim([loopHist.min()-
0.1*abs(loopHist.max()),loopHist.max()+0.1*abs(loopHist.max())])
            self.ConsPlot.axes.set_ylim([ConsHist.min()-
0.1*abs(ConsHist.max()), ConsHist.max()+0.1*abs(ConsHist.max())])
            self.ConsPlot.draw()

if __name__ == '__main__':
    # start matlab engine
    eng = matlab.engine.start_matlab()
    # call the GUI
    app = QtWidgets.QApplication(sys.argv)
    myWin = MyOPSITO_GUI()
    myWin.show()
    app.exec_()

```

C2 Python script of the Graphical User Interface of OPS-ITO

```

from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(445, 609)
        self.StartButton = QtWidgets.QPushButton(Form)
        self.StartButton.setGeometry(QtCore.QRect(60, 560, 81, 31))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.StartButton.setFont(font)
        self.StartButton.setObjectName("StartButton")
        self.InterruptButton = QtWidgets.QPushButton(Form)
        self.InterruptButton.setGeometry(QtCore.QRect(290, 560, 91,
31))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.InterruptButton.setFont(font)
        self.InterruptButton.setObjectName("InterruptButton")
        self.GeoGroupBox = QtWidgets.QGroupBox(Form)
        self.GeoGroupBox.setGeometry(QtCore.QRect(20, 70, 201, 131))
        self.GeoGroupBox.setObjectName("GeoGroupBox")
        self.DefaultGeoButton =
QtWidgets.QRadioButton(self.GeoGroupBox)
        self.DefaultGeoButton.setGeometry(QtCore.QRect(120, 30, 61,
17))
        self.DefaultGeoButton.setChecked(True)
        self.DefaultGeoButton.setObjectName("DefaultGeoButton")
        self.buttonGroup = QtWidgets.QButtonGroup(Form)
        self.buttonGroup.setObjectName("buttonGroup")
        self.buttonGroup.addButton(self.DefaultGeoButton)
        self.SizesLabel = QtWidgets.QLabel(self.GeoGroupBox)
        self.SizesLabel.setGeometry(QtCore.QRect(10, 20, 31, 21))
        self.SizesLabel.setObjectName("SizesLabel")
        self.OrdersLabel = QtWidgets.QLabel(self.GeoGroupBox)
        self.OrdersLabel.setGeometry(QtCore.QRect(10, 40, 41, 21))
        self.OrdersLabel.setObjectName("OrdersLabel")
        self.GeoPathLabel = QtWidgets.QLabel(self.GeoGroupBox)
        self.GeoPathLabel.setGeometry(QtCore.QRect(10, 80, 101, 21))
        self.GeoPathLabel.setObjectName("GeoPathLabel")
        self.PathOfGeoModelEdit =
QtWidgets.QLineEdit(self.GeoGroupBox)
        self.PathOfGeoModelEdit.setGeometry(QtCore.QRect(10, 100,
181, 20))
        self.PathOfGeoModelEdit.setObjectName("PathOfGeoModelEdit")
        self.SizesEdit = QtWidgets.QLineEdit(self.GeoGroupBox)
        self.SizesEdit.setGeometry(QtCore.QRect(50, 20, 61, 20))
        self.SizesEdit.setObjectName("SizesEdit")
        self.OrdersEdit = QtWidgets.QLineEdit(self.GeoGroupBox)
        self.OrdersEdit.setGeometry(QtCore.QRect(50, 40, 61, 20))
        self.OrdersEdit.setObjectName("OrdersEdit")
        self.NEIsLabel = QtWidgets.QLabel(self.GeoGroupBox)
        self.NEIsLabel.setGeometry(QtCore.QRect(10, 60, 41, 21))
        self.NEIsLabel.setObjectName("NEIsLabel")
        self.NEIsEdit = QtWidgets.QLineEdit(self.GeoGroupBox)
        self.NEIsEdit.setGeometry(QtCore.QRect(50, 60, 61, 20))
        self.NEIsEdit.setObjectName("NEIsEdit")
        self.UserInputGeoButton =
QtWidgets.QRadioButton(self.GeoGroupBox)
        self.UserInputGeoButton.setGeometry(QtCore.QRect(120, 50, 71,
17))
        self.UserInputGeoButton.setObjectName("UserInputGeoButton")

```

```

self.buttonGroup.addButton(self.UserInputGeoButton)
self.label = QtWidgets.QLabel(Form)
self.label.setGeometry(QQtCore.QRect(180, 10, 101, 21))
font = QtGui.QFont()
font.setPointSize(10)
font.setBold(True)
font.setWeight(75)
self.label.setFont(font)
self.label.setObjectName("label")
self.MatBCsGroupBox = QtWidgets.QGroupBox(Form)
self.MatBCsGroupBox.setGeometry(QQtCore.QRect(20, 200, 201,
171))
self.MatBCsGroupBox.setObjectName("MatBCsGroupBox")
self.MatTagLabel = QtWidgets.QLabel(self.MatBCsGroupBox)
self.MatTagLabel.setGeometry(QQtCore.QRect(10, 20, 41, 21))
self.MatTagLabel.setObjectName("MatTagLabel")
self.MatPropLabel = QtWidgets.QLabel(self.MatBCsGroupBox)
self.MatPropLabel.setGeometry(QQtCore.QRect(10, 50, 41, 21))
self.MatPropLabel.setObjectName("MatPropLabel")
self.MatPropEdit = QtWidgets.QLineEdit(self.MatBCsGroupBox)
self.MatPropEdit.setGeometry(QQtCore.QRect(60, 50, 61, 20))
self.MatPropEdit.setObjectName("MatPropEdit")
self.BCsLabel = QtWidgets.QLabel(self.MatBCsGroupBox)
self.BCsLabel.setGeometry(QQtCore.QRect(10, 90, 41, 21))
self.BCsLabel.setObjectName("BCsLabel")
self.BCsEdit = QtWidgets.QPlainTextEdit(self.MatBCsGroupBox)
self.BCsEdit.setGeometry(QQtCore.QRect(60, 80, 131, 41))
self.BCsEdit.setObjectName("BCsEdit")
self.LoadsLabel = QtWidgets.QLabel(self.MatBCsGroupBox)
self.LoadsLabel.setGeometry(QQtCore.QRect(10, 130, 41, 21))
self.LoadsLabel.setObjectName("LoadsLabel")
self.LoadsEdit = QtWidgets.QPlainTextEdit(self.MatBCsGroupBox)
self.LoadsEdit.setGeometry(QQtCore.QRect(60, 120, 131, 41))
self.LoadsEdit.setObjectName("LoadsEdit")
self.MatTagBox = QtWidgets.QComboBox(self.MatBCsGroupBox)
self.MatTagBox.setGeometry(QQtCore.QRect(60, 20, 101, 22))
self.MatTagBox.setObjectName("MatTagBox")
self.MatTagBox.addItem("")
self.OptParsGroupBox = QtWidgets.QGroupBox(Form)
self.OptParsGroupBox.setGeometry(QQtCore.QRect(20, 380, 201,
171))
self.OptParsGroupBox.setObjectName("OptParsGroupBox")
self.ConsValLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.ConsValLabel.setGeometry(QQtCore.QRect(10, 20, 41, 21))
self.ConsValLabel.setObjectName("ConsValLabel")
self.MMAParsLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.MMAParsLabel.setGeometry(QQtCore.QRect(10, 50, 51, 21))
self.MMAParsLabel.setObjectName("MMAParsLabel")
self.MMAParsEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.MMAParsEdit.setGeometry(QQtCore.QRect(60, 50, 61, 20))
self.MMAParsEdit.setObjectName("MMAParsEdit")
self.PenaltyLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.PenaltyLabel.setGeometry(QQtCore.QRect(10, 80, 41, 21))
self.PenaltyLabel.setObjectName("PenaltyLabel")
self.ConsValEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.ConsValEdit.setGeometry(QQtCore.QRect(60, 20, 31, 20))
self.ConsValEdit.setObjectName("ConsValEdit")
self.MaxIterEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.MaxIterEdit.setGeometry(QQtCore.QRect(150, 20, 31, 20))
self.MaxIterEdit.setObjectName("MaxIterEdit")
self.MaxIterLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.MaxIterLabel.setGeometry(QQtCore.QRect(100, 20, 41, 21))
self.MaxIterLabel.setObjectName("MaxIterLabel")
self.PenaltyEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.PenaltyEdit.setGeometry(QQtCore.QRect(70, 80, 31, 20))
self.PenaltyEdit.setObjectName("PenaltyEdit")
self.StpPntCheckBox =
QtWidgets.QCheckBox(self.OptParsGroupBox)
self.StpPntCheckBox.setGeometry(QQtCore.QRect(120, 80, 70, 17))
self.StpPntCheckBox.setChecked(True)
self.StpPntCheckBox.setObjectName("StpPntCheckBox")
self.ProjectionLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.ProjectionLabel.setGeometry(QQtCore.QRect(10, 110, 51, 21))

self.ProjectionLabel.setObjectName("ProjectionLabel")
self.PrjEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.PrjEdit.setGeometry(QQtCore.QRect(70, 110, 31, 20))
self.PrjEdit.setObjectName("PrjEdit")
self.StpPrjCheckBox =
QtWidgets.QCheckBox(self.OptParsGroupBox)
self.StpPrjCheckBox.setGeometry(QQtCore.QRect(120, 110, 70,
17))
self.StpPrjCheckBox.setChecked(True)
self.StpPrjCheckBox.setObjectName("StpPrjCheckBox")
self.ProjThresholdLabel =
QtWidgets.QLabel(self.OptParsGroupBox)
self.ProjThresholdLabel.setGeometry(QQtCore.QRect(10, 140, 71,
21))
self.ProjThresholdLabel.setObjectName("ProjThresholdLabel")
self.PrjThrEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.PrjThrEdit.setGeometry(QQtCore.QRect(80, 140, 31, 20))
self.PrjThrEdit.setObjectName("PrjThrEdit")
self.rminLabel = QtWidgets.QLabel(self.OptParsGroupBox)
self.rminLabel.setGeometry(QQtCore.QRect(120, 140, 21, 21))
self.rminLabel.setObjectName("rminLabel")
self.rminEdit = QtWidgets.QLineEdit(self.OptParsGroupBox)
self.rminEdit.setGeometry(QQtCore.QRect(150, 140, 31, 20))
self.rminEdit.setObjectName("rminEdit")
self.DimLabel = QtWidgets.QLabel(Form)
self.DimLabel.setGeometry(QQtCore.QRect(30, 20, 51, 21))
self.DimLabel.setObjectName("DimLabel")
self.OptTypeLabel = QtWidgets.QLabel(Form)
self.OptTypeLabel.setGeometry(QQtCore.QRect(30, 40, 51, 21))
self.OptTypeLabel.setObjectName("OptTypeLabel")
self.TopologyLabel = QtWidgets.QLabel(Form)
self.TopologyLabel.setGeometry(QQtCore.QRect(320, 70, 51, 21))
self.TopologyLabel.setObjectName("TopologyLabel")
self.ConsHistLabel = QtWidgets.QLabel(Form)
self.ConsHistLabel.setGeometry(QQtCore.QRect(300, 220, 91, 21))
self.ConsHistLabel.setObjectName("ConsHistLabel")
self.ObjHistLabel = QtWidgets.QLabel(Form)
self.ObjHistLabel.setGeometry(QQtCore.QRect(300, 390, 91, 21))
self.ObjHistLabel.setObjectName("ObjHistLabel")
self.progressBar = QtWidgets.QProgressBar(Form)
self.progressBar.setGeometry(QQtCore.QRect(300, 40, 118, 23))
self.progressBar.setProperty("value", 0)
self.progressBar.setObjectName("progressBar")
self.DimensionBox = QtWidgets.QComboBox(Form)
self.DimensionBox.setGeometry(QQtCore.QRect(80, 20, 69, 22))
self.DimensionBox.setObjectName("DimensionBox")
self.DimensionBox.addItem("")
self.DimensionBox.addItem("")
self.OptTypeBox = QtWidgets.QComboBox(Form)
self.OptTypeBox.setGeometry(QQtCore.QRect(80, 40, 69, 22))
self.OptTypeBox.setObjectName("OptTypeBox")
self.OptTypeBox.addItem("")
self.OptTypeBox.addItem("")
self.OptTypeBox.addItem("")
self.TopoWidget = QtWidgets.QWidget(Form)
self.TopoWidget.setGeometry(QQtCore.QRect(240, 100, 191, 121))
self.TopoWidget.setObjectName("TopoWidget")
self.ConsWidget = QtWidgets.QWidget(Form)
self.ConsWidget.setGeometry(QQtCore.QRect(240, 250, 191, 121))
self.ConsWidget.setObjectName("ConsWidget")
self.ObjWidget = QtWidgets.QWidget(Form)
self.ObjWidget.setGeometry(QQtCore.QRect(240, 420, 191, 121))
self.ObjWidget.setObjectName("ObjWidget")

self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)
def retranslateUi(self, Form):
    _translate = QtCore.QCoreApplication.translate
    Form.setWindowTitle(_translate("Form", "Form"))
    self.StartButton.setText(_translate("Form", "Start"))
    self.InterruptButton.setText(_translate("Form", "Interrupt"))
    self.GeomGroupBox.setText(_translate("Form", "geometric
model"))

```

```

self.DefaultGeoButton.setText(_translate("Form", "Default"))
self.SizesLabel.setText(_translate("Form", "Sizes"))
self.OrdersLabel.setText(_translate("Form", "Orders"))
self.GeoPathLabel.setText(_translate("Form", "Path of Geo
model:"))
self.SizesEdit.setText(_translate("Form", "5 3"))
self.OrdersEdit.setText(_translate("Form", "1 1"))
self.NElsLabel.setText(_translate("Form", "NEls"))
self.NElsEdit.setText(_translate("Form", "100 60"))
self.UserInputGeoButton.setText(_translate("Form", "User
input"))
self.label.setText(_translate("Form", "GUI of OPS-ITO"))
self.MatBCsGroupBox.setTitle(_translate("Form", "Material and
BCs"))
self.MatTagLabel.setText(_translate("Form", "MatTag"))
self.MatPropLabel.setText(_translate("Form", "MatProp"))
self.MatPropEdit.setText(_translate("Form", "2e11 0.3"))
self.BCsLabel.setText(_translate("Form", "BCs"))
self.BCsEdit.setPlainText(_translate("Form", "1 UX @(x,y)1\n"
"1 UY @(x,y)1"))
self.LoadsLabel.setText(_translate("Form", "Loads"))
self.LoadsEdit.setPlainText(_translate("Form", "2 FY @(x,y)-
3e6*(abs(y)<=0.2)"))
self.MatTagBox.setItemText(0, _translate("Form",
"Elasticisotropic"))
self.OptParsGroupBox.setTitle(_translate("Form",
"OptParameters"))
self.ConsValLabel.setText(_translate("Form", "ConsVal"))

```

```

self.MMAParsLabel.setText(_translate("Form", "MMAPars"))
self.MMAParsEdit.setText(_translate("Form", "2 1 1 2"))
self.PenaltyLabel.setText(_translate("Form", "Penalty"))
self.ConsValEdit.setText(_translate("Form", "0.5"))
self.MaxIterEdit.setText(_translate("Form", "200"))
self.MaxIterLabel.setText(_translate("Form", "MaxIter"))
self.PenaltyEdit.setText(_translate("Form", "3"))
self.StpPntCheckBox.setText(_translate("Form", "StpPenalty"))
self.ProjectionLabel.setText(_translate("Form", "Projection"))
self.PrjEdit.setText(_translate("Form", "5"))
self.StpPrjCheckBox.setText(_translate("Form", "StpPrj"))
self.ProjThresholdLabel.setText(_translate("Form",
"ProjThreshold"))
self.PrjThrEdit.setText(_translate("Form", "0.5"))
self.rminLabel.setText(_translate("Form", "rmin"))
self.rminEdit.setText(_translate("Form", "0.15"))
self.DimLabel.setText(_translate("Form", "Dimension"))
self.OptTypeLabel.setText(_translate("Form", "OptType"))
self.TopologyLabel.setText(_translate("Form", "Topology"))
self.ConsHistLabel.setText(_translate("Form", "Constraint
history"))
self.ObjHistLabel.setText(_translate("Form", "Objective history"))
self.DimensionBox.setItemText(0, _translate("Form", "2D"))
self.DimensionBox.setItemText(1, _translate("Form", "3D"))
self.OptTypeBox.setItemText(0, _translate("Form", "minComp"))
self.OptTypeBox.setItemText(1, _translate("Form", "minVol"))
self.OptTypeBox.setItemText(2, _translate("Form",
"minCompSimi"))

```

C3 The OPS-ITO tool written in MATLAB

C3.1 2D pre-process

```

function loop = Pre2D(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
Inputs = struct(Inputs);
mmaMoveLimits = str2double(strsplit(Inputs.MMAPars, ' '));
BCs = strsplit(Inputs.BCs, '\n');
numBCs = size(BCs,2);
for i = 1:numBCs
    BC_temp = strsplit(BCs{i}, ' ');
    Disp{i,1} = str2double(BC_temp[181]);
    Disp{i,2} = BC_temp[181];
    Disp{i,3} = str2func(BC_temp[3]);
end
Forces = strsplit(Inputs.Loads, '\n');
if isempty(Forces{end})% the end line is empty!
    numLoads = size(Forces,2)-1;
else
    numLoads = size(Forces,2);
end
for i = 1:numLoads
    force_temp = strsplit(Forces{i}, ' ');
    force{i,1} = str2double(force_temp[181]);
    force{i,2} = force_temp[181];
    force{i,3} = str2func(force_temp[3]);
end
MatProp = str2double(strsplit(Inputs.MatProp, ' '));
%% generate the mesh model
if strcmpi(Inputs.GeoInputType, 'Default')
    Sizes = str2double(strsplit(Inputs.Sizes, ' '));
    Orders = str2double(strsplit(Inputs.Orders, ' '));
    NEls = str2double(strsplit(Inputs.NEls, ' '));
    L = Sizes(1);D = Sizes(2);
    CtrlPts = zeros(4, 2, 2);
    CtrlPts(1 : 3, 1, 1) = [0; -D/2; 0];

```

```

    CtrlPts(1 : 3, 2, 1) = [L; -D/2; 0];
    CtrlPts(1 : 3, 1, 2) = [0; D/2; 0];
    CtrlPts(1 : 3, 2, 2) = [L; D/2; 0];
    CtrlPts(4, :, :) = 1;
    KntVect[181] = [0 0 1 1];
    KntVect[181] = [0 0 1 1];
    Surf = CreateNURBS(KntVect, CtrlPts);
    p=Orders(1);q=Orders(2);
    kx=1;ky=1;
    nelx=NEls(1); nely=NEls(2);
    Surf = KRefine(Surf, [nelx, nely], [p, q], [p-kx, q-ky]);
    Mesh = Mesh2D(Surf, 'VectorField');
elseif strcmpi(Inputs.GeoInputType, 'User input')
    GeoModelPath = Inputs.GeoModelPath;
    ModelDatas = load(GeoModelPath);
    Surf = ModelDatas.Surf;
    Mesh = Mesh2D(Surf, 'VectorField');
    nelx = Mesh.NEIDir(1); nely = Mesh.NEIDir(2);
end
if Inputs.StpPenalty == 1
    penalty = 1;
elseif Inputs.StpPenalty == 0
    penalty = str2double(Inputs.Penalty);
end
if Inputs.StpPrj == 1
    beta_HS = 0.1;
elseif Inputs.StpPrj == 0
    beta_HS = str2double(Inputs.Projection);
end
check_x = linspace(0,1,2*nelx+1);
check_y = linspace(0,1,2*nely+1);
node_x = check_x(2:2:end);
node_y = check_y(2:2:end);

```

```

R_Center = FindInterpFunc(Surf,{node_x, node_y});
R_Center = sparse(R_Center);
xloc_CtrlPts = reshape(Surf.CtrlPts3D(1, :, :), 1, []);
yloc_CtrlPts = reshape(Surf.CtrlPts3D(2, :, :), 1, []);
rmin = str2double(Inputs.rmin); HS_th =
str2double(Inputs.PrjThreshold);
%% modify the rho_e
if strcmpi(Inputs.OptType, 'minComp') ||
strcmpi(Inputs.OptType, 'minVol')
    loc_CPs = [xloc_CtrlPts', yloc_CtrlPts'];
    dist = pdist2(loc_CPs, loc_CPs);
    weight = rmin - dist;
    weight(weight < 0) = 0;
    weight = weight ./ (sum(weight, 2));
    weight = sparse(weight);
    rho_CPs = 0.5 * ones(Surf.NCtrlPts);
    rhoi_hat = weight * rho_CPs(:);
    rhoe_hat = reshape(R_Center * rhoi_hat, Mesh.NEIDir);
    rhoe_star = (tanh(beta_HS * HS_th) + tanh(beta_HS * (rhoe_hat -
HS_th))) / ...
    (tanh(beta_HS * HS_th) + tanh(beta_HS * (1 - HS_th)));
    prhoeStar_prhoeHat = diag(beta_HS * (1 -
tanh(beta_HS * (rhoe_hat - HS_th)).^2) / ...
    (tanh(beta_HS * HS_th) + tanh(beta_HS * (1 - HS_th))));
    rho_e = reshape(rhoe_star, nelx, nely);
    prhoe_prhoi = sparse(prhoeStar_prhoeHat * R_Center * weight);
    dist_plt = pdist2(loc_CPs, loc_CPs); weight_plt = rmin -
dist_plt; weight_plt(weight_plt < 0) = 0;
    weight_plt = weight_plt ./ (sum(weight_plt, 2)); weight_plt =
sparse(weight_plt);
    rhoi_plt_hat = reshape(weight_plt * rho_CPs(:), Surf.NCtrlPts);
    rhoi_plt = (tanh(beta_HS * HS_th) + tanh(beta_HS * (rhoi_plt_hat -
HS_th))) / ...
    (tanh(beta_HS * HS_th) + tanh(beta_HS * (1 - HS_th)));
    %% export the MMA parameters
    tnn = size(rho_CPs(:), 1);
    xy00 = rho_CPs(:);
else
    error('unsupported optimization type, please check')
end
numVar = length(xy00);
xval = xy00;
xold1 = xy00;
xold2 = xy00;
xmin = 0 * ones(tnn, 1);
xmax = ones(tnn, 1);
low = xmin;
upp = xmax;
numDV = length(low);
m = 1;
Var_num = tnn;
c = 1000 * ones(m, 1);
dd = ones(m, 1);
a0 = 1;
a = zeros(m, 1);
%% define BCs and Loads
f = sparse(zeros(Mesh.NDof, 1));
ConDispVals = []; LoadVals = [];
ConDispDofs = []; LoadDofs = [];
for i = 1:numBCs
    [UTemp, DofsTemp] = projDrchtBdryVals(Surf, Mesh, Disp{i, 3},
Disp{i, 1}, Disp{i, 2});
    DofsTemp(abs(UTemp) <= 0.2) = [];
    UTemp(abs(UTemp) <= 0.2) = [];
    ConDispVals = [ConDispVals; UTemp];
    ConDispDofs = [ConDispDofs; DofsTemp];
end
for i = 1:numLoads
    [LoadTemp, DofsTemp] = applyNewmannBdryVals(Surf, Mesh,
force{i, 3}, force{i, 1}, force{i, 2});
    LoadVals = [LoadVals; LoadTemp];
    LoadDofs = [LoadDofs; DofsTemp];
end
f(LoadDofs) = f(LoadDofs) + LoadVals;
ff = sparse(zeros(size(f)));
ff(1:2:end) = f(1:Surf.NNP);
ff(2:2:end) = f(Surf.NNP + 1:end);
%% generate the output datas
CtrlPts = reshape(Surf.CtrlPts3D, 3, []);
% export constrained CPs list
ConNdlDs = rem(ConDispDofs, Surf.NNP);
ConNdlDs(ConNdlDs == 0) = Surf.NNP;
ConDoflds = ceil(ConDispDofs / Surf.NNP);
ConDofList = zeros(Surf.NNP, 2);
for i = 1:length(ConNdlDs)
    ConDofList(ConNdlDs(i), ConDoflds(i)) = 1;
end
ConDofList = [(1:Surf.NNP)', ConDofList];
ConDofList(ConDofList(:, 2) + ConDofList(:, 3) == 0, :) = [];
% export loaded CPs list
LoadNdlDs = rem(LoadDofs, Surf.NNP);
LoadNdlDs(LoadNdlDs == 0) = Surf.NNP;
LoadDoflds = ceil(LoadDofs / Surf.NNP);
LoadDofList = zeros(Surf.NNP, 2);
for i = 1:length(LoadNdlDs)
    LoadDofList(LoadNdlDs(i), LoadDoflds(i)) = LoadVals(i);
end
LoadDofList = [(1:Surf.NNP)', LoadDofList];
LoadDofList(abs(LoadDofList(:, 2)) + abs(LoadDofList(:, 3)) == 0, :) = [];
EI = Mesh.EI;
rho_min = 1e-9;
rho_eHat = (1 - rho_min) * rho_e.^penalty + rho_min;
Weights = reshape(Surf.Weights, 1, []);
Const_value = str2double(strsplit(Inputs.ConstVal, ' '));
loop = 0;
loopHist = []; fvalHist = []; f0valHist = [];
NEN = Mesh.NEN;
PreOutput = {}; MMAPars = {}; VisDats = {}; IterPars = {};
PreOutput.CtrlPts = CtrlPts; PreOutput.ConDofList =
ConDofList; PreOutput.LoadDofList = LoadDofList;
PreOutput.MatProp = MatProp; PreOutput.D0 =
getElastMat(MatProp(1), MatProp(2),
'PlaneStress'); PreOutput.MatTag = Inputs.MatTag;
PreOutput.NEN = NEN; PreOutput.EI = EI; PreOutput.NEI = Mesh.NEI;
PreOutput.NCPS = prod(Surf.NCtrlPts);
PreOutput.NEIDir = Mesh.NEIDir; PreOutput.Order =
Surf.Order; PreOutput.EI = Mesh.EI;
PreOutput.KntVectX = Surf.KntVect[181]; PreOutput.KntVectY =
Surf.KntVect[181]; PreOutput.Weight = Weights(Mesh.EI);
PreOutput.rhoeHat = rho_eHat(:); PreOutput.OptType =
Inputs.OptType;
MMAPars.mmaMoveLimits = mmaMoveLimits; MMAPars.tnn = tnn;
MMAPars.xy00 = xy00; MMAPars.numVar = numVar; MMAPars.xval =
xval;
MMAPars.xold1 = xold1; MMAPars.xold2 = xold2; MMAPars.xmin =
xmin;
MMAPars.xmax = xmax; MMAPars.low = low; MMAPars.upp = upp;
MMAPars.numDV = numDV; MMAPars.m = m; MMAPars.Var_num =
Var_num;
MMAPars.c = c; MMAPars.dd = dd; MMAPars.a0 = a0; MMAPars.a = a;
VisDats.loc_x = reshape(xloc_CtrlPts, Surf.NCtrlPts); VisDats.loc_y =
reshape(yloc_CtrlPts, Surf.NCtrlPts);
VisDats.loop = loop; VisDats.loopHist = loopHist; VisDats.fvalHist =
fvalHist;
VisDats.f0valHist = f0valHist; VisDats.rhoi_plt =
rhoi_plt; VisDats.weight_plt = weight_plt;
IterPars.Const_value = Const_value; IterPars.rho_min =
rho_min; IterPars.rho_e = rho_e;
IterPars.weight = weight; IterPars.R_Center =
R_Center; IterPars.prhoe_prhoi = prhoe_prhoi;
IterPars.ff = ff; IterPars.penalty = penalty; IterPars.beta_HS = beta_HS;
IterPars.HS_th = HS_th; IterPars.Surf = Surf; IterPars.Mesh = Mesh;
IterPars.ACS_c_old = 0; IterPars.ACS_c = 0; IterPars.ACS_a = 0;
if strcmpi(Inputs.OptType, 'minVol')
    IterPars.ss = 0.8;
end

```

```
save('Output.mat','PreOutput','MMAPars','VisDats','IterPars');
```

```
end
```

C3.2 2D min_compliance model

```
function loop = SenOpt2DMinComp(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
SimOutput = load('SimOutput.mat');
PreOutput = load('Output.mat').PreOutput;
MMAPars = load('Output.mat').MMAPars;
VisDats = load('Output.mat').VisDats;
IterPars = load('Output.mat').IterPars;
Surf = IterPars.Surf;
Mesh = IterPars.Mesh;
Ke = reshape(SimOutput.Ke,Mesh.NEN*2,Mesh.NEN*2,[]);
rho_min = IterPars.rho_min;
weight = IterPars.weight;
R_Center = IterPars.R_Center;
beta_HS = IterPars.beta_HS;
HS_th = IterPars.HS_th;
prhoe_prhoi = IterPars.prhoe_prhoi;
VisDats.loop = VisDats.loop+1;
loop = VisDats.loop;
if VisDats.loop == 1
    IterPars.InitVe = sum(SimOutput.Ve);
end
d = SimOutput.u;
dx = d(:,1);dy = d(:,2);
d = reshape(d',1,[]);
ue = zeros(Mesh.NEI,2*Mesh.NEN);
ue(:,1:2:end) = dx(Mesh.EI);
ue(:,2:2:end) = dy(Mesh.EI);
[f0val,df0dx] =
calc_dCdx(IterPars.ff,d,IterPars.rho_e(:),ue,Ke,prhoe_prhoi,IterPars.p
enalty,IterPars.rho_min);
[fval,dfdx] =
calc_dVdx(IterPars.rho_e(:),prhoe_prhoi,SimOutput.Ve,IterPars.Const
_value);
if loop == 1
    IterPars.scl = f0val/10;
end
dgt0 = 5;
dgt = dgt0 - floor(log10([max(abs(df0dx(:))) max(abs(dfdx(:)))]));
f0val_mma = f0val/IterPars.scl;
df0dx = df0dx/IterPars.scl;
VisDats.loopHist = [VisDats.loopHist,VisDats.loop];
VisDats.fvalHist = [VisDats.fvalHist,fval];
VisDats.f0valHist = [VisDats.f0valHist,f0val];
[xmma,~,~,~,~,~,MMAPars.low,MMAPars.upp] = ...

mmasub(MMAPars.m,MMAPars.numDV,VisDats.loop,MMAPars.xval
,MMAPars.xmin,MMAPars.xmax,MMAPars.xold1,MMAPars.xold2, ...

f0val_mma,df0dx,fval,dfdx,MMAPars.low,MMAPars.upp,MMAPars.a
0,MMAPars.a,MMAPars.c,MMAPars.dd,MMAPars.mmaMoveLimits);
```

```
MMAPars.xold2 = MMAPars.xold1;
MMAPars.xold1 = MMAPars.xval;
MMAPars.xval = xmma;
if strcmpi(Inputs.OptType,'minComp')
    rho_CPs = MMAPars.xval;
    rho_CPs = reshape(rho_CPs,Surf.NCtrlPts);
    rhoi_hat = weight*rho_CPs(:);
    rhoe_hat = reshape(R_Center*rhoi_hat,Mesh.NEIDir);
    rhoe_star = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat-
HS_th)))/...
        (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
    rhoi_plt_hat =
reshape(VisDats.weight_plt*rho_CPs(:),Surf.NCtrlPts);
    VisDats.rhoi_plt =
(tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoi_plt_hat-HS_th)))/...
        (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
    prhoeStar_prhoeHat = sparse(diag(beta_HS*(1-
tanh(beta_HS*(rhoe_hat(:)-HS_th)).^2)/...
        (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
    IterPars.prhoe_prhoi = prhoeStar_prhoeHat*R_Center*weight;
    IterPars.rho_e = rhoe_star;%rho_e
end
rhoeHat = (1-rho_min)*IterPars.rho_e.^IterPars.penalty+rho_min;
PreOutput.rhoeHat = rhoeHat(:);
if rem(VisDats.loop,20) == 0 && VisDats.loop >= 100 ...
    && Inputs.StpPrj == 1 && IterPars.beta_HS <
str2double(Inputs.Projection)
    IterPars.beta_HS = IterPars.beta_HS + 1;
end
if rem(VisDats.loop,10) == 0 && VisDats.loop >= 10 ...
    && Inputs.StpPenalty == 1 && IterPars.penalty <
str2double(Inputs.Penalty)
    IterPars.penalty = IterPars.penalty + 1;
end
disp([' It: ' sprintf('%4i\t',VisDats.loop) ' Obj.: ' sprintf('%7.5f\t',f0val)
' S.t.: ' ...
    sprintf('%7.5f\t',fval)];
if rem(VisDats.loop,5) == 0
    fig = figure('visible','off');
    surf(VisDats.loc_x,VisDats.loc_y,VisDats.rhoi_plt);
    axis equal;
    caxis([0 1]); axis off; view(0,90);shading
interp;drawnow;colormap(flipud(gray));
    saveas(fig,[num2str(VisDats.loop),'_jpg']);
end
save('Output.mat','PreOutput','MMAPars','VisDats','IterPars');
end
```

C3.3 2D min_volume model (with Mises/D-P stress constraint)

```
function loop = SenOpt2DMinVol(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
SimOutput = load('SimOutput.mat');
PreOutput = load('Output.mat').PreOutput;
MMAPars = load('Output.mat').MMAPars;
VisDats = load('Output.mat').VisDats;
IterPars = load('Output.mat').IterPars;
AdjInfos = load('AdjInfos.mat').AdjInfos;
SimAdjOutput = load('SimAdjOutput.mat');
Surf = IterPars.Surf;
Mesh = IterPars.Mesh;
```

```
Ke = reshape(SimOutput.Ke,Mesh.NEN*2,Mesh.NEN*2,[]);
rho_min = IterPars.rho_min;
weight = IterPars.weight;
R_Center = IterPars.R_Center;
beta_HS = IterPars.beta_HS;
HS_th = IterPars.HS_th;
VisDats.loop = VisDats.loop+1;
loop = VisDats.loop;
if VisDats.loop == 1
    IterPars.InitVe = sum(SimOutput.Ve);
end
```

```

d = SimOutput.u;
dx = d(:,1);dy = d(:,2);
d = reshape(d',1,[]);
ue = zeros(Mesh.NEI,2*Mesh.NEN);
ue(:,1:2:end) = dx(Mesh.El);
ue(:,2:2:end) = dy(Mesh.El);
Lambda_KS = SimAdjOutput.u;
Lambda_dx = Lambda_KS(:,1);Lambda_dy = Lambda_KS(:,2);
Lambda_ue = zeros(Mesh.NEI,2*Mesh.NEN);
Lambda_ue(:,1:2:end) = Lambda_dx(Mesh.El);
Lambda_ue(:,2:2:end) = Lambda_dy(Mesh.El);
shp = SimOutput.shp; Stress = SimOutput.stress;
NEleGPs = prod(Surf.Order+1);
shp = reshape(shp,NEleGPs,2,[]);
shp = permute(shp,[2,1,3]);
Stress = reshape(Stress,3,[]);
edof = zeros(size(Mesh.El,1),size(Mesh.El,2)*2);
edof(:,1:2:end) = Mesh.El*2-1;
edof(:,2:2:end) = Mesh.El*2;
densPlot = Surf.CtrlPts4D; densPlot(3,:) = VisDatas.rhoi_plt;
Sw = BspineEval(Surf.KntVect, densPlot, Surf.uqKntVect);
[~, m, n] = size(Sw);
w = Sw(4, :, :);
S = bsxfun(@rdivide, Sw, w);
loc_x = reshape(S(1, :, :), m, n);
loc_y = reshape(S(2, :, :), m, n);
rhoi_Pars = reshape(S(3, :, :), m, n);
[f0val,df0dx] =
calc_dVdx(IterPars.rho_e(:),IterPars.prhoe_prhoi,SimOutput.Ve);
fval = AdjInfos.fval;
if length(IterPars.Const_value) == 1
    [dfdx,IterPars.ACS_c_old,IterPars.ACS_c,IterPars.ACS_a,SPars] =
    calc_dSdx(...

AdjInfos.ACS_c_old,AdjInfos.ACS_c,AdjInfos.ACS_a,Stress',ue,Lambda
_ue,shp,d,PreOutput.DO,Ke,IterPars.rho_e(:),...

IterPars.penalty,rho_min,edof,IterPars.prhoe_prhoi,IterPars.Const_v
alue,Surf,Mesh,PreOutput.MatProp,IterPars.ss);
elseif length(IterPars.Const_value) == 2% Drucker-Prager Stress index
    [dfdx,IterPars.ACS_c_old,IterPars.ACS_c,IterPars.ACS_a,SPars] =
    calc_dSdx_DP(...

AdjInfos.ACS_c_old,AdjInfos.ACS_c,AdjInfos.ACS_a,Stress',ue,Lambda
_ue,shp,d,PreOutput.DO,Ke,IterPars.rho_e(:),...

IterPars.penalty,rho_min,edof,IterPars.prhoe_prhoi,IterPars.Const_v
alue,Surf,Mesh,PreOutput.MatProp,...
    rhoi_Pars,IterPars.ss);
end
df0dx = df0dx/max(abs(df0dx));
dfdx = dfdx/max(abs(dfdx));
f0val = f0val/IterPars.InitVe;
VisDatas.loopHist = [VisDatas.loopHist,VisDatas.loop];
VisDatas.fvalHist = [VisDatas.fvalHist,fval];
VisDatas.f0valHist = [VisDatas.f0valHist,f0val];
[xmma,~,~,~,~,~,MMAPars.low,MMAPars.upp] = ...

mmasub(MMAPars.m,MMAPars.numDV,VisDatas.loop,MMAPars.xval
,MMAPars.xmin,MMAPars.xmax,MMAPars.xold1,MMAPars.xold2, ...

```

C3.4 2D min_volume model (adjoint method part)

```

function SenOpt2DMinVolAdj()
addpath(genpath(fullfile(pwd,'SRC')))
SimOutput = load('SimOutput.mat');
PreOutput = load('Output.mat').PreOutput;
VisDatas = load('Output.mat').VisDatas;
IterPars = load('Output.mat').IterPars;
Surf = IterPars.Surf;
Mesh = IterPars.Mesh;

```

```

f0val,df0dx,fval,dfdx,MMAPars.low,MMAPars.upp,MMAPars.a0,MM
APars.a,MMAPars.c,MMAPars.dd,MMAPars.mmaMoveLimits);
MMAPars.xold2 = MMAPars.xold1;
MMAPars.xold1 = MMAPars.xval;
MMAPars.xval = xmma;
rho_CP_s = MMAPars.xval;
rho_CP_s = reshape(rho_CP_s,Surf.NCtrlPts);
rhoi_hat = weight*rho_CP_s(:);
rhoe_hat = reshape(R_Center*rhoi_hat,Mesh.NEIDir);
rhoe_star = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat-
HS_th)))/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
rhoi_plt_hat = reshape(VisDatas.weight_plt*rho_CP_s(:),Surf.NCtrlPts);
VisDatas.rhoi_plt =
(tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoi_plt_hat-HS_th)))/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
prhoeStar_prhoeHat = diag(beta_HS*(1-tanh(beta_HS*(rhoe_hat(:)-
HS_th)).^2)/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
IterPars.prhoe_prhoi = prhoeStar_prhoeHat*R_Center*weight;
IterPars.rho_e = rhoe_star;%rho_e
rhoeHat = (1-rho_min)*IterPars.rho_e.^IterPars.penalty+rho_min;
PreOutput.rhoeHat = rhoeHat(:);
if rem(VisDatas.loop,20) == 0 && VisDatas.loop >= 100 ...
    && Inputs.StpPrj == 1 && IterPars.beta_HS <
str2double(Inputs.Projection)
    IterPars.beta_HS = IterPars.beta_HS + 1;
end
if rem(VisDatas.loop,10) == 0 && VisDatas.loop >= 10 ...
    && Inputs.StpPenalty == 1 && IterPars.penalty <
str2double(Inputs.Penalty)
    IterPars.penalty = IterPars.penalty + 1;
end
disp([' It.: ' sprintf('%4i\t',VisDatas.loop) ' Obj.: ' sprintf('%7.5f\t',f0val)
' S.t.: ' ...
    sprintf('%7.5f\t',fval)]);
if rem(loop,5) == 0
    fig = figure('visible','off');

surf(VisDatas.loc_x,VisDatas.loc_y,VisDatas.rhoi_plt);
axis equal;
caxis([0 1]); axis off; view(0,90);shading
interp;drawnow;colormap(flipud(gray));
saveas(fig,[num2str(loop),'.jpg']);
clf(fig);
surf(loc_x,loc_y,SPars);
if length(IterPars.Const_value) == 1% Mises Stress
    caxis([0 max(SPars(:))]);
elseif length(IterPars.Const_value) == 2% D-P Stress
    caxis([max(-1.5,min(SPars(:)) 0]);
end
axis equal;axis off; view(0,90);shading
interp;drawnow;colormap(jet);colorbar;
saveas(fig,['Stress',num2str(loop),'.jpg']);
end
save('Output.mat','PreOutput','MMAPars','VisDatas','IterPars');
end

```

```

AdjInfos = {};
VisDatas.loop = VisDatas.loop+1;
loop = VisDatas.loop;
if VisDatas.loop == 1
    IterPars.InitVe = sum(SimOutput.Ve);
end
d = SimOutput.u;
d = reshape(d',1,[]);

```

```

edof = zeros(size(Mesh.El,1),size(Mesh.El,2)*2);
edof(:,1:2:end) = Mesh.El*2-1;
edof(:,2:2:end) = Mesh.El*2;
shp = SimOutput.shp; Stress = SimOutput.stress;
NEleGPs = prod(Surf.Order+1);
shp = reshape(shp,NEleGPs,2,[]);
shp = permute(shp,[2,1,3]);
Stress = reshape(Stress,3,[]);
if length(IterPars.Const_value) == 1% Mises Stress

[AdjInfos.fval,AdjInfos.ACS_c_old,AdjInfos.ACS_c,AdjInfos.ACS_a,AdjI
nfos.LoadDatas] = calc_dSdxAdj(loop,...

IterPars.ACS_c_old,IterPars.ACS_c,IterPars.ACS_a,Stress',shp,d,PreOu
tput.DO,IterPars.rho_e(:),...

```

C3.5 3D min_compliance model

```

function loop = SenOpt3DMinComp(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
SimOutput = load('SimOutput.mat');
PreOutput = load('Output.mat').PreOutput;
MMAPars = load('Output.mat').MMAPars;
VisDatas = load('Output.mat').VisDatas;
IterPars = load('Output.mat').IterPars;
Volu = IterPars.Volu;
Mesh = IterPars.Mesh;
Ke = reshape(SimOutput.Ke,Mesh.NEN*3,Mesh.NEN*3,[]);
rho_min = IterPars.rho_min;
weight = IterPars.weight;
R_Center = IterPars.R_Center;
beta_HS = IterPars.beta_HS;
HS_th = IterPars.HS_th;
VisDatas.loop = VisDatas.loop+1;
loop = VisDatas.loop;
if VisDatas.loop == 1
    IterPars.InitVe = sum(SimOutput.Ve);
end
d = SimOutput.u;
dx = d(:,1);dy = d(:,2);dz = d(:,3);
d = reshape(d',1,[]);
ue = zeros(Mesh.NEI,3*Mesh.NEN);
ue(:,1:3:end) = dx(Mesh.El);
ue(:,2:3:end) = dy(Mesh.El);
ue(:,3:3:end) = dz(Mesh.El);
[f0val,df0dx] =
calc_dCdx(IterPars.ff,d,IterPars.rho_e(:),ue,Ke,IterPars.prhoe_prhoi,I
terPars.penalty,IterPars.rho_min);
[fval,dfdx] =
calc_dVdx(IterPars.rho_e(:),IterPars.prhoe_prhoi,SimOutput.Ve,IterP
ars.Const_value);
if loop == 1
    IterPars.scl = f0val/10;
end
dgt0 = 5;
dgt = dgt0 - floor(log10([max(abs(df0dx(:))) max(abs(dfdx(:)))]));
f0val_mma = f0val/IterPars.scl;
df0dx = df0dx/IterPars.scl;
VisDatas.loopHist = [VisDatas.loopHist,VisDatas.loop];
VisDatas.fvalHist = [VisDatas.fvalHist,fval];
VisDatas.f0valHist = [VisDatas.f0valHist,f0val];
[xmma,~,~,~,~,~,MMAPars.low,MMAPars.upp] = ...

mmasub(MMAPars.m,MMAPars.numDV,VisDatas.loop,MMAPars.xval
,MMAPars.xmin,MMAPars.xmax,MMAPars.xold1,MMAPars.xold2, ...

```

```

edof,IterPars.Const_value,Mesh,IterPars.ss);
elseif length(IterPars.Const_value) == 2% Drucker-Prager Stress index

[AdjInfos.fval,AdjInfos.ACS_c_old,AdjInfos.ACS_c,AdjInfos.ACS_a,AdjI
nfos.LoadDatas] = calc_dSdx_DPAAdj(loop,...

IterPars.ACS_c_old,IterPars.ACS_c,IterPars.ACS_a,Stress',shp,d,PreOu
tput.DO,IterPars.rho_e(:),IterPars.rho_min,...
edof,IterPars.Const_value,Mesh,IterPars.ss);
end
save('AdjInfos.mat','AdjInfos');
end

```

```

f0val,df0dx,fval,dfdx,MMAPars.low,MMAPars.upp,MMAPars.a0,MM
APars.a,MMAPars.c,MMAPars.dd,MMAPars.mmaMoveLimits);
MMAPars.xold2 = MMAPars.xold1;
MMAPars.xold1 = MMAPars.xval;
MMAPars.xval = xmma;
rho_CPs = MMAPars.xval;
rho_CPs = reshape(rho_CPs,Volu.NCtrlPts);
rhoi_hat = weight*rho_CPs(:);
rhoe_hat = reshape(R_Center*rhoi_hat,Mesh.NEIDir);
rhoe_star = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat-
HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
rhoi_plt_hat = reshape(VisDatas.weight_plt*rho_CPs(:),Volu.NCtrlPts);
VisDatas.rhoi_plt =
(tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoi_plt_hat-HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
prhoeStar_prhoeHat = sparse(diag(beta_HS*(1-
tanh(beta_HS*(rhoe_hat(:)-HS_th)).^2)/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
IterPars.prhoe_prhoi =
sparse(prhoeStar_prhoeHat*R_Center*weight);
IterPars.rho_e = rhoe_star;%rho_e
rhoeHat = (1-rho_min)*IterPars.rho_e.^IterPars.penalty+rho_min;
PreOutput.rhoeHat = rhoeHat(:);
if rem(VisDatas.loop,10) == 0 && VisDatas.loop >= 10 ...
    && Inputs.StpPrj == 1 && IterPars.beta_HS <
str2double(Inputs.Projection)
    IterPars.beta_HS = IterPars.beta_HS + 1;
end
if rem(VisDatas.loop,10) == 0 && VisDatas.loop >= 10 ...
    && Inputs.StpPenalty == 1 && IterPars.penalty <
str2double(Inputs.Penalty)
    IterPars.penalty = IterPars.penalty + 1;
end
disp([' It: ' sprintf('%4i\t',VisDatas.loop) ' Obj.: '
sprintf('%7.5f\t',VisDatas.f0valHist(end)) ' S.t.: ' ...
sprintf('%7.5f\t',VisDatas.fvalHist(end))]);

ParaPts = {linspace(0,1,2*Volu.NCtrlPts(1)),
linspace(0,1,2*Volu.NCtrlPts(2)), linspace(0,1,2*Volu.NCtrlPts(3))};
SRho =
refineRhoValue(Volu,ParaPts,permute(VisDatas.rhoi_plt,[4,1,2,3]));
VisDatas.rhoi_plt = SRho.rho;
save('Output.mat','PreOutput','MMAPars','VisDatas','IterPars');
end

```


Appendix D

D1 2D pre-process of OPS-ITO tool (with multi-pattern function)

```

function loop = Pre2D(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
Inputs = struct(Inputs);
mmaMoveLimits = str2double(strsplit(Inputs.MMAPars, ' '));
BCs = strsplit(Inputs.BCs, '\n');
numBCs = size(BCs,2);
for i = 1:numBCs
    BC_temp = strsplit(BCs{i}, ' ');
    Disp{i,1} = str2double(BC_temp[181]);
    Disp{i,2} = BC_temp[181];
    Disp{i,3} = str2func(BC_temp{3});
end
Forces = strsplit(Inputs.Loads, '\n');
if isempty(Forces{end})% the end line is empty!
    numLoads = size(Forces,2)-1;
else
    numLoads = size(Forces,2);
end
for i = 1:numLoads
    force_temp = strsplit(Forces{i}, ' ');
    force{i,1} = str2double(force_temp[181]);
    force{i,2} = force_temp[181];
    force{i,3} = str2func(force_temp{3});
end
MatProp = str2double(strsplit(Inputs.MatProp, ' '));
%% generate the mesh model
if strcmpi(Inputs.GeoInputType, 'Default')
    Sizes = str2double(strsplit(Inputs.Sizes, ' '));
    Orders = str2double(strsplit(Inputs.Orders, ' '));
    NEls = str2double(strsplit(Inputs.NEls, ' '));
    L = Sizes(1);D = Sizes(2);
    CtrlPts = zeros(4, 2, 2);
    CtrlPts(1 : 3, 1, 1) = [0; -D/2; 0];
    CtrlPts(1 : 3, 2, 1) = [L; -D/2; 0];
    CtrlPts(1 : 3, 1, 2) = [0; D/2; 0];
    CtrlPts(1 : 3, 2, 2) = [L; D/2; 0];
    CtrlPts(4, :, :) = 1;
    KntVect[181] = [0 0 1 1];
    KntVect[181] = [0 0 1 1];
    Surf = CreateNURBS(KntVect, CtrlPts);
    p=Orders(1);q=Orders(2);
    kx=1;ky=1;
    nelx=NEls(1); nely=NEls(2);
    Surf = KRefine(Surf, [nelx, nely], [p, q], [p-kx, q-ky]);
    Mesh = Mesh2D(Surf, 'VectorField');
elseif strcmpi(Inputs.GeoInputType, 'User input')
    GeoModelPath = Inputs.GeoModelPath;
    ModelDatas = load(GeoModelPath);
    Surf = ModelDatas.Surf;
    Mesh = Mesh2D(Surf, 'VectorField');
    nelx = Mesh.NEIDir(1); nely = Mesh.NEIDir(2);
end
if Inputs.StpPenalty == 1
    penalty = 1;
elseif Inputs.StpPenalty == 0
    penalty = str2double(Inputs.Penalty);
end
if Inputs.StpPrj == 1
    beta_HS = 0.1;
elseif Inputs.StpPrj == 0
    beta_HS = str2double(Inputs.Projection);
end
check_x = linspace(0,1,2*nelx+1);
check_y = linspace(0,1,2*nely+1);
node_x = check_x(2:2:end);
node_y = check_y(2:2:end);
R_Center = FindInterpFunc(Surf,{node_x, node_y});
R_Center = sparse(R_Center);
xloc_CtrlPts = reshape(Surf.CtrlPts3D(1, :, :), 1, []);
yloc_CtrlPts = reshape(Surf.CtrlPts3D(2, :, :), 1, []);
rmin = str2double(Inputs.rmin); HS_th =
str2double(Inputs.PrjThreshold);
%% modify the rho_e
if strcmpi(Inputs.OptType, 'minComp') ||
strcmpi(Inputs.OptType, 'minVol')
    loc_CPs = [xloc_CtrlPts, yloc_CtrlPts];
    dist = pdist2(loc_CPs, loc_CPs);
    weight = rmin-dist;
    weight(weight<0) = 0;
    weight = weight./(sum(weight,2));
    weight = sparse(weight);
    rho_CPs = 0.5*ones(Surf.NCtrlPts);
    rhoi_hat = weight*rho_CPs(:);
    rhoe_hat = reshape(R_Center*rhoi_hat, Mesh.NEIDir);
    rhoe_star = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat-
HS_th)))/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
    prhoeStar_prhoeHat = diag(beta_HS*(1-
tanh(beta_HS*(rhoe_hat(:)-HS_th)).^2)/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
    rho_e = reshape(rhoe_star, nelx, nely);
    prhoe_prhoi = sparse(prhoeStar_prhoeHat*R_Center*weight);
    dist_plt = pdist2(loc_CPs, loc_CPs); weight_plt = rmin-
dist_plt; weight_plt(weight_plt<0) = 0;
    weight_plt = weight_plt./(sum(weight_plt,2)); weight_plt =
sparse(weight_plt);
    rhoi_plt_hat = reshape(weight_plt*rho_CPs(:), Surf.NCtrlPts);
    rhoi_plt = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoi_plt_hat-
HS_th)))/...
    (tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
    %%export the MMA parameters
    tnn = size(rho_CPs(:),1);
    xy00 = rho_CPs(:);
    %-----multi-pattern pre part 1 start-----
    -----
elseif strcmpi(Inputs.OptType, 'minCompSimi')
    xloc_Eles = R_Center*xloc_CtrlPts;
    yloc_Eles = R_Center*yloc_CtrlPts;
    sideSldLyrs = 0;
    topSldLyrs = 0;

```

```

xDiv = 2;
yDiv = 1;
% SimGroupIDRef =
[[1;5],[2;6],[3;7],[4;8],[9;13],[10;14],[11;15],[12;16],[17;21],[18;22],[1
9;23],[20;24],[25;29]...
%      ,[26;30],[27;31],[28;32]];
SimGroupIDRef = {};
% SimGroupIDRef = {[1;2;3]};

if ~isempty(SimGroupIDRef)
    disp('self determined arrangement scheme')
    numOfVari = numel(SimGroupIDRef);
    disp(numOfVari)
else
    numOfVari = 2;
    disp(numOfVari)
end

NEls_RUC = [(Mesh.NEIDir(1)-topSldLyrs)/xDiv,(Mesh.NEIDir(2)-
2*sideSldLyrs)/yDiv];
ElMat = reshape(1:Mesh.NEI,Mesh.NEIDir);
El_RUC = mat2cell(ElMat(1:end-topSldLyrs,sideSldLyrs+1:end-
sideSldLyrs), ...
(Mesh.NEIDir(1)-topSldLyrs)/xDiv*ones(xDiv,1), ...
(Mesh.NEIDir(2)-sideSldLyrs*2)/yDiv*ones(yDiv,1));

rho_RUC = repmat(0.5*ones(NEls_RUC),1,numOfVari);
rho_e = 0.5*ones(Mesh.NEIDir);
loc_Els = [xloc_Els,yloc_Els];
dist = sparse(pdist2(loc_Els,loc_Els));
weight = rmin-dist;
weight(weight<0) = 0;
weight = sparse(weight./(sum(weight,2)));
rhoe_hat_RUC = weight*rho_e(:);% filter the size
rhoe_star_RUC =
(tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat_RUC-HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));% project the
density
prhoeStar_prhoeHat = sparse(diag(beta_HS*(1-
tanh(beta_HS*(rhoe_hat_RUC(-)-HS_th)).^2)/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)))));
rho_e = reshape(rhoe_star_RUC,nelx,nely);
rho_e(:,1:sideSldLyrs) = 1;
rho_e(:,end-sideSldLyrs+1:end) = 1;
rho_e(end+1-topSldLyrs:end,:) = 1;
prhoe_prhoi = prhoeStar_prhoeHat*weight;
rhoi_plt = rho_e;
weight_plt = weight;

tnn = numel(rho_RUC);
xy00 = rho_RUC(:);
%%-----multi-pattern pre part 1 end-----
else
    error('unsupported optimization type, please check')
end
numVar = length(xy00);
xval=xy00;
xold1 = xy00;
xold2 = xy00;
xmin=0*ones(tnn,1);
xmax=ones(tnn,1);
low = xmin;
upp = xmax;
numDV = length(low);
m = 1;
Var_num=tnn;
c=1000*ones(m,1);
dd=ones(m,1);
a0=1;
a=zeros(m,1);
%% define BCs and Loads
f = sparse(zeros(Mesh.NDof, 1));
ConDispVals = [];LoadVals = [];

ConDispDofs = [];LoadDofs = [];
for i = 1:numBCs
    [UTemp, DofsTemp] = projDrchltBdryVals(Surf, Mesh, Disp{i,3},
Disp{i,1}, Disp{i,2});
    DofsTemp(abs(UTemp)<=0.2) = [];
    UTemp(abs(UTemp)<=0.2) = [];
    ConDispVals = [ConDispVals; UTemp];
    ConDispDofs = [ConDispDofs; DofsTemp];
end
for i = 1:numLoads
    [LoadTemp,DofsTemp] = applyNewmannBdryVals(Surf, Mesh,
force{i,3}, force{i,1}, force{i,2});
    LoadVals = [LoadVals; LoadTemp];
    LoadDofs = [LoadDofs; DofsTemp];
end
f(LoadDofs) = f(LoadDofs) + LoadVals;
ff = sparse(zeros(size(f)));
ff(1:2:end) = f(1:Surf.NNP);
ff(2:2:end) = f(Surf.NNP+1:end);
%% generate the output datas
CtrlPts = reshape(Surf.CtrlPts3D,3,[]);
% export constrained CPs list
ConNdlDs = rem(ConDispDofs,Surf.NNP);
ConNdlDs(ConNdlDs==0) = Surf.NNP;
ConDofIds = ceil(ConDispDofs/Surf.NNP);
ConDofList = zeros(Surf.NNP,2);
for i = 1:length(ConNdlDs)
    ConDofList(ConNdlDs(i),ConDofIds(i)) = 1;
end
ConDofList = [(1:Surf.NNP)',ConDofList];
ConDofList(ConDofList(:,2)+ConDofList(:,3)==0,:) = [];
% export loaded CPs list
LoadNdlDs = rem(LoadDofs,Surf.NNP);
LoadNdlDs(LoadNdlDs==0) = Surf.NNP;
LoadDofIds = ceil(LoadDofs/Surf.NNP);
LoadDofList = zeros(Surf.NNP,2);
for i = 1:length(LoadNdlDs)
    LoadDofList(LoadNdlDs(i),LoadDofIds(i)) = LoadVals(i);
end
LoadDofList = [(1:Surf.NNP)',LoadDofList];
LoadDofList(abs(LoadDofList(:,2))+abs(LoadDofList(:,3))==0,:) = [];
El = Mesh.El;
rho_min = 1e-9;
rho_eHat = (1-rho_min)*rho_e.^penalty+rho_min;
Weights = reshape(Surf.Weights,1,[]);
Const_value = str2double(strsplit(Inputs.ConstVal,' '));
loop = 0;
loopHist = [];fvalHist = [];f0valHist = [];
NEN = Mesh.NEN;
PreOutput = {}; MMAPars = {}; VisDats = {}; IterPars = {};
PreOutput.CtrlPts = CtrlPts;PreOutput.ConDofList =
ConDofList;PreOutput.LoadDofList = LoadDofList;
PreOutput.MatProp = MatProp;PreOutput.DO =
getElastMat(MatProp(1), MatProp(2),
'PlaneStress');PreOutput.MatTag = Inputs.MatTag;
PreOutput.NEN = NEN;PreOutput.El = El;PreOutput.NEI = Mesh.NEI;
PreOutput.NCPs = prod(Surf.NCtrlPts);
PreOutput.NEIDir = Mesh.NEIDir;PreOutput.Order =
Surf.Order;PreOutput.El = Mesh.El;
PreOutput.KntVectX = Surf.KntVect[181];PreOutput.KntVectY =
Surf.KntVect[181];PreOutput.Weight = Weights(Mesh.El);
PreOutput.rhoeHat = rho_eHat(:);PreOutput.OptType =
Inputs.OptType;
MMAPars.mmaMoveLimits = mmaMoveLimits;MMAPars.tnn = tnn;
MMAPars.xy00 = xy00;MMAPars.numVar = numVar;MMAPars.xval =
xval;
MMAPars.xold1 = xold1;MMAPars.xold2 = xold2;MMAPars.xmin =
xmin;
MMAPars.xmax = xmax;MMAPars.low = low;MMAPars.upp = upp;
MMAPars.numDV = numDV;MMAPars.m = m;MMAPars.Var_num =
Var_num;
MMAPars.c = c;MMAPars.dd = dd;MMAPars.a0 = a0;MMAPars.a = a;

```

```

VisDats.loc_x = reshape(xloc_CtrlPts,Surf.NCtrlPts);VisDats.loc_y =
reshape(yloc_CtrlPts,Surf.NCtrlPts);
VisDats.loop = loop;VisDats.loopHist = loopHist;VisDats.fvalHist =
fvalHist;
VisDats.f0valHist = f0valHist;VisDats.rhoi_plt =
rhoi_plt;VisDats.weight_plt = weight_plt;
IterPars.Const_value = Const_value;IterPars.rho_min =
rho_min;IterPars.rho_e = rho_e;
IterPars.weight = weight;IterPars.R_Center =
R_Center;IterPars.prhoe_prhoi = prhoe_prhoi;
IterPars.ff = ff;IterPars.penalty = penalty;IterPars.beta_HS = beta_HS;
IterPars.HS_th = HS_th;IterPars.Surf = Surf;IterPars.Mesh = Mesh;
IterPars.ACS_c_old = 0;IterPars.ACS_c = 0; IterPars.ACS_a = 0;
%%-----multi-pattern pre part 2 start-----
if strcmpi(Inputs.OptType,'minCompSimi')
    IterPars.topSldLyrs = topSldLyrs; IterPars.sideSldLyrs = sideSldLyrs;
    IterPars.xDiv = xDiv; IterPars.yDiv = yDiv; IterPars.numOfVari =
numOfVari;

```

```

IterPars.NEls_RUC = NEls_RUC; IterPars.El_RUC = El_RUC;
IterPars.errCluster = Inf;
VisDats.loc_x = reshape(xloc_Eles,Mesh.NEIDir);VisDats.loc_y =
reshape(yloc_Eles,Mesh.NEIDir);

```

```
IterPars.SimGroupIDRef = SimGroupIDRef;
```

```

MMAPars.mmaMoveLimits = mmaMoveLimits*numOfVari;
% IterPars.rho_RUC = rho_RUC;
end
%%-----multi-pattern pre part 2 end-----
if strcmpi(Inputs.OptType,'minVol')
    IterPars.ss = 0.8;
end
save('Output.mat','PreOutput','MMAPars','VisDats','IterPars');
end

```

D2 2D min-compliance model (with multi-pattern function)

```

function loop = SenOpt2DMinComp(Inputs)
addpath(genpath(fullfile(pwd,'SRC')))
SimOutput = load('SimOutput.mat');
PreOutput = load('Output.mat').PreOutput;
MMAPars = load('Output.mat').MMAPars;
VisDats = load('Output.mat').VisDats;
IterPars = load('Output.mat').IterPars;
Surf = IterPars.Surf;
Mesh = IterPars.Mesh;
Ke = reshape(SimOutput.Ke,Mesh.NEN*2,Mesh.NEN*2,[]);
rho_min = IterPars.rho_min;
weight = IterPars.weight;
R_Center = IterPars.R_Center;
beta_HS = IterPars.beta_HS;
HS_th = IterPars.HS_th;
prhoe_prhoi = IterPars.prhoe_prhoi;
VisDats.loop = VisDats.loop+1;
loop = VisDats.loop;
if VisDats.loop == 1
    IterPars.InitVe = sum(SimOutput.Ve);
end
d = SimOutput.u;
dx = d(:,1);dy = d(:,2);
d = reshape(d',1,[]);
ue = zeros(Mesh.NEl,2*Mesh.NEN);
ue(:,1:2:end) = dx(Mesh.El);
ue(:,2:2:end) = dy(Mesh.El);
%%-----multi-pattern min-Comp part 1 start-----
if strcmpi(Inputs.OptType,"minCompSimi")
    xDiv = IterPars.xDiv;
    yDiv = IterPars.yDiv;
    numOfVari = IterPars.numOfVari;
    topSldLyrs = IterPars.topSldLyrs; sideSldLyrs = IterPars.sideSldLyrs;
    if VisDats.loop <= 100
        KeGPU = gpuArray(Ke); ueGPU = gpuArray(permute(ue,[3,2,1]));
        ueKe = squeeze(pagemtimes(ueGPU,KeGPU));
        Ce =
reshape(squeeze(pagemtimes(ueGPU,permute(ueKe,[1,3,2]))),Mesh.
NEIDir);
        Ce_RUC = mat2cell(Ce(1:end-topSldLyrs,sideSldLyrs+1:end-
sideSldLyrs), ...
        (Mesh.NEIDir(1)-
topSldLyrs)/xDiv*ones(1,xDiv),(Mesh.NEIDir(2)-
sideSldLyrs*2)/yDiv*ones(1,yDiv));
        Ce_RUC = cat(3,Ce_RUC{:});
        if isempty(IterPars.SimGroupIDRef)
            if loop == 1
                IterPars.splitClusterAssume = ones(size(Ce_RUC,3),2);
                IterPars.splitClusterAssume(:,2) = 1e7;

```

```

else
    for i = 1:numOfVari
        IterPars.splitCluMat(:,i) =
mean(Ce_RUC(:,i),IterPars.splitClusterAssume(:,1)==i),3);
    end
end
IterPars.splitClusterAssumeOld = IterPars.splitClusterAssume;
for i = 1:10
    if loop == 1
        IterPars.splitCluMat =
zeros([size(Ce_RUC,1:2),numOfVari]);% the last cluster is zero to
store void unit-cells
        for ii = 1:numOfVari
            IterPars.splitCluMat(:,ii) = rand(size(Ce_RUC(:,i)));
        end
    end
    [~, splitClusterAssumeTemp] =
KMeans(loop,IterPars.splitCluMat,IterPars.splitClusterAssume,Ce_RU
C,numOfVari);
    if
sum(splitClusterAssumeTemp(splitClusterAssumeTemp(:,1)<=numOf
Vari,2)) <= IterPars.errCluster
        IterPars.errCluster =
sum(splitClusterAssumeTemp(splitClusterAssumeTemp(:,1)<=numOf
Vari,2));
        IterPars.splitClusterAssume = splitClusterAssumeTemp;
    end
    if loop > 1
        mmaXval = [];
        for ii = 1:numOfVari
            mmaXvalTemp =
mean(IterPars.rho_RUC(:,i),IterPars.splitClusterAssumeOld(...
            IterPars.splitClusterAssume(:,1)==ii,1),3);
            mmaXval = [mmaXval;mmaXvalTemp(:)];
        end
        MMAPars.xval = mmaXval;
    end
end
end
end
end
if isempty(IterPars.SimGroupIDRef)
    simiGroupID = cell(1,IterPars.numOfVari);
    for i = 1:IterPars.numOfVari
        simiGroupID{i} = find(IterPars.splitClusterAssume(:,1)==i);
        disp(simiGroupID{i})
    end
end
else
    simiGroupID = IterPars.SimGroupIDRef;
    for i = 1:IterPars.numOfVari

```

```

disp(simiGroupld{i})
end
end
IterPars.simiGroupldOld = simiGroupld;
drhoeDiv_drhoerUC = [];
for ii = 1:length(simiGroupld)
    TempSimGroupld = simiGroupld[132];
    drhoeDiv_drhoerUC_temp =
sparse(zeros(Mesh.NEI,prod(IterPars.NEls_RUC)));
    for jj = TempSimGroupld'
        DivElList = IterPars.El_RUC(jj);
        for kk = 1:numel(DivElList)% loop the current div
            drhoeDiv_drhoerUC_temp(DivElList(kk),kk) = 1;
        end
    end
    drhoeDiv_drhoerUC =
[drhoeDiv_drhoerUC,drhoeDiv_drhoerUC_temp];
end
prhoe_prhoi = IterPars.prhoe_prhoi*drhoeDiv_drhoerUC;
end
%%-----multi-pattern min-Comp part 1 end-----
[f0val,df0dx] =
calc_dCdx(IterPars.ff,IterPars.rho_e(:),ue,Ke,prhoe_prhoi,IterPars.p
enalty,IterPars.rho_min);
[fval,dfdx] =
calc_dVdx(IterPars.rho_e(:),prhoe_prhoi,SimOutput.Ve,IterPars.Const
_value);
if loop == 1
    IterPars.scl = f0val/10;
end
dgt0 = 5;
dgt = dgt0 - floor(log10([max(abs(df0dx(:))) max(abs(dfdx(:)))]));
f0val_mma = f0val/IterPars.scl;
df0dx = df0dx/IterPars.scl;
VisDatas.loopHist = [VisDatas.loopHist,VisDatas.loop];
VisDatas.fvalHist = [VisDatas.fvalHist,fval];
VisDatas.f0valHist = [VisDatas.f0valHist,f0val];
[xmma,~,~,~,~,~,MMAPars.low,MMAPars.upp] = ...

mmasub(MMAPars.m,MMAPars.numDV,VisDatas.loop,MMAPars.xval
,MMAPars.xmin,MMAPars.xmax,MMAPars.xold1,MMAPars.xold2, ...

f0val_mma,df0dx,fval,dfdx,MMAPars.low,MMAPars.upp,MMAPars.a
0,MMAPars.a,MMAPars.c,MMAPars.dd,MMAPars.mmaMoveLimits);
MMAPars.xold2 = MMAPars.xold1;
MMAPars.xold1 = MMAPars.xval;
MMAPars.xval = xmma;
if strcmpi(Inputs.OptType,'minComp')
    rho_CPs = MMAPars.xval;
    rho_CPs = reshape(rho_CPs,Surf.NCtrlPts);
    rhoi_hat = weight*rho_CPs(:);
    rhoe_hat = reshape(R_Center*rhoi_hat,Mesh.NEIDir);
    rhoe_star = (tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_hat-
HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
    rhoi_plt_hat =
reshape(VisDatas.weight_plt*rho_CPs(:),Surf.NCtrlPts);

VisDatas.rhoi_plt =
(tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoi_plt_hat-HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th)));
prhoeStar_prhoeHat = sparse(diag(beta_HS*(1-
tanh(beta_HS*(rhoe_hat(:)-HS_th)).^2)/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
    IterPars.prhoe_prhoi = prhoeStar_prhoeHat*R_Center*weight;
    IterPars.rho_e = rhoe_star;%rho_e
%%-----multi-pattern min-Comp part 2 start-----
elseif strcmpi(Inputs.OptType,'minCompSimi')
    rho_RUC = MMAPars.xval;
    IterPars.rho_RUC =
reshape(rho_RUC,[IterPars.NEls_RUC,numOfVari]);
    rhoe_star_RUC = weight*drhoeDiv_drhoerUC*rho_RUC(:);
    rho_e =
reshape((tanh(beta_HS*HS_th)+tanh(beta_HS*(rhoe_star_RUC-
HS_th)))/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-
HS_th))),Mesh.NEIDir);% project the density
    prhoeStar_prhoeHat = sparse(diag(beta_HS*(1-
tanh(beta_HS*(rhoe_star_RUC(:)-HS_th)).^2)/...
(tanh(beta_HS*HS_th)+tanh(beta_HS*(1-HS_th))));
    rho_e(:,1:sideSldLyrs) = 1;
    rho_e(:,end-sideSldLyrs+1:end) = 1;
    rho_e(end+1-topSldLyrs:end,:) = 1;

    IterPars.rho_e = rho_e;
    IterPars.prhoe_prhoi = prhoeStar_prhoeHat*weight;

    VisDatas.rhoi_plt = rho_e;
%%-----multi-pattern min-Comp part 2 end-----
end
rhoeHat = (1-rho_min)*IterPars.rho_e.^IterPars.penalty+rho_min;
PreOutput.rhoeHat = rhoeHat(:);
if rem(VisDatas.loop,20) == 0 && VisDatas.loop >= 100 ...
    && Inputs.StpPrj == 1 && IterPars.beta_HS <
str2double(Inputs.Projection)
    IterPars.beta_HS = IterPars.beta_HS + 1;
end
if rem(VisDatas.loop,10) == 0 && VisDatas.loop >= 10 ...
    && Inputs.StpPenalty == 1 && IterPars.penalty <
str2double(Inputs.Penalty)
    IterPars.penalty = IterPars.penalty + 1;
end
disp([' It.: ' sprintf('%4i\t',VisDatas.loop) ' Obj.: ' sprintf('%7.5f\t',f0val)
'S.t.: ' ...
sprintf('%7.5f\t',fval)]);
if rem(VisDatas.loop,5) == 0
    fig = figure('visible','off');
    surf(VisDatas.loc_x,VisDatas.loc_y,VisDatas.rhoi_plt);
    axis equal;
    caxis([0 1]); axis off; view(0,90);shading
interp;drawnow;colormap(flipud(gray));
saveas(fig,[num2str(VisDatas.loop),'.jpg']);
end
save('Output.mat','PreOutput','MMAPars','VisDatas','IterPars');
end

```

Reference

- [1] Huerta S. Structural design in the work of Gaudi. *Architectural science review*. 2006;49:324-39.
- [2] Kotnik T, Schwartz J. The architecture of heinz isler. *Journal of the international association for shell and spatial structures*. 2011;52:185-90.
- [3] Westfall CW. *Architecture, liberty and civic order: architectural theories from Vitruvius to Jefferson and beyond*: Routledge; 2016.
- [4] Billington DP. *The tower and the bridge: the new art of structural engineering*: Princeton University Press; 1985.
- [5] Beghini LL, Beghini A, Katz N, Baker WF, Paulino GH. Connecting architecture and engineering through structural topology optimization. *Engineering Structures*. 2014;59:716-26.
- [6] Dean B, Dulac J, Petrichenko K, Graham P. *Global Status Report 2016: Towards zero-emission efficient and resilient buildings*. 2016.
- [7] Hansen S, Sadeghian P. Recycled gypsum powder from waste drywalls combined with fly ash for partial cement replacement in concrete. *Journal of Cleaner Production*. 2020;274:122785.
- [8] Jiang L, Li C, Wang C, Xu N, Chu H. Utilization of flue gas desulfurization gypsum as an activation agent for high-volume slag concrete. *Journal of Cleaner Production*. 2018;205:589-98.
- [9] Agreement P. Paris agreement. Report of the Conference of the Parties to the United Nations Framework Convention on Climate Change (21st Session, 2015: Paris) Retrieved December: HeinOnline; 2015. p. 2017.
- [10] Sasaki M, Itō T, Isozaki A. *Morphogenesis of flux structure*: Aa Publications; 2007.
- [11] Yan X, Bao D, Zhou Y, Xie Y, Cui T. Detail control strategies for topology optimization in architectural design and development. *Frontiers of Architectural Research*. 2022;11:340-56.
- [12] Dombernowsky P, Søndergaard A. *Unikabeton prototype. Fabricate: Making Digital Architecture*: Riverside Architectural Press; 2011. p. 56-61.
- [13] Jaillon L, Poon CS. Life cycle design and prefabrication in buildings: A review and case studies in Hong Kong. *Automation in Construction*. 2014;39:195-202.
- [14] Smith RE. *Prefab architecture: A guide to modular design and construction*: John Wiley & Sons; 2010.
- [15] Bos F, Wolfs R, Ahmed Z, Salet T. Additive manufacturing of concrete in construction: potentials and challenges of 3D concrete printing. *Virtual and physical prototyping*. 2016;11:209-25.
- [16] Nematollahi B, Xia M, Sanjayan J. Current progress of 3D concrete printing

- technologies. ISARC Proceedings of the international symposium on automation and robotics in construction: IAARC Publications; 2017.
- [17] Paul SC, Van Zijl GP, Tan MJ, Gibson I. A review of 3D concrete printing systems and materials properties: Current status and future research prospects. *Rapid Prototyping Journal*. 2018.
- [18] Khoshevis B. *Additive Fabrication Apparatus and Method*. 1996.
- [19] Seibert J, Rynerson M. *Solid free-form fabrication apparatus and method*. Google Patents; 2009.
- [20] Evjemo LD, Moe S, Gravdahl JT, Roulet-Dubonnet O, Gellein LT, Br V. Additive manufacturing by robot manipulator: An overview of the state-of-the-art and proof-of-concept results. 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA): IEEE; 2017. p. 1-8.
- [21] Tobi AM, Omar S, Yehia Z, Al-Ojaili S, Hashim A, Orhan O. Cost viability of 3D printed house in UK. *IOP Conference Series: Materials Science and Engineering*: IOP Publishing; 2018. p. 012061.
- [22] John Dias MR. *Design of the printing head for 3D printing from the concrete*. 2016.
- [23] Saunders J, Lißner M, Townsend D, Petrinic N, Bergmann J. Impact behaviour of 3D printed cellular structures for mouthguard applications. *Scientific Reports*. 2022;12:4020.
- [24] Liu H, Yang D, Hao P, Zhu X. Isogeometric analysis based topology optimization design with global stress constraint. *Computer Methods in Applied Mechanics and Engineering*. 2018;342:625-52.
- [25] Qiu W, Wang Q, Gao L, Xia Z. Evolutionary topology optimization for continuum structures using isogeometric analysis. *Structural and Multidisciplinary Optimization*. 2022;65:121.
- [26] Jahangiri HA, Jahangiri A. Combination of Isogeometric analysis and level-set method in topology optimization of heat-conduction systems. *Applied Thermal Engineering*. 2019;161:114134.
- [27] Hou W, Gai Y, Zhu X, Wang X, Zhao C, Xu L et al. Explicit isogeometric topology optimization using moving morphable components. *Computer Methods in Applied Mechanics and Engineering*. 2017;326:694-712.
- [28] Stoiber N, Kromoser B. Topology optimization in concrete construction: a systematic review on numerical and experimental investigations. *Structural and Multidisciplinary Optimization*. 2021;64:1725-49.
- [29] Vantighem G, De Corte W, Shakour E, Amir O. 3D printing of a post-tensioned concrete girder designed by topology optimization. *Automation in Construction*. 2020;112:103084.
- [30] Tsavdaridis KD. Applications of topology optimization in structural engineering: High-rise buildings and steel components. *Jordan Journal of Civil Engineering*. 2015;9:335-57.
- [31] Elmas P, ORHON AV, AKGUN Y. Potential Contributions of Topology Optimization for Building Structures: A Redesign Case Study on Saint Voukolos Church. *A+ Arch Design International Journal of Architecture and Design*. 2015;8:89-99.
- [32] Gray J, Moore K, Naylor B. OpenMDAO: An open source framework for multidisciplinary analysis and optimization. 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference 2010. p. 9101.

- [33] Gray JS, Hwang JT, Martins JR, Moore KT, Naylor BA. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*. 2019;59:1075-104.
- [34] Heath C, Gray J. OpenMDAO: framework for flexible multidisciplinary design, analysis and optimization methods. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA2012. p. 1673.
- [35] Talischi C, Paulino GH, Pereira A, Menezes IF. PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Structural and Multidisciplinary Optimization*. 2012;45:329-57.
- [36] Duarte LS, Celes W, Pereira A, Menezes IF, Paulino GH. PolyTop++: an efficient alternative for serial and parallel topology optimization on CPUs & GPUs. *Structural and Multidisciplinary Optimization*. 2015;52:845-59.
- [37] Aage N. TopOpt in PETSc: Exercises in large scale topology optimization. 2016.
- [38] Qu X, Pagaldipti N, Fleury R, Saiki J. Thermal topology optimization in optistruct software. 17th AIAA/ISSMO Multidisciplinary analysis and optimization conference2016. p. 3829.
- [39] Zhou Q, Shen W, Wang J, Zhou YY, Xie YM. Ameba: A new topology optimization tool for architectural design. *Proceedings of IASS Annual Symposia: International Association for Shell and Spatial Structures (IASS)*; 2018. p. 1-8.
- [40] Zuo ZH, Xie YM. A simple and compact Python code for complex 3D topology optimization. *Advances in Engineering Software*. 2015;85:1-11.
- [41] Jankovics D, Gohari H, Tayefeh M, Barari A. Developing topology optimization with additive manufacturing constraints in ANSYS®. *IFAC-PapersOnLine*. 2018;51:1359-64.
- [42] Dede EM. Multiphysics topology optimization of heat transfer and fluid flow systems. *proceedings of the COMSOL Users Conference2009*.
- [43] Salaimanimagudam MP, Suribabu CR, Murali G, Abid SR. Impact response of hammerhead pier fibrous concrete beams designed with topology optimization. *Periodica Polytechnica Civil Engineering*. 2020;64:1244-58.
- [44] Sigmund O. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization*. 2001;21:120-7.
- [45] Wang F, Lazarov BS, Sigmund O. On projection methods, convergence and robust formulations in topology optimization. *Structural and multidisciplinary optimization*. 2011;43:767-84.
- [46] Guest JK, Prévost JH, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International journal for numerical methods in engineering*. 2004;61:238-54.
- [47] Bendsoe MP, Sigmund O. *Topology optimization. Optimization of Structural and Mechanical Systems*: World Scientific; 2007. p. 161-94.
- [48] Guest JK. Topology optimization with multiple phase projection. *Computer Methods in Applied Mechanics and Engineering*. 2009;199:123-35.
- [49] Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*. 1998;16:68-75.

- [50] Jog CS, Haber RB. Stability of finite element models for distributed-parameter optimization and topology design. *Computer methods in applied mechanics and engineering*. 1996;130:203-26.
- [51] Diaz A, Sigmund O. Checkerboard patterns in layout optimization. *Structural optimization*. 1995;10:40-5.
- [52] Sigmund O. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*. 1997;25:493-524.
- [53] Bendsoe MP, Sigmund O. *Topology optimization: theory, methods, and applications*: Springer Science & Business Media; 2003.
- [54] Rozvany GI. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary optimization*. 2001;21:90-108.
- [55] Cheng GD, Guo X. ϵ -relaxed approach in structural topology optimization. *Structural optimization*. 1997;13:258-66.
- [56] Cheng G, Jiang Z. Study on topology optimization with stress constraints. *Engineering Optimization*. 1992;20:129-48.
- [57] Senhora FV, Giraldo-Londono O, Menezes IF, Paulino GH. Topology optimization with local stress constraints: a stress aggregation-free approach. *Structural and Multidisciplinary Optimization*. 2020;62:1639-68.
- [58] Rozvany G, Sobieszczanski-Sobieski J. New optimality criteria methods: forcing uniqueness of the adjoint strains by corner-rounding at constraint intersections. *Structural optimization*. 1992;4:244-6.
- [59] Duysinx P, Bendsøe MP. Topology optimization of continuum structures with local stress constraints. *International journal for numerical methods in engineering*. 1998;43:1453-78.
- [60] Bruggi M. On an alternative approach to stress constraints relaxation in topology optimization. *Structural and multidisciplinary optimization*. 2008;36:125-41.
- [61] Duysinx P, Sigmund O. New developments in handling stress constraints in optimal material distribution. 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization 1998. p. 4906.
- [62] Yang R, Chen C. Stress-based topology optimization. *Structural optimization*. 1996;12:98-105.
- [63] Qiu G, Li X. A note on the derivation of global stress constraints. *Structural and Multidisciplinary Optimization*. 2010;40:625-8.
- [64] Cottrell JA, Hughes TJ, Bazilevs Y. *Isogeometric analysis: toward integration of CAD and FEA*: John Wiley & Sons; 2009.
- [65] Hughes TJ, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*. 2005;194:4135-95.
- [66] Piegl L, Tiller W. *The NURBS book*: Springer Science & Business Media; 1996.
- [67] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S et al. Isogeometric analysis using T-splines. *Computer methods in applied mechanics and engineering*. 2010;199:229-63.
- [68] Gao J. *Isogeometric Topology Optimization: Methods, Applications and*

Implementations: Springer Nature; 2022.

- [69] Wang Y, Xu H, Pasini D. Multiscale isogeometric topology optimization for lattice materials. *Computer Methods in Applied Mechanics and Engineering*. 2017;316:568-85.
- [70] Gao J, Gao L, Luo Z, Li P. Isogeometric topology optimization for continuum structures using density distribution function. *International Journal for Numerical Methods in Engineering*. 2019;119:991-1017.
- [71] Xu J, Gao L, Xiao M, Gao J, Li H. Isogeometric topology optimization for rational design of ultra-lightweight architected materials. *International Journal of Mechanical Sciences*. 2020;166:105103.
- [72] Seo Y-D, Kim H-J, Youn S-K. Isogeometric topology optimization using trimmed spline surfaces. *Computer Methods in Applied Mechanics and Engineering*. 2010;199:3270-96.
- [73] Sigmund O, Maute K. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*. 2013;48:1031-55.
- [74] Drucker DC, Prager W. Soil mechanics and plastic analysis or limit design. *Quarterly of applied mathematics*. 1952;10:157-65.
- [75] Yang R, Chen C. Stress-based topology optimization. *Structural optimization*. 1996;12:98-105.
- [76] París J, Navarrina F, Colominas I, Casteleiro M. Topology optimization of continuum structures with local and global stress constraints. *Structural and Multidisciplinary Optimization*. 2009;39:419-37.
- [77] Le C, Norato J, Bruns T, Ha C, Tortorelli D. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*. 2010;41:605-20.
- [78] Holmberg E, Torstenfelt B, Klarbring A. Stress constrained topology optimization. *Structural and Multidisciplinary Optimization*. 2013;48:33-47.
- [79] Luo Y, Kang Z. Topology optimization of continuum structures with Drucker–Prager yield stress constraints. *Computers & Structures*. 2012;90:65-75.
- [80] Luo Y, Wang MY, Zhou M, Deng Z. Optimal topology design of steel–concrete composite structures under stiffness and strength constraints. *Computers & structures*. 2012;112:433-44.
- [81] Bruggi M, Duysinx P. Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization*. 2012;46:369-84.
- [82] Luo Y, Kang Z, Yue Z. Maximal stiffness design of two-material structures by topology optimization with nonprobabilistic reliability. *AIAA journal*. 2012;50:1993-2003.
- [83] Luo Y, Wang MY, Zhou M, Deng Z. Topology optimization of reinforced concrete structures considering control of shrinkage and strength failure. *Computers & Structures*. 2015;157:31-41.
- [84] Bruggi M, Duysinx P. A stress–based approach to the optimal design of structures with unilateral behavior of material or supports. *Structural and Multidisciplinary Optimization*. 2013;48:311-26.
- [85] Bruggi M, Taliercio A. Topology optimization of the fiber-reinforcement retrofitting existing structures. *International Journal of Solids and Structures*. 2013;50:121-36.
- [86] Jewett JL, Carstensen JV. Topology-optimized design, construction and experimental evaluation of concrete beams. *Automation in Construction*. 2019;102:59-67.

- [87] Bogomolny M, Amir O. Conceptual design of reinforced concrete structures using topology optimization with elastoplastic material modeling. *International journal for numerical methods in engineering*. 2012;90:1578-97.
- [88] Amir O. Stress-constrained continuum topology optimization: a new approach based on elasto-plasticity. *Structural and Multidisciplinary Optimization*. 2017;55:1797-818.
- [89] Pastore T, Mercuri V, Menna C, Asprone D, Festa P, Auricchio F. Topology optimization of stress-constrained structural elements using risk-factor approach. *Computers & Structures*. 2019;224:106104.
- [90] Kato J, Ramm E. Optimization of fiber geometry for fiber reinforced composites considering damage. *Finite Elements in Analysis and Design*. 2010;46:401-15.
- [91] Amir O, Sigmund O. Reinforcement layout design for concrete structures based on continuum damage and truss topology optimization. *Structural and Multidisciplinary Optimization*. 2013;47:157-74.
- [92] Amir O. A topology optimization procedure for reinforced concrete structures. *Computers & Structures*. 2013;114:46-58.
- [93] Zhang XS, Paulino GH, Ramos AS. Multi-material topology optimization with multiple volume constraints: a general approach applied to ground structures with material nonlinearity. *Structural and Multidisciplinary Optimization*. 2018;57:161-82.
- [94] Park J, Sutradhar A. A multi-resolution method for 3D multi-material topology optimization. *Computer Methods in Applied Mechanics and Engineering*. 2015;285:571-86.
- [95] Kato J, Lipka A, Ramm E. Multiphase material optimization for fiber reinforced composites with strain softening. *Structural and multidisciplinary optimization*. 2009;39:63-81.
- [96] Smarslik M, Mark P. Hybrid reinforcement design of longitudinal joints for segmental concrete linings. *Structural Concrete*. 2019;20:1926-40.
- [97] Smarslik M, Ahrens MA, Mark P. Toward holistic tension-or compression-biased structural designs using topology optimization. *Engineering Structures*. 2019;199:109632.
- [98] Gaganelis G, Jantos DR, Mark P, Junker P. Tension/compression anisotropy enhanced topology design. *Structural and Multidisciplinary Optimization*. 2019;59:2227-55.
- [99] Cai K. A simple approach to find optimal topology of a continuum with tension-only or compression-only material. *Structural and Multidisciplinary Optimization*. 2011;43:827-35.
- [100] Liang QQ, Xie YM, Steven GP. Topology optimization of strut-and-tie models in reinforced concrete structures using an evolutionary procedure: American Concrete Institute; 2000.
- [101] Kwak H-G, Noh S-H. Determination of strut-and-tie models using evolutionary structural optimization. *Engineering Structures*. 2006;28:1440-9.
- [102] Leu L-J, Huang C-W, Chen C-S, Liao Y-P. Strut-and-tie design methodology for three-dimensional reinforced concrete structures. *JOURNAL OF STRUCTURAL ENGINEERING-NEW YORK-*. 2006;132:929.
- [103] Lanes RM, Greco M, Guerra MBF. Strut-and-tie models for linear and nonlinear behavior of concrete based on topological evolutionary structure optimization (ESO). *Revista IBRACON de Estruturas e Materiais*. 2019;12:87-100.
- [104] Mezzina M, Palmisano F, Raffaele D. Designing simply supported RC bridge decks subjected to in-plane actions: Strut-and-tie model approach. *Journal of Earthquake*

Engineering. 2012;16:496-514.

[105] Hardjasaputra H. Evolutionary structural optimization as tool in finding strut-and-tie-models for designing reinforced concrete deep beam. *Procedia Engineering*. 2015;125:995-1000.

[106] Bruggi M. Generating strut-and-tie patterns for reinforced concrete structures using topology optimization. *Computers & Structures*. 2009;87:1483-95.

[107] Bruggi M. On the automatic generation of strut and tie patterns under multiple load cases with application to the aseismic design of concrete structures. *Advances in Structural Engineering*. 2010;13:1167-81.

[108] Bruggi M. A numerical method to generate optimal load paths in plain and reinforced concrete structures. *Computers & Structures*. 2016;170:26-36.

[109] Guest JK, Moen CD. Reinforced concrete design with topology optimization. 19th Analysis and Computation Specialty Conference 2010. p. 445-54.

[110] Victoria M, Querin OM, Martí P. Generation of strut-and-tie models by topology design using different material properties in tension and compression. *Structural and Multidisciplinary Optimization*. 2011;44:247-58.

[111] Xia Y, Langelaar M, Hendriks MA. A critical evaluation of topology optimization results for strut-and-tie modeling of reinforced concrete. *Computer-Aided Civil and Infrastructure Engineering*. 2020;35:850-69.

[112] Smarslik MJ. Optimization-based design of structural concrete using hybrid reinforcements: Ruhr-Universität Bochum; 2020.

[113] Francisco P, Faria L, Simões R. Multi-objective and multi-load topology optimization and experimental validation of homogenized coupled fluid flow and heat transfer and structural stiffness. *Structural and Multidisciplinary Optimization*. 2020;62:2571-98.

[114] Suresh K, Ramani A, Kaushik A. An adaptive weighting strategy for multi-load topology optimization. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference: American Society of Mechanical Engineers*; 2012. p. 1295-301.

[115] Li Y, Yang Q, Chang T, Qin T, Wu F. Multi-load cases topological optimization by weighted sum method based on load case severity degree and ideality. *Advances in Mechanical Engineering*. 2020;12:1687814020947510.

[116] Sokół T, Rozvany G. On the adaptive ground structure approach for multi-load truss topology optimization. 10th world congresses of structural and multidisciplinary optimization. 2013.

[117] Sokół T. Multi-load truss topology optimization using the adaptive ground structure approach. *Recent advances in computational mechanics*. 2015:9-16.

[118] Amir O, Shakour E. Simultaneous shape and topology optimization of prestressed concrete beams. *Structural and Multidisciplinary Optimization*. 2018;57:1831-43.

[119] Zhang Z, Yarlagadda T, Zheng Y, Jiang L, Usmani A. Isogeometric analysis-based design of post-tensioned concrete beam towards construction-oriented topology optimization. *Structural and Multidisciplinary Optimization*. 2021;64:4237-53.

[120] Zelickman Y, Amir O. Layout optimization of post-tensioned cables in concrete slabs. *Structural and multidisciplinary optimization*. 2021;63:1951-74.

[121] Vantighem G, Steeman M, Boel V, De Corte W. Multi-physics topology optimization

- for 3D-printed structures. Proceedings of IASS Annual Symposia: International Association for Shell and Spatial Structures (IASS); 2018. p. 1-8.
- [122] Vantighem G, De Corte W, Steeman M, Boel V. Density-based topology optimization for 3D-printable building structures. *Structural and Multidisciplinary Optimization*. 2019;60:2391-403.
- [123] Kirsch U. Optimum design of prestressed beams. *Computers & Structures*. 1972;2:573-83.
- [124] Quiroga AS, Arroyo MU. Optimization of prestressed concrete bridge decks. *Computers & structures*. 1991;41:553-9.
- [125] Lounis Z, Cohn M. Multiobjective optimization of prestressed concrete structures. *Journal of Structural Engineering*. 1993;119:794-808.
- [126] Lounis Z, Cohn MZ. Optimization of precast prestressed concrete bridge girder systems. *PCI Journal*. 1993;38:60-78.
- [127] Han S, Adamu A, Karihaloo B. Minimum cost design of multispan partially prestressed concrete beams using DCOC. *Engineering Optimization+ A35*. 1996;26:35-59.
- [128] Han S, Adamu A, Karihaloo B. Minimum cost design of multispan partially prestressed concrete T-beams using DCOC. *Structural optimization*. 1996;12:75-86.
- [129] Zhang Z, Yarlagadda T, Zheng Y, Jiang L, Usmani A. Isogeometric analysis-based design of post-tensioned concrete beam towards construction-oriented topology optimization. *Structural and Multidisciplinary Optimization*. 2021;64:4237-53.
- [130] Zhu M, Yang Y, Gaynor AT, Guest JK. Considering constructability in structural topology optimization. *Structures Congress 2014* 2014. p. 2754-64.
- [131] He L, Gilbert M. Rationalization of trusses generated via layout optimization. *Structural and Multidisciplinary Optimization*. 2015;52:677-94.
- [132] Torii AJ, Lopez RH, F Miguel LF. Design complexity control in truss optimization. *Structural and Multidisciplinary Optimization*. 2016;54:289-99.
- [133] Asadpoure A, Guest JK, Valdevit L. Incorporating fabrication cost into topology optimization of discrete structures and lattices. *Structural and Multidisciplinary Optimization*. 2015;51:385-96.
- [134] Kontovourkis O, Tryfonos G, Georgiou C. Robotic additive manufacturing (RAM) with clay using topology optimization principles for toolpath planning: the example of a building element. *Architectural Science Review*. 2020;63:105-18.
- [135] Søndergaard A, Amir O, Knauss M. Topology optimization and digital assembly of advanced space-frame structures. 2013.
- [136] Amir O, Mass Y. Topology optimization for staged construction. *Structural and Multidisciplinary Optimization*. 2018;57:1679-94.
- [137] Langelaar M. An additive manufacturing filter for topology optimization of print-ready designs. *Structural and multidisciplinary optimization*. 2017;55:871-83.
- [138] Gaynor AT, Guest JK. Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*. 2016;54:1157-72.
- [139] Allaire G, Dapogny C, Estevez R, Faure A, Michailidis G. Structural optimization under overhang constraints imposed by additive manufacturing technologies. *Journal of Computational Physics*. 2017;351:295-328.

- [140] Martens P, Mathot M, Bos F, Coenders J. Optimising 3D printed concrete structures using topology optimisation. *High Tech Concrete: Where Technology and Engineering Meet*: Springer; 2018. p. 301-9.
- [141] Wang W, Munro D, Wang CC, van Keulen F, Wu J. Space-time topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*. 2020;61:1-18.
- [142] Zuo Z. Topology optimization of periodic structures: RMIT University; 2009.
- [143] Thomas S, Li Q, Steven G. Finite periodic topology optimization with oriented unit-cells. *Structural and Multidisciplinary Optimization*. 2021;64:1765-79.
- [144] Jihong Z, Han Z, Chuang W, Lu Z, Shangqin Y, Zhang W. A review of topology optimization for additive manufacturing: Status and challenges. *Chinese Journal of Aeronautics*. 2021;34:91-110.
- [145] Morgan H, Cherry J, Jonnalagadda S, Ewing D, Sienz J. Part orientation optimisation for the additive layer manufacture of metal components. *The International Journal of Advanced Manufacturing Technology*. 2016;86:1679-87.
- [146] Johnson TE, Gaynor AT. Three-dimensional projection-based topology optimization for prescribed-angle self-supporting additively manufactured structures. *Additive Manufacturing*. 2018;24:667-86.
- [147] Qian X. Undercut and overhang angle control in topology optimization: a density gradient based integral approach. *International Journal for Numerical Methods in Engineering*. 2017;111:247-72.
- [148] Mass Y, Amir O. Topology optimization for additive manufacturing: Accounting for overhang limitations using a virtual skeleton. *Additive Manufacturing*. 2017;18:58-73.
- [149] Jared BH, Aguilo MA, Beghini LL, Boyce BL, Clark BW, Cook A et al. Additive manufacturing: Toward holistic design. *Scripta Materialia*. 2017;135:141-7.
- [150] Beghini A, Beghini LL, Baker WF. Applications of structural optimization in architectural design. *Structures Congress 2013: Bridging Your Passion with Your Profession 2013*. p. 2499-507.
- [151] Stromberg LL, Beghini A, Baker WF, Paulino GH. Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Structural and Multidisciplinary Optimization*. 2011;43:165-80.
- [152] Xie YM. Generalized topology optimization for architectural design. *Architectural Intelligence*. 2022;1:2.
- [153] Loos S, Wolk Svd, Graaf Nd, Hekkert P, Wu J. Towards intentional aesthetics within topology optimization by applying the principle of unity-in-variety. *Structural and Multidisciplinary Optimization*. 2022;65:185.
- [154] Jiang L, Usmani A. Towards scenario fires—modelling structural response to fire using an integrated computational tool. *Advances in structural engineering*. 2018;21:2056-67.
- [155] Jiang L, Usmani A. Computational performance of beam-column elements in modelling structural members subjected to localised fire. *Engineering Structures*. 2018;156:490-502.
- [156] Hughes TJ, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*. 2005;194:4135-95.

- [157] Jiang L, Jiang Y, Zhang Z, Usmani A. Thermal analysis infrastructure in OpenSees for fire and its smart application interface towards natural fire modelling. *Fire Technology*. 2021;1-26.
- [158] Cottrell J, Hughes T, Reali A. Studies of refinement and continuity in isogeometric structural analysis. *Computer methods in applied mechanics and engineering*. 2007;196:4160-83.
- [159] Rozvany GI. A critical review of established methods of structural topology optimization. *Structural and multidisciplinary optimization*. 2009;37:217-37.
- [160] Yang D, Liu H, Zhang W, Li S. Stress-constrained topology optimization based on maximum stress measures. *Computers & Structures*. 2018;198:23-39.
- [161] Moorman RB. Equivalent load method for analyzing prestressed concrete structures.: University of Missouri; 1952.
- [162] Costa G, Montemurro M, Pailhès J. Minimum length scale control in a NURBS-based SIMP method. *Computer Methods in Applied Mechanics and Engineering*. 2019;354:963-89.
- [163] Liu H, Yang D, Hao P, Zhu X. Isogeometric analysis based topology optimization design with global stress constraint. *Computer Methods in Applied Mechanics and Engineering*. 2018;342:625-52.
- [164] Cheng GD, Guo X. ϵ -relaxed approach in structural topology optimization. *Structural Optimization*. 1997;13:258-66.
- [165] Yang RJ, Chen CJ. Stress-based topology optimization. *Structural and Multidisciplinary Optimization*. 1996;12:98-105.
- [166] Duysinx P, Sigmund O. New developments in handling stress constraints in optimal material distribution. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization 1998.
- [167] Verbart A, Langelaar M, Keulen FV. A unified aggregation and relaxation approach for stress-constrained topology optimization. *Structural & Multidisciplinary Optimization*. 2017.
- [168] Amir O, Shakour E. Simultaneous shape and topology optimization of prestressed concrete beams. *Structural and Multidisciplinary Optimization*. 2018;57:1831-43.
- [169] Lazarov BS, Wang F, Sigmund O. Length scale and manufacturability in density-based topology optimization. *Archive of Applied Mechanics*. 2016;86:189-218.
- [170] SIMOGroup. SIMO-Package, a basic NURBS-based IsoGeometric Analysis package written in MATLAB. 2016.
- [171] Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*. 1987;24:359-73.
- [172] Lin TY, Burns NH. Design of prestressed concrete structures. 1981.
- [173] Zhang X, Xie YM, Zhou S. A nodal-based evolutionary optimization algorithm for frame structures. *Computer-Aided Civil and Infrastructure Engineering*. 2022.
- [174] Zienkiewicz OC, Taylor RL, Nithiarasu P, Zhu J. The finite element method: McGraw-hill London; 1977.
- [175] Hernandez CRB. Thinking parametric design: introducing parametric Gaudi. *Design Studies*. 2006;27:309-24.
- [176] Chang Z, Xu Y, Chen Y, Gan Y, Schlangen E, Šavija B. A discrete lattice model for

- assessment of buildability performance of 3D-printed concrete. *Computer-Aided Civil and Infrastructure Engineering*. 2021;36:638-55.
- [177] Lai Y, Zhang YJ, Liu L, Wei X, Fang E, Lua J. Integrating CAD with Abaqus: a practical isogeometric analysis software platform for industrial applications. *Computers & Mathematics with Applications*. 2017;74:1648-60.
- [178] Nguyen-Thanh N, Zhou K, Zhuang X, Areias P, Nguyen-Xuan H, Bazilevs Y et al. Isogeometric analysis of large-deformation thin shells using RHT-splines for multiple-patch coupling. *Computer Methods in Applied Mechanics and Engineering*. 2017;316:1157-78.
- [179] Nguyen VP, Anitescu C, Bordas SP, Rabczuk T. Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation*. 2015;117:89-116.
- [180] Wang Y, Wang Z, Xia Z, Poh LH. Structural design optimization using isogeometric analysis: a comprehensive review. *Computer Modeling in Engineering & Sciences*. 2018;117:455-507.
- [181] Benson D, Bazilevs Y, Hsu M-C, Hughes T. Isogeometric shell analysis: the Reissner–Mindlin shell. *Computer methods in applied mechanics and engineering*. 2010;199:276-89.
- [182] Gao J, Xiao M, Zhang Y, Gao L. A comprehensive review of isogeometric topology optimization: methods, applications and prospects. *Chinese Journal of Mechanical Engineering*. 2020;33:1-14.
- [183] Gao J, Wang L, Luo Z, Gao L. IgaTop: an implementation of topology optimization for structures using IGA in MATLAB. *Structural and Multidisciplinary Optimization*. 2021;64:1669-700.
- [184] McKenna F. OpenSees: a framework for earthquake engineering simulation. *Computing in Science & Engineering*. 2011;13:58-66.
- [185] Patzák B. OOFEM—an object-oriented simulation tool for advanced modeling of materials and structures. *Acta Polytechnica*. 2012;52.
- [186] Jiang J, Usmani A, Li G-Q. Modelling of steel-concrete composite structures in fire using OpenSees. *Advances in Structural Engineering*. 2014;17:249-64.
- [187] Jiang L, Usmani A. Computational performance of beam-column elements in modelling structural members subjected to localised fire. *Engineering Structures*. 2018;156:490-502.
- [188] Jiang L, Usmani A. Towards scenario fires—modelling structural response to fire using an integrated computational tool. *Advances in structural engineering*. 2018;21:2056-67.
- [189] Ousterhout JK. Tcl: An embeddable command language: Citeseer; 1989.
- [190] Zhu M, Scott MH. Modeling fluid–structure interaction by the particle finite element method in OpenSees. *Computers & Structures*. 2014;132:12-21.
- [191] Lu X, Xie L, Guan H, Huang Y, Lu X. A shear wall element for nonlinear seismic analysis of super-tall buildings using OpenSees. *Finite Elements in Analysis and Design*. 2015;98:14-25.
- [192] Noh NM, Liberatore L, Mollaioli F, Tesfamariam S. Modelling of masonry infilled RC frames subjected to cyclic loads: State of the art review and modelling with OpenSees. *Engineering Structures*. 2017;150:599-621.
- [193] Fuentes-Fernández L, Vallecillo-Moreno A. An introduction to UML profiles. *UML*

and Model Engineering. 2004;2:72.

[194] Scott MH, Kidarsa A, Higgins C. Development of bridge rating applications using OpenSees and Tcl. *Journal of Computing in Civil Engineering*. 2008;22:264-71.

[195] Zhu M, McKenna F, Scott MH. OpenSeesPy: Python library for the OpenSees finite element framework. *SoftwareX*. 2018;7:6-11.

[196] SIMOGroup. SIMO-Package, a basic NUBS-based IsoGeometric Analysis package written in MATLAB. 2016.

[197] Abaqus G. Abaqus 6.11. Dassault Systemes Simulia Corporation, Providence, RI, USA. 2011.

[198] Sigmund O, Maute K. Topology optimization approaches. *Structural and Multidisciplinary Optimization*. 2013;48:1031-55.

[199] Lazarov BS, Wang F, Sigmund O. Length scale and manufacturability in density-based topology optimization. *Archive of Applied Mechanics*. 2016;86:189-218.

[200] Verbart A, Langelaar M, Van Keulen F. A unified aggregation and relaxation approach for stress-constrained topology optimization. *Structural and Multidisciplinary Optimization*. 2017;55:663-79.

[201] Liu J, Gaynor AT, Chen S, Kang Z, Suresh K, Takezawa A et al. Current and future trends in topology optimization for additive manufacturing. *Structural and multidisciplinary optimization*. 2018;57:2457-83.

[202] Allaire G, Geoffroy-Donders P, Pantz O. Topology optimization of modulated and oriented periodic microstructures by the homogenization method. *Computers & Mathematics with Applications*. 2019;78:2197-229.

[203] Thomas S, Li Q, Steven G. Topology optimization for periodic multi-component structures with stiffness and frequency criteria. *Structural and Multidisciplinary Optimization*. 2020;61:2271-89.

[204] Huang X, Xie M. *Evolutionary topology optimization of continuum structures: methods and applications*: John Wiley & Sons; 2010.

[205] Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. *Computers & structures*. 1993;49:885-96.

[206] Kumar T, Suresh K. A density-and-strain-based K-clustering approach to microstructural topology optimization. *Structural and Multidisciplinary Optimization*. 2020;61:1399-415.

[207] Ryu N, Seo M, Min S. Multi-objective topology optimization incorporating an adaptive weighed-sum method and a configuration-based clustering scheme. *Computer Methods in Applied Mechanics and Engineering*. 2021;385:114015.

[208] Zhang C, Xu S, Liu J, Ma Y. Comprehensive clustering-based topology optimization for connectable multi-scale additive manufacturing structures. *Additive Manufacturing*. 2022;54:102786.

[209] MacQueen J. Classification and analysis of multivariate observations. *5th Berkeley Symp Math Statist Probability*1967. p. 281-97.

[210] Steinley D. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*. 2006;59:1-34.

[211] Zhang Z, Jiang L, Yarlagadda T, Zheng Y, Usmani A. OPS-ITO: Development of Isogeometric Analysis and Topology Optimization in OpenSEES for Free-Form Structural

Design. *Computer-Aided Design*. 2023;160:103517.

[212] Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*. 1987;24:359-73.