A COMPARATIVE STUDY ON COMPUTATIONAL METHODS FOR DETERMINING STRUT AND TIE MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CAN ALBAYRAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

APRIL 2025

Approval of the thesis:

A COMPARATIVE STUDY ON COMPUTATIONAL METHODS FOR DETERMINING STRUT AND TIE MODEL

submitted by **CAN ALBAYRAK** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Civil Engineering, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, **Graduate School of Natural and Applied Sciences** _____

Prof. Dr. Erdem Canbay
Head of the Department, **Civil Engineering** _____

Prof. Dr. Özgür Kurç
Supervisor, **Civil Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Yalın Arıcı
Civil Engineering, METU _____

Prof. Dr. Özgür Kurç
Civil Engineering, METU _____

Prof. Dr. Kağan Tuncay
Civil Engineering, METU _____

Prof. Dr. Afşin Sarıtaş
Civil Engineering, METU _____

Asst. Prof. Berat Feyza Soysal Albostan
Civil Engineering, Cankaya University _____

Date: 08.04.2025

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Can Albayrak

Signature :

ABSTRACT

A COMPARATIVE STUDY ON COMPUTATIONAL METHODS FOR DETERMINING STRUT AND TIE MODEL

Albayrak, Can
Master of Science, Civil Engineering
Supervisor: Prof. Dr. Özgür Kurç

April 2025, 199 pages

The Strut-and-Tie Model is used in the shear design of deep and short reinforced concrete members where traditional design methods are not applicable. A Strut and Tie Model is a truss analogy representing a reinforced concrete member's internal force flow. This method requires the engineer to have foresight into the internal flow of forces. Over time, many optimization methods have been suggested to determine the internal flow of forces within a reinforced concrete member. The Evolutionary Structural Optimization method aims to find the optimum material distribution by gradually and systematically eliminating relatively inadequate elements from the reinforced concrete member for a predefined set of external loads and boundary conditions. This method involves modeling the reinforced concrete member with a mesh of finite elements. In this study, the Evolutionary Structural Optimization method is utilized to find the optimum finite element layout of several examples– which can show the internal force flow within a reinforced concrete member. Three different analytical elements are used to define each member example: truss elements, membrane elements, and made-up brick elements comprising six truss elements. The resulting optimized element layout of the example models using three

analytical elements is examined. The performance of the three analytical elements is compared in terms of their results, as well as their robustness, effectiveness, and feasibility. Overall, the brick element is observed to give the most consistent outcomes based on the determined criteria.

Keywords: Strut-and-Tie Model, Internal Force Flow, Evolutionary Structural Optimization, Analytical Elements.

ÖZ

ÇUBUK ANALOJİSİ MODELİNİN BELİRLENMESİNE YÖNELİK HESAPLAMA YÖNTEMLERİ ÜZERİNE KARŞILAŞTIRMALI BİR ÇALIŞMA

Albayrak, Can
Yüksek Lisans, İnşaat Mühendisliği
Tez Yöneticisi: Prof. Dr. Özgür Kurç

Nisan 2025, 199 sayfa

Çubuk Analojisi Modeli, geleneksel tasarım yöntemlerinin uygulanamadığı derin ve kısa betonarme elemanların kesme tasarımı için kullanılmaktadır. Bir Çubuk Analojisi Modeli, betonarme bir elemanın iç kuvvet akışını temsil eder. Bu yöntem, mühendisin iç kuvvetlerin akışına ilişkin bir öngörüye sahip olmasını gerektirir. Zaman içinde, bir betonarme elemanı içindeki iç kuvvet akışını belirlemek amacıyla birçok optimizasyon yöntemi önerilmiştir. Evrimsel Yapısal Optimizasyon yöntemi, önceden tanımlanmış bir dizi dış yük ve sınır koşulu için, betonarme yapıdan nispeten yetersiz elemanları kademeli ve sistematik bir şekilde eleyerek optimum malzeme dağılımını bulmayı amaçlar. Bu yöntemde betonarme yapı, sonlu elemanlar metodu kullanılarak modellenir. Bu çalışmada, Evrimsel Yapısal Optimizasyon yöntemi, kullanılan örneklerin optimum sonlu eleman düzenini bulmak için kullanılmıştır. Optimizasyon sonucu bulunan sonlu eleman düzeni iç kuvvet akışını gösterebilir. Her bir örnek betonarme yapıyı tanımlamak için üç farklı analitik eleman kullanılmıştır: çubuk eleman, membran eleman ve altı çubuk elemandan oluşan tuğla eleman. Üç analitik eleman kullanılarak oluşturulan örnek modellerin

optimize edilmiş eleman düzeni incelenmiştir. Sonuçlar doğrultusunda, üç analitik elemanın performansı karşılaştırılmıştır. Karşılaştırma; dayanıklılık, etkinlik ve uygulanabilirlik kriterlerine göre yapılmıştır. Belirlenen kriterlere göre, tuğla elemanın en tutarlı sonuçları verdiği gözlemlenmiştir.

Anahtar Kelimeler: Çubuk Analojisi Modeli, İç Kuvvet Akışı, Evrimsel Yapı Optimizasyon, Analitik Elemanlar.

To My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Özgür Kurç, for his invaluable guidance and insightful feedback throughout this research journey. His expertise was instrumental in shaping this thesis.

On a personal note, I am grateful for the lifelong, unconditional love and support of my parents, Emine and Osman Albayrak. I am also thankful to my girlfriend, Belya Lüleci, for her support and understanding during the challenging times of this endeavor. Finally, I would like to thank my friends and colleagues for their help during this process.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The Strut-and-Tie Modeling method is incorporated in special reinforced concrete members where traditional sectional design methods are not applicable. The classical beam theory relies on the Euler-Bernoulli hypothesis, which states that the plane sections remain plane. The classical plane section assumption of flexural theory, however, is not valid for every type of member. These special circumstances occur when a member's strain distribution in its cross-section is substantially nonlinear due to concentrated loadings and geometrical discontinuities. The traditional design methods also assume that the shear stresses are uniform at the cross-section of a member. This is also not applicable for deep and short members. Many structural components employed in bridge construction, for example, such as pier caps, corbels, beams with holes, deep footings, pile caps, etc.; require an alternative approach in which the internal force flow is being considered in more detail.

In a nutshell, the Strut-and-Tie Model is a truss analogy that represents the internal force flow within a structural member. It is a truss model within the reinforced concrete member that can successfully transfer the applied loads to the supports. The compressive truss elements represent the concrete and are called struts. The tensile truss elements represent the steel reinforcement and are called ties. The Strut-and-Tie Model denotes the concrete proportions and the steel reinforcement placement in the member.

The Strut-and-Tie Modelling method requires the engineer to have foresight into the internal force flow. A Strut-and-Tie Model can easily be proposed for simple models where the stress paths are straightforward and hence very easy for the engineer to guess. However, as the number of concentrated forces, supports and the complexity of the geometry increases, the internal force flow of the structure becomes harder to

predict. Hence, many methods have been suggested over time to predict the internal force flow of any given reinforced concrete member.

The design of these special structural components mentioned above requires a step-by-step application of a series of different methods. Hence, possible important milestones to be achieved in the future in this vast research area of Strut-and-Tie Modelling can be listed as follows: A special type of analysis to forecast/envision the stress path within the described domain, an optimization method to determine the internal force flow within the described domain, a composite tool to build a truss model within the described domain, and finally a complete design tool that can design the structure.

Although many different structural optimization methods have been suggested and developed, not one method has been agreed upon to be favored among the many others, and the research in the area remains to be scattered. Because of this, these optimization methods remain to only offer merely an inspiration of what the internal force flow might look like.

Current optimization methods used in preliminary strut-and-tie model design lack comprehensive comparison, especially in terms of how different analytical elements perform within a chosen framework. While the Evolutionary Structural Optimization method gained traction within the context of the Strut-and-Tie Method, there is little research comparing the performance of different finite elements. Without a clear understanding of which elements perform best in Evolutionary Structural Optimization, engineers may choose inefficient methods for designing load paths, leading to suboptimal material use. Therefore, a comparative study on every aspect of these methods must be performed on a large scale to essentially favor a single method to move forward.

In this thesis, the Evolutionary Structural Optimization method is studied. The method aims to find the optimum distribution of material layout by gradually decreasing the number of analytical elements in a finite element model of a reinforced concrete member for a predefined set of external loads and boundary

conditions. This iterative method involves modeling the reinforced concrete member with a mesh of finite elements. The relatively inadequate analytical elements are eliminated gradually and systematically from the model step-by-step.

1.1 Objectives and Scope

The objective of this study is to compare the performances of three different finite elements within the Evolutionary Structural Optimization method. It is aimed to determine the best type of finite element for detecting the internal force flow within a selected reinforced concrete member and, ultimately, for designing a suitable Strut-and-Tie model in reinforced concrete structures. This study aims to help guide future research on Strut-and-Tie Modeling and provide a strong base for further development.

The Structural Evolutionary Optimization method is utilized to find the optimum material layout of a number of examples– which can show the internal force flow within the objective structural component. Three different analytical elements are used to define the reinforced concrete member: a truss element, a membrane element, and a made-up brick element that consists of six truss elements.

This thesis focuses on the performance of these three analytical elements in a stress-based evolutionary structural optimization. The resulting optimized material layout of the example models using three analytical elements is examined. The Evolutionary Structural Optimization method was selected due to its simplicity, common usage, and compatibility with the Strut-and-Tie method.

The main focus is on the design of reinforced concrete superstructure elements of a bridge structure, such as pier caps. Therefore, this study will be limited to two-dimensional reinforced concrete members. Likewise, the optimization process will be based on a single static load case applied to a two-dimensional reinforced concrete structure. The study will not cover nonlinear material properties or three-dimensional model geometry.

This study adopts the direct element elimination method. In this approach, redundant elements are systematically removed from the system in each iteration until a steady state is achieved. Direct elimination was chosen for its simplicity and efficiency in reducing material density. It is expected to accelerate the optimization process, enabling the structure to reach its optimum state more quickly.

Two sets of reinforced concrete member examples are incorporated to test the candidate finite elements. The first set of examples consists of arbitrary rectangular shapes and is used for the preliminary evaluation and comparison of the candidate finite elements. The second set of examples consists of various trapezoidal pier caps, which are used to assess the candidate finite elements based on their performance with real-life examples. Gmsh (Geuzaine & Remacle, 2024) is used to discretize these examples into quadrilateral finite elements. All examples are discretized into three levels of mesh refinement: coarse, medium and fine. Additionally, two distinct meshing strategies are applied for the second set of examples across all mesh refinements. The connectivity information is exported to a MATLAB Script.

Key data of each example, including material and geometric properties, Gmsh scripts, and boundary conditions are systematically organized in an Excel spreadsheet. An Excel VBA macro is utilized to export the properties of the selected example into a MATLAB Script for analysis and optimization procedures. Finally, these exported scripts are all combined within one of three MATLAB Projects, specific to each candidate finite element, where the Evolutionary Structural Optimization takes place, and the results are saved.

CHAPTER 2

LITERATURE REVIEW

This section denotes the key sources reviewed during the preparation of this thesis. Along with research literature, design codes, reports and textbooks were examined to develop a comprehensive understanding of the Strut-and-Tie Model (STM) methodology and its applications in bridge engineering. The literature review also explores optimization methods used for designing Strut-and-Tie Models, focusing on identifying the most suitable approaches for comparing finite elements. These studies provide the foundation for establishing the framework of this comparative study.

The review begins with an overview of the Strut-and-Tie Model methodology and its fundamental principles. It then transitions to its specific applications in bridge components, highlighting how STM has been utilized in bridge design. Finally, the discussion focuses on optimization methods for deriving Strut-and-Tie Models in bridge components, examining the variations in element elimination strategies, selection criteria, and methodological approaches. This structure is planned to progress from general concepts to specific techniques.

2.1 Strut-and-Tie Modeling

The Strut-and-Tie Model is a procedure to design a reinforced concrete structure where concentrated forces are applied, or geometric discontinuities occur. (American Association of State Highway and Transportation Officials, 2023, Section 5). Traditional design methods use the Euler-Bernoulli hypothesis, which states that plane sections remain plane. This assumption, however, is not valid for structural members that have nonlinear strain distribution at its cross-section due to geometrical discontinuities and concentrated loads in the member. Hence, the Strut-

and-Tie Model method is proposed as an alternative to classical sectional design approach for the design of such structural components. According to the AASHTO (2023, Section 5), a structural element's actual flow of forces at regions near discontinuities should be examined in more detail, using a Strut-and-Tie Model.

A Strut-and-Tie Model is essentially a truss analogy to represent complex stress patterns within a reinforced concrete structure. A truss model is built within the reinforced concrete member. The model transfers the external loads applied on the reinforced concrete structure to the supports.

The use of truss models for reinforced concrete was first suggested by Ritter (1899) for the shear design of reinforced concrete structures. Ritter's model was expanded by Mörsch (1909) to cover the design of beams under torsion. This truss analogy method has been significantly improved by various researchers (Kupfer 1964; Leonhardt 1965; Lampert and Thurlimann 1971) throughout the 60s and 70s. A truss model that can be used for the design of prestressed reinforced concrete structures was suggested by Collins and Mitchell (1980).

At this point, it should be noted that the truss model was only applicable for the design of regions of a reinforced concrete structure where the Bernoulli hypothesis is valid. It was put forward by Marti (1985) that the Strut-and-Tie Model, which was a generalization of the truss analogy method, can also be used for the design of disturbed regions of a reinforced concrete structure. Finally, the truss analogy method was extended by Schlaich et al. (1987) to offer a consistent Strut-and-Tie Model approach for the complete design of reinforced concrete structures. The Strut-and-Tie Model was used by Ramirez (1994) for the design of pretensioned reinforced concrete structures. The Strut-and-Tie Model was utilized by Sanders and Breen (1997) for the design of post-tensioned anchorage zones.

A Strut-and-Tie Model has three components: struts, ties, and nodes. Truss elements that undergo compressive stresses are called struts, the elements that are in tensile stresses are ties, and the region at the intersection of the struts and ties are denoted

as nodes. The struts represent the concrete material which is strong in compression. The ties represent the reinforcement steel which supports the tensile internal forces.

Schlaich et al. (1987) observed that most reinforced concrete structural components transfer the external loads through compressive stress areas. These compressive areas are linked by tensile stress areas. These tensile fields can be reinforcing bars, prestressing tendons, or concrete tensile stress fields. This observation validates the Strut-and-Tie Model method. What the Strut-and-Tie model does is simplify these compression and tensile fields within the reinforced concrete structure, by concentrating them into discrete members.

In a nutshell, the Strut-and-Tie Model shows the stress patterns of a structure in a truss system of struts, ties, and nodes. Kwak and Noh (2006) further states that a Strut-and-Tie model should effectively describe the behavior of a structure.

A strut-and-tie design workflow begins with defining the geometry of the reinforced concrete member. The second step would be to determine the locations of the B and D regions (Colorito et al. 2017). The B-region of a structure is where Bernoulli's plane sections remain plane hypotheses can be assumed valid (AASHTO, 2023, Section 5.7.1.1). The B stands for beam or Bernoulli. These regions are often designed with extreme care and accuracy by using the sectional forces to extract the internal state of stresses (Schlaich et al. 1987). The D-region denote the part of a structure where the traditional design methods are not applicable. D-regions are located where the strain distributions are nonlinear. The D stands for discontinuity (Schlaich et al. 1987).

2.2 Bridge Components

D-regions usually occur near concentrated loads, corners, bends, and openings. These types of geometric discontinuities and concentrated loadings are spotted in structural bridge components. Therefore, the Sturt-and-Tie Model method is of high importance in bridge engineering.

AASHTO (2023, Section 5) states that the Strut-and-Tie Model approach should be incorporated for the design of thick footings, box culverts, pile caps or deep pier cap beams (AASHTO, 2023, Section 5). Different bridge components require different approaches to achieve a consistent strut-and-tie model. AASHTO (2023, Section 5) states that three dimensional Strut-and-Tie Model should be used when a simultaneous shear in a second direction should be considered in a reinforced concrete structure. These type of bridge components include prestressed interior diaphragm of a box girder bridge, pier diaphragms, corbels and beam ledges, brackets and corbels, and pile caps. A pile cap, for example, is a thick concrete slab to distribute loads from columns, piers, or bridge abutments to a group of piles driven into the ground.

A three-dimensional strut-and-tie method to design and verify four-pile caps was presented by Souza et al. (2009). A displacement-based three-dimensional Compatibility Strut-and-Tie Method (C-STM) to model the behavior of shear-critical four-pile caps was proposed by Abdul-Razzaq and Farhood (2017). A three-dimensional compatibility strut-and-tie method to model the behavior of four-pile caps was proposed by Dey and Karthik (2019).

These studies show that the Strut-and-Tie Model is utilized across various types of bridge components. They highlight the vast applicability of this method and its versatility in addressing different areas of design in bridge construction. Three-dimensional Strut-and-Tie Models are mostly preferred when designing reinforced concrete pier cap structures, where the width is significantly larger than height.

In this study, however, the pier caps and pier cap like structures, such as rectangle beams, and rectangle beams with openings are more focused. These structural components can be sufficiently idealized to two-dimensional reinforced concrete members. Many studies have used a two-dimensional strut-and-tie analogy to examine pier caps.

AASHTO (2023, Section 5) states that the Strut-and-Tie Model should be incorporated for the design of structural members where the distance between a

concentrated external load location and the supports is less than twice as much the member depth. The STM should be considered for the design of deep footings and pile caps or other situations in which the distance between the centers of applied load and the supporting reactions is less than two times the member depth.

This condition is likely to be encountered in deep beams. The depth of a deep beam is significantly higher compared to its span length. Deep beams exceed the limit of the traditional beam theory since their shear distribution is not uniform over their depth. Deep beams are often come across in structures like transfer girders, bents, and pier caps. AASHTO (2023, Section 5) underlines that the traditional methods cannot be used for the design of deep beams, hence the Strut-and-Tie Model methos should be incorporated instead.

This study focuses on the performance of deciding a Strut-and-Tie Model for pier caps using different optimization methods. The example models used for the evaluation of alternate optimization methods are rectangular and hammerhead-shaped pier caps. Studies that demonstrate the application of Strut-and-Tie Modelling in the design of reinforced concrete pier caps are summarized below. They highlight the importance of this approach for bridge design.

The Strut-and-Tie Model method for designing a statically indeterminate deep beam with openings was experimentally evaluated by Garber et al. (2014). Simplified Strut-and-Tie Model provisions and a guidebook to support the design of precast bridge components such as hammerhead-shaped pier and bent caps were developed by Williams et al. (2014). A Strut-and-Tie Model for estimating the ultimate shear strength of the reinforced concrete hammerhead pier cap was proposed by Vinayak et al. (2016). A specialized strut-and-tie methodology, along with a graphical solution algorithm and computer code to simplify and improve the accuracy of shear strength predictions for deep pier caps, was developed by Baniya and Guner (2019). The Strut-and-Tie Evaluation Program to simplify and support the design of rectangular and hammerhead bent caps was developed by Vicksman et al. (2020). Reinforced concrete hammerhead pier caps with varying shear span-to-depth ratios

were tested and a modified STM for improved capacity estimation is proposed by Dawood and Abdul-Razzaq (2021). The structural performance of hammerhead piers using Strut-and-Tie methods was examined by Wang et al. (2024).

Most of these studies focus solely on calculating the thickness of an already established strut and tie elements. They do not offer any optimization approach or an algorithm to initially determine the truss system. Baniya and Guner (2019) offers an optimization approach on a rectangular pier cap example. This method is not used for determining the Strut-and-Tie Model. It is a tool developed for post Strut-and-Tie Model and used to determine the reinforcement.

These studies mainly focus on rectangular shapes, with limited exploration of hammerhead-shaped pier caps. The Strut-and-Tie Models proposed for hammerhead pier caps are often standard truss models without alternative designs. Garber et al. (2014) uses a basic FEA model to identify high-stress regions and validate the Strut-and-Tie Models. In most other studies on hammerhead pier caps, the chosen models rely on the design engineer's judgment rather than a systematic optimization process. The design of such a bridge component is not taken into consideration within the domains of an optimization problem.

2.3 Topology Optimization

In a nutshell, most structural bridge components (such as pier caps) have nonlinear distribution of shear stresses and longitudinal strains over their depth. Hence, constant values of factored sectional forces cannot be used for the design of such reinforced concrete structures. The force flow within these members is to be examined in more detail using a Strut-and-Tie Model.

Building a Strut-and-Tie Model is straightforward for simply shaped reinforced concrete members where the internal flow of forces can be easily visualized. However, as the geometry of a reinforced concrete member gets more complicated,

and the number of external loads increases, the force flow within a reinforced concrete member can get too complicated to be guessed by a design engineer.

Load path methods based on elastic stress distributions were used for the design of Strut-and-Tie Models (Kwak and Noh, 2006; Liang et al., 2000). These methods are examined in detail by Marti (1985) and Schlaich et al. (1987). A linear elastic finite element analysis is performed on the reinforced concrete structure. The internal forces and stresses of the structure are extracted and illustrated as trajectories. These trajectory patterns show the path of the forces from the loaded edges through the structure to the supports. The Strut-and-Tie Model is then built by aligning the struts and ties with these stress trajectories. Kwak and Noh (2006) and Liang et al., (2000) point out that these patterns are often intricate, especially for concrete members with complex loading conditions and geometries, requiring further simplification to ensure practical applicability.

As a result of this, topology optimization procedures have been commonly utilized to refine the design domain and extract a distinctive internal force path. Topology optimization aims to find the most efficient material distribution of a design domain while satisfying predetermined objectives and specific constraints. Various optimization approaches were utilized by researchers and engineers for the purpose of designing Strut-and-Tie Model.

The homogenization-based optimization was examined by Bendsøe and Kikuchi (1988), Suzuki and Kikuchi (1991), Díaz and Bendsøe (1992), Díaz and Kikuchi (1992), Tenek and Hagiwara (1993), Bendsøe et al. (1995), Ma et al. (1995), and Krog and Olhoff (1999). This method optimizes a continuum design domain by redistributing composite material with microstructure.

The density function approach was studied by Mlejnek and Schirrmacher (1993) and Yang and Chuang (1994). This approach is based on shifting the density of the materials in the design domain to achieve an efficient distribution.

These methods were primarily used in mechanical engineering. Their implementation is complex and often applied to disks, plates, or small mechanical parts with anisotropic material properties. The layouts derived from material distribution at the end of the optimization process are not suitable for producing a Strut-and-Tie Model.

A simpler approach, such as Evolutionary Structural Optimization, has been found sufficient and more appropriate for the design of reinforced concrete structures using the Strut-and-Tie Model. This method is straightforward and produces refined results that closely resemble a Strut-and-Tie Model. Since the final design will be Strut-and-Tie Model comprised of truss elements, a more direct method is understandably preferred. The Finite Element Analysis Method is utilized for this optimization type. The material distribution of the structure is represented by finite elements. Redundant finite elements are gradually removed from the reinforced concrete member during the optimization process.

There are many variations of Evolutionary Structural Optimization approaches in the design field. These methods are grouped based on the element elimination method, element elimination criterion, and the type of finite elements used in the procedure.

2.3.1 Evolutionary Structural Optimization: Elimination Method

In Evolutionary Structural Optimization, the element removal method can be examined under two headings: Hard-kill and soft-kill approaches. The Hard-kill approach directly removes redundant elements from the design domain at each iteration of the optimization procedure. The basis of the Evolutionary Structural Optimization method was established by Rodriguez and Seireg (1985), Atrek (1989), Rozvany et al. (1992), and Xie and Steven (1993). Alternative ESO methods that are proposed over time are discussed in the following sections.

The soft-kill method, on the other hand, gradually removes underutilized elements throughout the iterations, rather than directly deleting it. This method is a variation

of the classical Evolutionary Structural Optimization and is also called Smooth Evolutionary Structural Optimization.

The early studies on the soft-kill method were conducted by Baumgartner et al. (1992). This approach was formulated by Mattheck (1997) for the solution of a stressed design domain. Smooth Evolutionary Structural Optimization was conducted by Li et al. (2000) by gradually decreasing the thickness of the underutilized elements in the reinforced concrete member. The classical and smooth Evolutionary Structural Optimization methods were compared by Almeida et al. (2013) by decreasing the stiffness of underutilized elements.

Another variant approach is the Bidirectional Evolutionary Structural Optimization. This method introduces the concept of re-adding elements back into the design domain. The Bidirectional Evolutionary Structural Optimization can utilize both soft-kill and hard-kill approaches.

Adding elements to certain regions of the design domain for the purpose of alleviating high stresses were initially proposed to be added within Evolutionary Structural Optimization by Querin (1997). Comparative studies between the BESO and ESO methods were conducted by Querin et al. (1998) and Yang et al. (1999).

2.3.2 Evolutionary Structural Optimization: Elimination Criterion

An Evolutionary Structural Optimization, where redundant plane stress elements are gradually removed from a design domain according to their Von Mises stresses, was proposed by Xie and Steven (1993). Xie and Steven (1997) further stated that an ideal material distribution of a structure can be determined at a configuration where stress in every part is relatively at the same level.

Von Mises stresses were used by the authors mentioned above (Querin et al., 1998; Yang et al., 1999; Li et al., 2000; Almeida et al., 2013) as criteria for determining underutilized elements. An optimization-based strut-and-tie model for deep beams with web openings using element stresses was developed by Guan (2005). Stress

constraints on reinforced concrete members were utilized by Liang et al. (1999) for Evolutionary Structural Optimization.

Zhao et al. (1998) suggested that using the element displacements as the removal criterion instead of element stresses could be a better choice, considering that stresses are actually derived from displacements and are usually less accurate in a typical finite element analysis. This deduction is also backed by the fact that stresses are not required to be continuous in displacement based finite element formulations.

Element sensitivity numbers to identify redundant elements were suggested by Chu et al. (1996) and Liang et al. (2000). This number is calculated according to the effect of an element's removal on the displacement of a constrained node. Element sensitivity numbers were also used as element removal criteria by Liang et al. (2000) and Liang et al. (2002), along with a performance index parameter for selecting the most efficient element distribution. Elements with the least virtual strain energy during the iterations were proposed to be removed by Kwak and Noh (2006).

2.3.3 Evolutionary Structural Optimization: Analytical Elements

Quadrilateral plane stress membrane element is the most commonly utilized analytical element. This element has been consistently employed in all the studies referenced above. The common approach to enhance the capability of a Strut-and-Tie Model tool is to build upon the existing method that uses the classical plane stress element. By implementing new functions, constraints and new algorithms, the defects of the plane stress element are sought to be eliminated. The results are also expected to be improved.

It is suggested by recent studies that there is a trend in searching for an alternative finite element for the optimization problem. A new element, a brick element comprised of six truss elements, was suggested by Kwak and Noh (2006). They stated that since the resulting element distribution of the structural optimization of a model resembles a truss-like structure, it is only appropriate to use truss elements to

discretize the reinforced concrete member. A triangular plane-stress element with higher-order modes was utilized by Almeida et al. (2013). A micro-truss element with 8 nodes was proposed by Zhong et al. (2017).

Although Evolutionary Structural Optimization is widely used to determine Strut-and-Tie Models for bridge component design, a standardized approach has yet to be established. This is due to the various Evolutionary Structural Optimization methods available, each with its own distinct approach.

There is a clear search for a new direction, as many new methods have been proposed to improve the Evolutionary Structural Optimization methos. These suggestions mainly focus on either introducing alternative finite elements for discretizing reinforced concrete structures or selecting different elimination criteria to achieve the minimization objective. However, an extensive study comparing these proposed methods is lacking.

In this study, the most promising Evolutionary Structural Optimization variants are selected for evaluation. Various methods are analyzed and categorized into two groups: analytical elements and elimination criteria. After a comprehensive literature review based on their popularity and reported success, candidate methods for comparison in this thesis are identified. The stress-based and displacement-based elimination criteria using brick and membrane elements are examined. Additionally, as a novel approach to Evolutionary Structural Optimization, the use of truss elements is proposed. These methods are compared based on their performance in optimizing reinforced concrete pier cap structures, aiming to determine the most effective Evolutionary Structural Optimization approach for this application.

Furthermore, it is observed that most previous studies focus solely on rectangular pier caps, neglecting hammerhead shapes. No existing research explores two-dimensional finite element models of hammerhead pier caps for use in an Evolutionary Structural Optimization procedure. This study addresses these gaps by proposing different meshing strategies for discretizing hammerhead pier caps for the optimization process. This way, the effects of mesh refinement and meshing

approaches on various Evolutionary Structural Optimization methods are also examined. These approaches provide a framework for developing alternative two-dimensional finite element analysis models of reinforced concrete pier caps and incorporating different Evolutionary Structural Optimization methods, which have not been explored in previous research to this extent.

This study aims to consolidate research in this field, which is currently fragmented across various topics, and establish a more structured and a certain approach. By doing so, it is hoped to lay the groundwork for future advancements, such as the development of a fully automated design tool.

CHAPTER 3

METHODOLOGY AND IMPLEMENTATION

Different approaches to Evolutionary Structural Optimization are examined and compared in this comprehensive study. These approaches are the use of three different analytical elements within an optimization procedure, by using two different element elimination criteria.

This chapter explains the steps of the Evolutionary Structural Optimization of a reinforced concrete member. Meanwhile, the computer code implementation is thoroughly explained.

Three software packages were employed to facilitate each step of the workflow. The process began with geometry generation and mesh discretization at various densities using Gmsh (Geuzaine & Remacle, 2024). In the second step, boundary conditions for the reinforced concrete member and material properties of the reinforced concrete were defined using Excel VBA. Finally, the process concluded with finite element analysis and Evolutionary Structural Optimization performed in MATLAB. At the end of each iteration, the current state of the example was reported to track the optimization progress.

3.1 Workflow

The stress-based and displacement-based evolutionary structural optimization methods are incorporated for the comparison of three analytical elements. These elements are Membrane, Truss, and Brick. The flowchart diagram for the stress-based and displacement-based evolutionary structural optimization procedures are given in Figure 3.1 and 3.2, respectively.

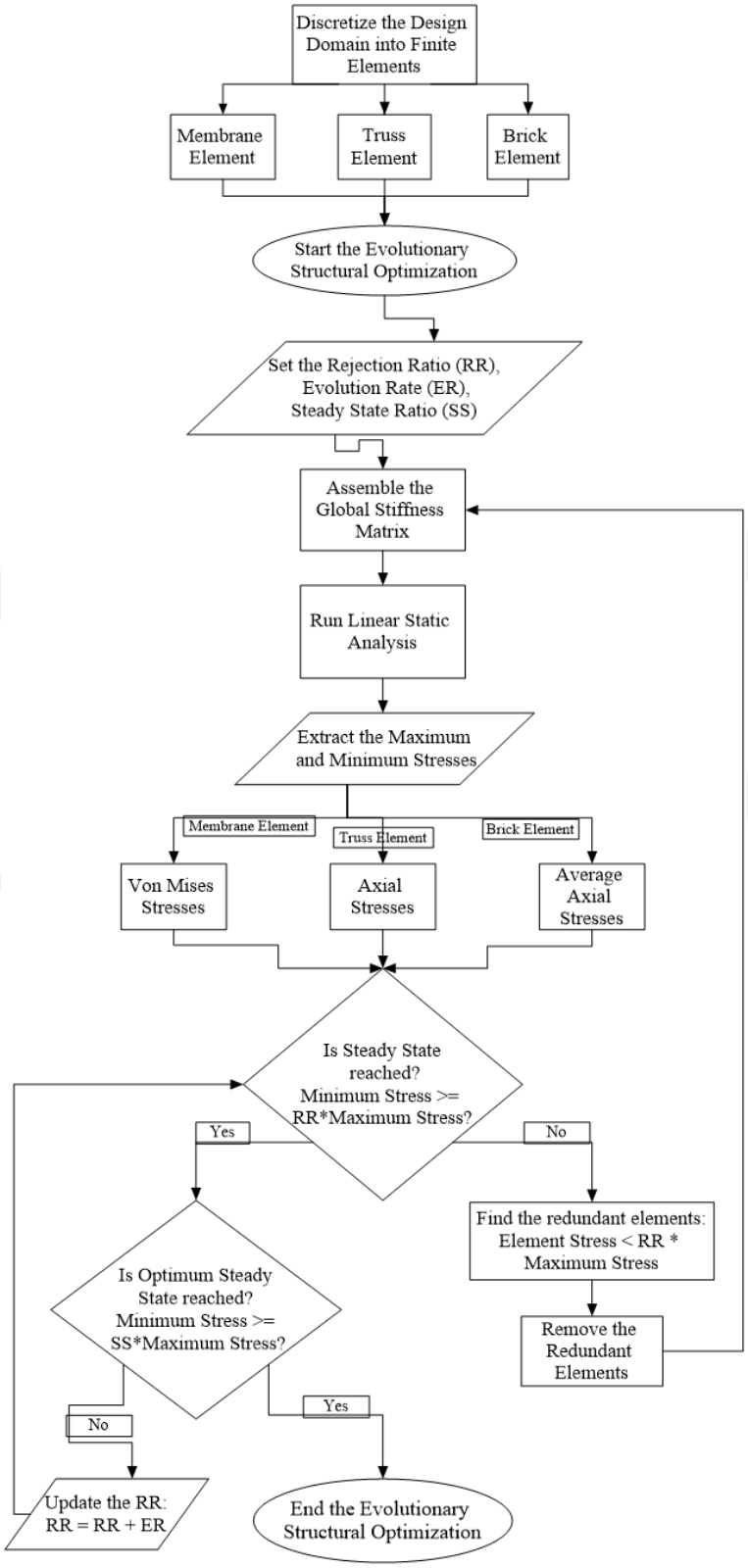


Figure 3.1: Stress-based Evolutionary Structural Optimization Flowchart Diagram

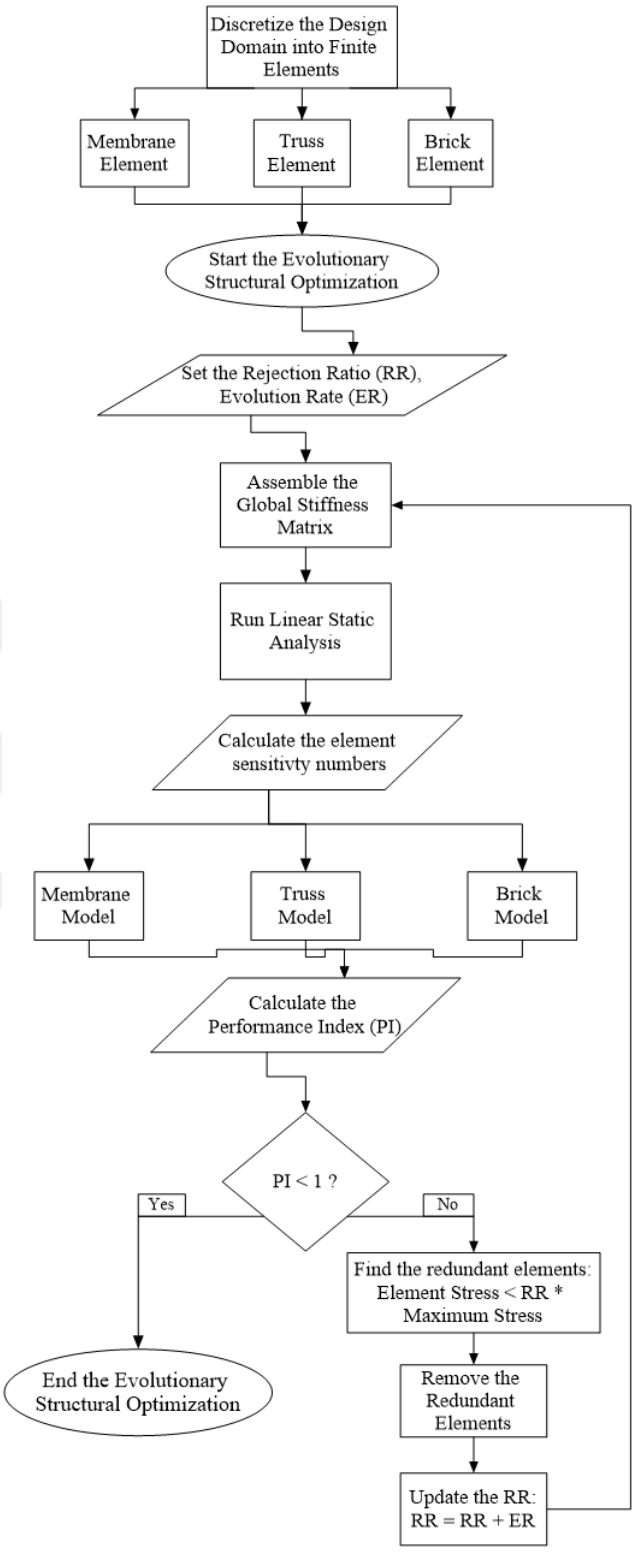


Figure 3.2: Displacement-based Evolutionary Structural Optimization Flowchart Diagram

3.2 Pre-processing- Gmsh and Excel VBA

The Evolutionary Structural Optimization method requires multiple finite element analyses in each iteration until the optimum state is achieved. To support this process, a structured mesh of the reinforced concrete member is necessary. Gmsh is used to generate the mesh for the examples analyzed (Geuzaine & Remacle, 2024). In this study, example geometries are defined within .geo text files, which also specify the mesh density and methodology. All example structures are discretized into quadrilateral elements using a structured mesh generated in Gmsh. The mesh information is then exported to a MATLAB script. For further information on the Gmsh models and the meshing strategies for different types of example geometry, please refer to the Appendix A.1.

Additional information required for finite element analysis in MATLAB is defined in an Excel sheet. The input data includes the material properties, locations of the supports, locations and magnitudes of the external loads, and the corresponding units. This information is organized into columns, with each row representing a specific example model. Input data are entered into the corresponding cells for each model. An Excel VBA macro is then executed to generate a MATLAB script that contains the input data from the selected row. For further information on the excel macro routine for organizing model properties and exporting them into a MATLAB script, please refer to the Appendix A.2.

3.3 Finite Element Analysis and Evolutionary Structural Optimization- MATLAB

In Evolutionary Structural Optimization (ESO) using the Finite Element Method (FEM), the material distribution of the structure to be optimized is represented by finite analytical elements. A finite element model is constructed by incorporating the structure's boundary conditions and external loads. The process begins with an initial linear static analysis. Based on these initial results, the optimization procedure is

initiated. This involves the gradual removal of redundant elements in each iteration until an optimal steady state is achieved. At this stage, the distribution of the remaining elements reveals the internal force flow within the reinforced concrete member. Since the finite elements represent the ideal material distribution required to transfer external loads to the structure's supports, their behavior is critically important. Three types of analytical elements are utilized for this task.

Three MATLAB Projects are created for three types of finite elements. The projects are named after the analytical element model they create: Brick, Membrane, and Truss. These projects create the finite element model by using the mesh script (exported from Gmsh model) and the inputData struct from the inputData Excel sheet, conduct a series of finite element analyses, and perform evolutionary structural optimization. The MATLAB procedure is examined in two parts: Finite Element Analysis and Evolutionary Structural Optimization.

3.3.1 Finite Element Analysis

The workflow of the Finite Element Analysis of each element model is generally identical with a few key differences. The Finite Element Analysis routine is saved in a script, where the necessary functions are called for the analysis. The pseudocode of this script is presented below.

Meshing

CALL inputScript
Import the input script from Excel to load project parameters.

CALL readGmsh(inputData.gmsh)
Extract the node coordinates and connectivity matrices from the Gmsh file.
OUTPUT: meshData containing node coordinates and connectivity.

CALL createGraph(meshData) RETURNING G
Generate a graph G using the extracted mesh data.
G.Nodes represent mesh points, and G.Edges represent connections between them.

CALL mergeEdges(G) RETURNING G
Note: This function applies only to Truss and Brick Model.
Merge overlapping edges in Graph G and updates the graph structure.

CALL plotGraph(G)
Visualize graph G to verify its correctness.

CALL createTable(meshData, G) RETURNING T
Note: This function applies only to Brick and Membrane Model.
Generate a table T where each row represents a single brick element.

Linear Static Analysis

CALL setSupports(G, inputData.supports) RETURNING G
Mark the fixed nodes.

CALL setDof(G) RETURNING G
Number the free degrees of freedom.

CALL setLoads(G, inputData.loads) RETURNING G
Define the external loads to the corresponding nodes in Graph G.

CALL markElements(T, G) RETURNING T
Mark the elements that are directly connected to a fixed or loaded node.
Note: This function applies only to Brick and Membrane Model.

CALL elasticity(G, inputData.materialProperties.Ec)
Assign the elasticity of each element.
OUTPUT: G for Brick and Truss Model, T for Membrane Model.

CALL globalStiffness(G) RETURNING K_structure
Assemble the global stiffness matrix of the structure.

CALL getDisplacements(G, K_structure, inputData) RETURNING G
Solve the system for displacements.

CALL getStresses(G, inputData) RETURNING G
Extract the axial stresses of truss elements.
Note: This function only applies to Brick and Truss Model.

CALL sortBrickStresses(T, G) RETURNING T
Find the average truss axial stresses for each brick element.
Note: This function only applies to Brick Model.

CALL getStressesAtCenter(G, T, inputData) RETURNING T
Extract the stresses at the center of the membrane elements.
Note: This function only applies to Membrane Model.

Graphs

CALL plotAxialStresses
INPUTS: G, inputData.outputUnits, the title of the graph.
Plots the axial stresses of the truss elements.

CALL plotVonMisesStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the Von Mises stresses at the center of the membrane elements.

CALL plotPrincipalStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the principal stresses at the center of the membrane elements.

The *inputScript.m* file is executed to generate the inputData struct, and the corresponding mesh script is run to obtain the node coordinates and node connectivity. In this section, nodes and analytical elements are created. MATLAB's Graph and Table data structures are utilized for this purpose.

Graph objects in MATLAB represent undirected graphs, where edges connect nodes without any direction. Graph objects are ideal for modeling one-dimensional elements connected to each other at nodes. Table arrays, on the other hand, store column-oriented data. Each column is treated as a variable, which allows for storing data of varying types and sizes, as long as all columns have the same number of rows.

A Finite Element Model is created in MATLAB, by incorporating the Graph and Table object. A Linear Static Analysis is then performed using the initial geometry of the structure. The results from this analysis are saved for the optimization process. This initial step in the overall optimization procedure is examined separately for the three analytical element models.

3.3.1.1 Truss Model

The Strut-and-Tie Method (STM) is a truss analogy used to represent the flow of forces. Various optimization methods are commonly employed by structural engineers to design strut-and-tie models, which consist of truss elements. Hence, using truss elements for discretizing the reinforced concrete member is suggested.

If a fully automated design tool were to be developed in the future—where the user inputs the reinforced concrete member, boundary conditions, and external loads to

generate a complete strut-and-tie model—starting with truss elements would streamline the process. This approach eliminates the need to transform remaining membrane elements into a truss model later in the workflow.

A one-dimensional 2-node linear truss element is utilized. Each node has two degrees of freedom. The degrees of freedom of a truss element are illustrated in Figure 3.3.

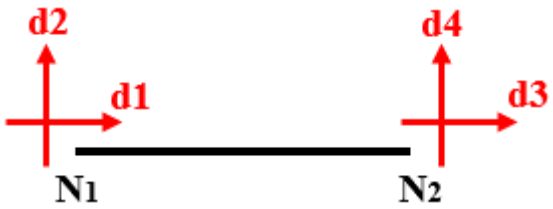


Figure 3.3: The Local Degrees of Freedom of a Truss Element

The elasticity of each truss element is determined using the modulus of elasticity of the concrete. The cross-sectional area of the truss elements is not considered explicitly; instead, all truss elements are assumed to have a unit area. The modulus of elasticity is available in the inputData struct that is exported into the MATLAB Project using the Excel VBA macro.

Meshing

In this section, the readGmsh function creates the nodes and truss elements. The mesh script contains the coordinates matrix of the nodes and the connectivity matrix of the quadrilateral mesh elements. The readGmsh function further discretizes the quadrilateral elements into truss elements. The sides of the quadrilateral elements are defined as truss elements and two additional truss elements are added to the diagonals of the quadrilateral elements. This function also removes overlapping truss elements at the sides.

A graph object of G is created. G includes two tables: Edges and Nodes. G.Nodes represents the nodes and G.Edges represents the truss elements. Each node and edge are assigned a unique ID number. The G.Nodes table stores the coordinates, the degrees of freedom in both directions, external load values, and displacements of the nodes. Each row in the G.Nodes table denotes a node. The G.Edges table stores the

IDs of the end-nodes, the elasticity and axial stresses of truss elements. Each row in the G.Edges table denotes a truss element. The configuration of the G.Nodes and G.Edged tables can be seen in Figure 3.4 and Figure 3.5, respectively. The Truss model is shown in Figure 3.6.

G | G.Nodes | G.Edges

G.Nodes

	1 X	2 Y	3 ID
1	-3000	0	1
2	3000	0	2
3	3000	2400	3
4	-3000	2400	4
5	-2500	0	5

Figure 3.4: G.Nodes Table

G | G.Nodes | G.Edges

G.Edges

	1 EndNodes		2 ID
1	1	5	1
2	1	32	2
3	1	33	3
4	2	15	4
5	2	16	5
6	2	63	6

Figure 3.5: G.Edges Table

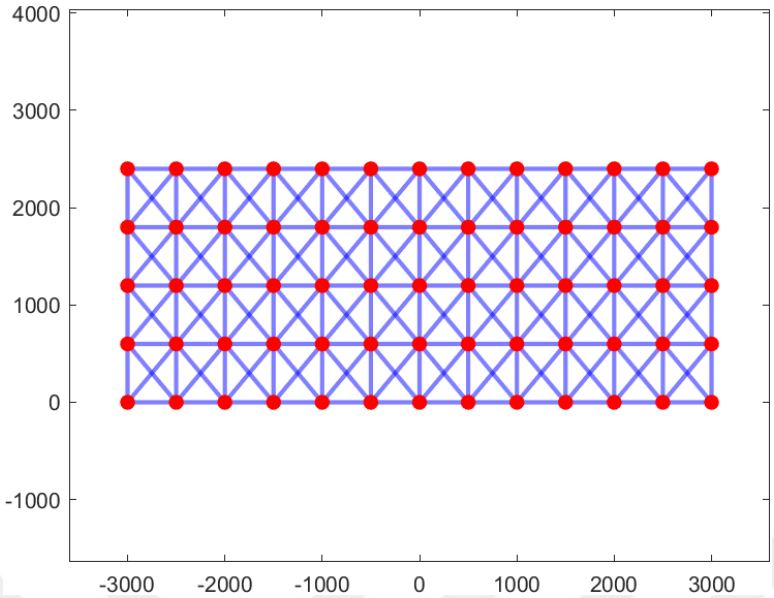


Figure 3.6: Truss Model

Linear Static Analysis

Once the data structure is created, the rest of the procedure is conducted by utilizing G. Step by step, the attributes of the nodes and truss elements are processed on G.Edges and G.Nodes. The x-coordinate ranges of the support regions are available in the inputData struct. Likewise, the locations and magnitudes of the external loads are also present. Using this information, the nodes that are fixed in any direction and applied external loads are marked. Consequently, the truss elements that are directly connected to these nodes are also marked. This information will come in handy in the evolutionary structural optimization. Stresses are excessively high in such regions in a finite element analysis. Hence the stresses of these elements will be ignored during the optimization procedure.

The next step is to construct the stiffness matrix for the entire structure. The free degrees of freedom are numbered in G.Nodes. The elasticity of each truss element is entered in G.Edges. The global stiffness matrix is assembled by integrating Graph G.

Finally, the displacements are solved and saved in G.Nodes; and the axial stresses are extracted and saved in G.Edges. The final configuration of G.Nodes and G.Edges can be seen in Figure 3.7 and Figure 3.8.

G | G.Nodes | G.Edges

G.Nodes

	1 X	2 Y	3 ID	4 DOF_X	5 DOF_Y	6 SupportNode_X	7 SupportNode_Y	8 SupportNode	9 Load_X	10 Load_Y	11 LoadNode	12 Displacement_X	13 Displacement_Y
1	-3000	0	1	0	0	1	1	1	0	0	0	0	0
2	3000	0	2	0	0	1	1	1	0	0	0	0	0
3	3000	2400	3	1	2	0	0	0	0	0	0	-46.2697	-11.4247
4	-3000	2400	4	3	4	0	0	0	0	0	0	46.2697	-11.4247
5	-2500	0	5	0	0	1	1	1	0	0	0	0	0
6	-2000	0	6	5	6	0	0	0	0	0	0	-10.1336	-36.8962
7	-1500	0	7	7	8	0	0	0	0	-400	1	-12.3442	-66.9056
8	-1000	0	8	9	10	0	0	0	0	-400	1	-9.6561	-83.5975
9	-500	0	9	11	12	0	0	0	0	-400	1	-4.9266	-91.7099
10	0	0	10	13	14	0	0	0	0	0	0	6.8030e-14	-87.2853

Figure 3.7: Final Configuration of G.Nodes in Truss Model

G | G.Nodes | G.Edges

G.Edges

	1 EndNodes		2 ID	3 SupportEdge	4 LoadEdge	5 Elasticity	6 AxialStress
1	1	5	1	1	0	28567	0
2	1	32	2	1	0	28567	-102161534
3	1	33	3	1	0	28567	-177261480
4	2	15	4	1	0	28567	0
5	2	16	5	1	0	28567	-102161534
6	2	63	6	1	0	28567	-177261480
7	3	18	7	0	0	28567	-19648995
8	3	19	8	0	0	28567	-16374162
9	3	65	9	0	0	28567	25577259
10	4	29	10	0	0	28567	-16374162

Figure 3.8: Final Configuration of G.Edges in Truss Model

3.3.1.2 Brick Model

Kwak and Noh (2006) proposed a new analytical finite element called the brick element. This element is not commonly used and is underexplored, in contrast to the use of the membrane element.

A brick element consists of six truss elements: four representing the sides of a quadrilateral element and two forming its diagonals. The side truss elements are designed to resist direct (normal) forces and stresses, while the diagonal elements are intended to resist shear forces and stresses.

Element elimination during the optimization process is performed by treating each brick element as the smallest unit of the system. The configuration of the brick element is illustrated in Figure 3.9.

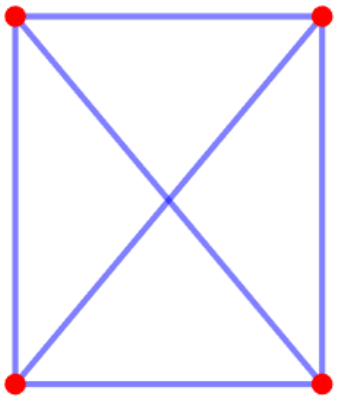


Figure 3.9: The Brick Element

The brick element offers a novel approach compared to the truss element model. While resembling the quadrilateral shape and bilinear shape functions of the membrane element, it is specifically designed to enhance the robustness of a reinforced concrete member discretized with truss elements.

Meshing

The Brick Model resembles the Truss Model. The reinforced concrete member is discretized into truss elements just like the Truss Model. Each quadrilateral element is replaced with 6 truss elements: 4 truss elements at the sides and 2 truss elements at the diagonals. The overlapping truss elements are also removed.

Unlike the Truss model, the G.Edges in the Brick Model do not represent the smallest unit of analytical elements. Therefore, a table object, T, is created. Each row in T contains information about a single brick element. The table T includes the IDs, node connectivity, and truss connectivity of each brick element. The truss connectivity matrix denotes the truss (G.Edges) IDs that comprises a brick element.

The workflow of the Brick Model recalls the association of truss elements to the brick elements. Hence, G.Edges for the Brick Model includes the ID of the brick elements that they form. Since the overlapping truss elements are removed at the beginning of the procedure, a single truss element can belong to two adjacent brick elements. In that case, two brick IDs are denoted in G.Edges table. This can be seen in Figure 3.10. The configuration of table T can be seen in Figure 3.11.

T | G | G.Edges | G.Nodes

G.Edges

	1 EndNodes		2 ID	3 brickID
15	5	33	15	[1;5]
16	5	36	16	5
17	6	7	17	9
18	6	33	18	5
19	6	36	19	[5;9]
20	6	39	20	9

Figure 3.10: G.Edges in Brick Model

T | G | G.Edges | G.Nodes

48x3 table

	1 ID	2 nodeConnectivity				3 trussConnectivity					
1 brick1	1	1	5	33	32	1	2	3	14	15	115
2 brick2	2	32	33	34	31	111	112	113	115	116	117
3 brick3	3	31	34	35	30	108	109	110	113	114	120
4 brick4	4	30	35	29	4	10	11	12	105	106	110
5 brick5	5	5	6	36	33	13	15	16	18	19	118

Figure 3.11: Table T in Brick Model

Linear Static Analysis

The linear static analysis is conducted in a similar manner to the Truss Model. G.Nodes and G.Edges are processed using the inputData struct. Then, the global stiffness matrix is solved for nodal displacements. The axial stresses of each truss element are extracted.

However, a brick element is comprised of 6 truss elements. Hence, sortBrickStress function is utilized to track down the truss elements that are forming each brick element. Average axial stresses of these truss elements are calculated and saved in Table T. The final configuration of table T can be seen in Figure 3.12.

T | G | G.Edges | G.Nodes

48x5 table

	1 ID	2 nodeConnectivity				3 trussConnectivity						4 ignoreElements	5 avgAxialStress
1 brick1	1	1	5	33	32	1	2	3	14	15	115	1	1.6783e+08
2 brick2	2	32	33	34	31	111	112	113	115	116	117	0	1.1399e+08
3 brick3	3	31	34	35	30	108	109	110	113	114	120	0	6.0376e+07
4 brick4	4	30	35	29	4	10	11	12	105	106	110	0	1.9730e+07
5 brick5	5	5	6	36	33	13	15	16	18	19	118	1	270194067
6 brick6	6	33	36	37	34	117	118	119	121	122	126	0	1.4660e+08
7 brick7	7	34	37	38	35	120	122	123	124	125	129	0	8.3823e+07
8 brick8	8	35	38	28	29	101	102	103	106	107	125	0	55584132
9 brick9	9	6	7	39	36	17	19	20	22	23	127	1	1.4961e+08
10 brick10	10	36	39	40	37	126	127	128	130	131	135	0	1.5012e+08

Figure 3.12: Final Configuration of Table T in Brick Element

3.3.1.3 Membrane Model

The plane stress membrane element is the most commonly used plane stress element in the field of Evolutionary Structural Optimization. Membrane elements are used to model two-dimensional design domains where the in-plane forces are significant. Out-of-plane bending moments are ignored.

A 4-node bilinear quadrilateral element is utilized. Bilinear shape functions are used to interpolate the nodal displacements to the points located anywhere within the

element. The displacement within the element is mathematically expressed in Equation 3.2.

$$u(x,y) = \sum_{i=1}^{4} N_i(x,y)\, u_i \tag{3.2}$$

$N_i(x,y)$ are the bilinear shape functions and u_i are the displacements at nodes. Figure 3.13 illustrates the 4-node bilinear quadrilateral element (Yang, 2018).

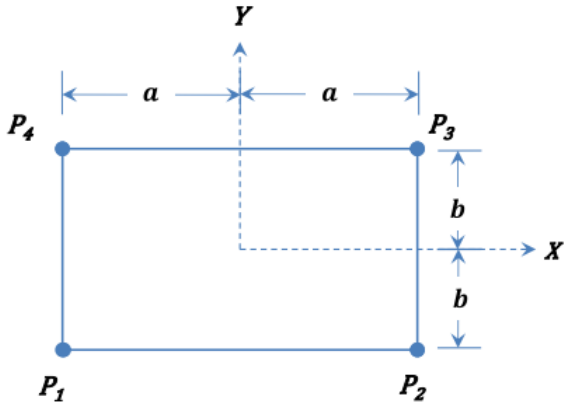


Figure 3.13: 4-node Bilinear Quadrilateral Membrane Element (Yang, 2018)

Each node has 2 degrees of freedom: displacements in the x and y direction. The shape functions are linear in each direction. Equation 3.3 show the shape functions given for each point in the element respectively.

$$N(X,Y) = \begin{cases} N_1(X,Y) = \dfrac{1}{4}(1-X)(1-Y) \\ N_2(X,Y) = \dfrac{1}{4}(1+X)(1-Y) \\ N_3(X,Y) = \dfrac{1}{4}(1+X)(1+Y) \\ N_4(X,Y) = \dfrac{1}{4}(1-X)(1+Y) \end{cases} \tag{3.3}$$

Where X, Y ϵ [-1, 1] are the natural coordinates of the element. The elasticity matrix is constructed using the modulus of elasticity and Poisson's ratio of the concrete. These material properties are extracted from the inputData struct. The elasticity matrix for the plane stress membrane element is presented in Figure 3.14.

```
% Construct the elasticity matrix
D = E/(1-poissonRatio^2)*[1, poissonRatio, 0;
    poissonRatio, 1, 0;
    0, 0, 0.5*(1-poissonRatio)];
```

Figure 3.14: The Elasticity Matrix of Membrane Element in MATLAB

Stress recovery is performed using a single Gauss point, which provides stress values at the element's center. The Von Mises stresses and principal stresses are reported at the center of the element.

Meshing

A graph G is also created in the Membrane Model using similar approaches as the previous Models. Since the element used in this model is a 4-node quadrilateral membrane element, there is no need to further discretize the existing mesh. The connectivity matrix is directly adapted to the G.Edges table. Each row in G.Edges table represents the sides of the quadrilateral plane stress elements. Since every edge belongs to a single membrane element, overlapping edges are not removed.

G.Nodes is configured similarly to the previously mentioned Models. G.Edges include the edge IDs and the IDs of the membranes each of them belong to. The G.Edges table can be seen in Figure 3.15. The resulting Membrane model is shown in Figure 3.16.

G | T | G.Edges | G.Nodes

G.Edges

	1 EndNodes		2 ID	3 membraneID
1	1	5	1	1
2	1	32	145	1
3	2	15	45	45
4	2	16	93	45
5	3	18	96	48

Figure 3.15: G.Edges in Membrane Model

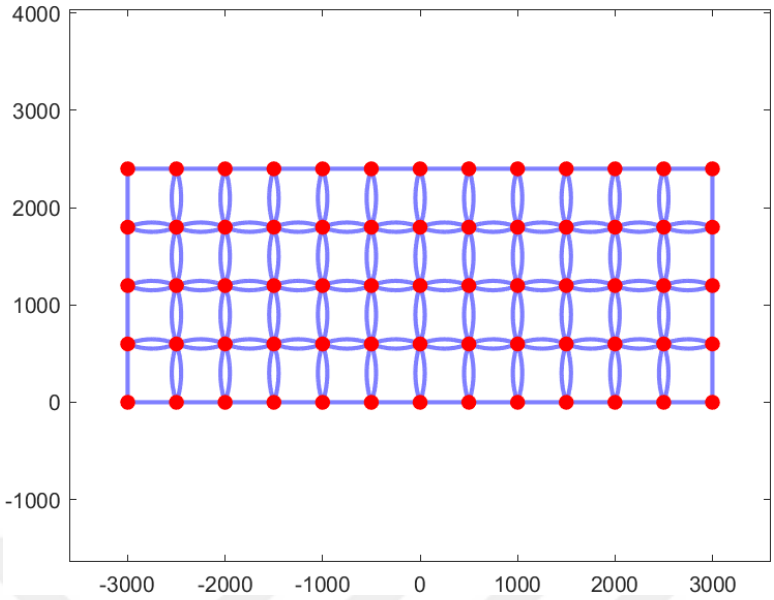


Figure 3.16: Membrane Model

Similar to the Brick Model, the edges in the Membrane Model do not represent the smallest unit of analytical elements. Therefore, a table object of T is created to correlate the G.Edges to actual quadrilateral membrane elements. Each row in T contains information about a single membrane element. The table includes the membrane ID and its node connectivity. The configuration of Table T can be seen in Figure 3.17.

G | T | G.Edges | G.Nodes

48x2 table

	1 ID	2 nodeConnectivity			
1 membrane1	1	1	5	33	32
2 membrane2	2	32	33	34	31
3 membrane3	3	31	34	35	30
4 membrane4	4	30	35	29	4
5 membrane5	5	5	6	36	33

Figure 3.17: Table T in Membrane Model

Linear Static Analysis

Table T stores the elasticity matrices and the node connectivity for all membrane elements. The G.Nodes table contains the degrees of freedom and coordinates for each node. Using this information, the global stiffness matrix is constructed by assembling the stiffness from all elements.

The in-plane stresses at the center of the membrane elements are extracted using a single Gauss point. The stress extraction equation is given in Equation 3.4.

$$\sigma = D * B * u \tag{3.4}$$

Where D is the constitutive matrix, B is the derivatives of the shape functions evaluated at the integration points, and u is the displacement vector of each membrane element.

The Von Mises stresses at the center are calculated using the in-plane stress values and saved in Table T. The Von Mises stress equation for a plane stress membrane element is given in Equation 3.5.

$$\sqrt{\sigma_{xx}^2 - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^2 + 3\sigma_{xy}^2} \tag{3.5}$$

Likewise, the principal stresses at the center are also calculated and saved in Table T. In addition, the absolute maximum principal stress is found and saved in Table T as well. The equation for finding the principal stresses is given in Equation 3.6.

$$\sigma_1, \sigma_2 = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_{xx} - \sigma_{yy}}{2}\right)^2 + \sigma_{xy}^2} \tag{3.6}$$

The final configuration of Table T in Membrane Model can be seen in Figure 3.43.

T | G | G.Edges | G.Nodes

48x8 table

	1 ID	2 nodeConnectivity				3 ignoreElements	4 elasticity	5 stressesAtCenter	6 principalStresses	7 absolutePrincipalStresses	8 vonMisesStresses
1 membrane1	1	1	5	33	32	1	[2.9225e...	[-406.5621;-4.5...	-4.6793e+03	4.6793e+03	4.5328e+03
2 membrane2	2	32	33	34	31	0	[2.9225e...	[77.7510;-2.48...	-2.5661e+03	2.5661e+03	2.6484e+03
3 membrane3	3	31	34	35	30	0	[2.9225e...	[-62.0729;-1.05...	-1.2956e+03	1.2956e+03	1.3957e+03
4 membrane4	4	30	35	29	4	0	[2.9225e...	[-143.7793;-20...	-332.8774	332.8774	323.5263
5 membrane5	5	5	6	36	33	1	[2.9225e...	[-3.8766e+03;-...	-7.5960e+03	7.5960e+03	7.5391e+03
6 membrane6	6	33	36	37	34	0	[2.9225e...	[142.7889;-2.1...	-3.5051e+03	3.5051e+03	4.4525e+03
7 membrane7	7	34	37	38	35	0	[2.9225e...	[-524.2321;-87...	-2.1576e+03	2.1576e+03	2.6198e+03
8 membrane8	8	35	38	28	29	0	[2.9225e...	[-733.9336;-21...	-1.1118e+03	1.1118e+03	1.2043e+03
9 membrane9	9	6	7	39	36	1	[2.9225e...	[-906.0764;1.1...	2.5878e+03	2.5878e+03	4.3019e+03
10 membrane10	10	36	39	40	37	0	[2.9225e...	[-1.1017e+03;-...	-3.6539e+03	3.6539e+03	5.2169e+03

Figure 3.18: Final Configuration of Table T in Membrane Model

3.3.2 Evolutionary Structural Optimization

The Evolutionary Structural Optimization method is utilized to reduce the finite element distribution of a design domain to extract the internal force path of the reinforced concrete structure. The remaining elements that denote the force path can then be used to build a Strut-and-Tie Model.

To achieve meaningful results at the end of the optimization process, it is crucial to establish a clear and accurate method for determining which elements to eliminate. The elimination criterion defines the elements deemed redundant and identifies those that are subject to removal.

Hence, two element elimination criteria are utilized in the optimization process. These are stress-based criterion and displacement-based (performance-based) criterion. These two approaches to the Evolutionary Structural Optimization method are examined in the following sections. The methodology is explained, and the implementation is described.

3.3.2.1 Stress-based Optimization

Element stresses are chosen as the governing parameter for elimination. Analytical elements with stress values below a certain threshold are considered redundant and

subject to removal. Stress-based elimination is both direct and straightforward, making it easy to understand and implement. Since the goal of the Strut-and-Tie Method is to detect internal force flow, using the stresses of elements within the reinforced concrete member is suggested.

Ultimately, Evolutionary Structural Optimization is a compliance minimization problem. The stress-based evolutionary structural optimization problem is described in Equation 3.7.

$$\begin{aligned} &minimize\ W = \sum_{e=1}^{n} w_e \\ &subject\ to\ \sigma^{max} - \sigma^{*} \leq 0 \end{aligned} \tag{3.7}$$

Where W is the total weight of the structure, n is the number of finite elements in the reinforced concrete member, w_e is the weight of the e^{th} element, σ^{max} is the maximum stress value in the initial geometry of the structure under the given loads, σ^{*} is the prescribed stress limit. The minimization of the structure's weight is achieved by removing the redundant elements from the reinforced concrete member. The element removal must be performed gradually to avoid instabilities and potential singularities in the stiffness matrix.

Firstly, a preliminary linear static analysis is conducted on the structure in its initial state. The internal stress of each finite element is then calculated and recorded.

Three parameters are introduced for the determination and gradual removal of redundant elements: Rejection Ratio (RR), Evolution Rate (ER), and Steady State Ratio (SS). RR describes the desired stress levels among the finite elements at each iteration. It is used to determine the redundant elements at each step. The minimum (σ_{min}) and maximum stress value (σ_{max}) in the reinforced concrete member is determined.

$$\sigma_{min} < RR * \sigma_{max} \tag{3.8}$$

If Equation 3.8 is satisfied, that means there are redundant elements in the reinforced concrete member. The stresses of each element are compared with the maximum stress value.

$$\sigma_e < RR * \sigma_{max} \tag{3.9}$$

The elements that satisfy Equation 3.9 are redeemed as redundant and removed from the reinforced concrete member. The structure is reanalyzed, and the stresses of the remaining elements are calculated. This continues until Equation 3.8 is not satisfied. This means that all the element stresses in the reinforced concrete member are within the RR threshold. This concludes the iteration. Then, the next iteration begins.

$$RR = RR + ER \tag{3.10}$$

RR is increased by ER. Since the element stress threshold is increased, new redundant elements must be determined and eliminated. The iterations continue until RR reaches the value of SS. This means that all the elements in the reinforced concrete member have relatively the same stress levels that are above a certain ratio.

The Evolution Rate controls the pace of the optimization process, ensuring a gradual progression. Xie and Steven (1993) stated that large RR and ER values can lead to structural singularity due to excessive element removal. Liang et al (2000) stated that although a smaller number of elements deleted per iteration would mean a more accurate solution obtained by the optimization, the computational cost would be immensely increased. Hence, an efficient value for RR and ER must be determined. A typical value of 1%, as recommended by Xie and Steven (1993), is selected for both parameters.

Further decreasing these parameters will slow down the optimization as fewer elements will be identified as redundant in each iteration. Hence, lower values should only be considered if the structure becomes singular in the early stages of the optimization process.

The Steady State Ratio (SS) is used to define the threshold for determining the optimum state. SS is set to be 50%, to make sure that enough number of elements are removed from the reinforced concrete member to achieve a refined element distribution. To track the optimization history, the element stresses are plotted and saved during each iteration in which elements are eliminated.

The element elimination criteria differ across each Model. In Truss Model, elimination is based on the axial stresses of each truss element. In the Brick Model, elimination is based on the average axial stresses of the six truss elements that form a brick element. In the Membrane Model, elimination is based on the either Von Mises stresses or principal stresses of each membrane element.

In addition to the three parameters that were introduced at the beginning of this section, the Element Removal Ratio (ERR), is introduced for the Truss Model. It is used to limit the maximum number of elements that can be removed in a single iteration and introduced to increase the robustness of the Truss Model. It is aimed to diminish the risk of singularities during the course of the optimization by setting a limit to the maximum number of elements that can be removed at a step. The maximum number of redundant elements is determined by multiplying the number of remaining elements in the model with the ERR. Once the redundant elements are determined, they are sorted in ascending order of their absolute axial stresses. That way, when the maximum number of element deletion limit is applied, the least critical elements in the redundant elements group are set to be deleted. There is a potential weakness in this approach, though.

A potential weakness of this approach is when the applied loading is distributed symmetrically in reinforced concrete structures. In this case, the element stresses are also expected to be symmetric. Without a maximum number of element deletion limit, it can be assumed that equal number of elements will be gradually deleted from each side of the model. However, when a scalar maximum number of element deletion limit is applied, it is possible that more elements are removed from either side of the model. To prevent this, an additional check is included in the element deletion function. The redundant elements are determined, sorted in ascending order of their absolute axial stress values. The maximum number of elements to be deleted is n. The first n rows are selected for removal. Then the stresses of these n elements are checked. If there is a group of elements that have relatively higher axial stresses than the rest of the elements, they are also excluded from deletion. That way, groups

of elements that have the same stress values in the model are deleted all together. This aims to preserve the symmetry of the model.

3.3.2.2 Displacement-based Optimization

The displacement-based evolutionary structural optimization problem is described in Equation 3.11.

$$\begin{aligned} &minimize\ W = \sum_{e=1}^{n} w_e \\ &subject\ to\ u_j\ - u_j^* \leq 0, j = 1, \dots, m \end{aligned} \tag{3.11}$$

Where W is the total weight of the structure, n is the number of elements in the model, w_e is the weight of the e^{th} element, u_j is the displacement of the j^{th} constrained node, u_j^* is the prescribed limit of u_j, and m is the number of constraint nodes.

The goal of the displacement-based optimization is to gradually remove elements from the reinforced concrete member that have the least effect on the displacement of the constrained nodes. The redundant elements are determined based on element sensitivity numbers. At every iteration, unit virtual loads are applied on the constrained node, along with the real external loads, and the system is analyzed for both real and virtual displacements. These displacement values are all stored in G.Nodes. Element sensitivity numbers are calculated using Equation 3.12.

$$\alpha_e = \left| u_{ej} * k_e * u_e \right| \tag{3.12}$$

Where α_e is the sensitivity number of eth element, u_{ej} is the displacement vector of the e^{th} element under virtual loads, k_e is the e^{th} element's stiffness matrix, u_e is the displacement vector of e^{th} element under real loads. At every iteration of the optimization, sensitivity numbers of every element in the model are calculated.

Similar to the stress-based optimization method, Rejection Ratio (RR) and Evolution Rate (ER) parameters are incorporated in the procedure. The maximum element sensitivity number at the iteration is extracted. Elements that satisfy Equation 3.13

are determined as redundant and set for removal from the reinforced concrete member.

$$\alpha_e < RR * \alpha_{max} \tag{3.13}$$

Instead of the Steady State Ratio (SS) a new parameter called the Performance Index is introduced by Liang et al. (1999) for the stopping criterion of the optimization process. The Performance Index is calculated using Equation 3.14.

$$PI = \frac{u_{0j} * W_0}{u_{ij} * W_i} \tag{3.14}$$

Where u_{0j} is the displacement of the constrained node under real loads in the initial state of the structure, w_0 is the weight of the initial structure, u_{ij} is the displacement of the constrained node under real loads in each iteration, w_0 is the weight of the structure in each iteration.

The Performance Index is calculated at the end of every iteration of the optimization procedure. The iterations continue until the Performance Index is smaller than 1. Apart from being the stopping criterion for the optimization process, the Performance Index is also suggested to be used when selecting the most efficient element distribution generated at the end of every iteration. The resulting element distribution that has the largest PI value is selected to be the best possible topology of the reinforced concrete member.

3.3.2.3 Element Removal in MATLAB

Each MATLAB Project (Truss, Brick, and Membrane Model) includes two optimization routines— one for stress-based and one for displacement-based optimization— along with their corresponding functions. After the initial Finite Element Analysis, the model is ready for the optimization procedure. The pseudocode for the element removal algorithm of the stress-based evolutionary optimization is given below.

Initialize Parameters

SET RR
Set the initial Rejection Ratio.

SET ER
Set the Evolution Rate.

SET SS
Set the Optimum Steady State Criteria.

Optimization

REPEAT

REPEAT

CALL extractExtremeStresses(T)
Extract the absolute maximum and minimum stresses of the system.
OUTPUT: iMaxStress and iMinStress, containing the absolute maximum and minimum stress value at the iteration, respectively.

Element Elimination

CALL deleteLowStressElements(G, T, iMaxStress, RR) RETURNING G & T
Remove the redundant elements.
OUTPUT: nDeletedElements containing the number of redundant elements in the iteration.

Linear Static Analysis

CALL setDof(G) RETURNING G
Number the free degrees of freedom.

CALL globalStiffness(G) RETURNING K_structure
Assemble the global stiffness matrix of the structure.

CALL getDisplacements(G, K_structure, inputData) RETURNING G
Solve the system for displacements.

CALL getStresses(G, inputData) RETURNING G
Extract the axial stresses of truss elements.
Note: This function only applies to Brick and Truss Model.

CALL sortBrickStresses(T, G) RETURNING T
Find the average truss axial stresses for each brick element.
Note: This function only applies to Brick Model.

CALL getStressesAtCenter(G, T, inputData) RETURNING T
Extract the stresses at the center of the membrane elements.
Note: This function only applies to Membrane Model.

Graphs

CALL plotAxialStresses
INPUTS: G, inputData.outputUnits, the title of the graph.
Plots the axial stresses of the truss elements.

CALL plotVonMisesStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the Von Mises stresses at the center of the membrane elements.

CALL plotPrincipalStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the principal stresses at the center of the membrane elements.

UNTIL iMinStress >= RR*iMaxStress

COMPUTE RR = ER + RR
Update the rejection ratio

UNTIL iMinStress >= SS*iMaxStress

The pseudocode for the element removal algorithm of the displacement-based evolutionary optimization is given below.

Initialize Parameters

SET RR
Set the initial Rejection Ratio.

SET ER
Set the Evolution Rate.

Optimization

REPEAT

CALL performanceIndex(G, T, initialState)
Find the Performance Index of the current state of the system.
OUTPUT: PI, the Performance Index in the iteration.

Compute the Sensitivity Numbers

CALL sensitivityNumber(G, T) RETURNING T
Find the sensitivity number of each element.

CALL extractExtremeSensitivityNumbers(T)
Extract the maximum and minimum sensitivity number of the system.
OUTPUT: maxSensitivityNumber and minSensitivityNumber, containing the maximum and minimum sensitivity number at the iteration, respectively.

Element Elimination

CALL deleteLowSensitivityNumberElements(G, T, maxSensitivityNumber, RR) RETURNING G & T
Remove the redundant elements.
OUTPUT: nDeletedElements containing the number of redundant elements in the iteration.

Linear Static Analysis

CALL setDof(G) RETURNING G
Number the free degrees of freedom.

CALL globalStiffness(G) RETURNING K_structure
Assemble the global stiffness matrix of the structure.

CALL getDisplacements(G, K_structure, inputData) RETURNING G
Solve the system for displacements.

CALL getStresses(G, inputData) RETURNING G
Extract the axial stresses of truss elements.
Note: This function only applies to Brick and Truss Model.

CALL sortBrickStresses(T, G) RETURNING T
Find the average truss axial stresses for each brick element.
Note: This function only applies to Brick Model.

CALL getStressesAtCenter(G, T, inputData) RETURNING T
Extract the stresses at the center of the membrane elements.
Note: This function only applies to Membrane Model.

Graphs

CALL plotAxialStresses
INPUTS: G, inputData.outputUnits, the title of the graph.
Plots the axial stresses of the truss elements.

CALL plotVonMisesStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the Von Mises stresses at the center of the membrane elements.

CALL plotPrincipalStressesAtCenter
INPUTS: G, T, inputData.outputUnits, The title of the graph.
Plots the principal stresses at the center of the membrane elements.

UNTIL PI < 1

In each iteration where element elimination occurs, the stress diagram of the current structure is plotted and saved. The percentage of the remaining elements in the structure is also reported in these graphs, along with the Rejection Ratio for stress-based optimization and the Performance Index for displacement-based optimization. The elapsed time for the loop is also recorded. The time spent drawing the stress diagrams and processing and recording the stress of elements is not accounted for.

Additional functions and checks are included in the Truss Model Project. Precautionary functions are applied after each element elimination and before the subsequent linear static finite element analysis to prevent singularities in the global stiffness matrix. removeFreeNodes(G) function removes the free nodes formed upon element deletion. The function finds the nodes that are not associated with any edge object and removes them from Graph G. Hence preventing zero diagonals in the global stiffness matrix. Likewise, the removeFreeTrusses(G) function removes a single or a group of edge objects that are disconnected from the structure.

Before analyzing the structure at an iteration, the global stiffness matrix is checked for zero diagonals. If found any, the row numbers are associated with the free degrees of freedom. Since the free degrees of freedom were numbered and stored in G.Nodes, the corresponding free nodes can be detected. These nodes are marked as support nodes in that direction, which makes sure that when the degrees of freedom are numbered again, they are excluded.

Various example models from different studies are collected for this comparison. They are organized in an Excel sheet, and their properties are exported to a MATLAB script. The models are discretized into quadrilateral elements in Gmsh, and the mesh is also exported to MATLAB. These scripts are then used to perform a series of linear static analyses within an optimization process. The selected models and their results are presented in the next two chapters.

CHAPTER 4

PRELIMINARY TESTS

This chapter presents the results of several example problems that are put through different types of Evolutionary Structural Optimization procedures. The ultimate goal is to identify the finite element type that consistently provides the most efficient material distribution, revealing meaningful load paths for the structure, while also demonstrating compatibility across a wide range of structural examples, including rectangles, beams with openings, and pier caps.

The first section introduces the example models used in this study, illustrates the geometry of the models, and explains the boundary conditions and external loadings. As a first case, the Truss Model, where the structural domain is discretized with truss elements, is introduced. The preliminary analyses were conducted to test this model and to see if the model's behavior in a strut-and-tie optimization problem is convincing enough when compared with the traditional Membrane Model and the relatively new Brick Model. The results of The Truss Model for a couple of test cases were presented and findings were discussed.

In the following section, the Membrane and Brick Model comparisons are presented. As their names imply, the Membrane Model utilizes quadrilateral membrane elements whereas in the Brick Model six truss elements are used as a quadrilateral element to discretize the structural domain. A deductive approach is embraced when presenting the results for these models. A few preliminary tests are conducted among the Brick and Membrane Models to determine the most suitable element elimination criteria for both these models. Once the element elimination method is decided, the comparison between Membrane and Brick Models is fully conducted and the results are presented in the next chapter.

4.1 Example Models

The example models used in this study are divided into two subcategories based on their purpose in this comparative study. Sample Problems are used for preliminary performance evaluation of the models. The geometric and material properties of these problems are derived from Liang et al. (2000). The Real-Life Examples are for examining how these models behave in actual design problems. The geometric and material properties of these models have been taken from multiple sources (Mitchell and Collins, 2013; Bedru 2014; Dawood and Abdul-Razzaq, 2021). Each of these models are discretized across three levels of mesh sizes: coarse, medium, and fine mesh.

4.1.1 Sample Problems

The first set of Sample Problems is used to assess the robustness and versatility of the proposed models. This set consists of three rectangular deep beam examples. All Sample Problems involve simply supported rectangular concrete deep beams with varying configurations.

Rectangle Example 1 features two external loads applied at the bottom of the beam. Rectangle Example 2 includes two web openings, and two external loads applied at the top of the beam. Rectangle Example 3 contains a large opening near the bottom-left corner, with a single external load applied at the top of the beam. The geometries, boundary conditions and external loadings of Rectangle Example 1, 2 and 3 are shown in Figures 4.1, 4.3, and 4.5, respectively (Liang et al., 2000). All dimensions are in millimeters. The remaining element distribution obtained by Liang et al. (2000) is shown in Figures 4.2, 4.4, and 4.6 (a), respectively. Rectangle Example 3 and 4 are also optimized by Kwak and Noh (2006) using brick elements. The remaining element distribution of Rectangle Example 3 obtained by Kwak and Noh (2006) is shown in Figure 4.6 (b).

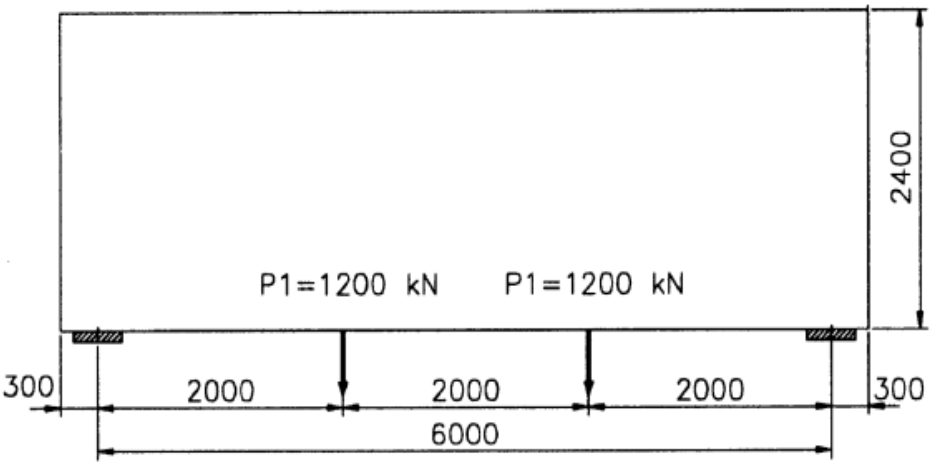


Figure 4.1: The geometry of Rectangle Example 1 (Liang et al., 2000)

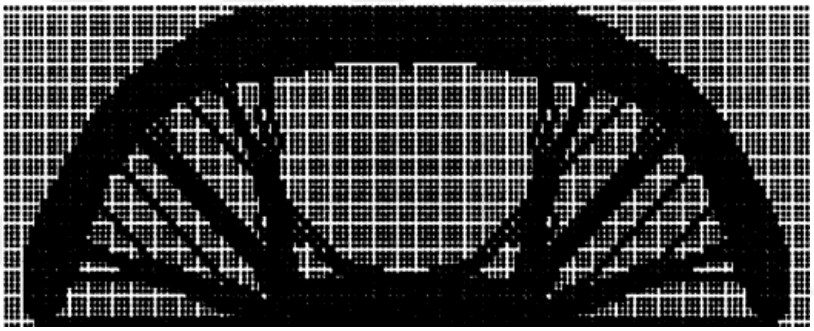


Figure 4.2: The element distribution of Rectangle Example 1 at the end of the optimization (Liang et al., 2000)

Three discretized Rectangle Example 1 models are generated for three levels of mesh refinement. The size of a single quadrilateral element is 200, 100, and 50 mm for coarse, medium, and fine mesh refinements, respectively.

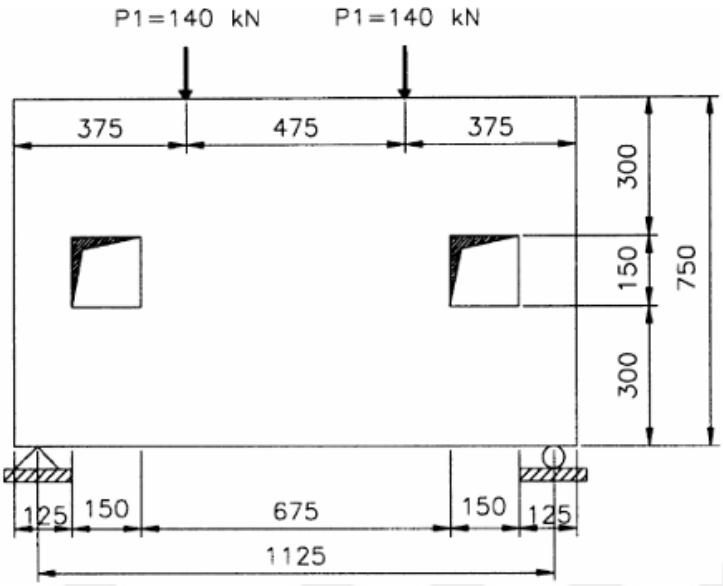


Figure 4.3: The geometry of Rectangle Example 2 (Liang et al., 2000)

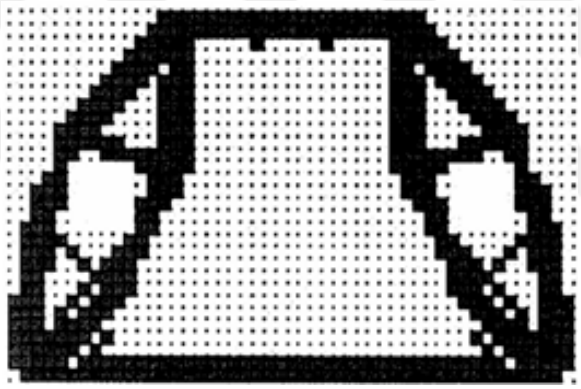


Figure 4.4: The element distribution of Rectangle Example 2 at the end of the optimization (Liang et al., 2000)

A single discretized Rectangle Example 2 model is generated. This is due to the dimensions of the openings. More information on the mesh can be found in Appendix A.1.2. The size of a single quadrilateral element is 25 mm.

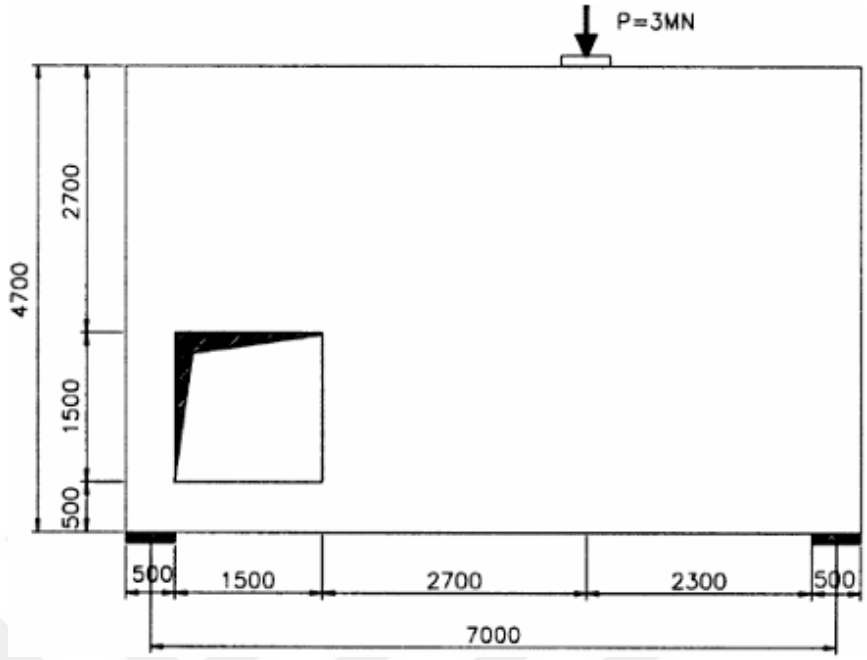
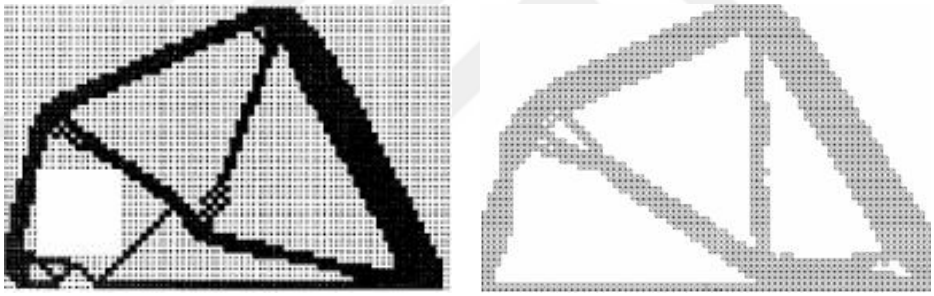


Figure 4.5: The geometry of Rectangle Example 3 (Liang et al., 2000)



(a) (b)

Figure 4.6: The element distribution of Rectangle Example 3 at the end of the optimization: (a) (Liang et al., 2000); (b) (Kwak and Noh, 2006)

A single discretized Rectangle Example 3 model is generated. This is due to the dimensions of the opening. More information on the mesh can be found in Appendix A.1.2. The size of a single quadrilateral element is 100 mm.

The second set of Sample Problems is used to evaluate the performance of the proposed models in reinforced concrete members with increasing slenderness. This set consists of a simple rectangular beam, Rectangle Example 4, presented in four variations. All Rectangle Example 4 models are simply supported, and have an external load applied at the midpoint of the top surface. All beam dimensions remain constant, except for the length. The span-to-depth ratio (L/D) is 2, 3, 4, and 5 for Rectangle Example 4a, 4b, 4c, and 4d. The depth of the beam is 1000 mm, and the applied load is 1200 kN. The geometry, boundary conditions, and external loadings of these models are illustrated in Figure 4.7 (Liang et al., 2000). The element distribution obtained by Liang et al. (2000) is shown in Figure 4.8 for L/D equal to 2, 3, 4, and 5, respectively. The element distribution obtained by Kwak and Noh (2006) is shown in Figure 4.9 for L/D equal to 2, 3, 4, and 6, respectively. Kwak and Noh (2006) selected the span to depth ratio as 6 for the Rectangle Example 4d. Since the sample problems are taken mostly from Liang et al. (2000), the span to depth ratio of 5 is selected for this study.

For each Rectangle Example 4 model, three discretized models are generated for three levels of mesh refinement. For Rectangle Example 4a, the size of a single quadrilateral element is 250, 100, and 50 mm for coarse, medium, and fine mesh refinements, respectively. For Rectangle Example 4b, the size of a single quadrilateral element is 187.5, 100, and 50 mm for coarse, medium, and fine mesh refinements, respectively. For Rectangle Example 4c, the size of a single quadrilateral element is 250, 100, and 50 mm for coarse, medium, and fine mesh refinements, respectively. For Rectangle Example 4c, the size of a single quadrilateral element is 208.3, 100, and 50 mm for coarse, medium, and fine mesh refinements, respectively. For some models, different element sizes have to be selected to make sure that a node is generated at the midpoint of the structure.

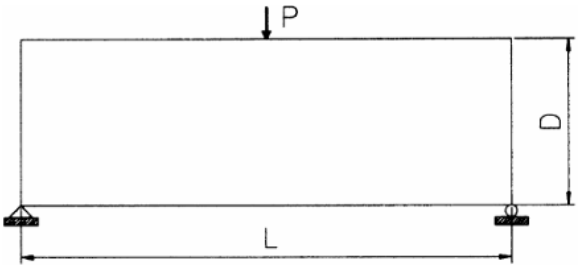
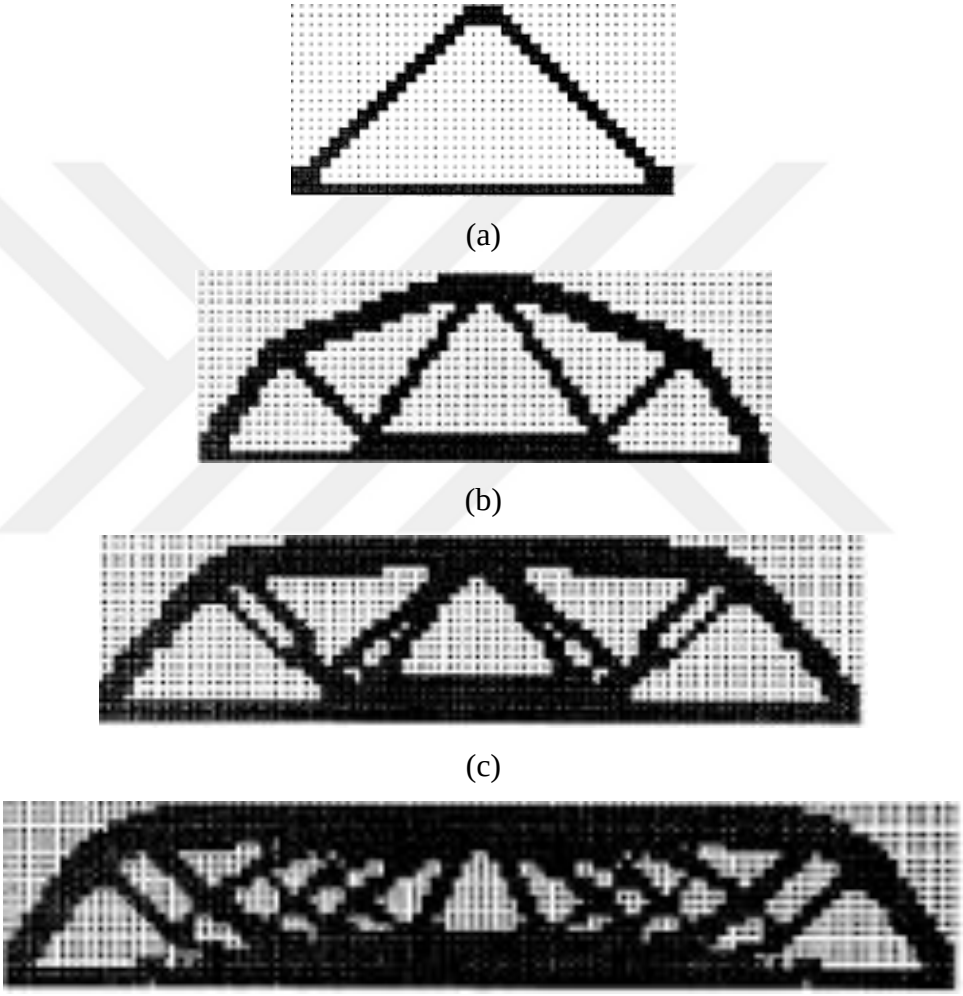


Figure 4.7: The geometry of Rectangle Example 4 (Liang et al., 2000)



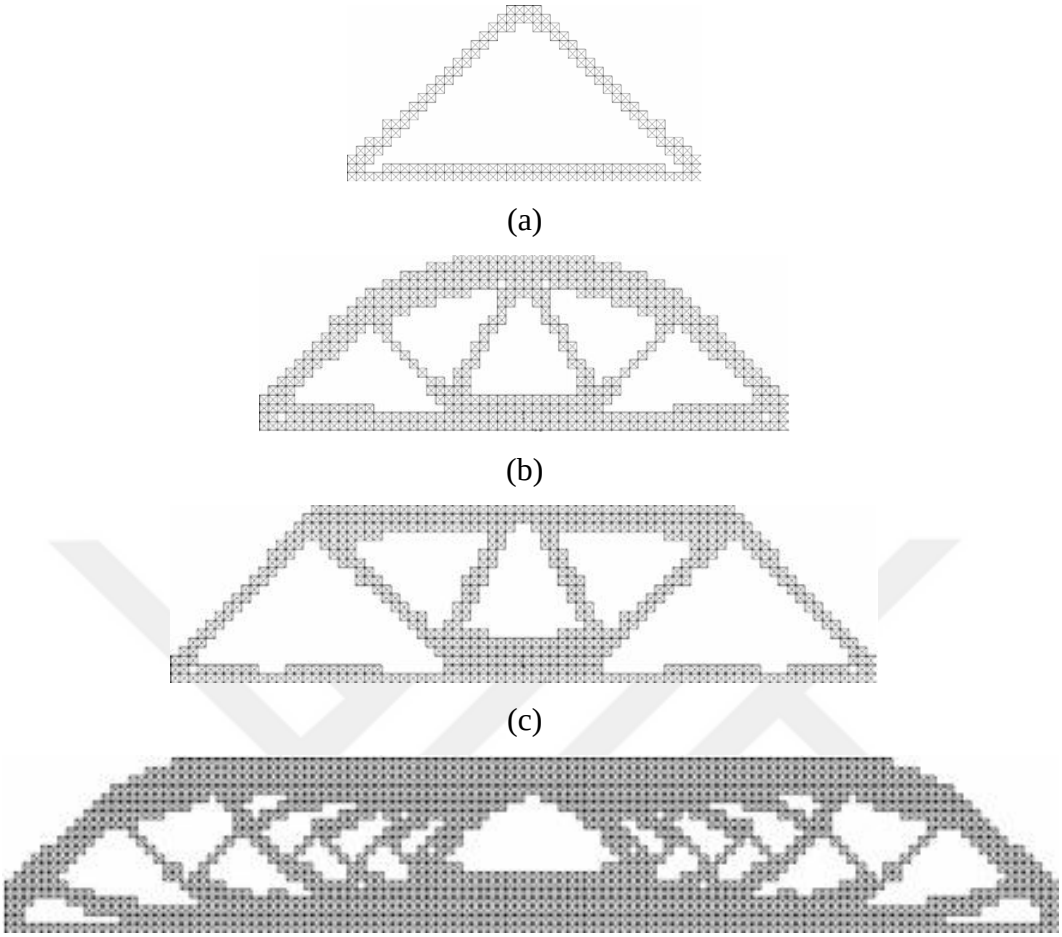
(a)

(b)

(c)

(d)

Figure 4.8: The element distribution of Rectangle Example 4 at the end of the optimization (Liang et al., 2000): (a) (L/D = 2); (b) (L/D = 3); (c) (L/D = 4); (d) (L/D = 5)



(a)

(b)

(c)

(d)

Figure 4.9: The element distribution of Rectangle Example 4 at the end of the optimization (Kwak and Noh, 2006): (a) (L/D = 2); (b) (L/D = 3); (c) (L/D = 4); (d) (L/D = 6)

4.1.2 Real-Life Examples

Real-life examples are analyzed to evaluate the candidate finite element models and their overall performance in practical applications. Additionally, the goal of these analyses are to observe the response of the elements to different mesh sizes and configurations.

The cases chosen here are trapezoidal-shaped hammerhead pier caps. Hence, these domains are not discretized using structured rectangular mesh. The reinforced concrete member is discretized into quadrilateral finite elements. These elements are then modeled as 4-node plane stress elements or further discretized into 1-dimensional elements to be modeled as truss elements. To discretize a trapezoidal shape into quadrilateral elements, two mesh strategies are used: mesh 1 and mesh 2.

In mesh 1, the trapezoidal shape is divided into three quadrilateral shapes by drawing two lines from the bottom corners to the top of the hammerhead shape. In mesh 2, the trapezoidal shape is divided into two quadrilateral shapes by drawing a line from side corner to the other. The real-life examples are not discretized into identical squares due to their hammerhead shapes. Each quadrilateral element has different sizes in lengths and interior angles. Hence, parametric formulation is used to make sure that the variation of element sizes are minimum. The mesh refinements are then determined by applying a mesh size factor. More information regarding the mesh of the real-life examples can be found in Appendix A.1.3.

The geometric and material properties of Pier Cap 1, 2, and 3 are taken from Dawood and Abdul-Razzaq (2021). The dimensions of these three models are identical except for their length. The length values of Pier Cap 1, 2, and 3 are 810, 1180, and 1550 mm, respectively. Figure 4.10 illustrates Pier Cap 1 (Dawood and Abdul-Razzaq, 2021). All dimensions are in millimeters. P is 2000 kN. The proposed Strut-and-Tie Model by Dawood and Abdul-Razzaq (2021) is also shown in Figure 4.11.

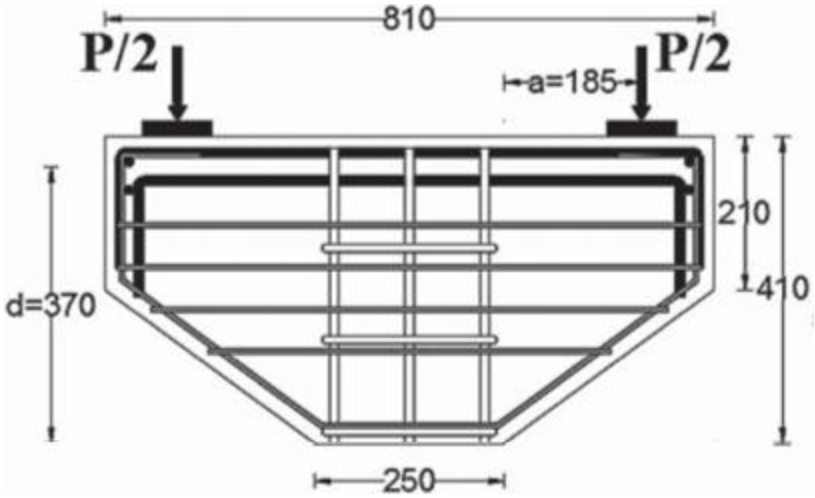


Figure 4.10: The geometry of Pier Cap 1 (Dawood and Abdul-Razzaq, 2021)

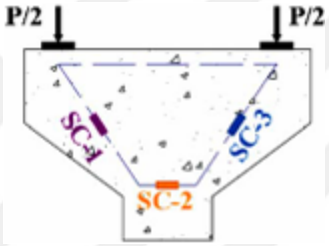


Figure 4.11: The proposed Strut-and-Tie Model of Pier Cap 1 (Dawood and Abdul-Razzaq, 2021)

The geometric and material properties of Pier Cap 4 are taken from Bedru (2014). Figure 4.12 illustrates the dimensions of Pier Cap 4 and Figure 4.13 shows the external loads and the proposed Strut-and-Tie Model (Bedru, 2014). All dimensions are in meters.

Figure 4.12: Dimensions of Pier Cap 4 (Bedru, 2014)

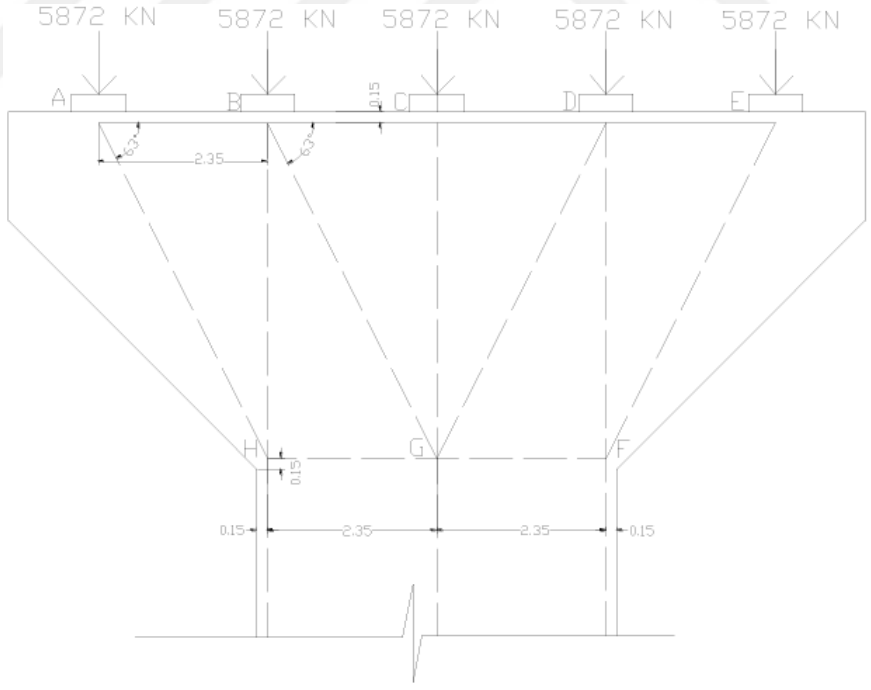


Figure 4.13: External Loadings and the proposed Strut-and-Tie Model of Pier Cap 4 (Bedru, 2014)

The geometric and material properties of Pier Cap 5 are taken from Mitchell and Collins (2013). Figure 4.14 illustrates the dimensions of Pier Cap 5 (Mitchell and Collins, 2013). Figure 4.15 shows the Strut-and-Tie Model proposed by Mitchell and Collins (2013).

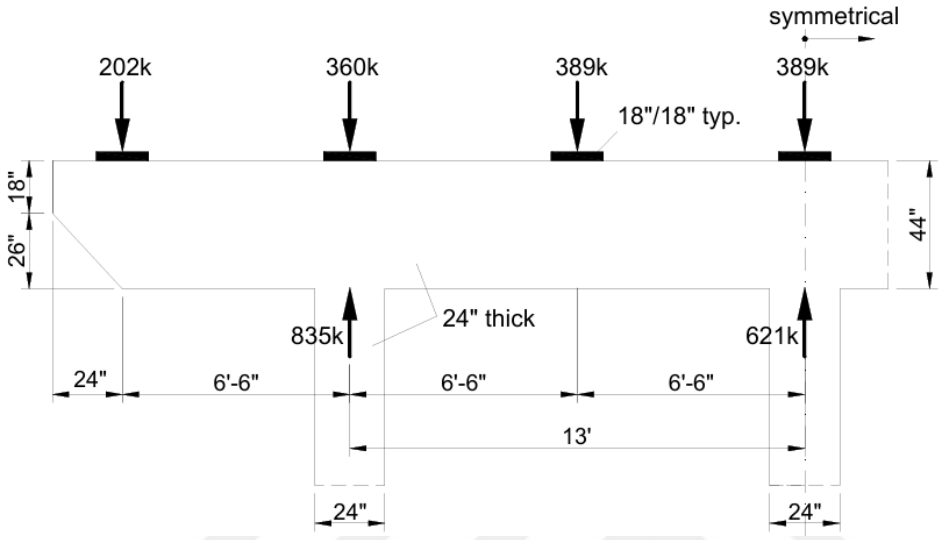


Figure 4.14: Dimensions of Pier Cap 5 (Mitchell and Collins, 2013)

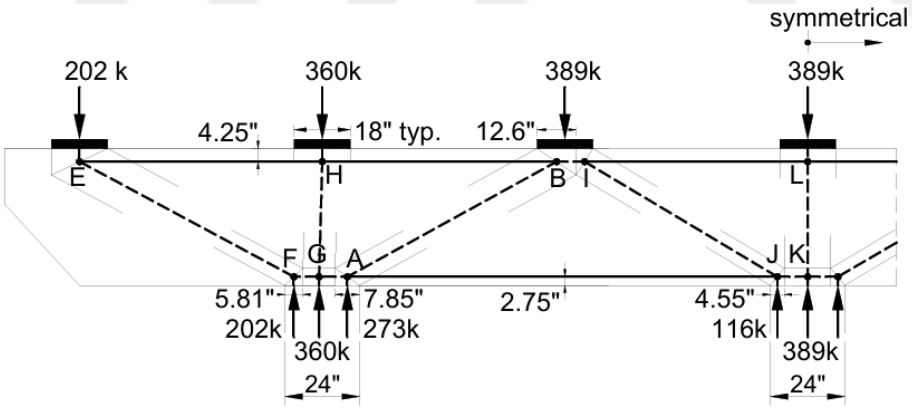


Figure 4.15: The proposed Strut-and-Tie Model of Pier Cap 5 (Mitchell and Collins, 2013)

4.2 Results

4.2.1 Truss Model

The first preliminary study focused on the recently suggested Truss Model, as its application in the Evolutionary Structural Optimization method is not encountered in the literature. The Strut-and-Tie Models consist of truss elements. Hence, the Truss Model is suggested to discretize the reinforced concrete member with truss elements. If a refined element distribution can be achieved at the end of the optimization, the model can be used directly as a Strut-and-Tie Model. A pre-testing phase was deemed necessary to assess whether the Truss Model was a viable and valid candidate for further testing with the full set of examples.

The truss element proved to be a challenging choice for optimization problems due to its inherent complexities. The Truss Model is prone to yield singularity warnings and errors in the linear static analyses within the optimization process. Removal of truss elements has the risk of creating an unbalanced structure. There is also the possibility that the remaining element distribution at the end of an iteration can do rigid body motion. Despite these complications, after implementing precautionary measures to prevent singularities during the optimization process, the truss element demonstrated a degree of success in identifying the load path for a simple hammerhead pier cap.

The stress-based evolutionary structural optimization is employed for the evaluation of the Truss Model. The Rejection Ratio (RR) and the Evolution Rate (ER) are set to 1%. The Steady State Ratio (SS) is 50%. At each iteration, redundant truss elements are eliminated from the structure to achieve a level of stress homogeneity. Once all the truss element stresses are in the same threshold, which is denoted by RR, RR is increased by ER and the iterations start over. The optimization process is to be continued until the RR is equal to SS. At every iteration where redundant

elements are removed, the axial stresses are plotted. The current Rejection Ratio, and the iteration number within the RR is given in the graphs.

The Truss Model was unable to generate an adequate element distribution to delineate a meaningful load path for Sample Problems. For reference, the Rectangle Example 1 Medium Mesh result is presented below. The initial axial stress diagram of Rectangle Example 1 is given in Figure 4.16. Figure 4.17 shows the remaining truss element distribution of medium mesh of Rectangle Example 1 model.

The optimization process continued until all the element stresses in the model were within 3% of the maximum stress value. The analysis cannot continue further due to singularities. A refined element distribution that can directly be used as a Strut-and-Tie Model is not generated, and the number of remaining elements is still too high to distinguish a meaningful load path within the reinforced concrete structure.

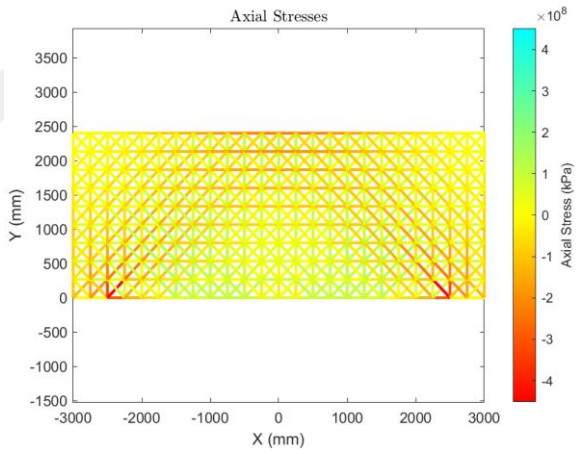


Figure 4.16: Rectangle Example 1- Medium Mesh

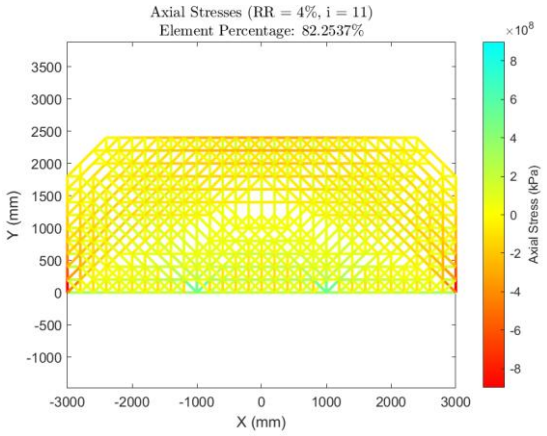


Figure 4.17: Rectangle Example 1- Medium Mesh: Truss Model Result

Since a successful result cannot be obtained from Sample Problems, Pier Cap 1 with a medium mesh resolution was selected for the initial testing of Truss Model. The Strut-and-Tie Model suggested by Dawood and Abdul-Razzaq (2021) is shown in Figure 4.18. The axial stress distribution of the remaining elements at the end of the optimization is shown in Figure 4.19.

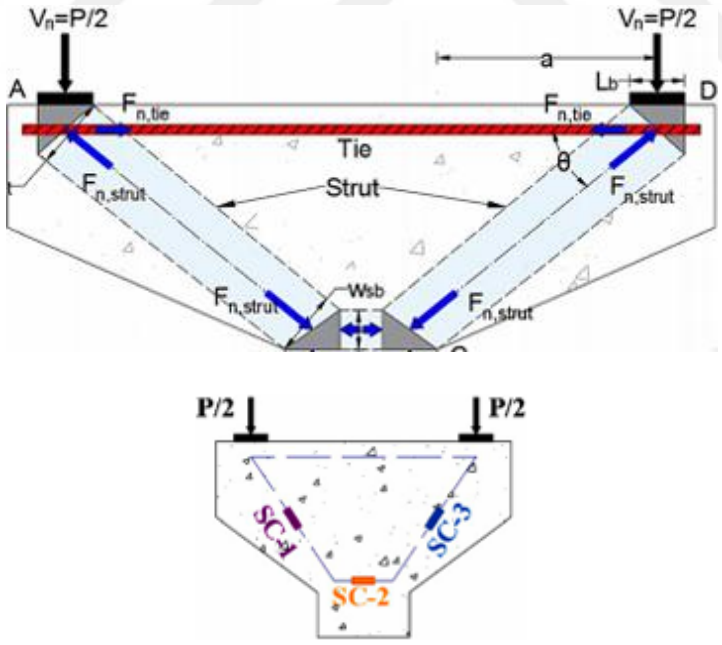


Figure 4.18: Proposed Strut-and-Tie Model in Pier Cap (Dawood and Abdul-Razzaq, 2021)

Two compression regions are accumulated at each side of the pier cap, while the tensile region is concentrated at the top of the structure. The remaining element distribution shows consistency with the Strut-and-Tie Model proposed by Dawood and Abdul-Razzaq (2021). Although there are some scattered truss element remnants in the middle, critical compressive zones can be distinguished. The element distribution at the compression regions demonstrates the bottle-shaped behavior of a strut element. Tensile elements within the strut area can be seen. One of the reasons for proposing the Truss Model was to generate a refined truss element distribution at the end of an optimization process, one that can be directly used as a Strut-and-Tie Model.

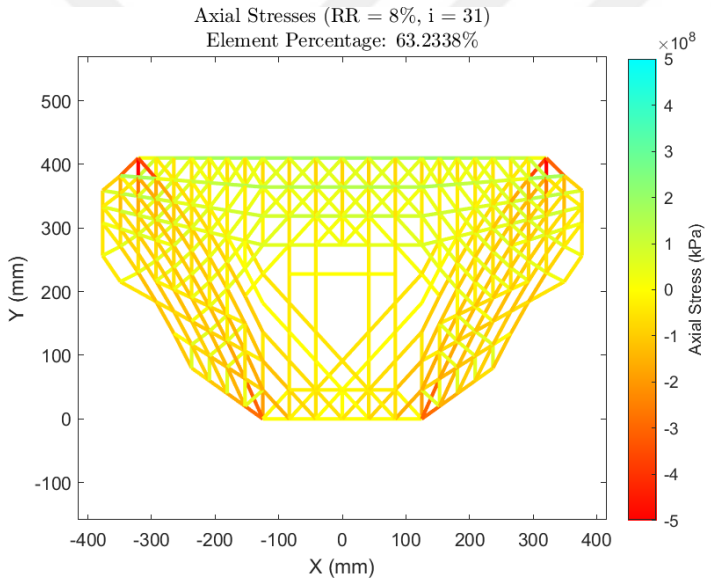


Figure 4.19: Axial Stresses of the Optimized Pier Cap 1- Medium Mesh

However, this initial result indicates that a post-processing routine is still necessary to build a Strut-and-Tie Model. To further stabilize the model during the optimization process, an element deletion limit was introduced. By slowing down the elimination of elements, this measure aims to delay or prevent matrix singularity until a meaningful load path emerges. The element deletion limit is defined by the parameter, “Element Removal Ratio (ERR)”, which represents the ratio of the maximum number of elements that can be removed in a single iteration to the total

number of elements in the structure at that stage. ERR is set to 1%. The results of Pier Cap 1 with the mentioned modifications are presented in Figure 4.20. The resulting structure shows that by employing a scalar limit to the number of elements that can be deleted per iteration, the symmetry of the element distribution is distorted. Elements are scattered towards one side of the reinforced concrete member. It can be seen that the compressive element groups on the right is thicker than those on the left. Also, there are additional perpendicular compression trusses in the middle of the remaining element distribution, which are also accumulated towards the right of the pier cap.

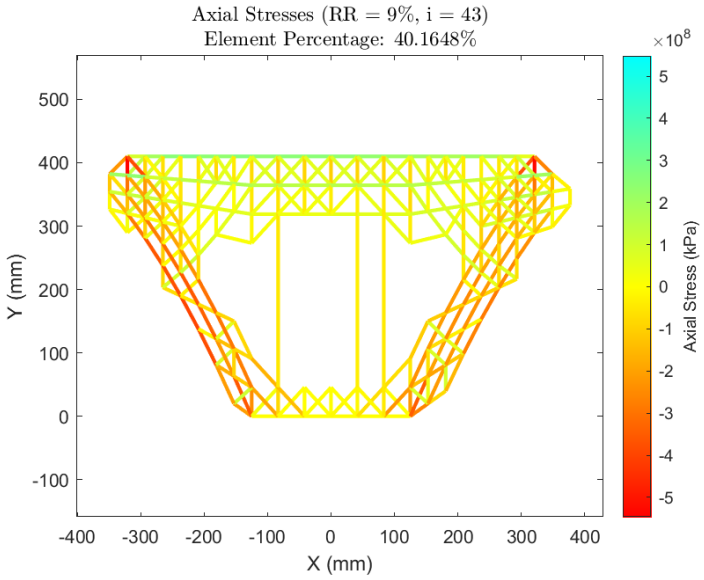


Figure 4.20: Pier Cap 1- Medium Mesh 1 (with ERR)

Hence, the rest of the optimization tests are conducted without an element removal cap. The tests are continued with coarse meshes of Pier Cap 1, 2, and 3. This time, the results of mesh 1 and mesh 2 models for each example are compared to see Truss Model's sensitivity to different types of mesh.

Pier Cap 1

The axial stress diagram at the beginning of the optimization and the resulting element distribution at the end of the optimization procedure of Pier Cap 1 Coarse Mesh 1 and 2 are presented in Figure 4.21.

Dawood and Abdul-Razzaq (2021) proposed the same Strut-and-Tie Model (Figure 4.18) for Pier Caps 1, 2, and 3. The model depicts two strut elements carrying the external force to the bottom of the structure. A tie element connects these struts at the top of the pier cap. It is examined whether the remaining element concentration will point out to similar compression and tensile regions within the reinforced concrete member, using two mesh strategies.

The remaining element distribution of the Pier Cap 1 Mesh 1 model (Figure 4.21 (c)) appears scattered. However, the critical internal force flow can still be traced through the truss elements. This model provides insight into the compressive and tensile regions within the reinforced concrete structure. However, the element distribution is not sufficiently refined to clearly define a Strut-and-Tie Model. The Pier Cap 1 Mesh 2 model (Figure 4.21 (d)) can go further in the optimization process to a point where a single truss element remains at the compressive regions. As discussed at the beginning of this section, Truss Model is suggested expecting to end up with a refined element distribution that can directly be used as a Strut-and-Tie Model. In Pier Cap 1 Mesh 2 model, strut elements can be directly employed as struts in a Strut-and-Tie Model.

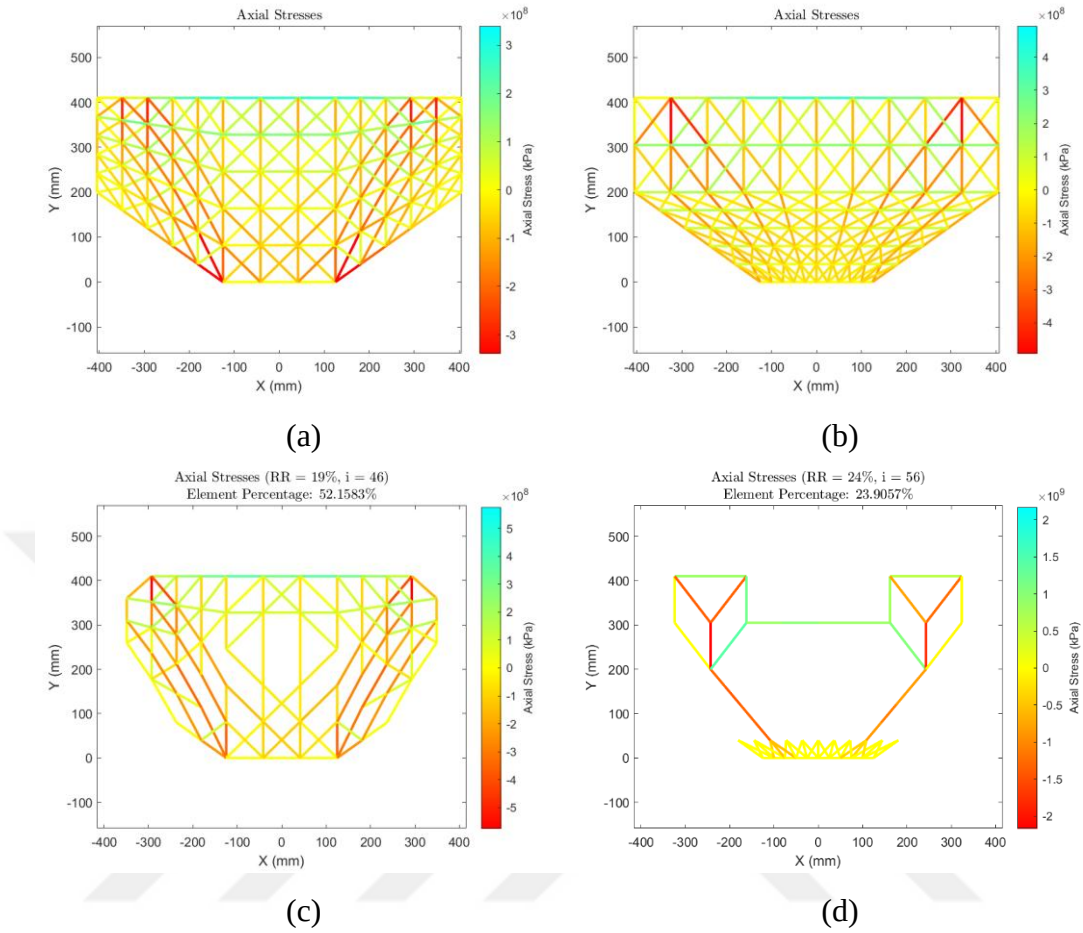


Figure 4.21: Pier Cap 1- Coarse Mesh: (a) Initial Stress Diagram of Mesh 1 (b) Initial Stress Diagram of Mesh 2 (c) Mesh 1 Result; (d) Mesh 2 Result

Pier Cap 2

The axial stress diagram at the beginning of the optimization and the resulting element distribution at the end of the optimization procedure of Pier Cap 2 Coarse Mesh 1 and 2 are presented in Figure 4.22.

In this case, the mesh 1 model (Figure 4.22 (c)) has dense element distribution. The critical compressive and tensile regions can clearly be detected from the high stressed truss elements. However, the remaining number of elements in the model is still high. The element distribution can be used to examine the internal force flow of the reinforced structure, not to guide designing a Strut-and-Tie Model. Figure 4.22 (d)

denotes additional compressive element clusters in the middle of Pier Cap 2 Mesh 2 model. This will also change the configuration of the tie elements.

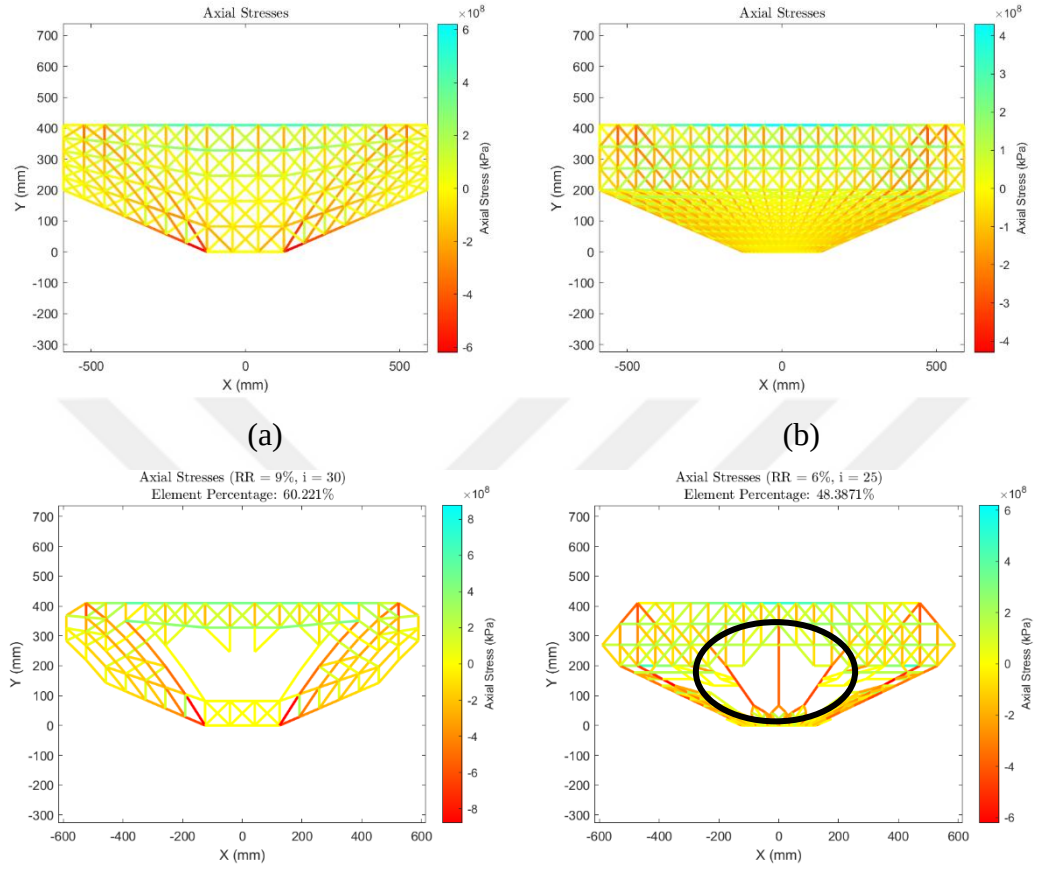


Figure 4.22: Pier Cap 2- Coarse Mesh: (a) Initial Stress Diagram of Mesh 1 (b) Initial Stress Diagram of Mesh 2 (c) Mesh 1 Result; (d) Mesh 2 Result

Pier Cap 3

The axial stress diagram at the beginning of the optimization and the resulting element distribution at the end of the optimization procedure of Pier Cap 3 Coarse Mesh 1 and 2 are presented in Figure 4.23.

The increasing length of the Pier Cap 3 meant an increase in the slenderness of the structure. Pier Cap 3 Mesh 1 model (Figure 4.23 (c)) copes with this by introducing an additional tensile region below the top of the structure. This tensile region is connected to the supports by another compressive element cluster in the middle.

These groups are highlighted in Figure 4.23 (c). Although the high stresses truss elements give an idea on the critical load distribution, the number of remaining elements in the model is still too much to guide designing a Strut-and-Tie Model.

Figures 4.22 (d) and 4.23 (d) also show that a concentrated element distribution can only be achieved at the bottom of the structure for Mesh 2. The rest of the remaining elements are accumulated to the top of the structure. Mesh 1 models shows a more homogenic element distribution throughout the reinforced concrete member (Figure 4.22 (c) and Figure 4.23 (c)).

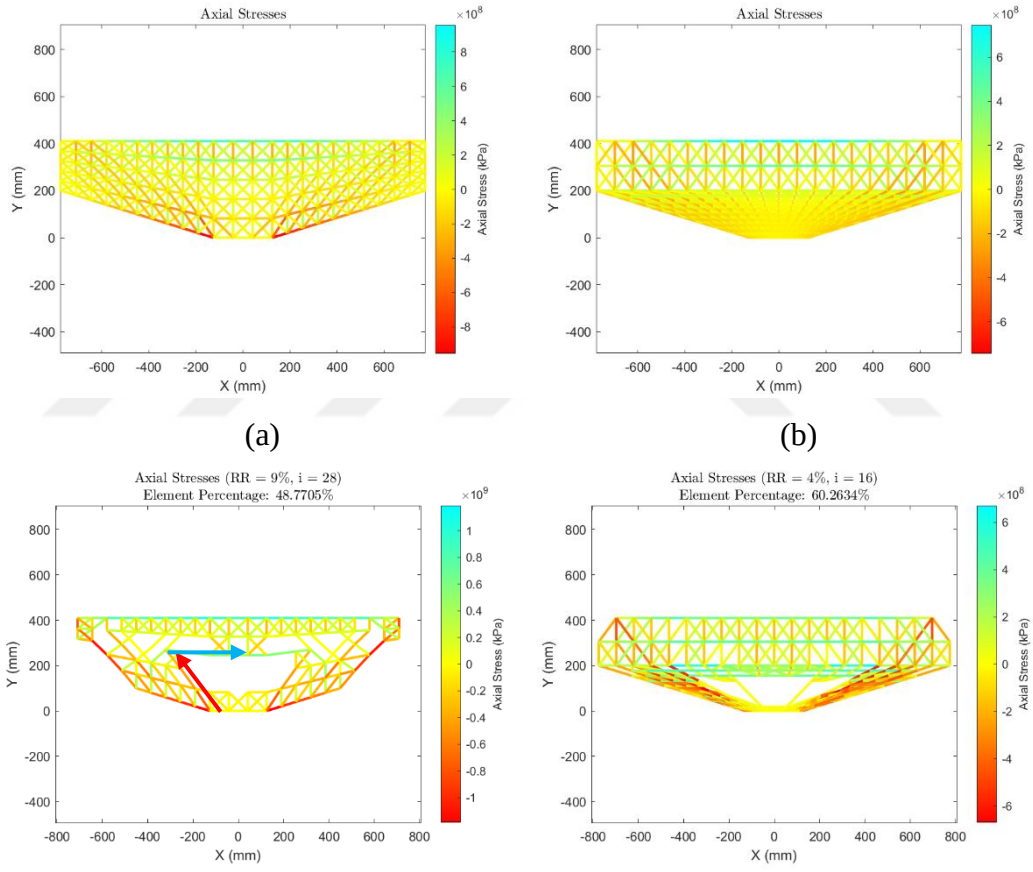


Figure 4.23: Pier Cap 3- Coarse Mesh: (a) Initial Stress Diagram of Mesh 1 (b) Initial Stress Diagram of Mesh 2 (c) Mesh 1 Result; (d) Mesh 2 Result

The preliminary tests of the Truss Model generated inconsistent results among the example models. The proposed model completely failed in giving a stable element

distribution at the end of an optimization procedure with Sample Problems. The results were unable to point out the internal force flow of the structure.

Some of the Real-Life Example models were able to yield successful results. At best, the remaining truss elements were too scattered across the plane to be directly converted into a Strut-and-Tie Model. This necessitates an intermediate algorithm, negating a significant advantage the truss element was expected to offer over the membrane and brick elements.

The Element Removal Ratio (ERR) increased the number of iterations of the optimization process, while not being able to yield consistent positive outcomes. It is also observed that ERR has the potential to disrupt the behavior of a structure, particularly for symmetric load distribution conditions.

Due to the inconsistent nature of the results and the truss elements tendency to produce unstable structures within the optimization procedure, despite the numerous precautions to prevent singularity errors, the Truss Model is deemed as unsuccessful. And it is not found worthy enough to continue further with further testing and development of the element. Hence, the Truss Model is eliminated from this comparative study. The rest of the study is focused on the use of Brick and Membrane Models for forming Strut and Tie Models.

4.2.2 Brick and Membrane Model

Before arriving at the final comparison between the Brick and Membrane Models, a preliminary study was conducted to refine the methodology and validate the suitability of each approach. This preliminary test aims to find the most efficient element elimination criterion for both Brick and Membrane Model, preparing them for their final comparison.

The preliminary study is conducted between stress-based and displacement-based (performance-based) optimization methods to determine the most suitable element

removal criterion for the Membrane and Brick Model. The Sample Problems served as the basis for this preliminary evaluation.

Firstly, the results of stress-based and displacement-based optimization are presented for both Brick and Membrane Models. The displacement-based and stress-based results of Brick and Membrane Models are compared per element model. The Brick and Membrane results are compared with each other as well. Then, the displacement-based optimization is more focused on as the performance index is examined and evaluated in detail. The element distribution at an iteration where the performance index is largest is compared with an element distribution at a different iteration within the same optimization procedure. The goal is to find out whether the performance index can indeed be used as an indicative for the most efficient element distribution.

Finally, the performance index history is recorded and examined for every Sample Problem. This is done so to see if a correlation can be made with the iteration number and the performance index trend. Also, whether the performance index can be used as a stopping criterion is examined.

4.2.2.1 Displacement-based and Stress-based Optimization Results

This section shows the element distributions derived from the displacement-based and stress-based evolutionary structural optimization methods using Sample Models. The performance index parameter is utilized in displacement-based optimization procedure. Hence, the element distribution at the iteration that has the highest performance index is chosen for comparison. The performance index value and the iteration number are displayed in the plot title.

The stress-based results are hand-picked according to some conditions. Usually, an optimization procedure aims for the minimization of the weight of a structure. Hence, the element distribution is chosen from an iteration that is towards the end of the optimization process. As the weight of the structure decreases (as elements are

removed from the model), the remaining element distribution becomes more refined in demonstrating the internal force path in the structure more clearly. Usually, after a certain number of iterations in the optimization process, the compressive and tensile regions begin to be clearly observed. And after the element distribution becomes the clearest, over-deletion happens, and the structure becomes unstable, meaning that there are not enough number of elements left in the model to carry the external loads.

Another principal is to follow the element stress levels. A jump on the overall stress values of elements in the model can indicate the disruption of cluster of elements that were crucial in transferring the internal forces. The current RR value of the element distribution and the iteration number are displayed on the plot title for stress-based results. Opposite to the displacement-based results, the iteration number counts the iteration within the same RR value. So, once all the elements in the model have reached the RR threshold in an iteration, the RR is increased by the ER value, and the iteration number resets as the optimization procedure continues with the new RR value.

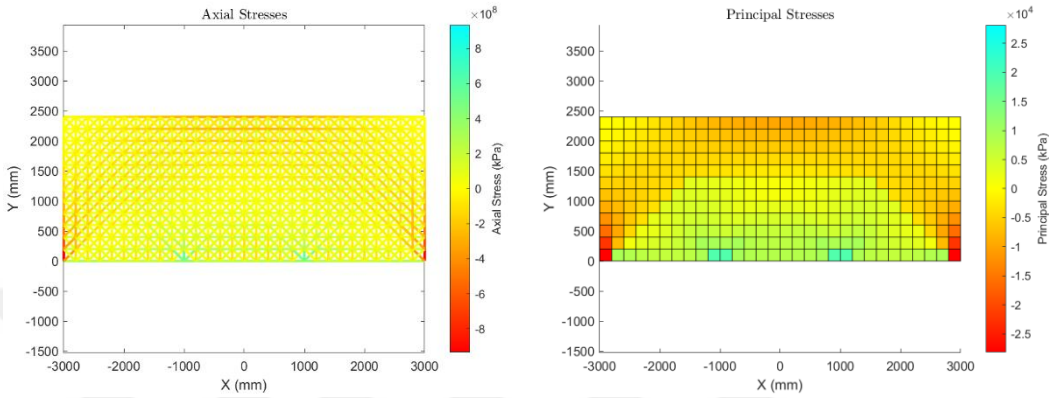
It should also be noted that although the Von Mises stresses are used as element removal criterion for the Membrane Model, the principal stresses of the resulting element distribution are plotted. This is done so to distinguish the compressive and tensile regions.

Rectangle Example 1

The initial axial and principal stress diagrams of the coarse mesh Brick and Membrane Rectangle Example 1 models are shown in Figure 4.24. The optimum element distribution of Rectangle Example 1 obtained by Liang et al. (2000) is presented in Figure 4.25. The optimization results for these models are presented in Figure 4.26.

Liang et al. (2000) used 50 mm rectangular quadrilateral elements to discretize the reinforced concrete structure. In this study, Rectangle Example 1 is discretized with

a coarser mesh, where quadrilateral elements are 200 mm in size. All four optimization methods successfully converge to a refined element distribution with fewer elements. The Rejection Ratio and Evolution Rate are set to 1% for both stress-based and displacement-based optimization models.



(a) (b)

Figure 4.24: Rectangle Example 1- Coarse Mesh: (a) Brick Model; (b) Membrane Model

Previous results showed that imposing an element removal limit during optimization can disrupt symmetry and lead to incorrect solution convergence. Therefore, no such limit is applied in this study. As a result, solutions are obtained within a few iterations. On average, optimization runs take less than a minute, excluding time spent plotting graphs.

Brick and Membrane Models produce similar results (Figure 4.26) across different element elimination criteria. All four optimization methods provide clearer tensile element distribution compared to the optimum solution from Liang et al. (2000) (Figure 4.25). The tensile elements are especially well-defined, making it easier to identify tie locations for Strut-and-Tie Model development.

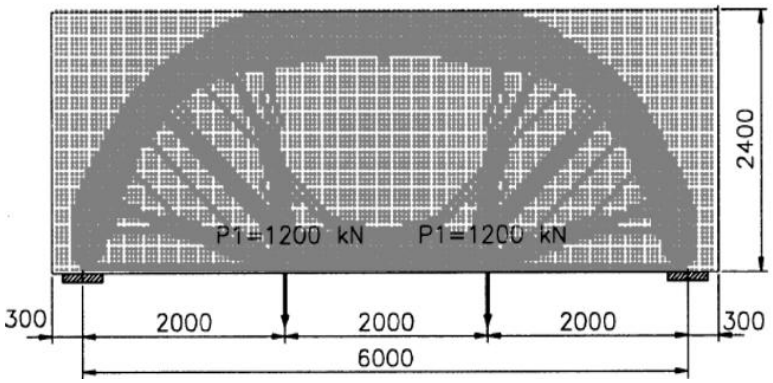
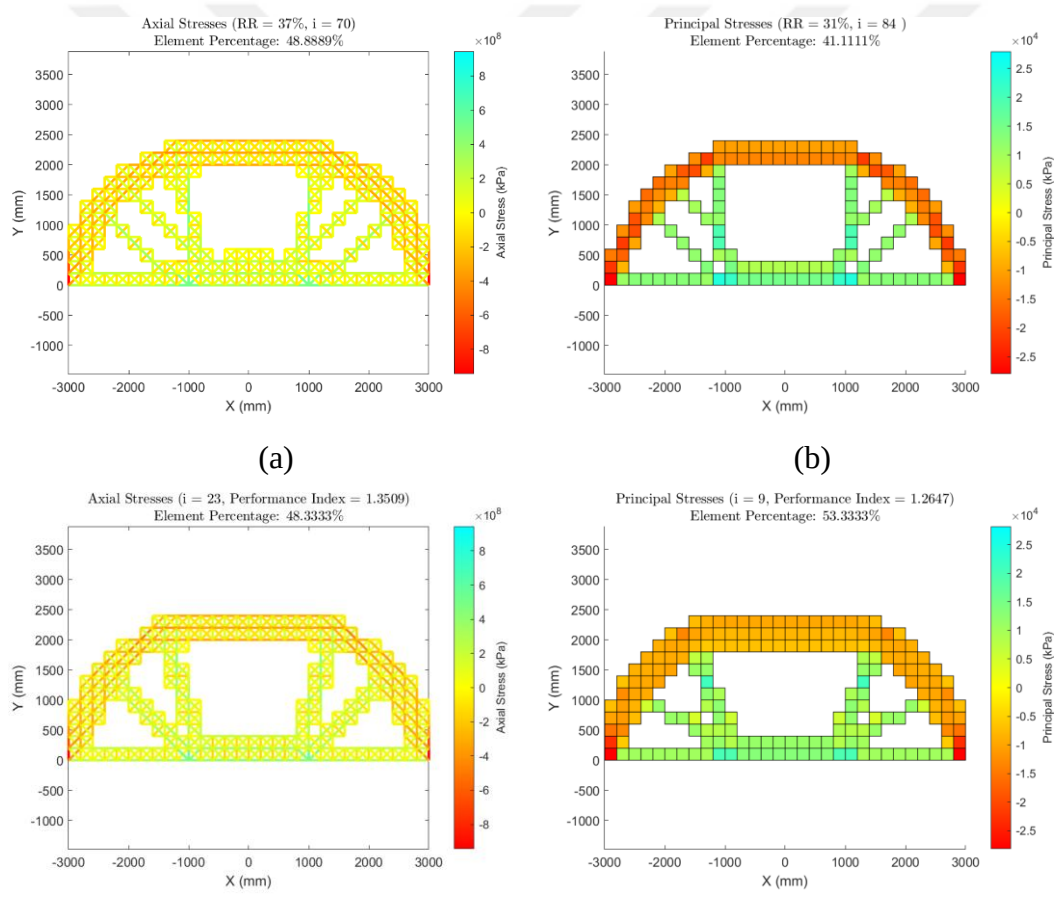


Figure 4.25: The optimum element distribution of Rectangle Example 1 (Liang et al., 2000)



(a) (b)

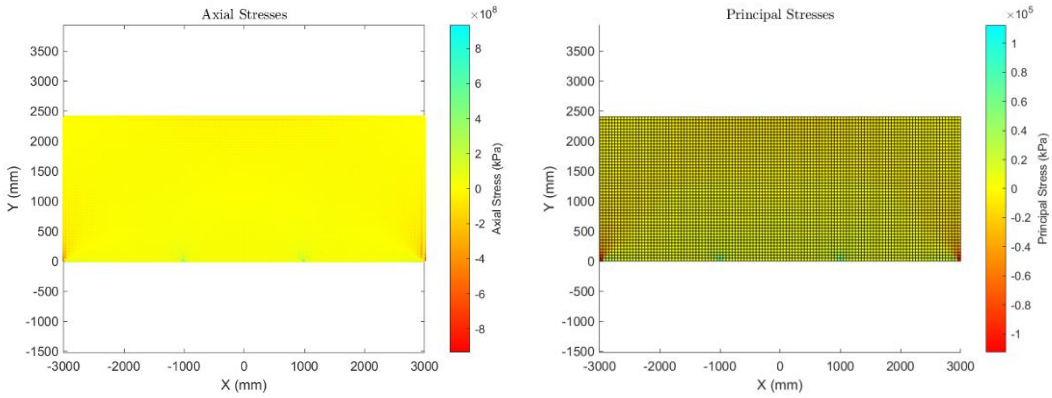
(c) (d)

Figure 4.26: Rectangle Example 1- Coarse Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

The medium and fine mesh results also provide a refined element distribution, effectively depicting internal force flow within the reinforced concrete structure. All four methods clearly distinguish compressive and tensile element groups, which can serve as a guide for developing a Strut-and-Tie Model. A constant Rejection Ratio (RR) and Evolution Rate (ER) of 1% is used for stress-based optimization across different examples and mesh refinement levels. However, for displacement-based optimization, RR and ER need to be adjusted as mesh refinement increases. Using the same values leads to excessive element removal in the early iterations, causing instabilities that prematurely terminate the optimization without reaching an optimal element distribution. In contrast, stress-based optimization can safely remove large numbers of redundant elements in the initial iterations without disrupting the process.

For the medium mesh Rectangle Example 1, RR and ER are set to 0.1%, while for the fine mesh model, they are reduced to 0.01% at the beginning of the displacement-based optimization process. This adjustment also impacts the number of iterations and the runtime of the optimization, particularly in fine mesh models. The fine mesh stress-based optimization procedure took an average of 30 minutes, whereas the displacement-based optimization required over 30 hours.

The initial axial and principal stress diagrams of the fine mesh Brick and Membrane Rectangle Example 1 are shown in Figure 4.27.



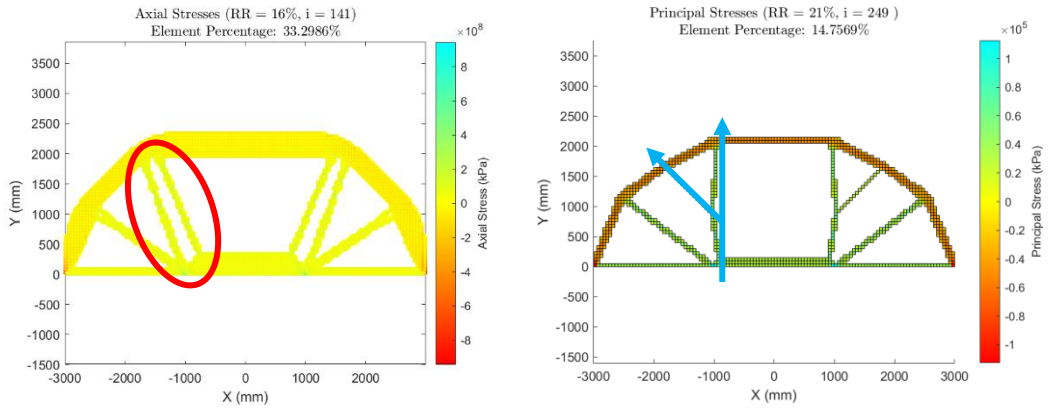
(a) (b)

Figure 4.27: Rectangle Example 1- Fine Mesh: (a) Brick Model; (b) Membrane Model

It should be noted that both models achieved a more refined element distribution and clearly demonstrated internal force flow compared to the configuration obtained by Liang et al. (2000) (Figure 4.25). The weight minimization objective was also met, as the optimal model configurations contain fewer remaining elements. Both element distributions are straightforward enough to be used as a guide for determining a Strut-and-Tie Model.

The stress-based solutions of the Brick and Membrane Models exhibit some differences. The Brick Model clearly shows two tensile element groups on each side of the reinforced concrete structure, as highlighted in Figure 4.28. The presence of a double line of tensile elements in the middle suggests that the tie element in this region ought to carry a higher tensile force.

In contrast, the Membrane Model displays additional tensile element clusters branching from the perpendicular tensile groups in the middle, as highlighted in Figure 4.28. However, these branching tensile elements do not contribute to the model's stability. Also, such a truss configuration would likely cause numerical issues, as it involves a truss element directly connected to another.



(a) (b)

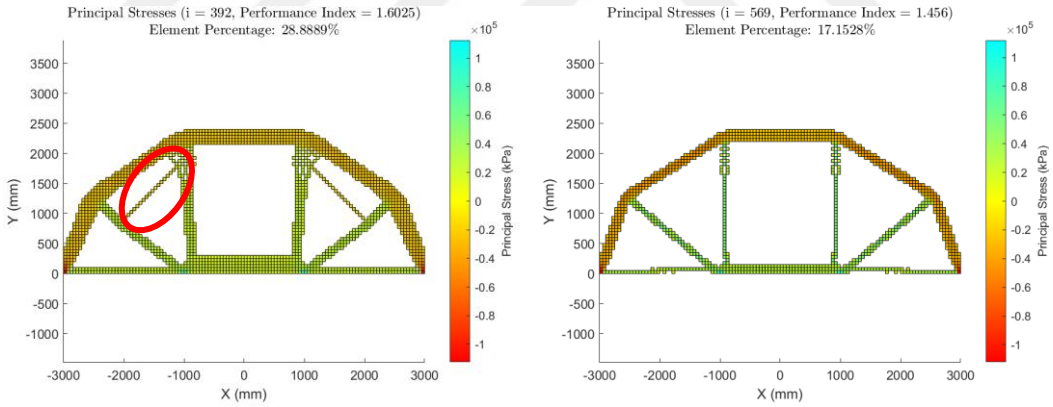
Figure 4.28: Rectangle Example 1- Stress-based Fine Mesh Results: (a) Brick Model; (b) Membrane Model

It is also observed that displacement-based optimization does not always produce the most optimal element distribution with the highest performance index value. The

optimal element distribution with the highest performance index is presented in Figure 4.29, along with an alternative configuration obtained during the same optimization process.

The highest performance index model contains redundant compressive element clusters between the two tensile regions, as highlighted in Figure 4.29. While compressive elements in this region were being removed during optimization, some remained in the structure, becoming isolated from the main arch-shaped compressive cluster above them. The performance index peaked during one of these iterations.

Although these isolated compressive elements connect the tensile regions, they do not contribute to the model's stability. A strut element directly connected to tie elements would create instability in the truss model. While the element distribution helps visualize internal force flow and guide Strut-and-Tie Model development, a clearer configuration with fewer elements is also available within the optimization process, as shown in Figure 4.29.



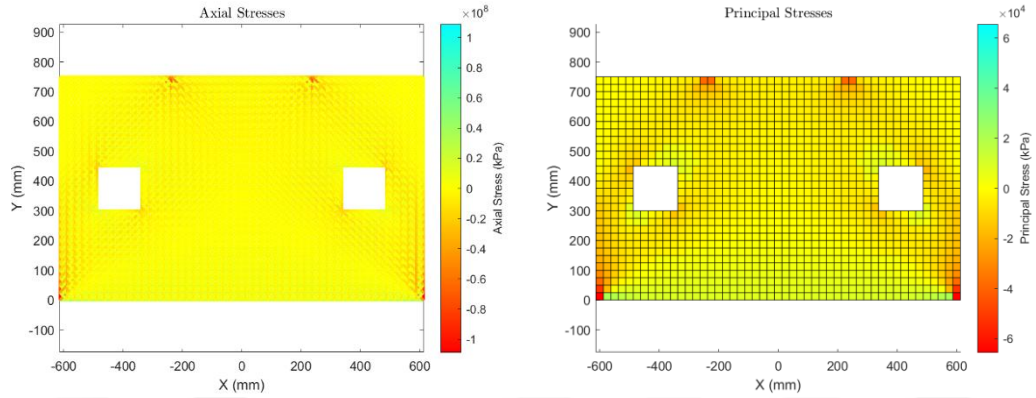
(a) (b)

Figure 4.29: Rectangle Example 1- Displacement-based Fine Mesh Membrane Model Results: (a) Highest Performance Index Model; (b) Alternative Model

Rectangle Example 2

The initial axial and principal stress diagrams of Brick and Membrane Rectangle Example 2 models are shown in Figure 4.30. The optimum element distribution of

Rectangle Example 2 obtained by Liang et al. (2000) is presented in Figure 4.31. The optimization results for these models are presented in Figure 4.32.



(a) (b)

Figure 4.30: Rectangle Example 2: (a) Brick Model; (b) Membrane Model

In this study, Rectangle Example 2 is discretized using 25 mm rectangular quadrilateral elements, consistent with Liang et al. (2000). The stress-based optimization routines successfully achieved the optimum element distribution with RR and ER set to 1%. However, the displacement-based method required a slower process, reducing RR and ER to 0.1% to prevent excessive element removal in the early stages of optimization.

All four methods produce an element distribution that clearly defines the internal force flow within the structure. Refined clusters of compressive and tensile elements are formed, providing a reliable guide for Strut-and-Tie Model development. The displacement-based optimization models (Figure 4.32 (c) and (d)) result in fewer remaining elements. However, except for the displacement-based Brick Model (Figure 4.32 (c)), the locations of compressive and tensile elements are nearly identical.

The stress-based Brick and Membrane Models (Figure 4.32 (a) and (b)) exhibit similar element distributions. The remaining element configuration closely matches that of Liang et al. (2000) (Figure 4.31), with an improvement— the scattered

element distribution near the bottom corners observed in Liang et al. (2000) is not repeated here.

The stress-based Brick Model (Figure 4.32 (a)) better meets the weight minimization objective, as it contains fewer remaining elements. Additionally, the stress levels across the structure are more uniform compared to the stress-based Membrane Model (Figure 4.32 (b)).

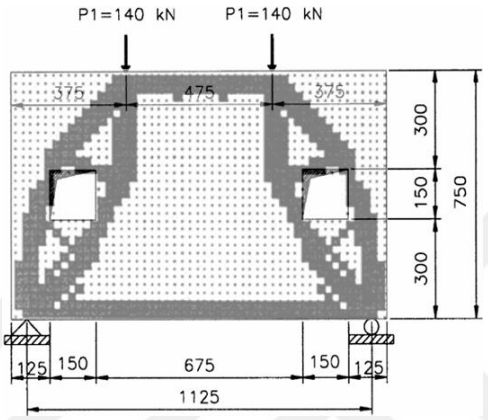


Figure 4.31: The optimum element distribution of Rectangle Example 2 (Liang et al., 2000)

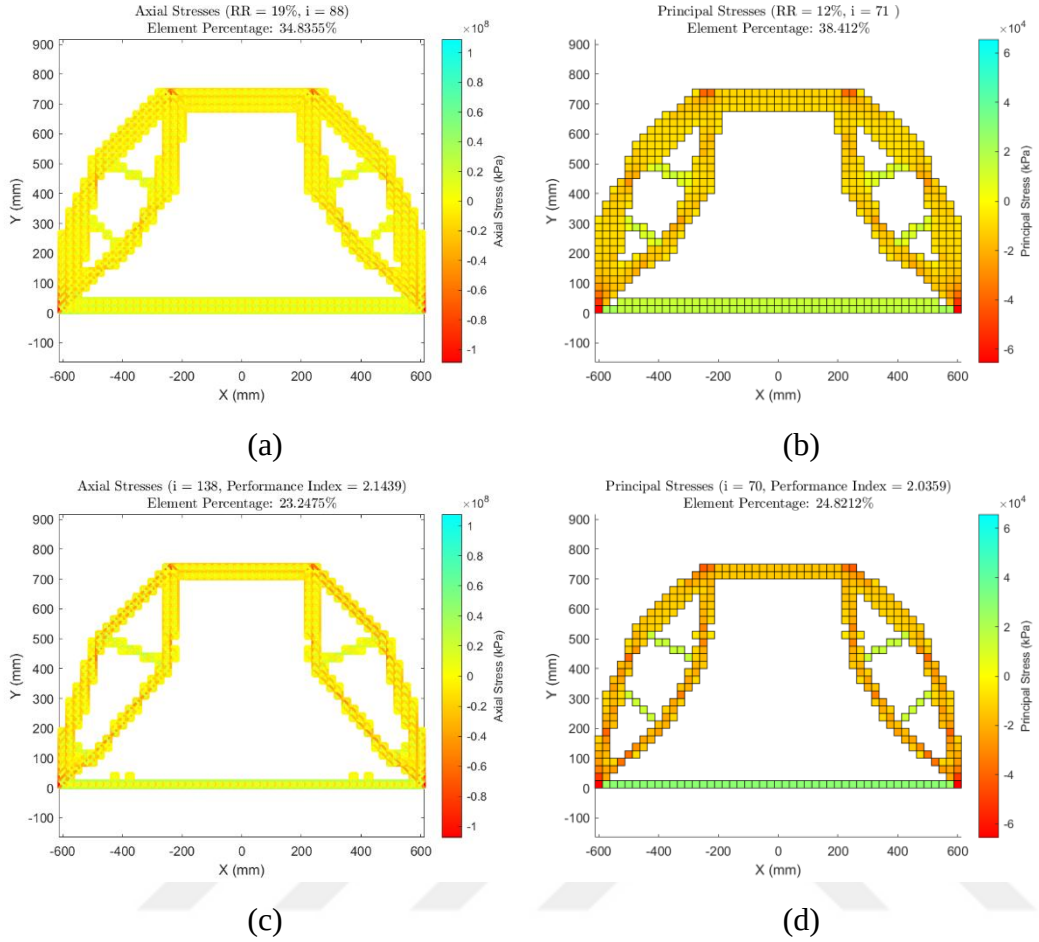


Figure 4.32: Rectangle Example 2 Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

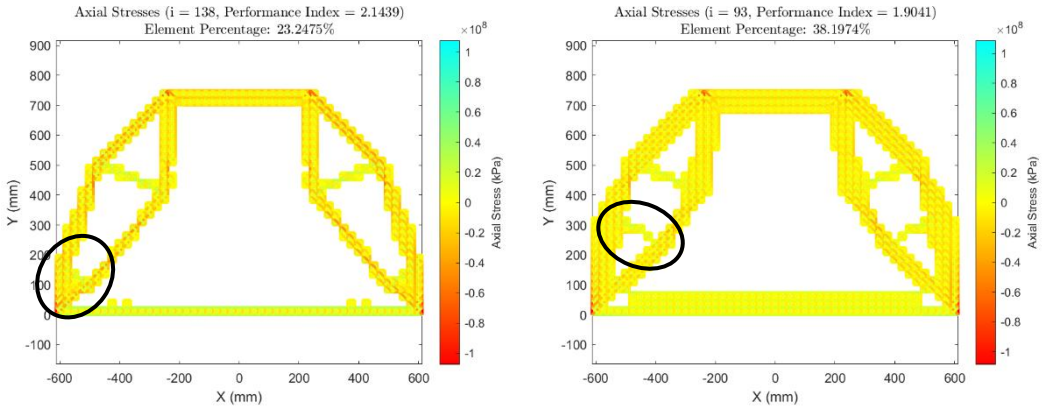
The optimal element distribution with the highest performance index is shown in Figure 4.33, along with an alternative configuration obtained during the same optimization process. The Brick Model produces the highest performance index result that differs from the optimum element distribution obtained by the other three approaches.

During optimization, the tensile element groups beneath the openings were removed at a certain stage. Instead, a shorter tensile element group emerged closer to the bottom of the structure, connecting the compressive elements in that region, as highlighted in Figure 4.33 (a). An alternative element distribution appears in an

earlier iteration of the same optimization routine. This configuration, similar to the results of other methods, retains the tensile element groups directly beneath the openings, as highlighted in Figure 4.33 (b). Considering the length of the compressive and tensile element groups, a truss model based on the alternative configuration (Figure 4.33 (b)) would be more stable and efficient.

As the optimization progresses rapidly, the tensile elements beneath the openings are prematurely deleted. Since the number of elements before this stage is high, the performance index is initially calculated as lower. Only in later iterations, when more elements are removed, does the performance index reach its maximum.

This once again highlights the limitations of displacement-based optimization compared to stress-based optimization. RR and ER values must be adjusted for each case to prevent excessive element removal, whereas stress-based optimization consistently converges to the optimal element distribution with fixed RR and ER values. Additionally, the direct influence of the remaining element percentage on the performance index calculation can sometimes lead to overlooking an efficient element distribution that emerges earlier in the process.



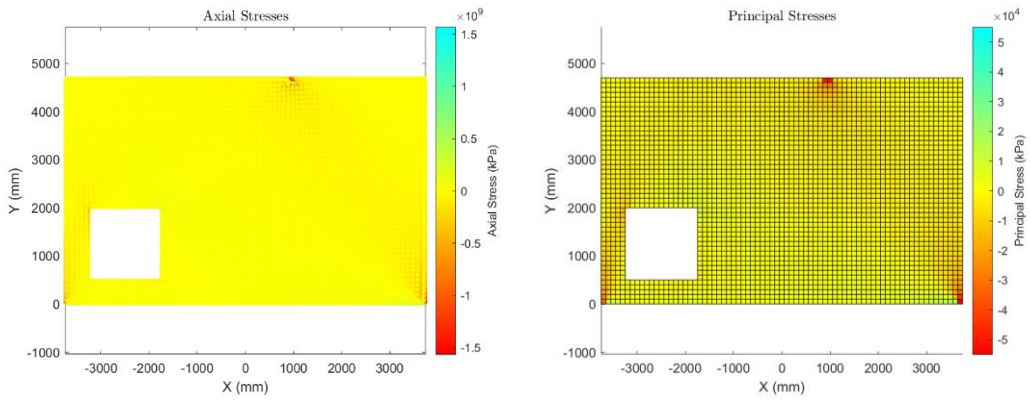
(a) (b)

Figure 4.33: Rectangle Example 2 Displacement-based Brick Model Results: (a) Highest Performance Index Model; (b) Alternative Model

Rectangle Example 3

Along with Liang et al. (2000), Kwak and Noh (2006) also used the Rectangle Example 3 model to test the Brick element. Liang et al. (2000) discretized the reinforced structure with an opening using 100 mm square plane stress elements, while Kwak and Noh (2006) used 100 mm square brick elements.

In this study, Rectangle Example 3 is discretized into 100 mm square quadrilateral elements. The initial axial and principal stress diagrams of the model are shown in Figure 4.34. The optimum element distributions obtained by Liang et al. (2000) and Kwak and Noh (2006) are presented in Figure 4.35. The optimum element distributions of all four models in this study are given in Figure 4.36.



(a) (b)

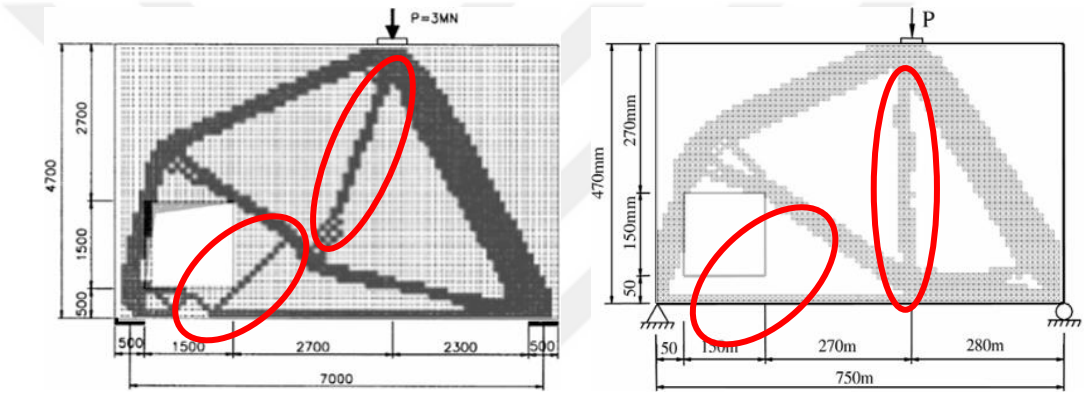
Figure 4.34: Rectangle Example 3: (a) Brick Model; (b) Membrane Model

The displacement-based optimization required slower progression to prevent excessive element removal in the early iterations. To ensure stability and avoid structural failure, the RR and ER values were adjusted to 0.01%. This significantly impacted the optimization speed. While the stress-based optimization is completed in an average of 20 minutes, the displacement-based optimization took over two hours to reach a conclusion.

The results from both stress-based and displacement-based optimizations more closely resemble the optimum element configuration obtained by Kwak and Noh

(2006) rather than Liang et al. (2000). The differences between the solutions of Liang et al. (2000) and Kwak and Noh (2006) are highlighted in Figure 4.35.

In all four approaches, a compression zone originates from the load application point and extends downward perpendicularly. To the left of the structure, the compressive elements follow a deviator path around the opening. A diagonal tensile element group then connects the perpendicular compression zone to the deviatoric compression zone, as highlighted in Figure 4.36 (d). This truss configuration is stable, eliminating the need for an additional element group in the bottom-right corner of the opening.



(a) (b)

Figure 4.35: The optimum element distribution of Rectangle Example 3 by: (a) Liang et al. (2000); (b) Kwak and Noh (2006)

The Membrane Models (Figure 4.36 (b) and (d)) produce identical element configurations across both element elimination criteria. In contrast, the Brick Models (Figure 4.36 (a) and (c)) show differences in the compression zone at the top left of the structure. The displacement-based Brick Model (Figure 4.36 (c)) features a thicker compressive element cluster extending toward the upper-left corner of the opening, while the stress-based Brick Model (Figure 4.36 (a)) refines the compression force flow, dividing it into two thinner compressive clusters.

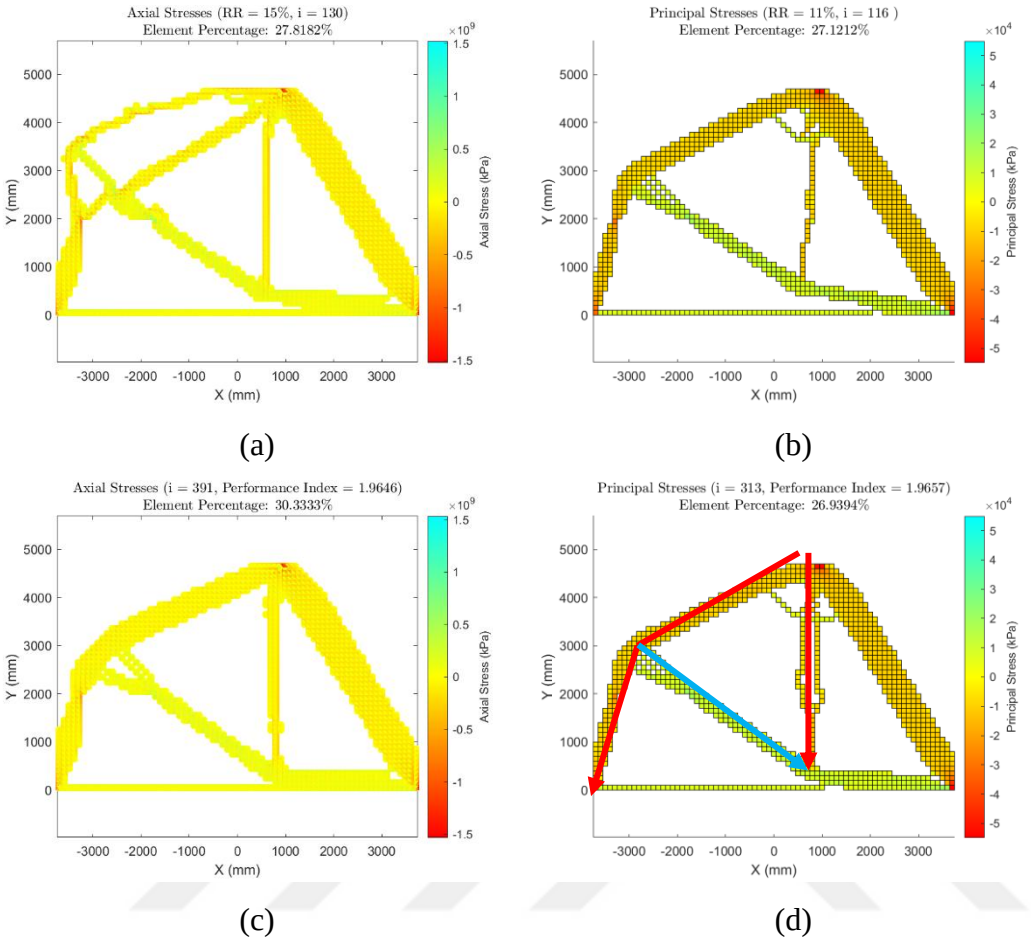


Figure 4.36: Rectangle Example 3 Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

This refinement results in a clearer triangular formation above the opening. This configuration is also consistent with an alternative element distribution obtained by Kwak and Noh (2006) using a coarser mesh. The corresponding element configuration is shown in Figure 4.37, where the triangular region is highlighted.

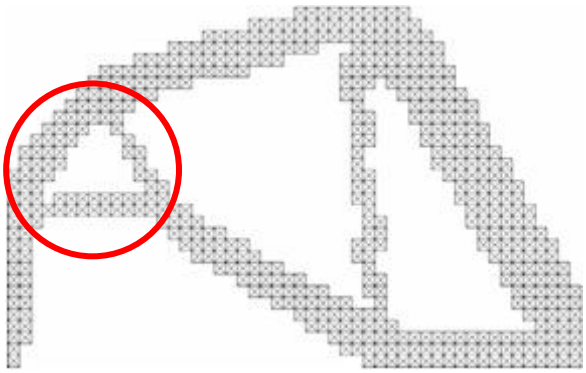


Figure 4.37: Alternative element distribution by Kwak and Noh (2006)

Both Brick Models (Figure 4.36 (a) and (c)) exhibit a triangular truss system above the opening, where the concrete is prone to cracking under the applied load. The stress-based optimization model (Figure 4.36 (a)) identifies an additional strut element that further strengthens the concrete around the opening.

In contrast, both Membrane Models (Figure 4.36 (b) and (d)) do not display this configuration. Instead, a checkerboarding pattern appears at the corresponding location, preventing the formation of distinct elements that define a clear triangular structure. This is shown in Figure 4.38.

Figure 4.38: Checkerboard pattern in Membrane Model

Rectangle Example 4a

The Rectangle Example 4a results were nearly identical across all four optimization approaches and mesh refinement levels. The results also closely resemble the

optimum element configurations obtained by Liang et al. (2000) and Kwak and Noh (2006). Both studies identified two diagonal compressive element fields connected by a tensile element group at the bottom, forming a basic Strut-and-Tie Model.

The element distributions obtained by Liang et al. (2000) and Kwak and Noh (2006) are shown in Figure 4.39. The results of this example can be found in Appendix B and C. To achieve a more refined element distribution, the Steady State Ratio in the coarse mesh stress-based model was increased to 65% due to the low number of elements in the coarse mesh configuration. A higher SS value was necessary to meet the weight minimization objective.

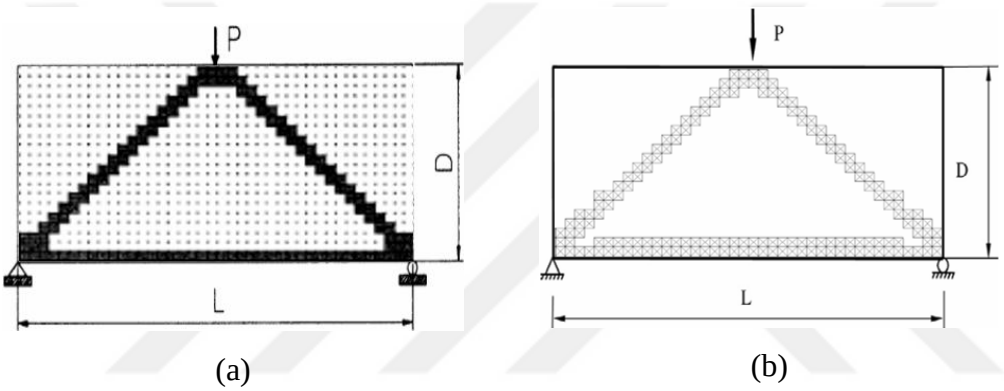
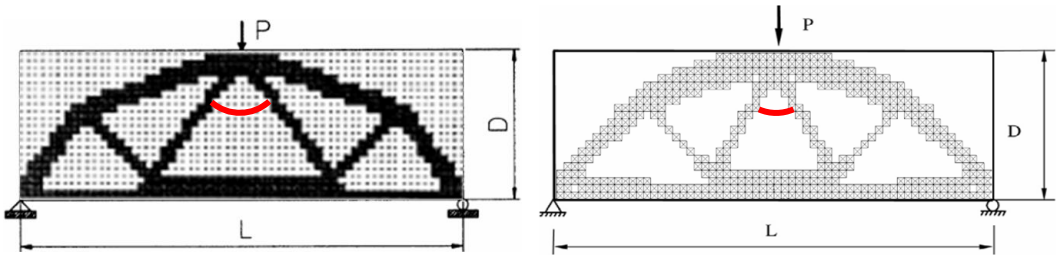


Figure 4.39: The optimum element distribution of Rectangle Example 4a by: (a) Liang et al. (2000); (b) Kwak and Noh (2006)

Rectangle Example 4b

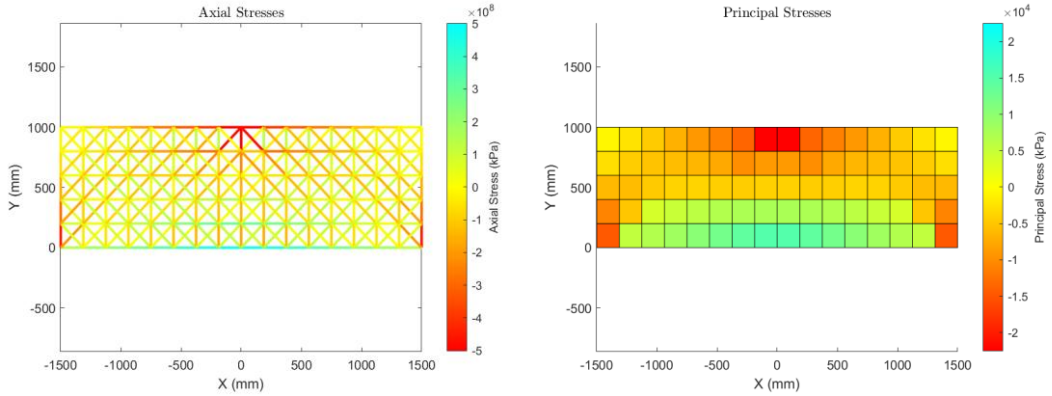
Liang et al. (2000) and Kwak and Noh (2006) obtained similar optimum element distributions for the Rectangle Example 4b. The results show identical configurations, except for the Brick Model, where the compressive element groups in the middle form a narrower angle. These results are presented in Figure 4.40



(a) (b)

Figure 4.40: The optimum element distribution of Rectangle Example 4b by: (a) Liang et al. (2000); (b) Kwak and Noh (2006)

The initial axial and principal stress diagrams for the coarse mesh Rectangle Example 4b structure are shown in Figure 4.41. The coarse mesh results from stress-based and displacement-based optimization using Brick and Membrane elements are presented in Figure 4.42. The RR and ER values are set to 0.1% in displacement-based optimization routines to prevent over deletion of elements in the earlier stages.

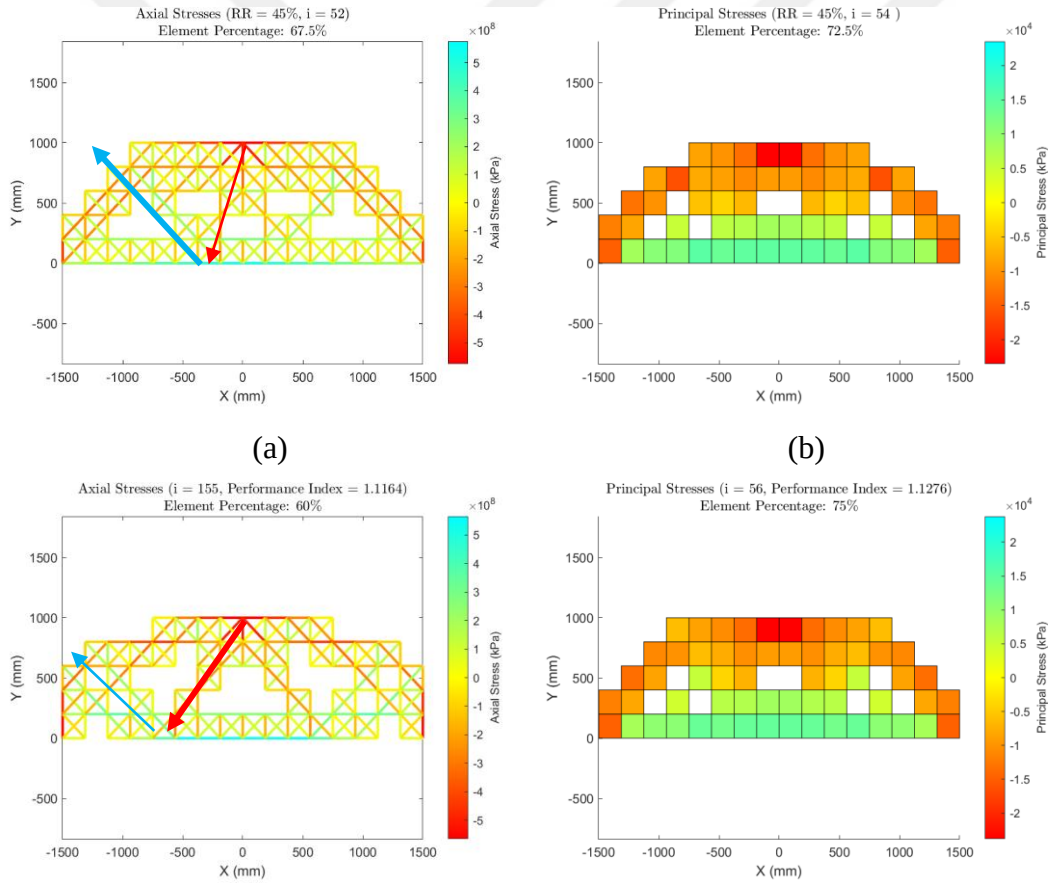


(a) (b)

Figure 4.41: Rectangle Example 4b Coarse Mesh: (a) Brick Model; (b) Membrane Model

The mesh configuration is not refined enough to produce a detailed element distribution that clearly defines the load path within the structure. However, the Brick Model (Figure 4.41 (a) and (c)) performs better than the Membrane Model (Figure 4.41 (b) and (d)) in highlighting force flow direction and key compression and tensile element clusters.

The stress-based Brick Model (Figure 4.41 (a)) distinctly shows the inclined tensile element group, which can serve as a direct guide for designing a Strut-and-Tie Model and positioning tie elements. However, the compressive elements in the middle are less defined due to the limited number of Brick elements. In contrast, the displacement-based Brick Model (Figure 4.41 (c)) exhibits the opposite behavior. The compressive element groups in the middle are more visible, making them useful for designing strut elements in a Strut-and-Tie Model. However, the inclined tensile elements appear less distinct. The tensile force path in the region is shown by fragmented groups of tensile elements. These element distributions are highlighted in Figure 4.41.



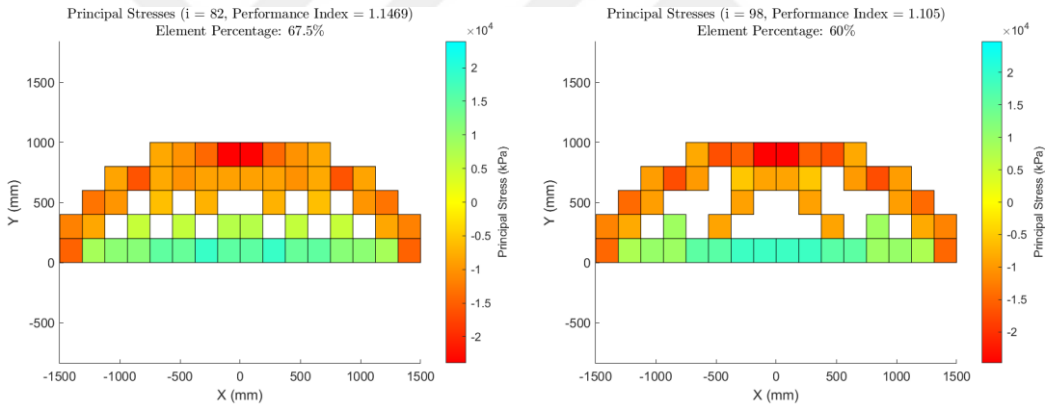
(a) (b)

(c) (d)

Figure 4.42: Rectangle Example 4b Coarse Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

The Membrane Model fails to indicate the internal force flow within the structure at the current mesh refinement level. The remaining element distribution from both optimization approaches (Figure 4.41 (b) and (d)) does not provide a clear guide for developing a Strut-and-Tie Model. The tendency of Membrane elements to form a checkerboarding pattern may prevent the model from producing a refined element distribution in coarser meshes.

However, there is a clearer remaining element configuration observed within the displacement-based optimization. The displacement-based Membrane Model with the highest performance index, along with an alternative model with a lower performance index, is shown in Figure 4.42. The alternative model (Figure 4.42 (b)) more effectively highlights the critical element distribution within the structure and can serve as a reference for Strut-and-Tie Model development.



(a) (b)

Figure 4.43: Rectangle Example 4b Displacement-based Coarse Mesh Membrane Model Results: (a) Highest Performance Index Model; (b) Alternative Model

The initial axial stress diagram of medium mesh Rectangle Example 4b is shown in Figure 4.44 The stress-based and displacement-based Brick Model results are also shown in Figure 4.45.

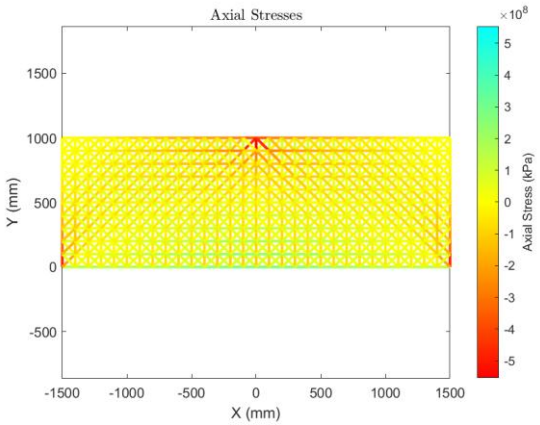
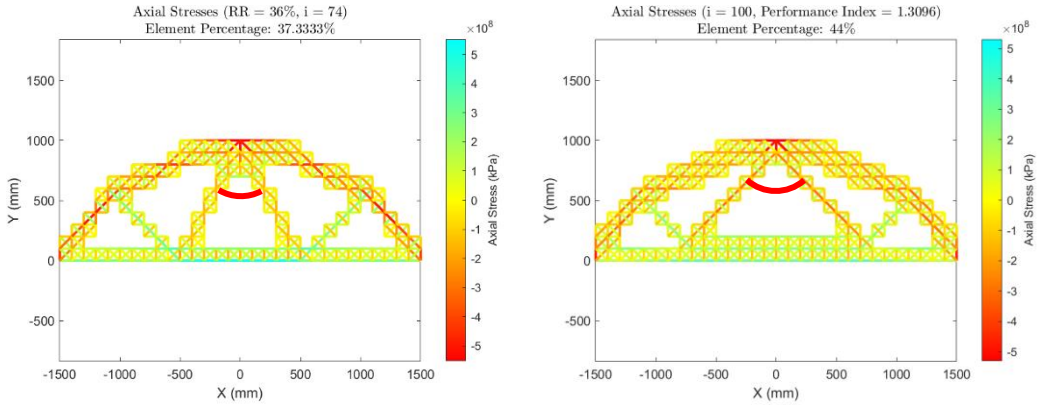


Figure 4.44: Rectangle Example 4b Medium Mesh: Brick Model

The Brick Model results exhibit similar element distributions, with the primary difference being the top angle between the compressive element groups in the middle. The stress-based Brick Model (Figure 4.45 (a)) shows a narrower angle, resembling the element distribution obtained by Kwak and Noh (2006) (Figure 4.40 (b)), while the displacement-based model (Figure 4.45 (b)) presents a wider angle, similar to the distribution obtained by Liang et al. (2000) (Figure 4.40 (a)). These angles are highlighted in Figure 4.45.

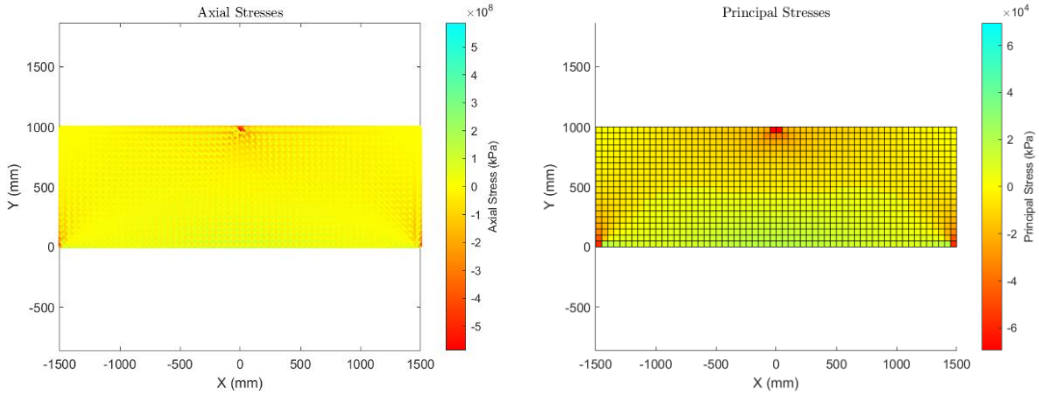
The structural behavior varies slightly depending on the angle configuration. In the stress-based Brick Model (Figure 4.45 (a)), the steeper compressive element groups result in lower tensile forces in the region at the bottom of the triangle. In contrast, the displacement-based Brick Model (Figure 4.45 (b)) features more slanted compressive elements, increasing the tensile force in that region. As a result, the tensile element cluster at the bottom is thicker. These element configurations shows the consistency of the Brick Models among themselves.



(a) (b)

Figure 4.45: Rectangle Example 4b Medium Mesh Brick Model Results: (a) Stress-based; (b) Displacement-based

The initial axial and principal stress diagrams of fine mesh Rectangle Example 4b is shown in Figure 4.46. The stress-based and displacement-based Brick and Membrane Model results are also shown in Figure 4.47.



(a) (b)

Figure 4.46: Rectangle Example 4b Fine Mesh: (a) Brick Model; (b) Membrane Model

The results show variations compared to the element configurations obtained by Liang et al. (2000) and Kwak and Noh (2006), and differ from their coarser meshed counterparts. The stress-based Membrane Model (Figure 4.47 (b)) exhibits a similar element distribution to previous results, closely resembling a truss model where the locations of compressive struts and tensile ties can be easily identified. In contrast,

the stress-based Brick Model (Figure 4.47 (a)) is more irregular. While the compressive regions in the middle and the inclined tensile element distribution are consistent with previous results, additional compressive and tensile elements remain in the optimal configuration. These element distributions are highlighted in Figure 4.47. As mesh refinement increases, the load paths become more detailed throughout the optimization process. The Brick Model can still aid in determining a Strut-and-Tie Model with the guidance of other Brick Model (Figure 4.41 (a) and Figure 4.45 (a)) results. These findings indicate that repeating the optimization process with different mesh refinement levels improves the accuracy of the results.

As slenderness increases, and with higher mesh refinement and a greater number of elements, the results begin to resemble continuum-like structures rather than a truss system. This effect is more pronounced in displacement-based results. The displacement-based Brick Model (Figure 4.47 (c)) features compressive elements in the middle of the structure, similar to previous results. However, the displacement-based Membrane Model (Figure 4.47 (d)) exhibits multiple compressive element clusters, similar to the stress-based Brick Model (Figure 4.47 (a)).

More evidently, both displacement-based models contain reverse arch tensile element clusters. In all models, the top of the structure forms an arch to support the applied loads and transfer them to the base. If an external force were applied at the bottom in the opposite direction, the force would then be carried by the reverse arch tensile element groups. Such a configuration cannot function as tie elements in a Strut-and-Tie Model. Despite this, the continuum-like element distribution remains useful in visualizing the internal force flow within the structure.

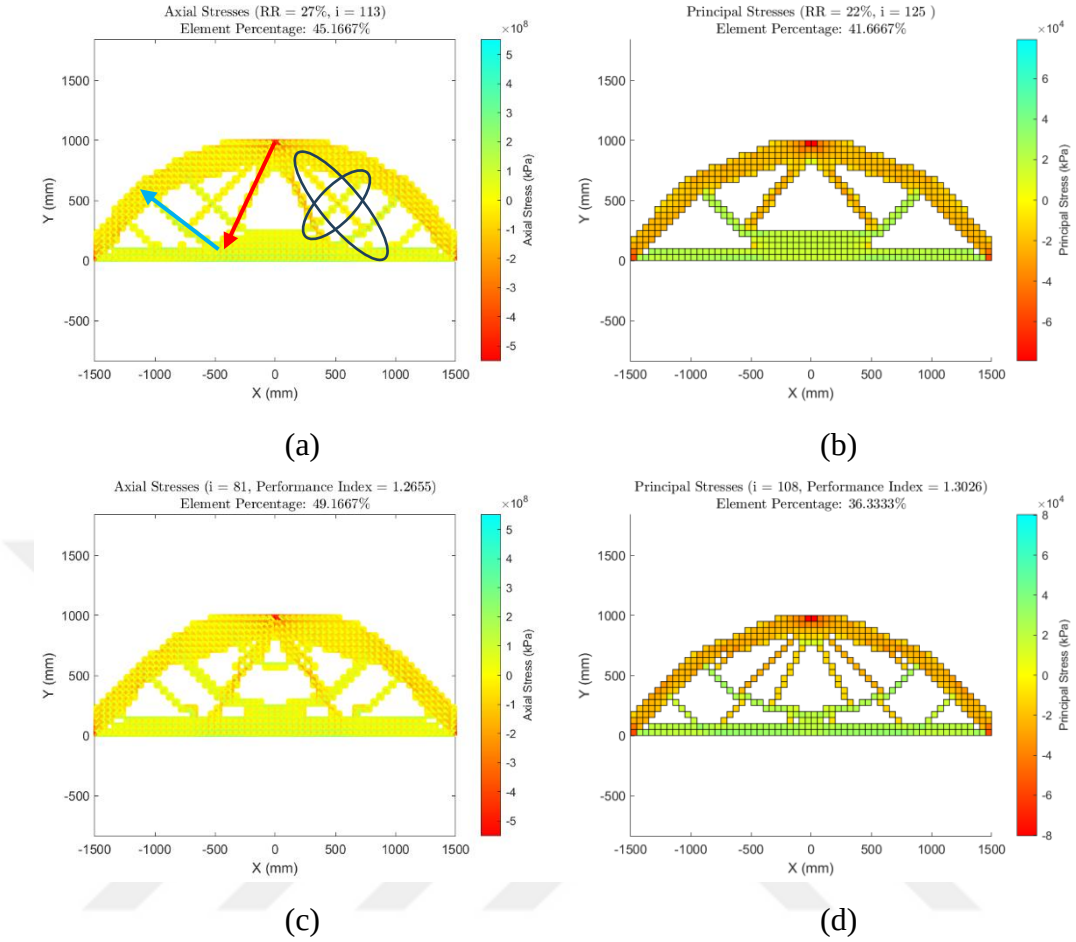
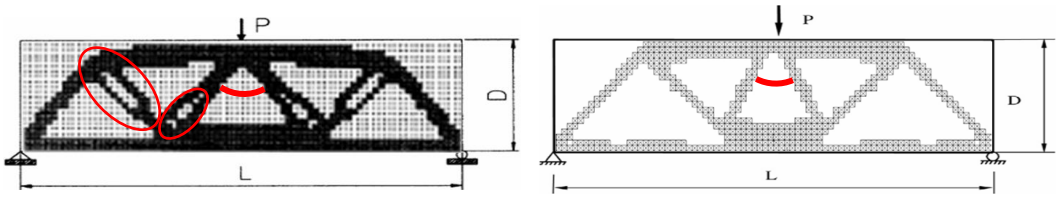


Figure 4.47: Rectangle Example 4b Fine Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

Rectangle Example 4c

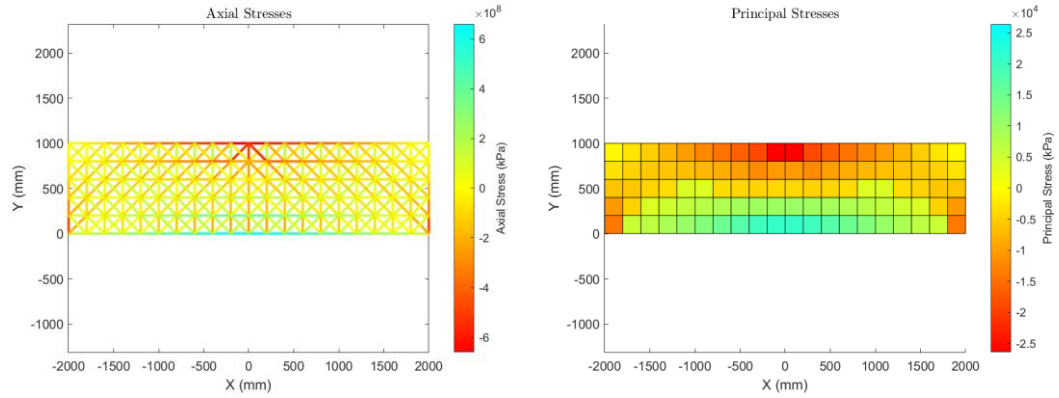
Liang et al. (2000) and Kwak and Noh (2006) obtained similar optimum element distributions for Rectangle Example 4c. As in Rectangle Example 4b, Kwak and Noh's (2006) configuration features compressive elements in the middle with a narrower angle than Liang et al. (2000). Additionally, the Brick Model produces a more concentrated element distribution. These results are shown in Figure 4.48.



(a) (b)

Figure 4.48: The optimum element distribution of Rectangle Example 4c by: (a) Liang et al. (2000); (b) Kwak and Noh (2006)

The initial axial and principal stress diagrams for the coarse mesh Rectangle Example 4c structure are shown in Figure 4.49. The coarse mesh results from stress-based and displacement-based optimization using Brick and Membrane elements are also presented in Figure 4.50. To prevent excessive element removal in the early stages, the RR and ER values in displacement-based optimization routines are set to 0.1%.



(a) (b)

Figure 4.49: Rectangle Example 4c Coarse Mesh: (a) Brick Model; (b) Membrane Model

Both stress-based models produce a refined element distribution. The Brick Model (Figure 4.50 (a)) remains consistent with its medium mesh counterpart (Figure 4.53 (a)), as both feature compressive and tensile elements in nearly the same locations. The stress-based Membrane Model (Figure 4.50 (b)), however, shows slight variations in tensile element placement compared to its medium (Figure 4.53 (b)) and fine mesh (Figure 4.56 (b)) counterparts. In the stress-based models, the

compressive elements in the middle form a triangle with the tensile region at the bottom. In Membrane Model, the tensile section of this triangle is shorter than in the other models, as highlighted in Figure 4.50. Despite these differences, the stress-based Membrane Model still provides valuable guidance for developing a Strut-and-Tie Model for the reinforced structure.

The optimum element configurations from the displacement-based models are more irregular and less distinct compared to the stress-based results. While the Membrane Model (Figure 4.50 (d)) provides insight into the internal force flow, a finer mesh is needed to clearly distinguish critical element clusters.

The displacement-based Brick Model (Figure 4.50 (c)) also shows a scattered element distribution. However, since each brick element consists of six truss elements, critical elements can be more easily traced along the truss structure. This allows for a better understanding of the load path within the reinforced concrete structure.

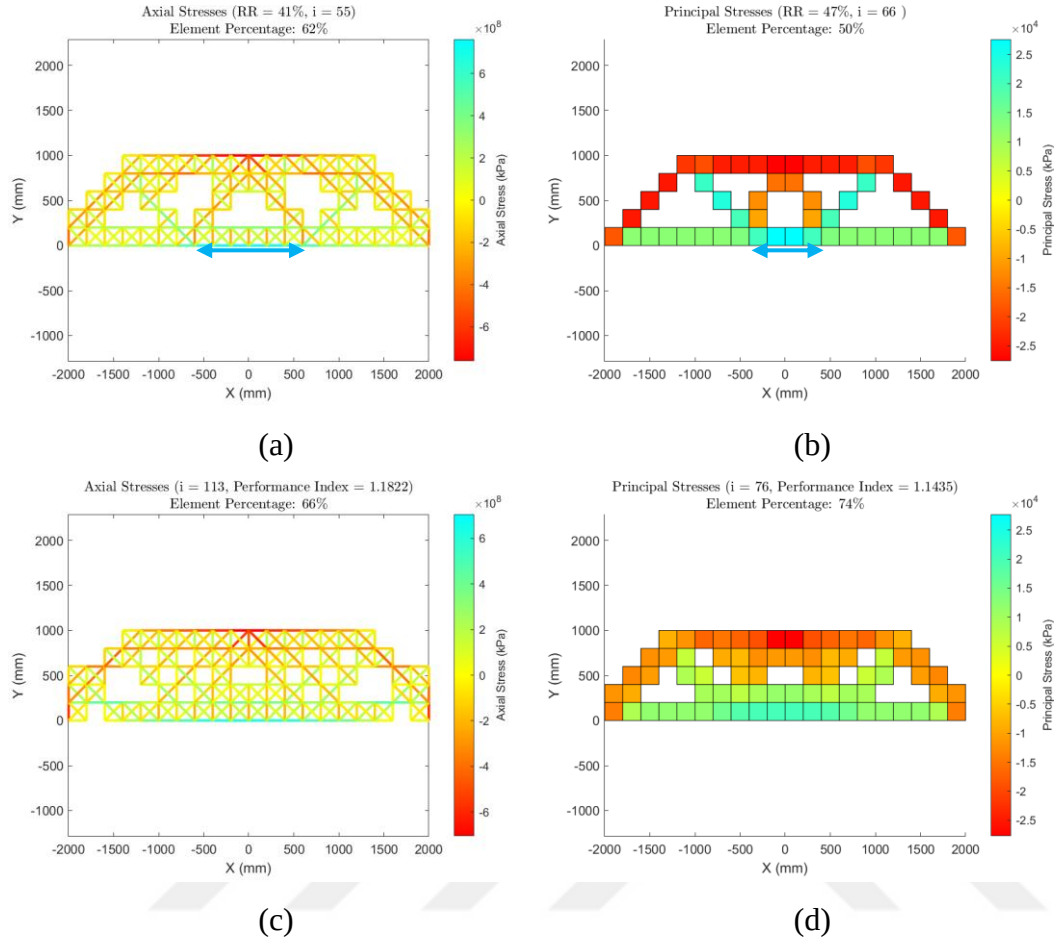
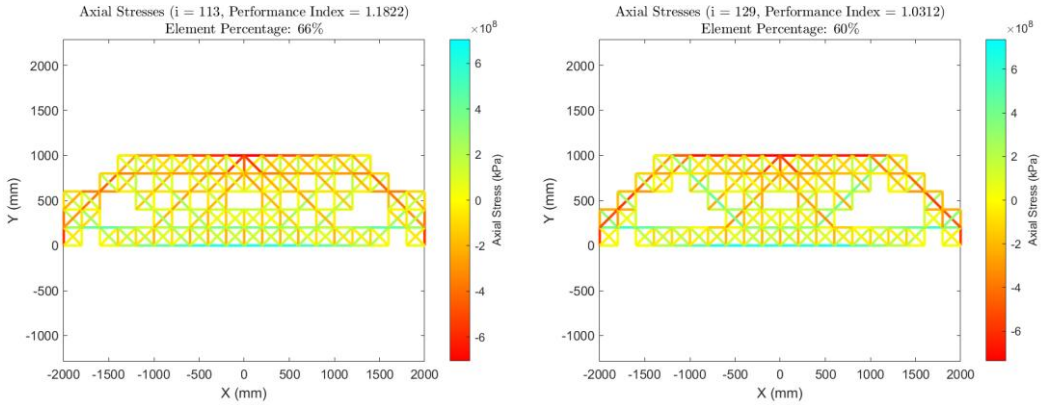


Figure 4.50: Rectangle Example 4c Coarse Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

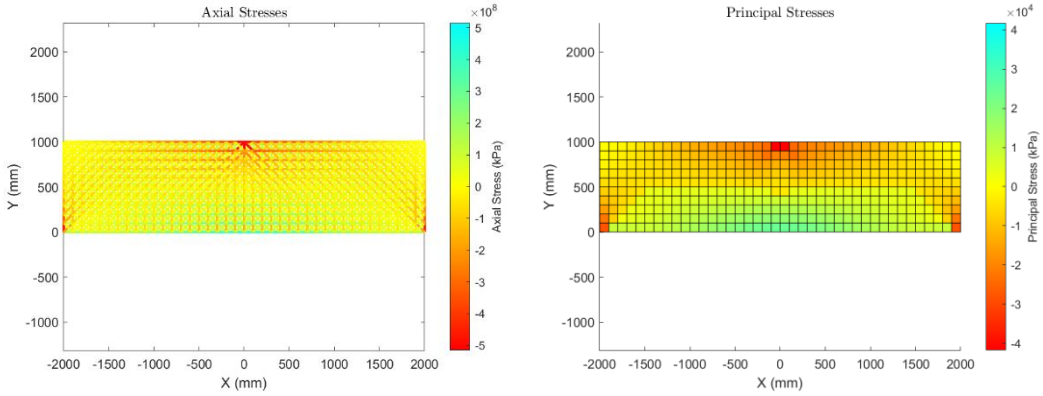
For the Brick Model, an alternative result with a more refined and clearer element distribution was produced during the same displacement-based optimization process. The displacement-based coarse mesh Brick Model result with the highest performance index, along with an alternative configuration, is shown in Figure 4.51. The alternative model (Figure 4.51 (b)) contains fewer elements and provides a clearer representation of the internal force flow within the reinforced structure. This makes it a more effective guide for developing a Strut-and-Tie Model.



(a) (b)

Figure 4.51: Rectangle Example 4c Displacement-based Coarse Mesh Brick Model Results: (a) Highest Performance Index Model; (b) Alternative Model

The initial axial and principal stress diagrams for the medium mesh Rectangle Example 4c structure are shown in Figure 4.52. The stress-based and displacement-based Brick and Membrane Model results are also shown in Figure 4.53.



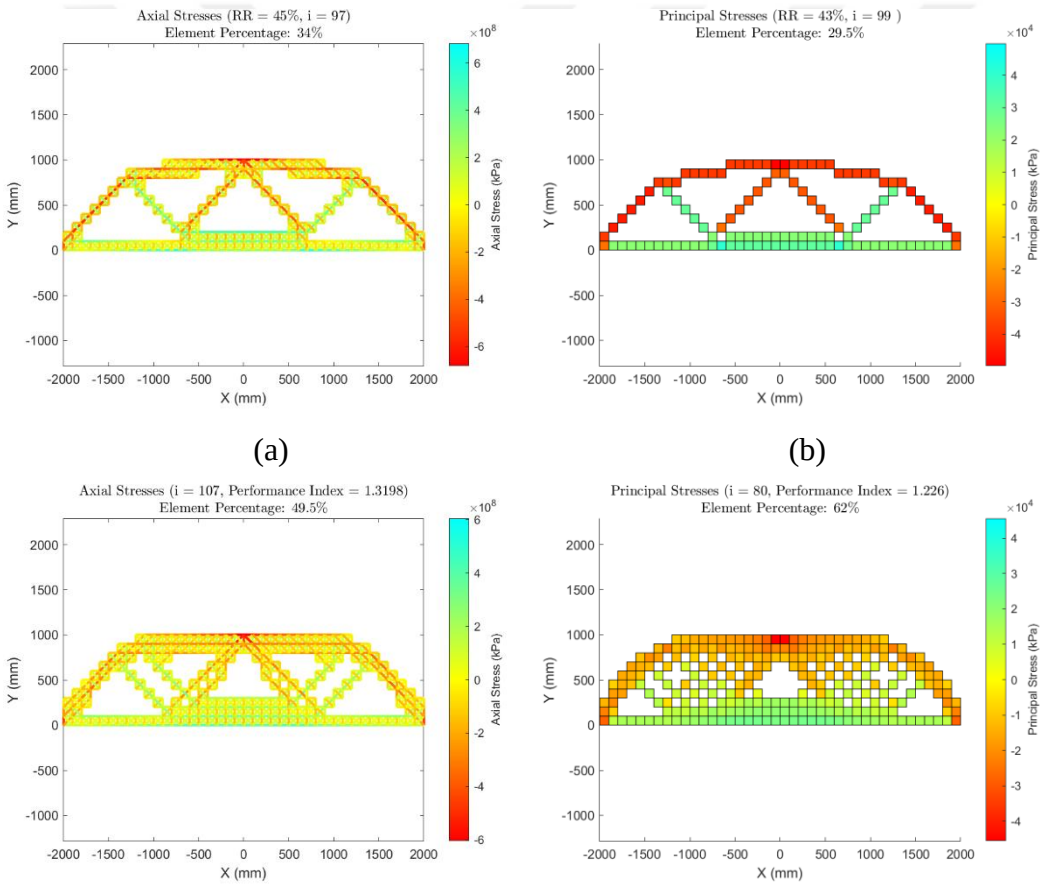
(a) (b)

Figure 4.52: Rectangle Example 4c Medium Mesh: (a) Brick Model; (b) Membrane Model

The stress-based results (Figure 4.53 (a) and (b)) exhibit highly refined element distributions, with identical critical compressive and tensile element configurations in both models. These results also show consistency with the optimum element distributions obtained by Liang et al. (2000) and Kwak and Noh (2006) (Figure 4.48), with further refinement.

The displacement-based Brick Model (Figure 4.53 (c)), however, does not provide as clear a critical element distribution as the stress-based methods. Its separated compressive and tensile element groups resemble the element distribution obtained by Liang et al. (2000). Despite this, it can still serve as a reference for designing a Strut-and-Tie Model.

In contrast, the displacement-based Membrane Model (Figure 4.53 (d)) suffers from checkerboarding behavior. Without 1D elements, such as the truss elements within a Brick element, the critical compressive and tensile forces cannot be effectively traced. As a result, this model is not suitable for developing a Strut-and-Tie Model.



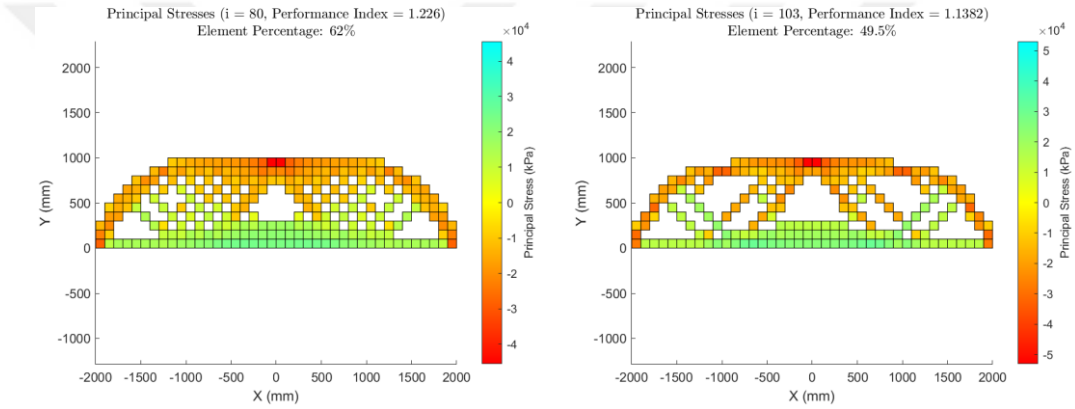
(a) (b)

(c) (d)

Figure 4.53: Rectangle Example 4c Medium Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

An alternative element distribution emerges during the displacement-based optimization of the medium mesh Membrane Model. The displacement-based medium mesh Membrane Model result with the highest performance index, along with an alternative configuration, is presented in Figure 4.54.

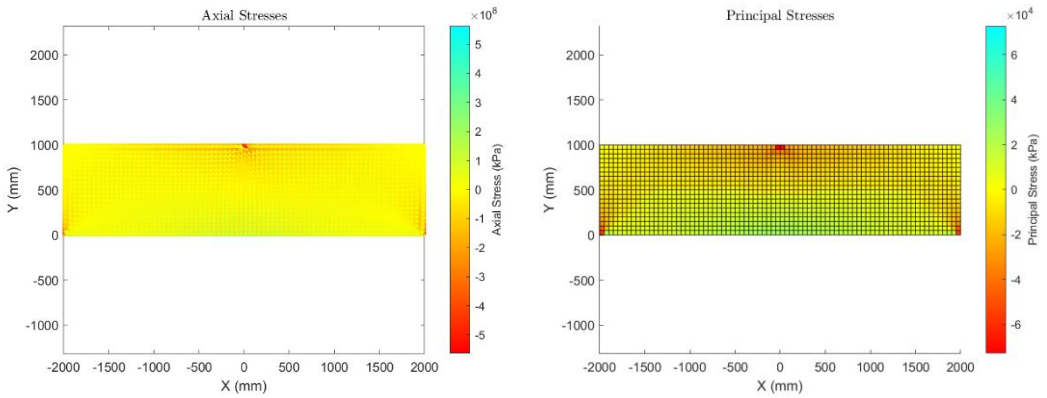
The alternative model (Figure 4.54 (b)) contains fewer remaining elements and produces a more refined element distribution. While the critical compressive and tensile regions are better defined, they are still not as clear as those obtained through stress-based optimization. Additionally, remnants of the checkerboarding pattern from previous iterations continue to affect the alternative structure.



(a) (b)

Figure 4.54: Rectangle Example 4c Displacement-based Coarse Mesh Membrane Model Results: (a) Highest Performance Index Model; (b) Alternative Model

The initial axial and principal stress diagrams for the fine mesh Rectangle Example 4c structure are shown in Figure 4.55. The stress-based and displacement-based Brick and Membrane Model results are also shown in Figure 4.56.



(a) (b)

Figure 4.55: Rectangle Example 4c Fine Mesh: (a) Brick Model; (b) Membrane Model

Similar to Rectangle Example 4b (Figure 4.47 (a) and (c)), the Brick Model (Figure 4.56 (a)) exhibits a continuum-like element distribution as the structure becomes slenderer and the number of finite elements increases with finer meshing. Reverse arch tensile element groups emerge, disrupting the diagonal compressive elements and leading to an irregular compressive element distribution that no longer resembles a truss-like formation. As a result, compressive forces accumulate in a perpendicular element cluster in the middle of the structure. While the element distribution provides insights into internal force flow, it is advisable to refer to coarse or medium mesh stress-based results when designing a Strut-and-Tie Model.

The stress-based Membrane Model (Figure 4.56 (b)) maintains consistency across all three mesh refinements, with compressive and tensile element distributions nearly identical to those in the coarse and medium meshes. While checkerboarding patterns appeared in earlier stages of the optimization, remnants of this phenomenon can still be observed. The voids within the central compression elements are a result of redundant elements being removed with the checkerboarding behavior during the optimization, as highlighted in Figure 4.56 (b). However, this behavior did not disrupt the optimization process. The remaining element distribution is well-refined and provides a clear guide for designing a Strut-and-Tie Model.

The displacement-based optimization results (Figure 4.56 (c) and (d)) also show consistency across different mesh refinements. As slenderness increases and refinement improves, both models tend to resemble continuum-like structures rather than truss systems. The Brick Model's remaining element configuration (Figure 4.56 (c)) follows a continuum-like pattern, with tensile element groups converging toward an arch-like geometry. However, the structure can still be interpreted as a truss model. The compressive elements in the middle form truss-like shapes, and by disregarding small clusters of deviated compressive and tensile elements, a truss structure similar to previous results emerges. One notable difference is the narrower angle of the compressive elements, which resembles the solution obtained by Kwak and Noh (2006). Additionally, the split tensile groups align with the solution reported by Liang et al. (2000).

The Membrane Model (Figure 4.56 (d)) exhibits a different element distribution. Instead of a single set of compressive element groups in the middle, the compression force is transferred through two sets of compressive element clusters. The outer compression zone supports the tensile element distribution, giving it a more arch-like form. The resulting model serves as a useful indicator of the internal load path within the structure rather than providing direct guidance for a Strut-and-Tie Model.

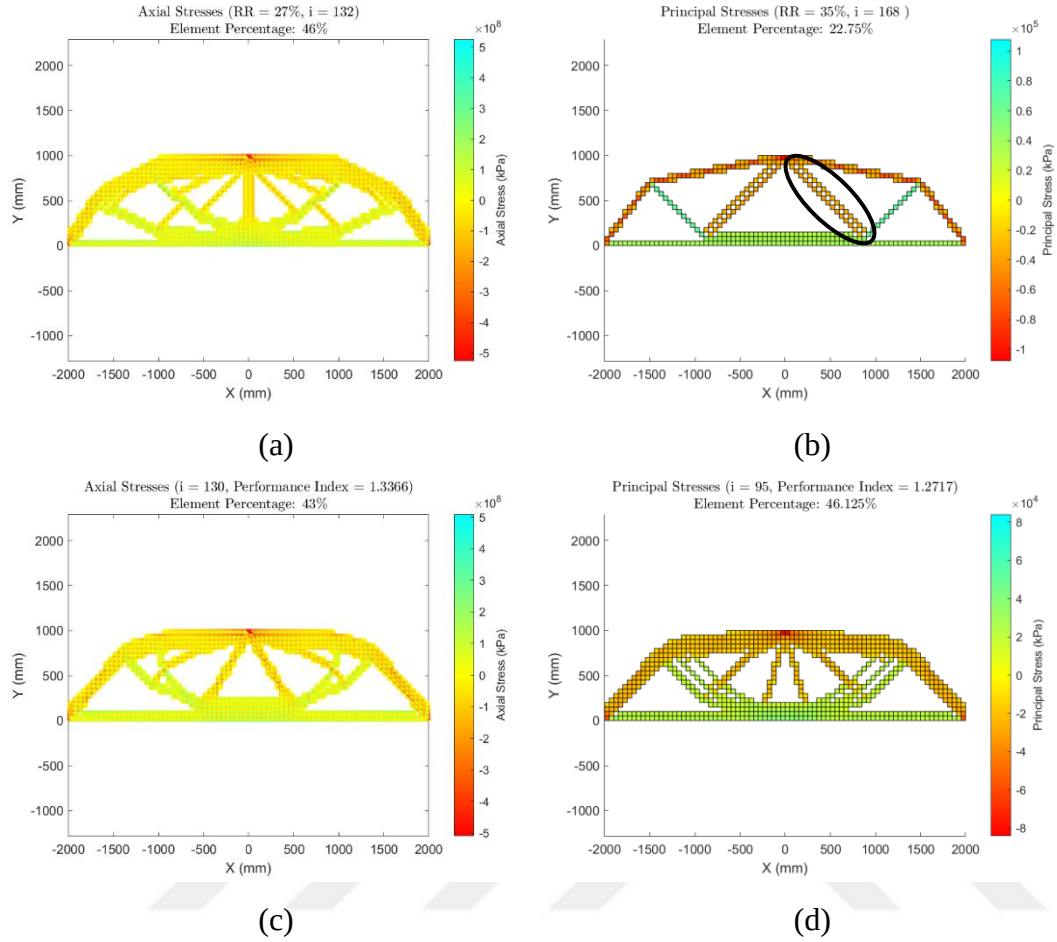
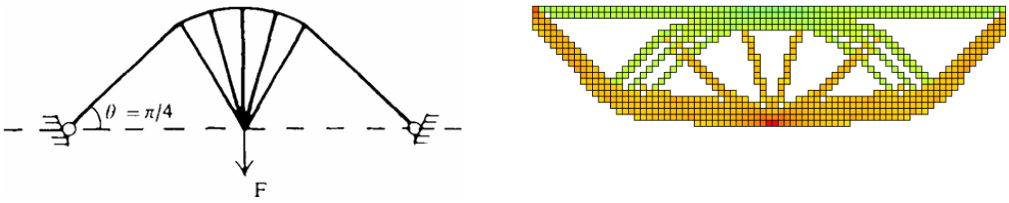


Figure 4.56: Rectangle Example 4c Fine Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

The resulting configuration of the displacement-based Membrane Model (Figure 4.56 (d)) closely resembles an upside-down Michell-type structure. This typical structural layout optimization problem was analytically solved by Hemp (1973). The Michell structure and the upside-down displacement-based fine mesh Membrane Model of Rectangle Example 4c are shown in Figure 4.57. The Michell structure illustration is adapted from Xie and Steven (1992).

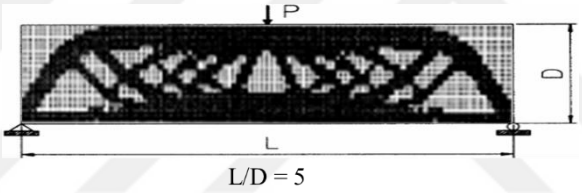


(a) (b)

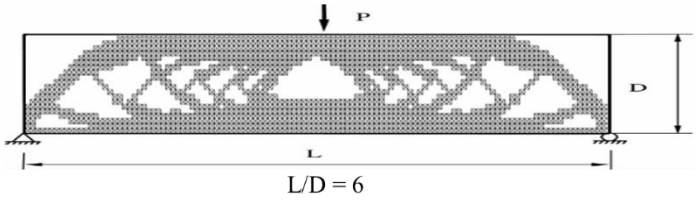
Figure 4.57: (a) Michell Type Structure; (b) Upside-down Membrane Model

Rectangle Example 4d

Liang et al. (2000) and Kwak and Noh (2006) obtained similar optimum element distributions for Rectangle Example 4d. The key difference is that Kwak and Noh (2006) used a span-to-depth ratio of 6, while Liang et al. (2000) set it at 5. In this study, the span-to-depth ratio is also set to 5. The results from previous studies are presented in Figure 4.58.



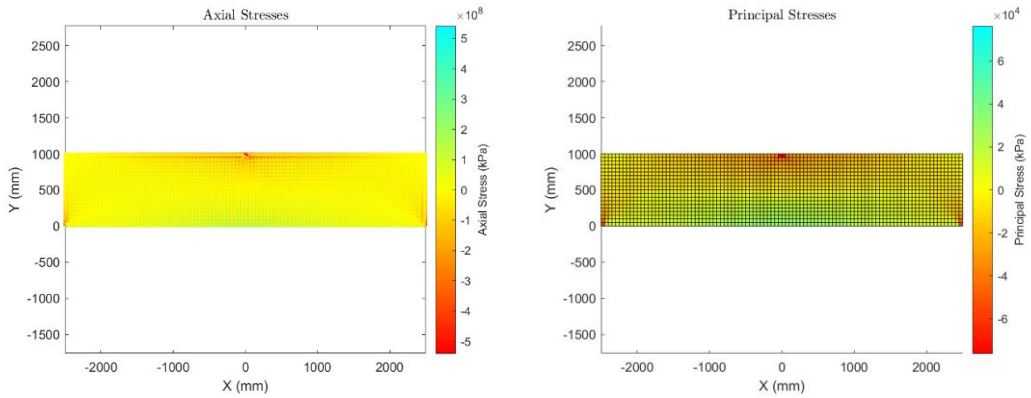
(a)



(b)

Figure 4.58: The optimum element distribution of Rectangle Example 4c by: (a) Liang et al. (2000); (b) Kwak and Noh (2006)

The initial axial and principal stress diagrams for the medium mesh Rectangle Example 4d structure are shown in Figure 4.59. The stress-based and displacement-based Brick and Membrane Model results are also shown in Figure 4.60.



(a) (b)

Figure 4.59: Rectangle Example 4d Medium Mesh: (a) Brick Model; (b) Membrane Model

The stress-based optimization procedures result in truss-like structures for both the Brick and Membrane Models (Figure 4.60 (a) and (b)). Both feature two sets of compressive and tensile elements on each side of the structure. The remaining element distribution is well-refined, with clearly distinguishable compressive and tensile zones, making it a useful guide for developing a Strut-and-Tie Model. The inclination of the elements shows slight variations, which are highlighted in the axial and principal stress diagrams of the structure.

The displacement-based optimization results for the Brick Model (Figure 4.60 (c)) resemble a continuum-like structure rather than a truss-like configuration. The element distribution closely aligns with the optimum solutions obtained by Liang et al. (2000) and Kwak and Noh (2006) (Figure 4.58). This model effectively illustrates the internal force flow within a slender reinforced concrete structure. In contrast, the displacement-based Membrane Model (Figure 4.60 (d)) lacks the refinement needed to clearly represent neither the internal force flow nor a truss-like element distribution due to checkerboarding patterns.

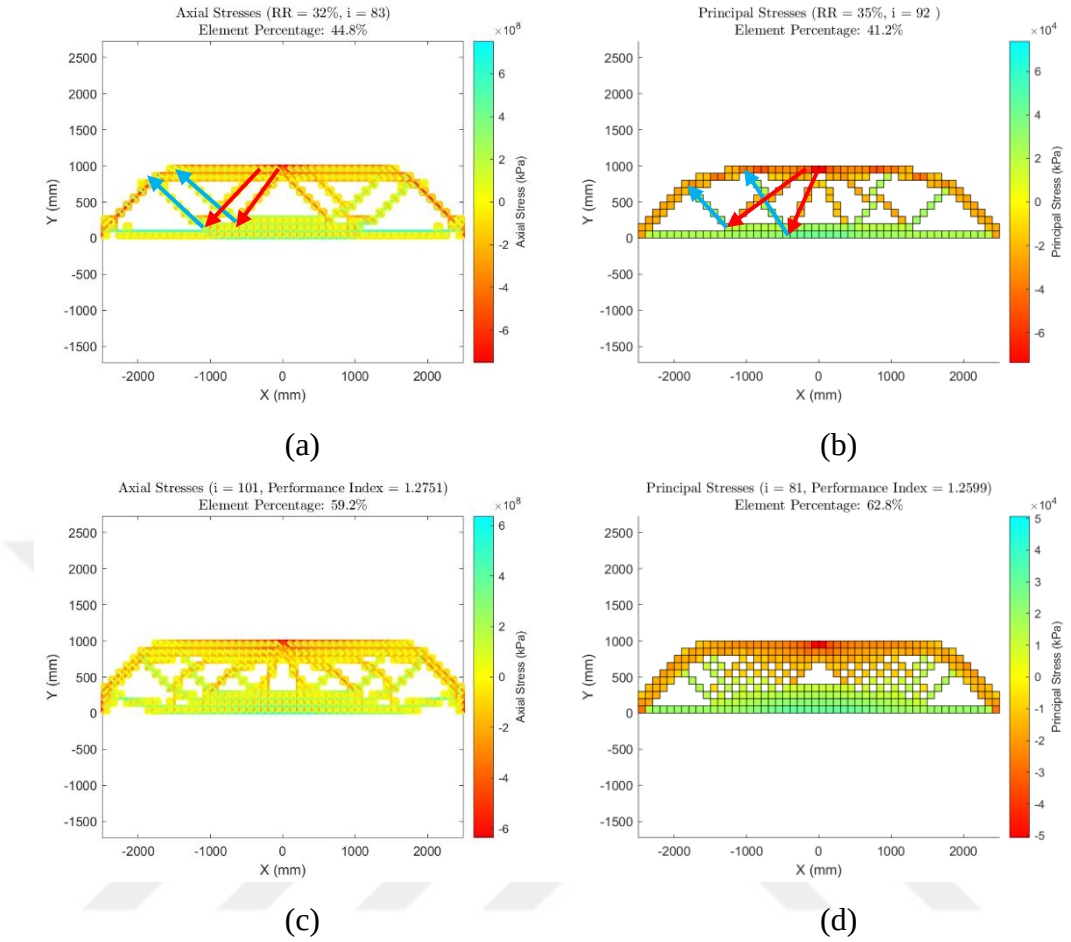
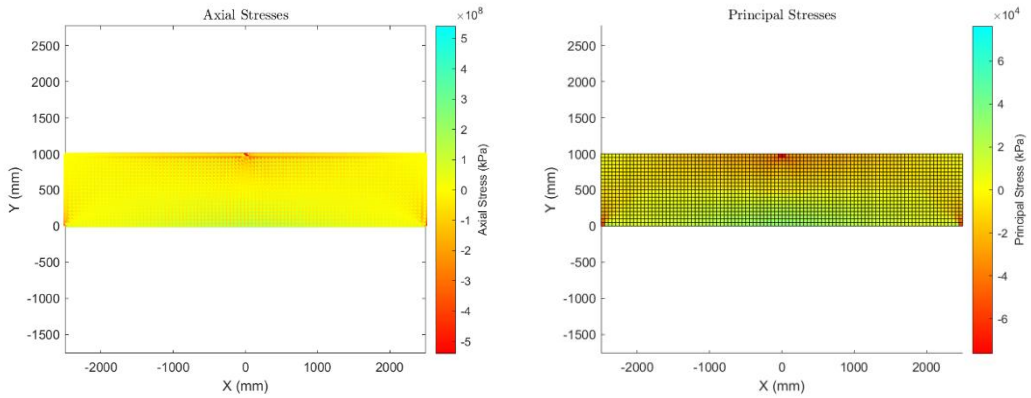


Figure 4.60: Rectangle Example 4d Medium Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

The initial axial and principal stress diagrams for the fine mesh Rectangle Example 4d structure are shown in Figure 4.61. The stress-based and displacement-based Brick and Membrane Model results are also shown in Figure 4.62.



(a) (b)

Figure 4.61: Rectangle Example 4d Fine Mesh: (a) Brick Model; (b) Membrane Model

The stress-based Brick Model (Figure 4.62 (a)) exhibits a continuum-like structure, closely resembling the optimum element distributions obtained by Liang et al. (2000) and Kwak and Noh (2006) (Figure 4.58). However, the remaining element distribution is not suitable for directly constructing a Strut-and-Tie Model, as it does not represent a truss system. Instead, it can be used to trace the internal force flow within the reinforced concrete structure. As the structure becomes slenderer, the influence of flexural beam theory increases, making it a more dominant approach over Strut-and-Tie Modelling. The stress-based Membrane Model (Figure 4.62 (b)) produces an element distribution similar to the previous results of Rectangle Example 4c (Figures 4.50 (b), 4.53 (b), 4.56 (b)). However, due to the greater length of the reinforced concrete structure in this case, the compressive and tensile element clusters contain a larger number of elements. Additionally, redundant elements remain between the inclined compressive and tensile groups, as marked in Figure 4.62. These elements do not contribute to the structure's stability and are remnants of the checkerboarding behavior observed in earlier iterations.

The displacement-based Brick Model (Figure 4.62 (c)) also illustrates the internal force flow within the reinforced concrete structure. However, its element distribution is less detailed compared to its stress-based counterpart and the results obtained by Liang et al. (2000) and Kwak and Noh (2006) (Figure 4.58). This is due to excessive

element removal in the early stages of optimization, despite the RR and ER values being reduced to 0.1%. This further highlights the robustness of the stress-based approach compared to the displacement-based method.

As expected for this element type, the displacement-based Membrane Model (Figure 4.62 (d)) produces a truss-like system rather than a continuum-like structure. Its element distribution resembles that of the stress-based medium mesh Brick Model (Figure 4.60 (a)) and can serve as a guide for developing a Strut-and-Tie Model. However, given the slenderness of the structure, using both element models together would provide a more comprehensive examination of the internal force flow.

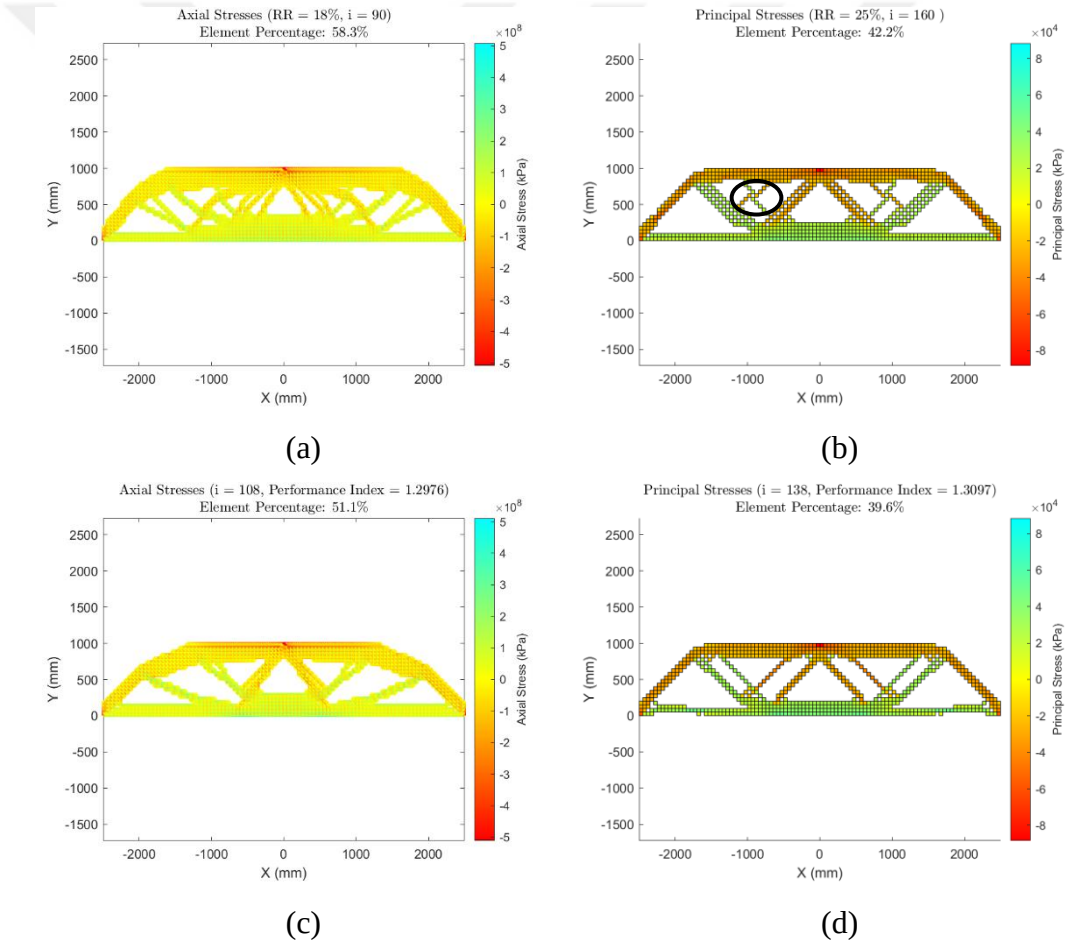
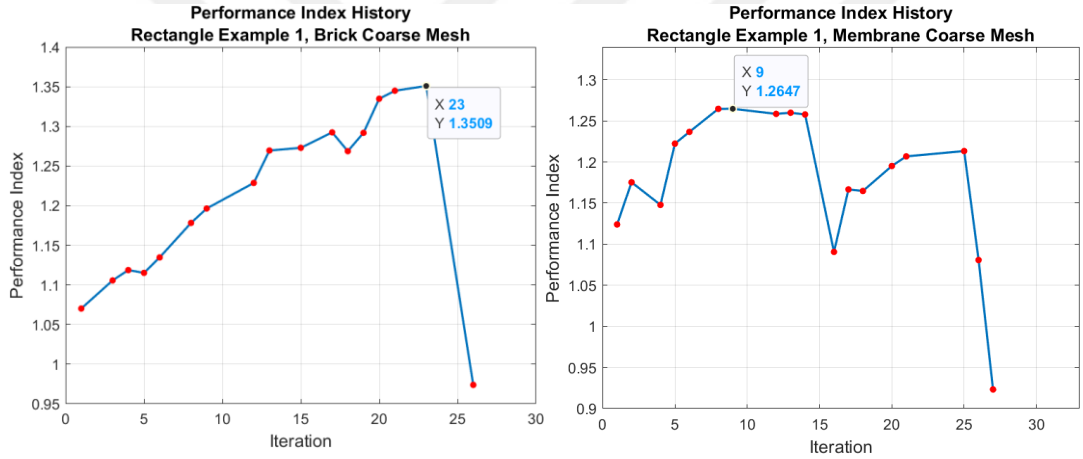


Figure 4.62: Rectangle Example 4d Fine Mesh Results: (a) Stress-based Brick Model; (b) Stress-based Membrane Model; (c) Displacement-based Brick Model; (d) Displacement-based Membrane Model

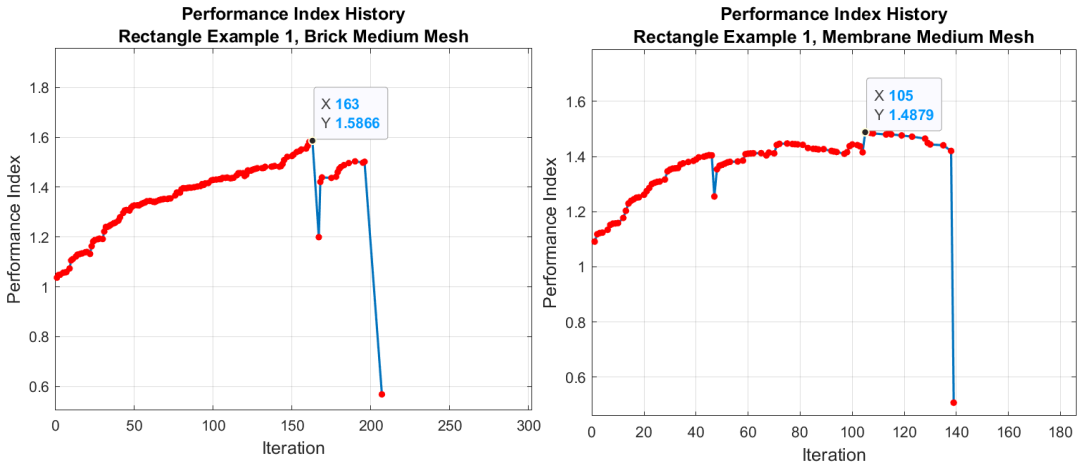
4.2.2.2 Performance Index History

The performance index history of the models throughout the iterations of the displacement-based evolutionary structural optimization procedure has been recorded. This data is plotted to examine the behavior of the performance index during the gradual element elimination process of a structural model. The iteration with the highest performance index value is marked. The graphs show at which point of the displacement-based optimization the highest performance index value occurs. The resulting trends provide insights into the overall progression of the performance index. The Performance Index History graphs for all the Sample Problems are presented in Figures 4.63 - 4.79.



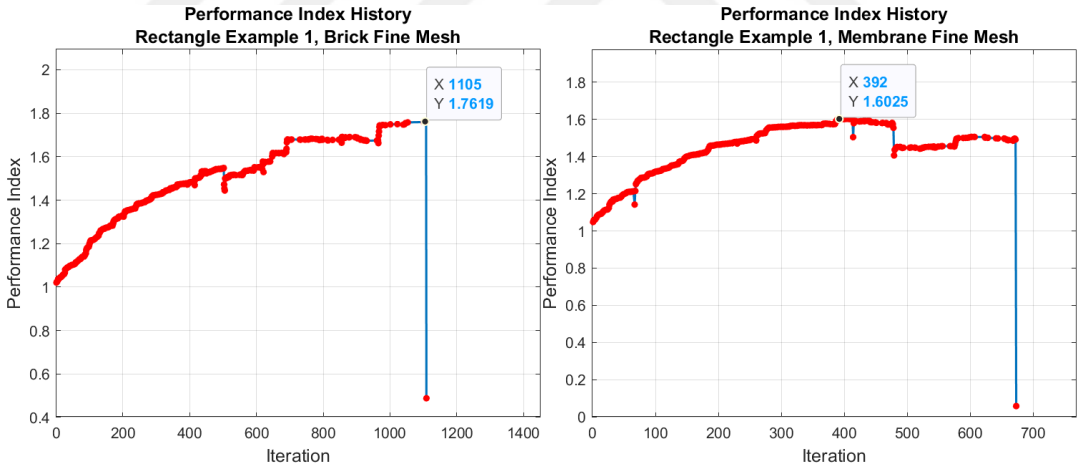
(a) (b)

Figure 4.63: Performance Index History of Rectangle Example 1 Coarse Mesh: (a) Brick Model; (b) Membrane Model



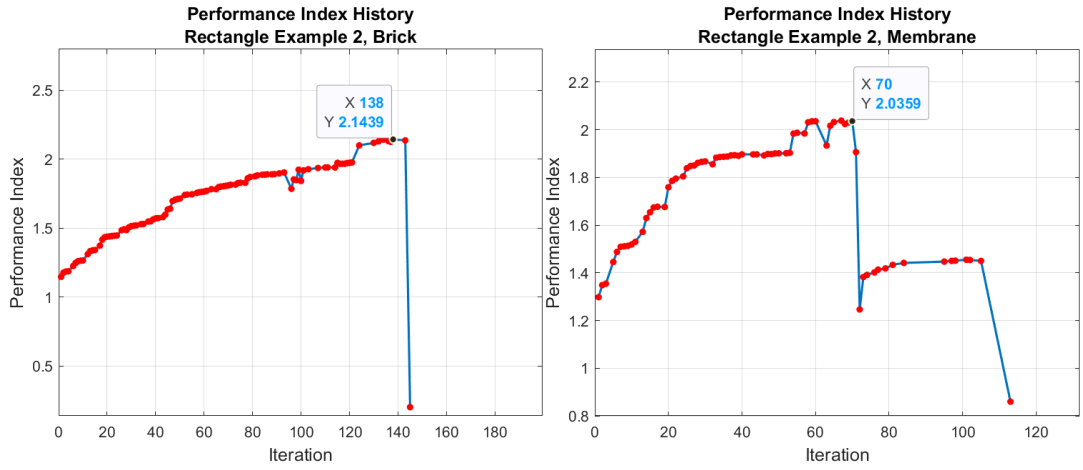
(a) (b)

Figure 4.64: Performance Index History of Rectangle Example 1 Medium Mesh: (a) Brick Model; (b) Membrane Model



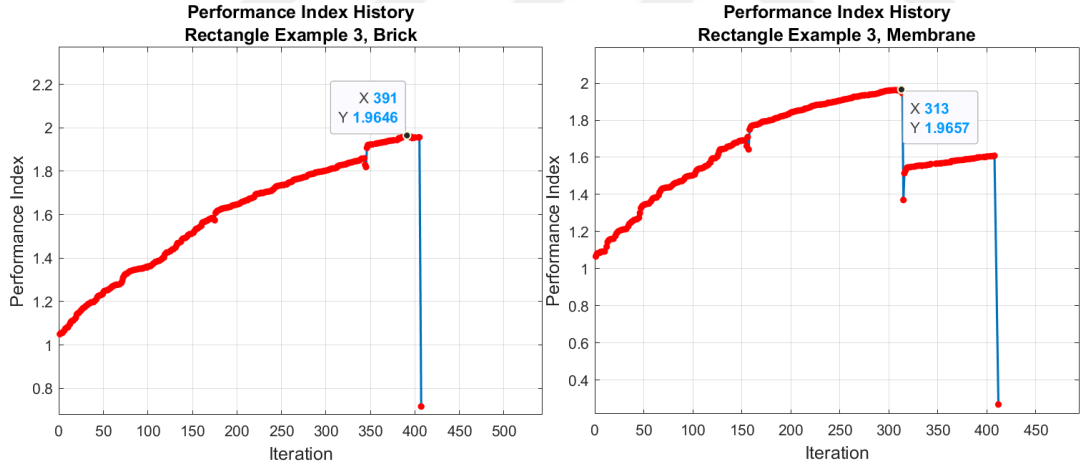
(a) (b)

Figure 4.65: Performance Index History of Rectangle Example 1 Fine Mesh: (a) Brick Model; (b) Membrane Model



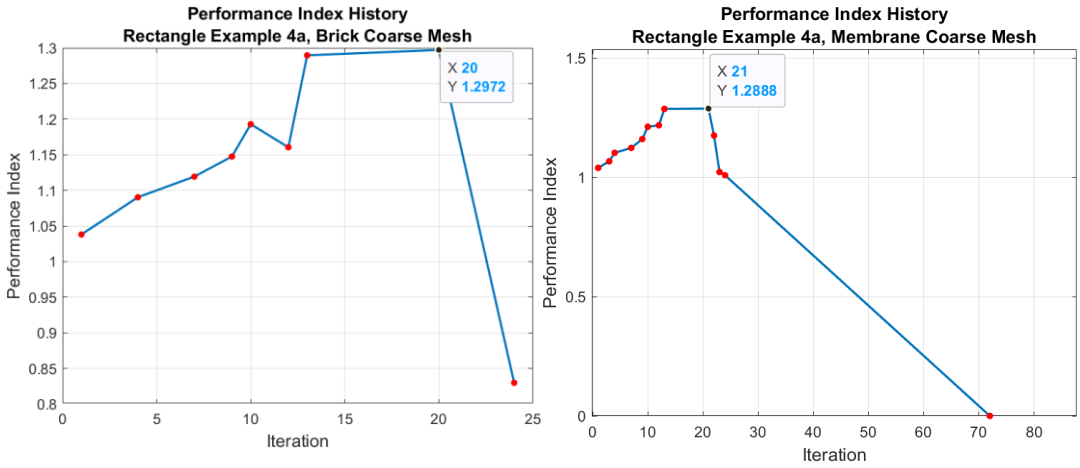
(a) (b)

Figure 4.66: Performance Index History of Rectangle Example 2: (a) Brick Model; (b) Membrane Model



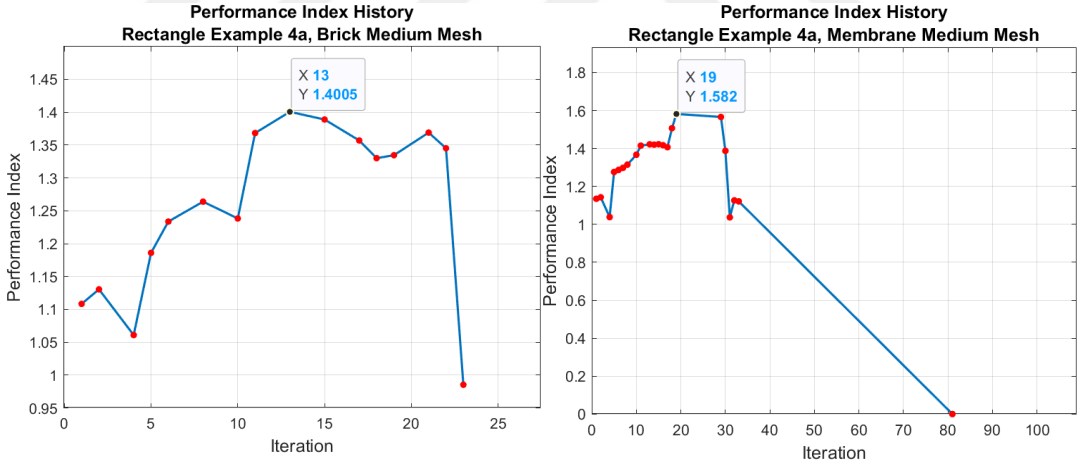
(a) (b)

Figure 4.67: Performance Index History of Rectangle Example 3: (a) Brick Model; (b) Membrane Model



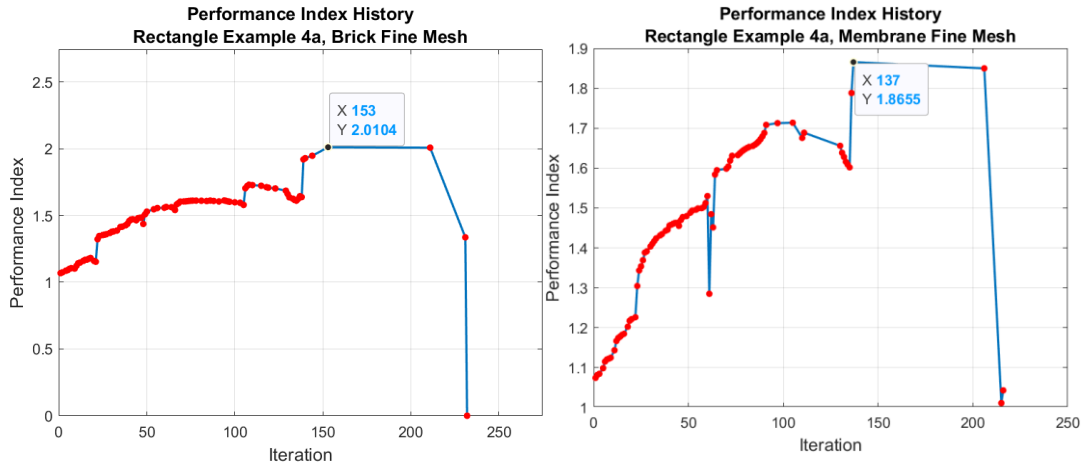
(a) (b)

Figure 4.68: Performance Index History of Rectangle Example 4a Coarse Mesh: (a) Brick Model; (b) Membrane Model



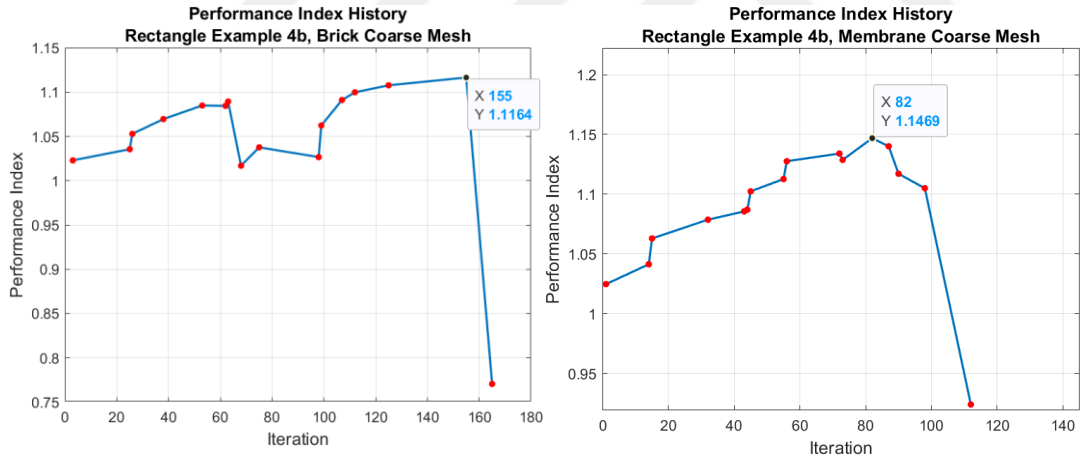
(a) (b)

Figure 4.69: Performance Index History of Rectangle Example 4a Medium Mesh: (a) Brick Model; (b) Membrane Model



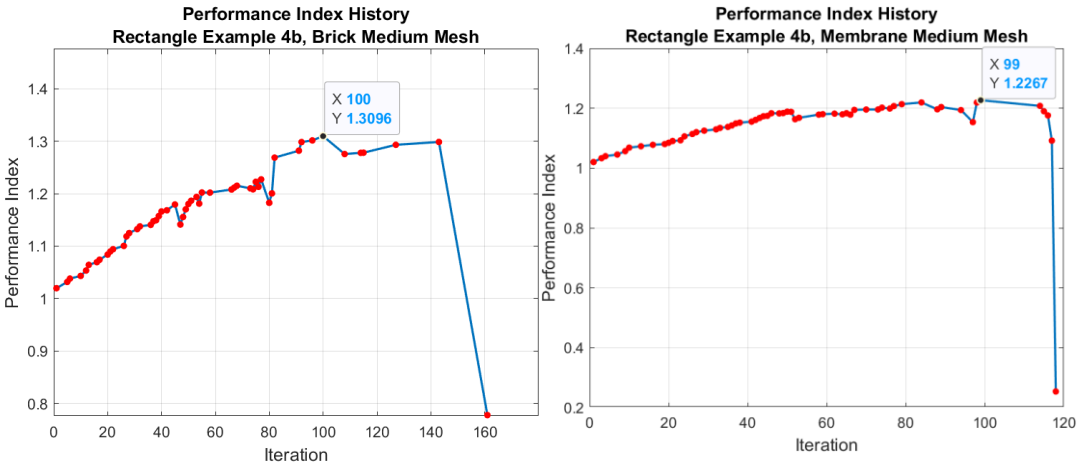
(a) (b)

Figure 4.70: Performance Index History of Rectangle Example 4a Fine Mesh: (a) Brick Model; (b) Membrane Model



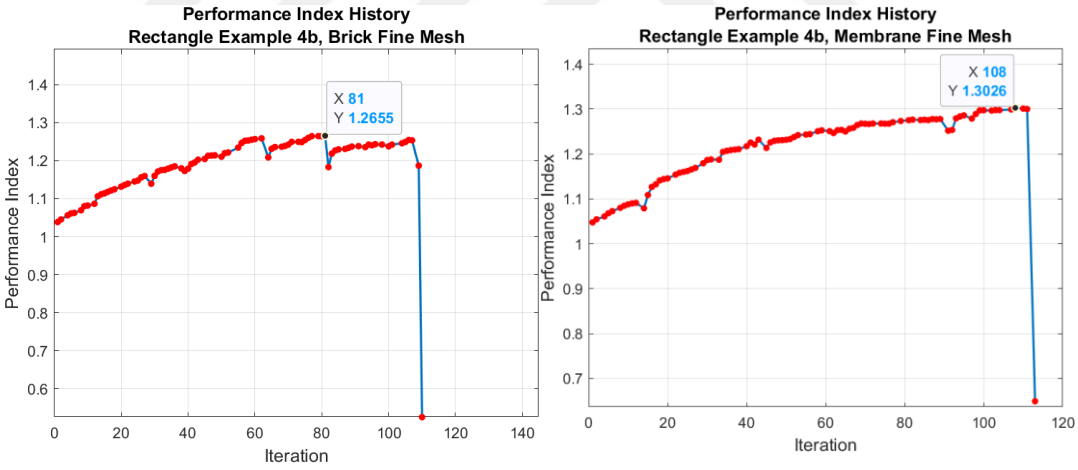
(a) (b)

Figure 4.71: Performance Index History of Rectangle Example 4b Coarse Mesh: (a) Brick Model; (b) Membrane Model



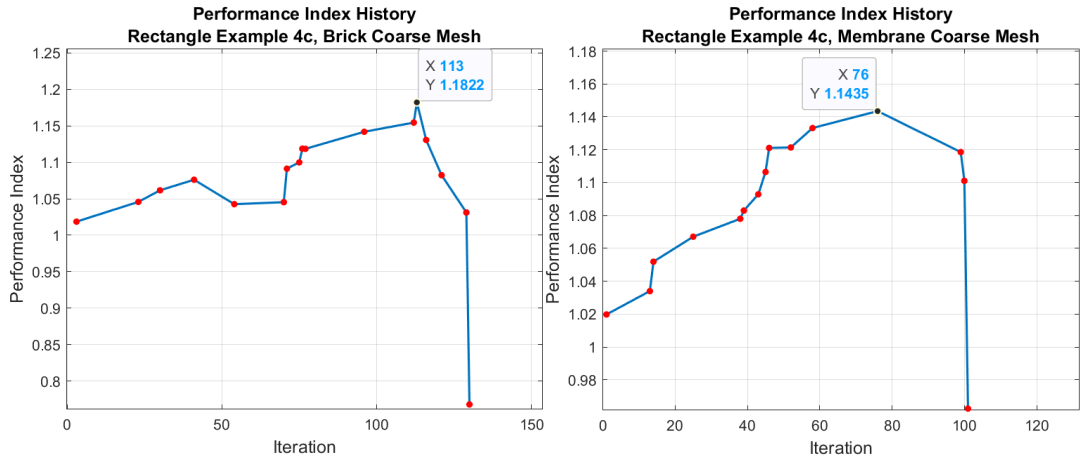
(a) (b)

Figure 4.72: Performance Index History of Rectangle Example 4b Medium Mesh: (a) Brick Model; (b) Membrane Model



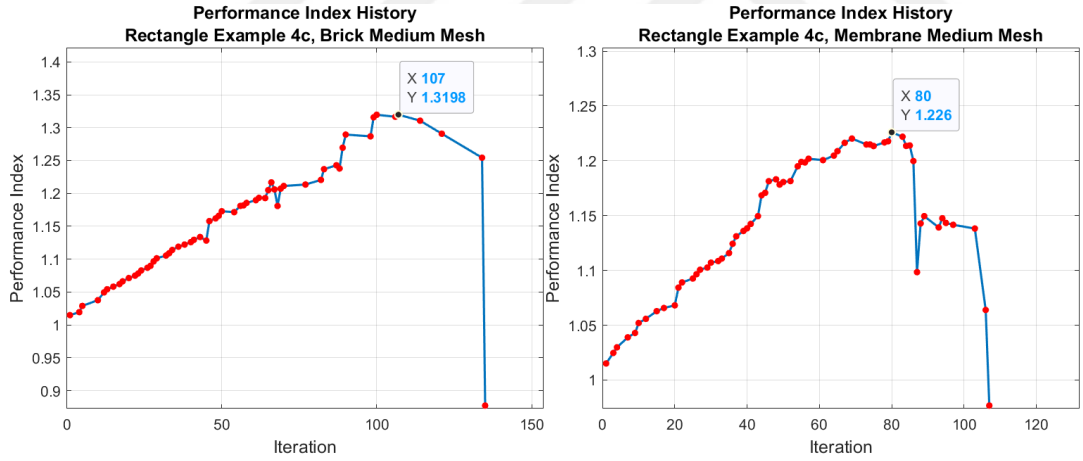
(a) (b)

Figure 4.73: Performance Index History of Rectangle Example 4b Fine Mesh: (a) Brick Model; (b) Membrane Model



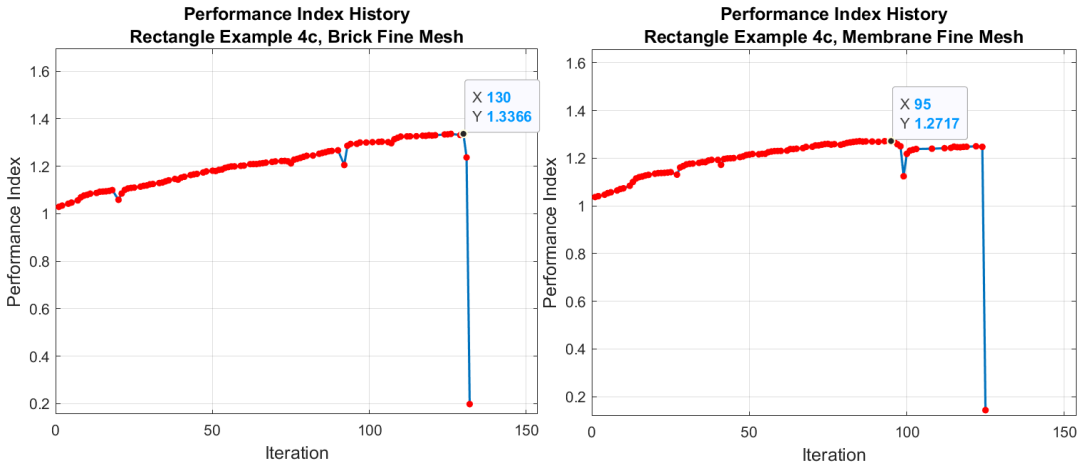
(a) (b)

Figure 4.74: Performance Index History of Rectangle Example 4c Coarse Mesh: (a) Brick Model; (b) Membrane Model



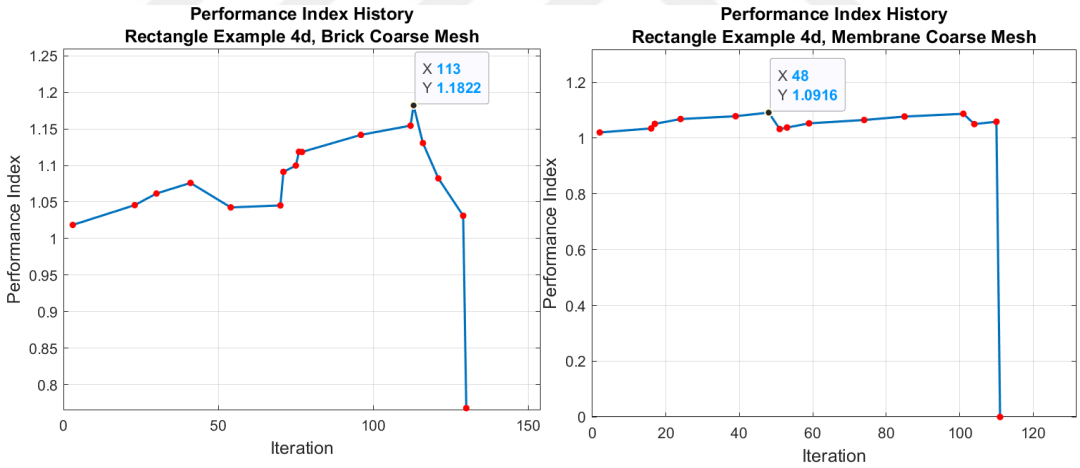
(a) (b)

Figure 4.75: Performance Index History of Rectangle Example 4c Medium Mesh: (a) Brick Model; (b) Membrane Model



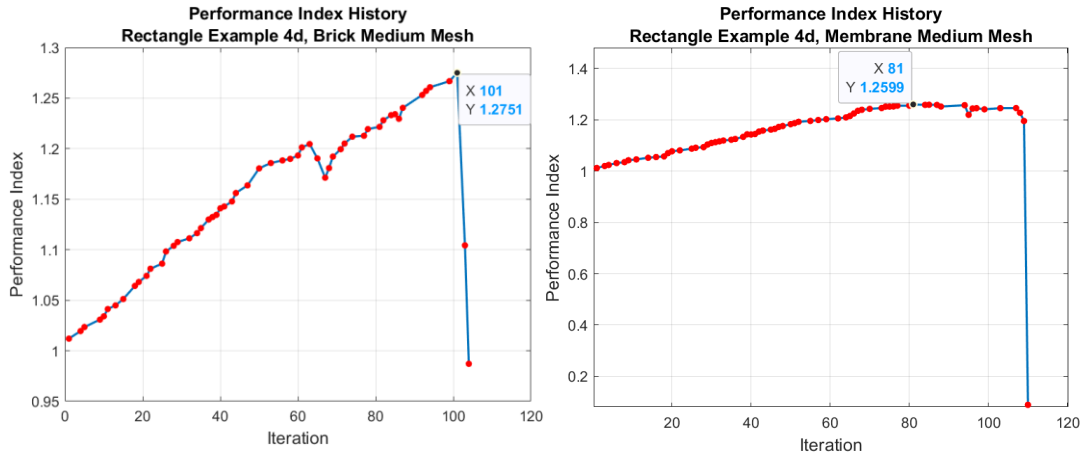
(a) (b)

Figure 4.76: Performance Index History of Rectangle Example 4c Fine Mesh: (a) Brick Model; (b) Membrane Model



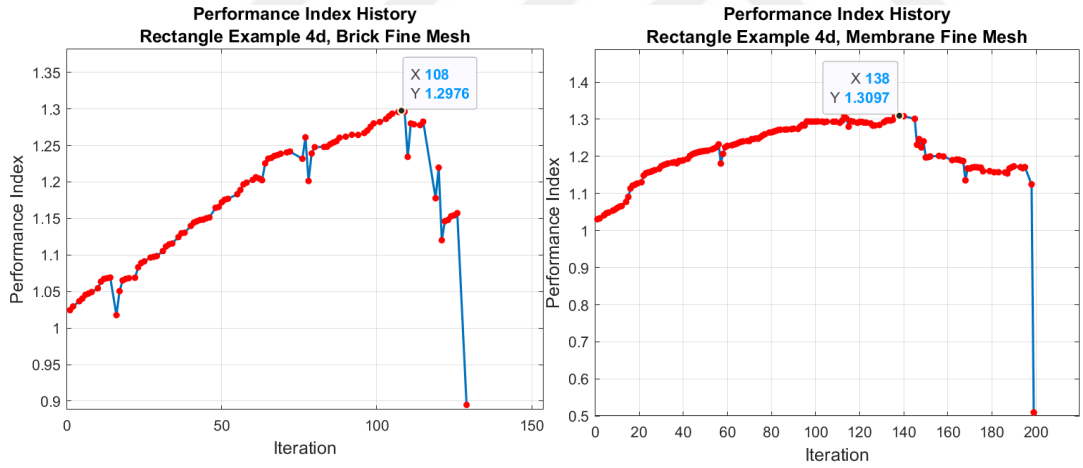
(a) (b)

Figure 4.77: Performance Index History of Rectangle Example 4d Coarse Mesh: (a) Brick Model; (b) Membrane Model



(a) (b)

Figure 4.78: Performance Index History of Rectangle Example 4d Medium Mesh: (a) Brick Model; (b) Membrane Model



(a) (b)

Figure 4.79: Performance Index History of Rectangle Example 4d Fine Mesh: (a) Brick Model; (b) Membrane Model

The trend of the performance index history curve is examined. This is done so to decide whether the maximum performance index value can be used as a stopping criterion. Apart from some exceptions (Figures 4.68 (b), 4.74 (b), 4.77 (b)) the Performance Index history of both models shows that the performance curve

demonstrates one or more peaks before reaching the maximum value. The iteration corresponding to the largest performance index occurs randomly. If the optimization procedure is stopped when the performance index begins to drop, the element distribution with a bigger performance index can be missed out. Hence, the performance index should not be used as a stopping criterion. Moreover, the previous results show that the performance index is not always a sufficiently robust parameter to reliably identify the best material distribution. Since the remaining number of elements directly affect the performance index, more refined and stable element distributions may be overlooked. Many optimized element distributions with lower performance index values were often more statically stable and demonstrated clearer load paths.

It is also deduced that the displacement-based optimization method is less robust compared to the stress-based optimization method. As mesh refinement increased, the displacement-based method required a progressively smaller Rejection Ratio (RR), whereas the stress-based method was able to maintain a constant RR of 1% across all analyses. Stress-based optimization produces clearer element distributions compared to displacement-based optimization, more suitable for effectively distinguishing tensile and compressive regions and designing a Strut-and-Tie Model.

Hence, the rest of the study will consider only the stress-based Evolutionary Structural Optimization Method when comparing the Brick and Membrane Models. The Real-Life Examples are utilized for this purpose. The stress-based optimization results of Real-Life Examples using Brick and Membrane Elements are given in the next chapter.

CHAPTER 5

BRICK VS MEMBRANE COMPARISON

This chapter presents the comparison between the element distributions of the Brick and Membrane Models. Based on the results that were discussed in the previous section, the stress-based evolutionary structural optimization method was preferred over the displacement-based method for the main comparative study.

For the Brick Model, which is composed of truss elements, axial stresses were naturally selected as the most appropriate measure. Previous studies show that Von Mises stresses are commonly used when evaluating membrane element stresses in stress-based evolutionary structural optimization (Almeida et al., 2013; Guan, 2005; Li et al., 2000; Liang et al., 1999; Querin et al., 1998; Xie and Steven, 1993). Hence, Von Mises stresses were determined to be the most suitable elimination criteria for the Membrane Model.

The results are given side by side for easier comparison. The Brick Model results are given on the left, and the Membrane Model results are given on the right. The Brick Model results depict the axial stresses of the remaining critical elements in the reinforced concrete member. The Membrane Model results depict the principal stresses of the remaining critical elements in the reinforced concrete member although the redundant membrane element eliminations were taken place based on the Von Mises stresses. The principal stresses are preferred to distinguish the compressive and tensile membrane element distribution in the model.

The Rejection Ratio (RR), Evolution Rate (ER), and the Steady-State Criterion (SS) is set to 1%, 1%, and 50% for every stress-based evolutionary structural optimization procedure. After the finite element analysis, the element stresses are checked to see whether they are below the RR*maximum stress threshold. Iterations continue until all elements are within the range of [RR*maximum stresses, maximum stresses].

Then, the RR is updated, and the iteration number is reset. The results also show the Rejection Ratio and the iteration number of the selected element distribution.

The stress-based evolutionary structural optimization does not provide a performance parameter for the selection of the most efficient element distribution. Hence, the results that are depicted in this chapter are manually selected. In a nutshell, evolutionary structural optimization is a weight minimization problem. Hence, the element distributions obtained further into the analysis are mostly chosen. Care is also taken that the selected results include refined clusters of compression and tension elements that can demonstrate the internal force flow within the structure and point out the possible locations of the strut and tie elements. The maximum stress values are also examined throughout the iterations. Usually towards the end of the optimization procedure, when critical elements that are representing load paths are deleted, stresses increase at a higher rate compared to its incline through the previous steps.

5.1 Real-Life Examples

In the second phase of this comprehensive comparison study, real-life examples are analyzed to evaluate the candidate finite elements and their overall performance in practical applications. The primary objective is to assess how these elements behave in real design problems rather than solely focusing on their robustness. Additionally, this phase aims to observe the response of the elements to different mesh configurations. To achieve this, five sets of trapezoidal hammerhead pier cap examples are selected. Due to their trapezoidal shapes, these domains are not meshed using a structured rectangular finite element approach. Each pier cap example is analyzed using two distinct mesh methodologies across three levels of refinement, similar to the previous tests. The selected results are presented in Figures 5.1 - 28.

Pier Cap 1

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 1 coarse mesh models (Mesh 1 and Mesh 2) are shown in Figure 5.1. The Strut-and-Tie Model suggested by Dawood and Abdul-Razzaq (2021) is presented in Figure 5.2. The optimization results for these models are presented in Figure 5.3.

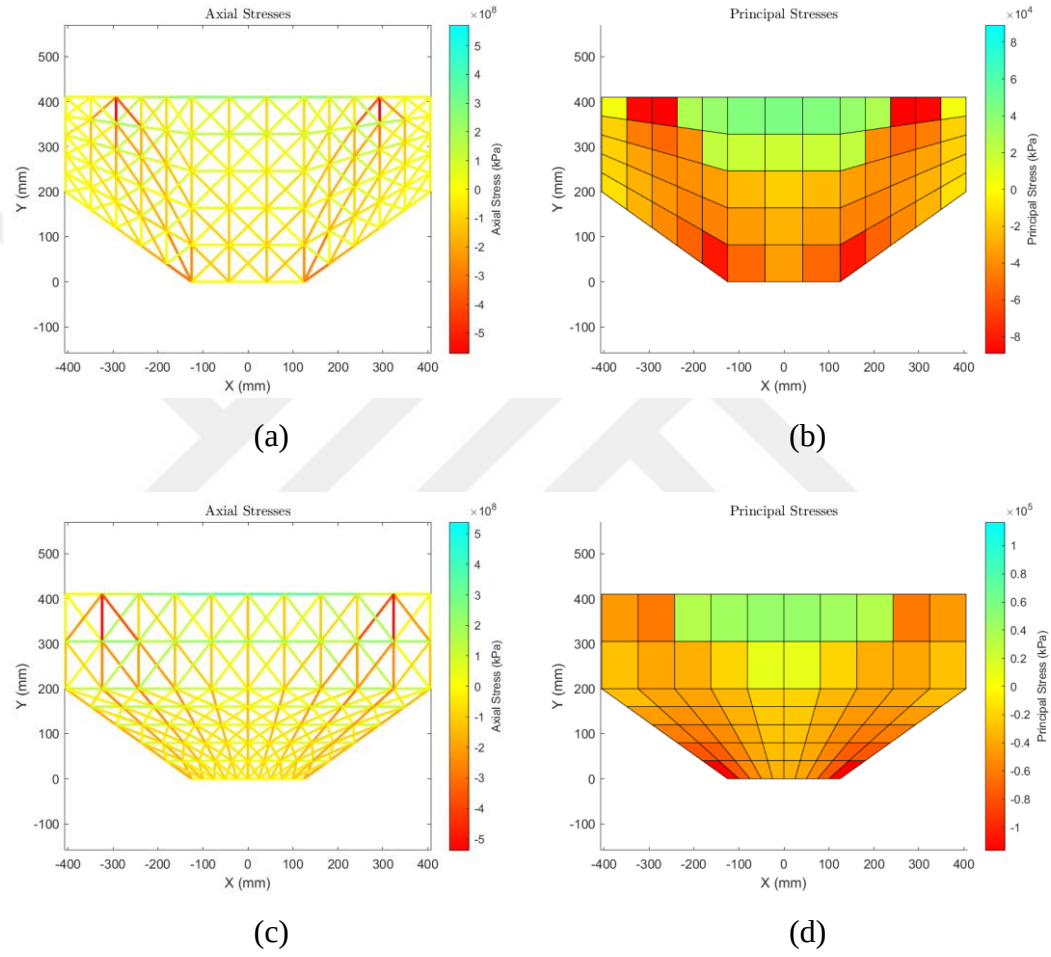


Figure 5.1: Pier Cap 1- Coarse Mesh: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

Both models demonstrate similar element distributions at the end of the optimization procedure for both types of mesh. The Mesh 1 results are identical (Figure 5.3 (a) and (b)). These results agree with the Strut-and-Tie model (Figure 5.2) proposed by Dawood and Abdul-Razzaq (2021). As the brick element is composed of 6 truss elements, the behavior of the struts can be seen in more detail. The Brick Model

(Figure 5.3 (a)) exhibits the bottle-shaped behavior of a strut element. The brick elements follow the stress trajectories, the tensile stresses that occur transversely within the strut bottle can be observed.

Mesh 2 and Mesh 1 results are consistent. Mesh 2 results have a higher percentage of remaining element distribution at the end of the optimization. In addition, the element percentage is more reduced in Membrane Model (Figure 5.3 (d)), as the Brick Model (Figure 5.3 (c)) retains more elements than the Membrane Model. The Brick Model reached the optimum steady-state before decreasing the number of the elements as much as the Membrane Model.

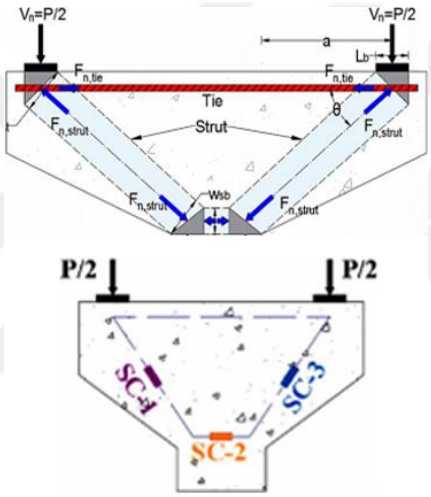
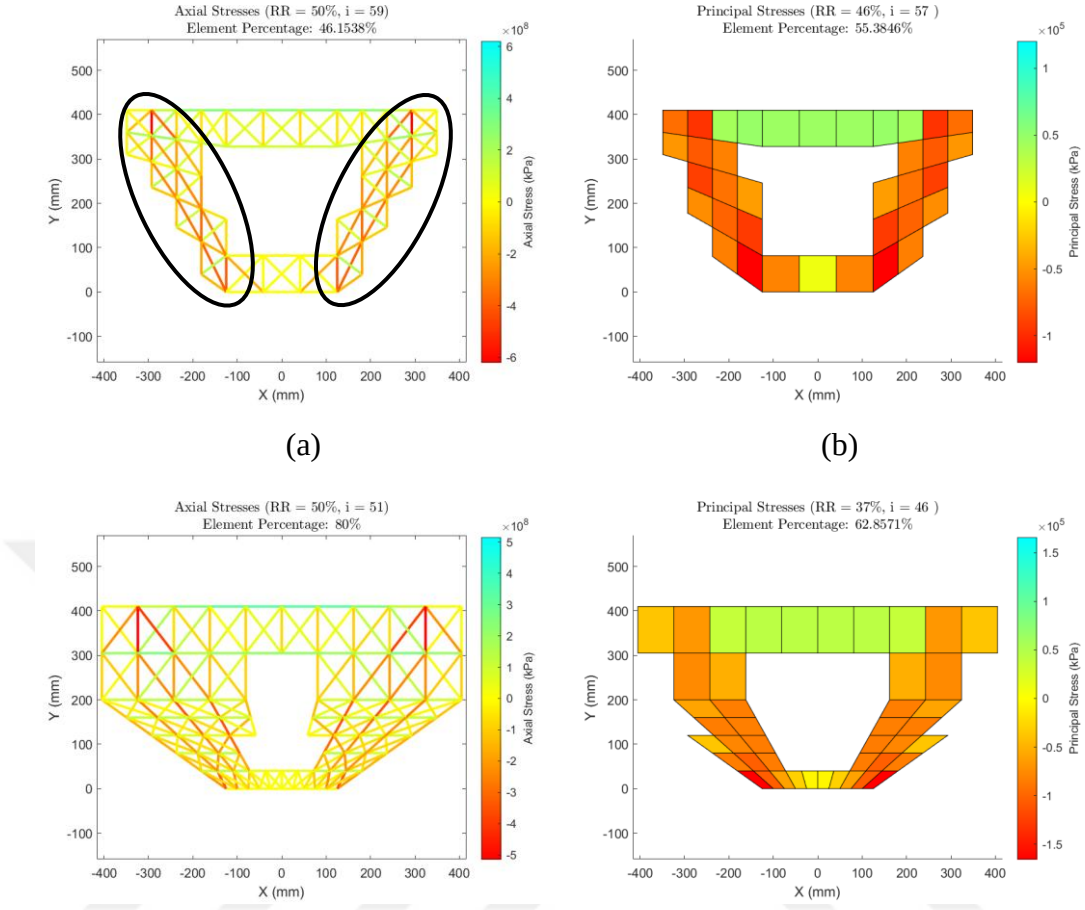


Figure 5.2: Pier Cap 1: STM by Dawood and Abdul-Razzaq (2021)



(a) (b)

(c) (d)

Figure 5.3: Pier Cap 1- Coarse Mesh Results: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

For all stress-based optimizations conducted using different methods and elements, the steady-state criterion (SS) was consistently set at 50%. For this specific example, a steady-state criterion of 75% was applied to evaluate whether the Brick Model could develop a more refined element distribution. The resulting distribution is shown in Figure 5.4. The Mesh 2 Brick Model was able to match the Mesh 1 (Figure 5.2 (a)) result with this configuration.

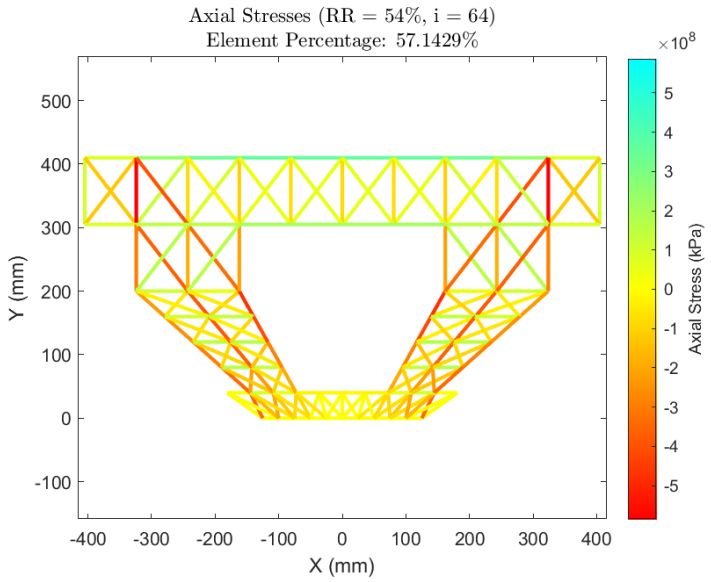


Figure 5.4: Pier Cap 1- Coarse Mesh 2 (SS=75%)

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 1 fine mesh models (Mesh 1 and Mesh 2) are shown in Figure 5.5. The optimization results for these models are presented in Figure 5.6. The fine mesh models for both Brick and Membrane Model yield similar element distribution. Mesh 1 and 2 Brick Models (Figure 5.6 (a) and (c)) has less elements than the Membrane Models (Figure 5.6 (b) and (d)) at the end of the optimization. The results largely agree with the Strut-and-Tie model proposed by the (Figure 5.2) proposed by Dawood and Abdul-Razzaq (2021). Pier Cap 1 results denote that stress-based optimization method is not sensitive to different mesh configuration for both types of elements. The Mesh 1 and 2 model results can directly be used as guidance for designing a Strut-and-Tie Model.

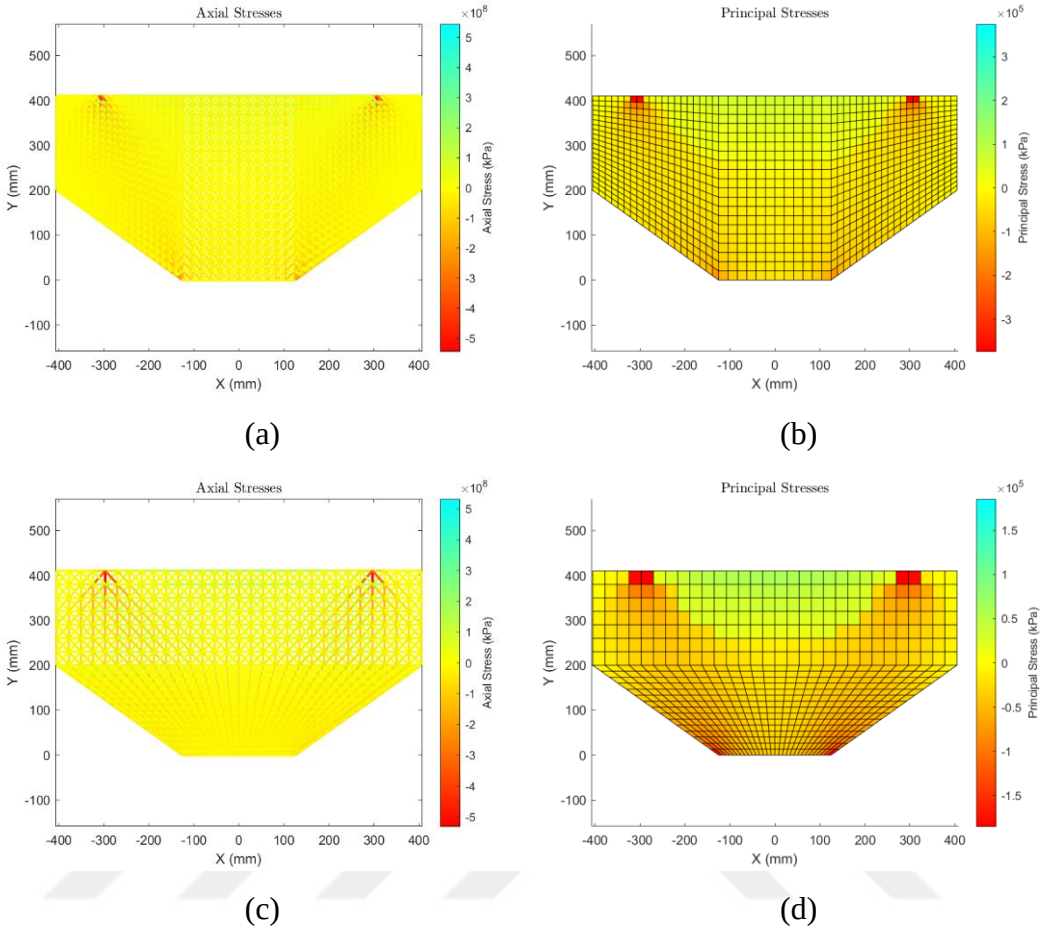


Figure 5.5: Pier Cap 1- Fine Mesh: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

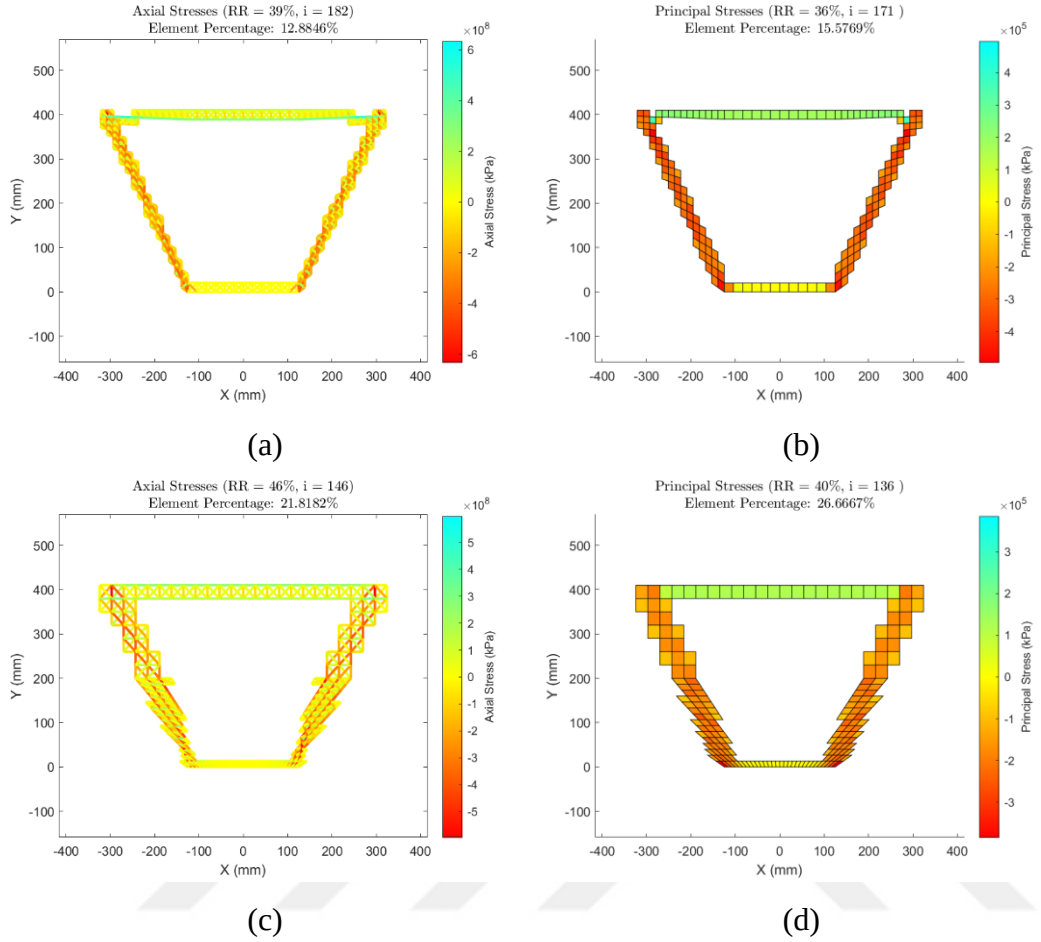
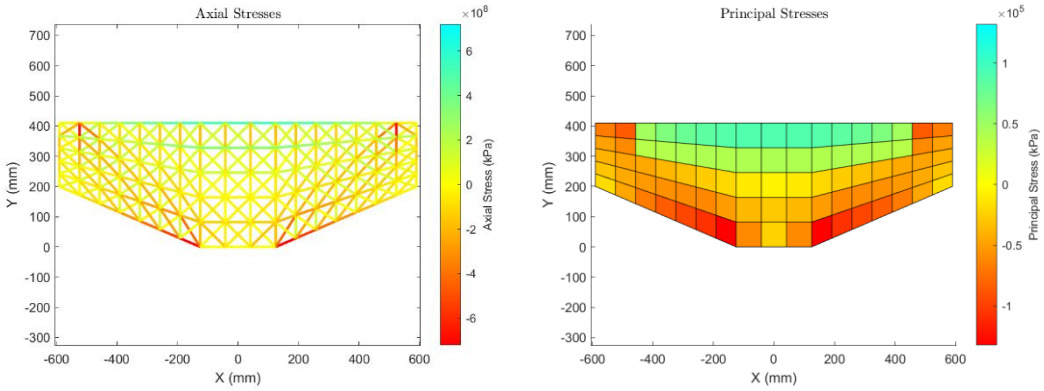


Figure 5.6: Pier Cap 1- Fine Mesh Results: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

Pier Cap 2

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 2 coarse mesh models (Mesh 1) are shown in Figure 5.7. The Strut-and-Tie Model suggested by Dawood and Abdul-Razzaq (2021) is presented in Figure 5.8. The optimization results for these models are presented in Figure 5.9.



(a) (b)

Figure 5.7: Pier Cap 2- Coarse Mesh 1 Results: (a) Brick Model; (b) Membrane Model

The coarse mesh 1 models for both Brick (Figure 5.9 (c)) and Membrane Model (Figure 5.9 (d)) yield almost identical element distribution at the end of the optimization procedure. The weight minimization objective is successfully completed. The achieved element distribution clearly demonstrates the internal force flow within the structure and can be used to design the strut and tie elements. The results are consistent with the Strut-and-Tie model (Figure 5.8) proposed by Dawood and Abdul-Razzaq (2021). As Dawood and Abdul-Razzaq (2021) offered the same Strut-and-Tie Model set up for all three Pier Cap examples.

Pier Cap 2 has a longer top length than Pier Cap 1. This increase in slenderness did not affect the optimized model configuration. However, in the earlier iteration, Brick Model demonstrated an additional cluster of compression elements towards the middle of the structure. This is highlighted in Figure 5.9 (a). Membrane Model did not exhibit such behavior throughout the optimization process Figure 5.9 (b).

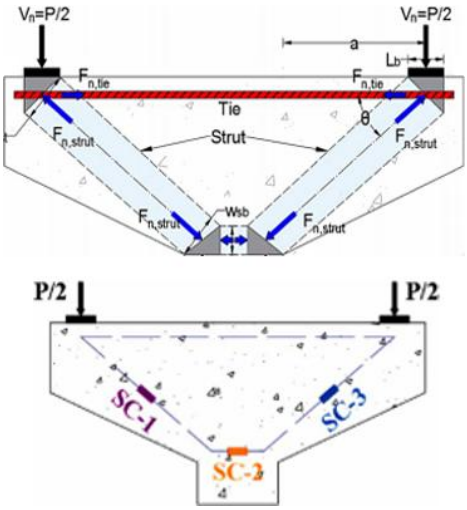


Figure 5.8: Pier Cap 2: STM by Dawood and Abdul-Razzaq (2021)

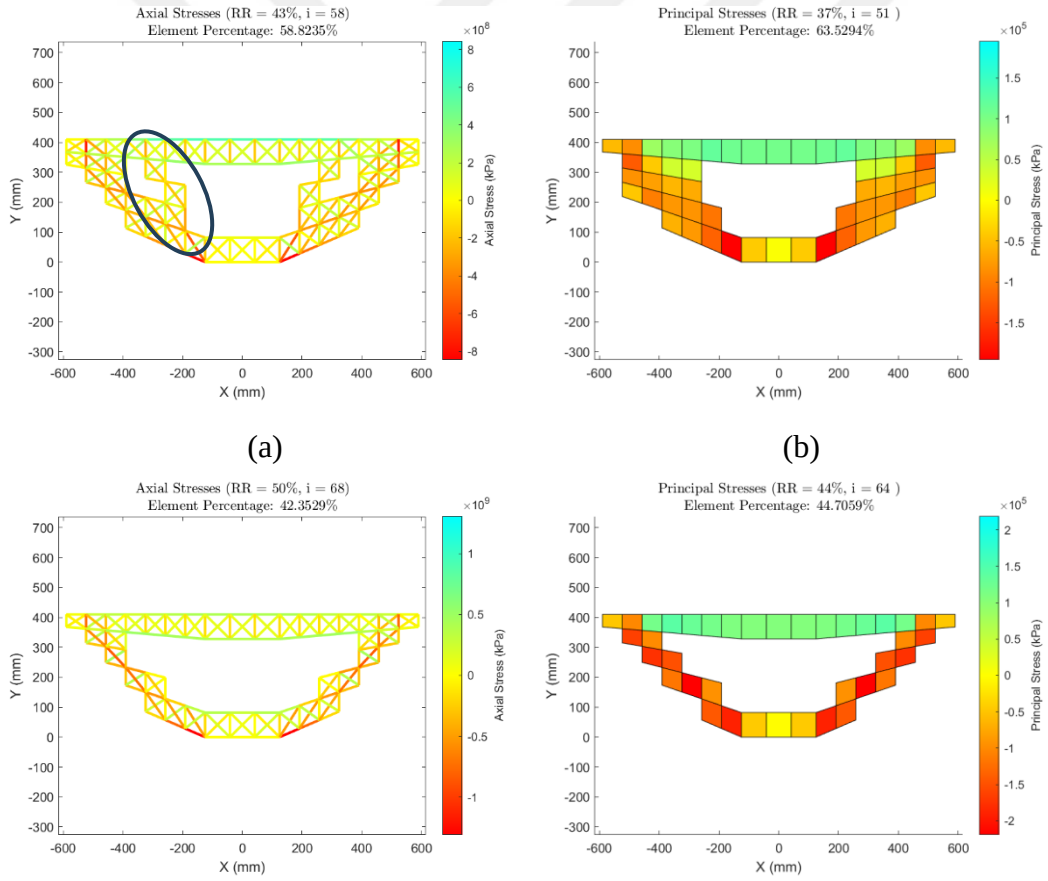
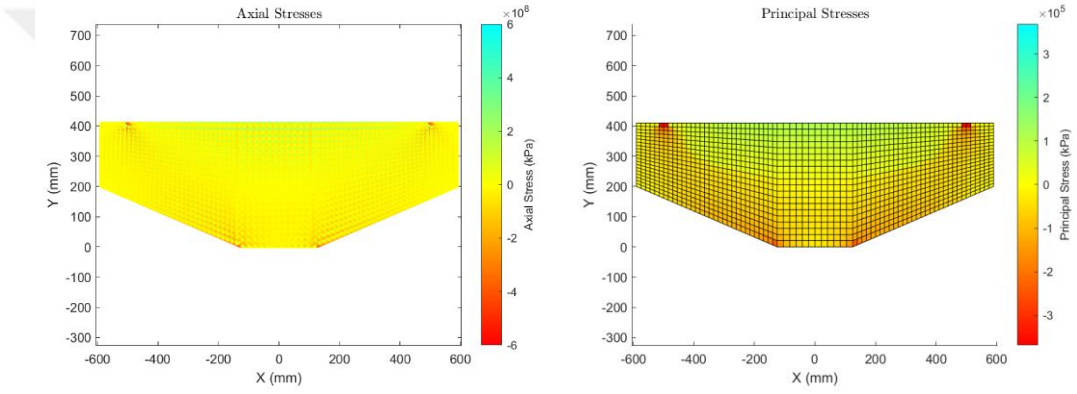


Figure 5.9: Pier Cap 2- Coarse Mesh 1: (a) Earlier Brick Model; (b) Earlier Membrane Model; (c) Brick Model; (d) Membrane Model

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 2 fine mesh models (Mesh 1) are shown in Figure 5.10. The optimization results for these models are presented in Figure 5.11. Similar to the coarse mesh results (Figure 5.9 (c) and (d)), the Brick and Membrane Models give matching results at the end of the optimization procedure. The element distributions clearly demonstrate the internal force flow within the structure and are consistent with the Strut-and-Tie model (Figure 5.8) proposed by Dawood and Abdul-Razzaq (2021). In the earlier iterations of the optimization procedure, both Brick and Membrane Model captures an additional compressive element cluster at the middle of the structure.



(a) (b)

Figure 5.10: Pier Cap 2- Fine Mesh 1: (a) Brick Model; (b) Membrane Model

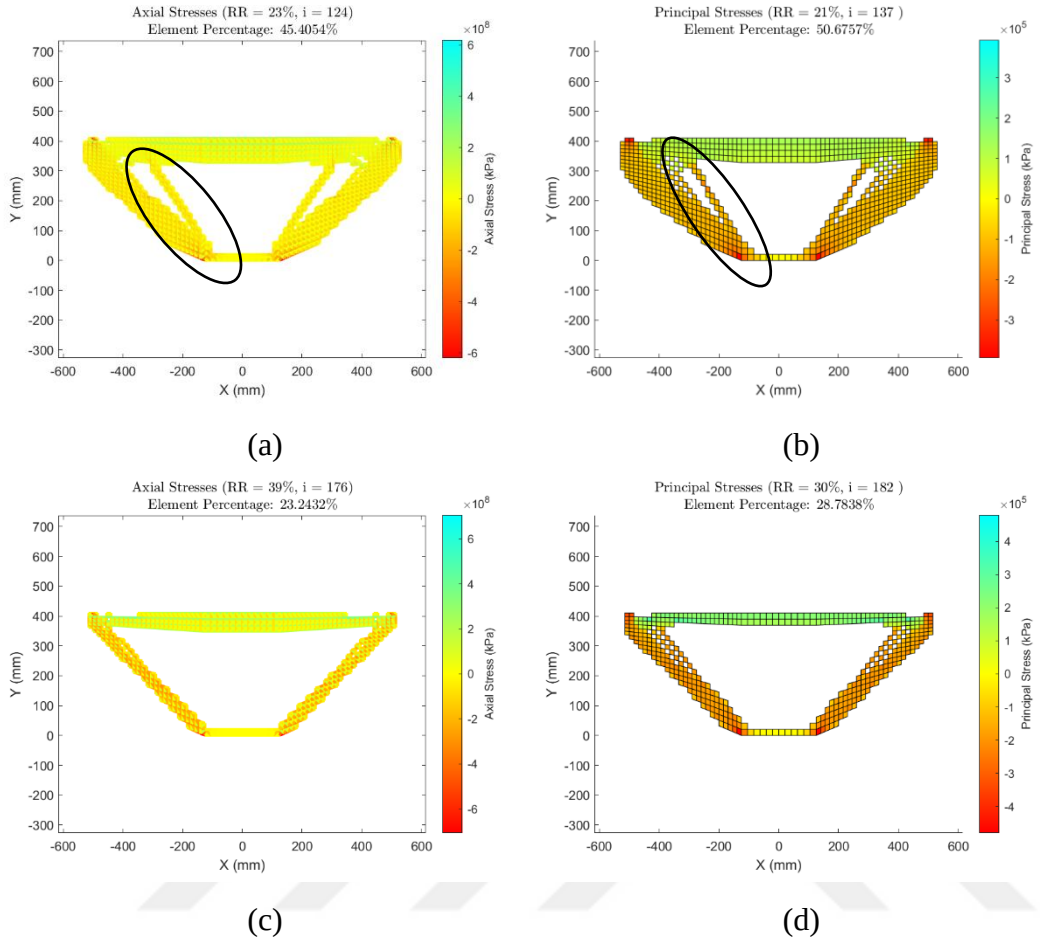
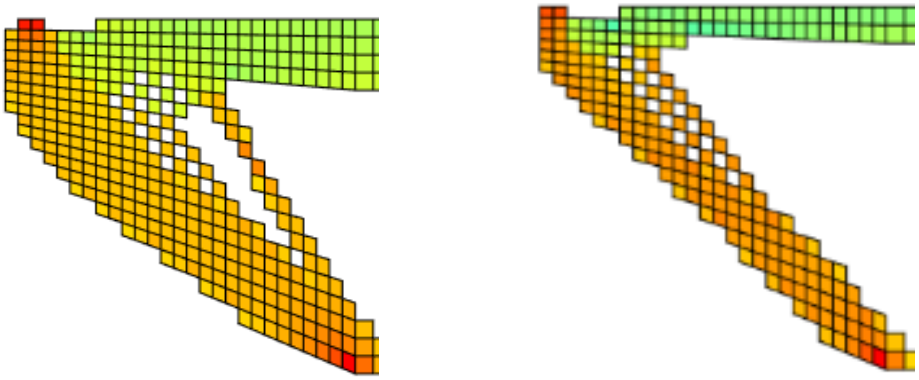


Figure 5.11: Pier Cap 2- Fine Mesh 1 Results: (a) Earlier Brick Model; (b) Earlier Membrane Model; (c) Brick Model; (d) Membrane Model

The Membrane Model exhibits checkerboarding at the earlier and the final element configuration. These checkerboarding patterns are highlighted in Figure 5.12 below. Brick Model shows no checkerboarding behavior. Although the checkerboarding pattern does not disrupt the element distribution, the internal force flow is clearly seen.



(a) (b)

Figure 5.12: Pier Cap 2 Fine Mesh 1- Checkerboarding: (a) Earlier Membrane Model; (b) Membrane Model

These results demonstrate that an intermediate model configuration can also be used to determine a Strut-and-Tie Model. The final optimization result, where the number of elements is most reduced to meet the minimization objective, can be directly referenced as a Strut-and-Tie Model. However, intermediate steps provide valuable insight into the internal force flow and can suggest alternative Strut-and-Tie configurations.

The results also show that the Brick Model produces refined and alternative element distributions, even with coarse mesh refinement. In contrast, the Membrane Model requires a finer mesh to accurately capture the internal force flow in intermediate optimization steps. The load paths obtained with the Brick Model using coarse meshes often require higher mesh refinement in the Membrane Model to achieve comparable results. This highlights the efficiency and robustness of the Brick Model in generating clear and refined load paths with less computational effort.

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 2 medium and fine mesh models (Mesh 2) are shown in Figure 5.13. The optimization results for these models are presented in Figure 5.14.

Results indicate that the Membrane Model is more effective in meeting the minimization objective. Both medium and coarse mesh Membrane Models (Figure

5.14 (b) and (d)) retain fewer elements at the end of the optimization. The Mesh 2 Membrane results align with the corresponding Mesh 1 models. In both cases, the Membrane Models are consistent with the Strut-and-Tie Model proposed by Dawood and Abdul-Razzaq (2021) (Figure 5.8). The element distribution is refined and clearly identifies compression and tensile regions.

The Brick Model, however, shows more sensitivity to different mesh types. The remaining elements tend to cluster toward the top of the pier cap, where the finite element dimensions are more structured and organized. The medium mesh model exhibits separation in the compressive region, with additional compressive element clusters emphasized in Figure 5.14 (c). The fine Mesh 2 Brick Model also introduces new tensile element regions, as shown in Figure 5.14 (a). These tensile regions are diagonally oriented toward the outer compressive elements.

This element orientation cannot be dismissed, as it suggests that increasing the top length of the pier cap leads to the formation of new compressive internal force paths. This branching in the compressive region resembles the behavior observed in earlier iterations of the fine Mesh 1 results for the Brick and Membrane Models (Figure 5.11 (a) and (b)) and the coarse Mesh 1 Brick Model (Figure 5.9 (a)). This pattern appears more clearly in the Pier Cap 3 results which are presented below. Still, these findings indicate that the Membrane Model is less sensitive to mesh type compared to the Brick Model.

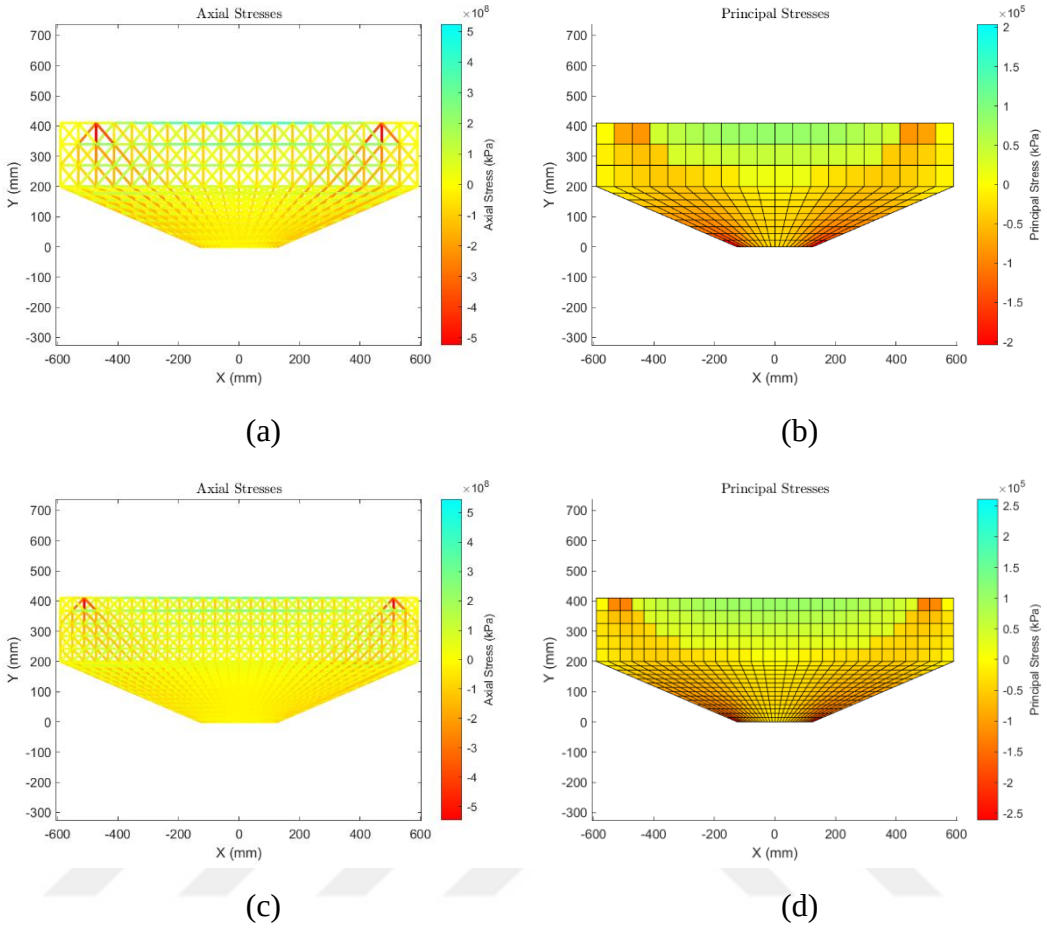


Figure 5.13: Pier Cap 2- Mesh 2: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

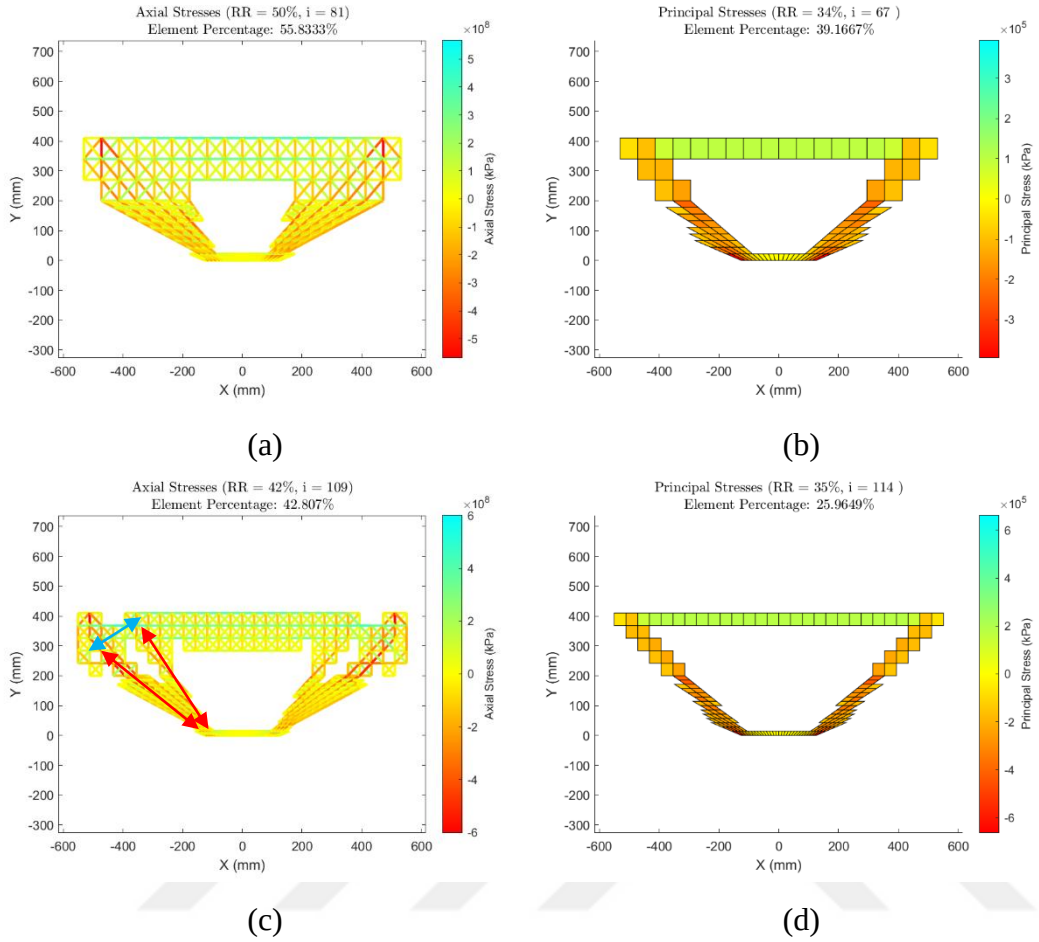


Figure 5.14: Pier Cap 2- Mesh 2 Results: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

Pier Cap 3

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 3 medium and fine mesh models (Mesh 1) are shown in Figure 5.15. The Strut-and-Tie Model suggested by Dawood and Abdul-Razzaq (2021) is presented in Figure 5.16. The optimization results for these models are presented in Figure 5.17.

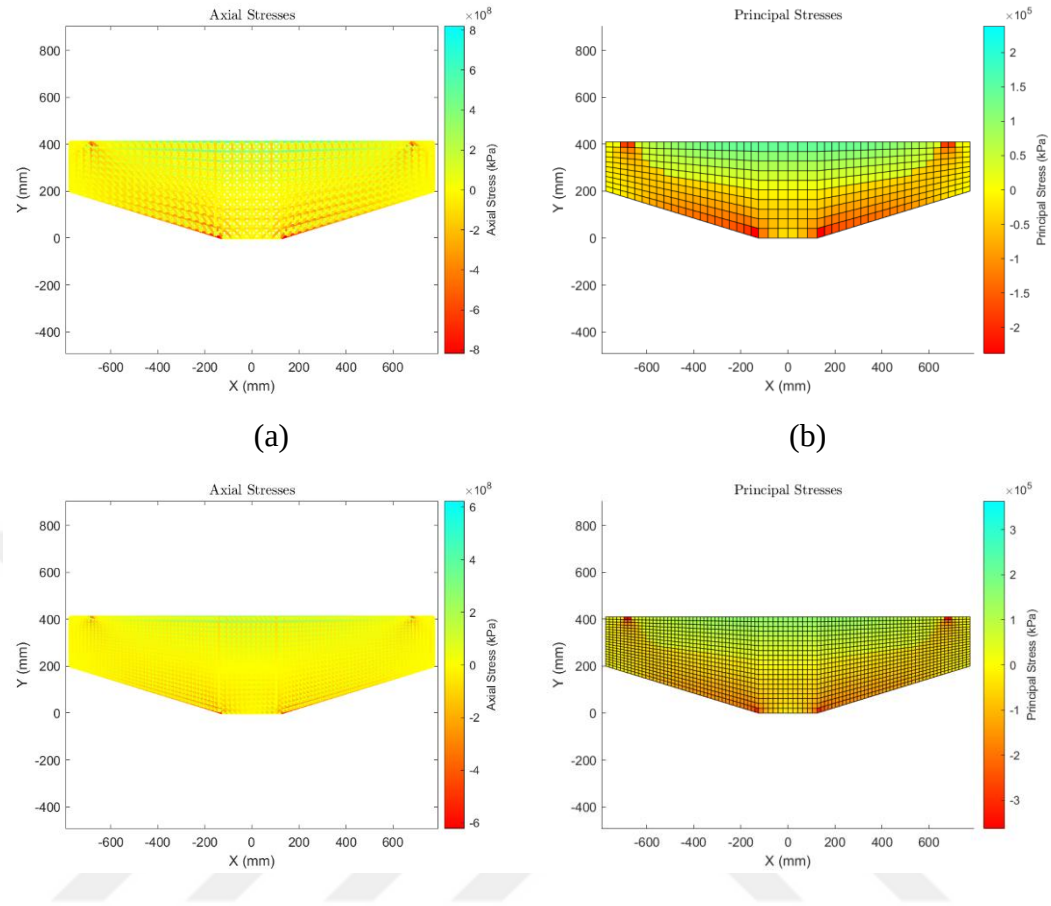


Figure 5.15: Pier Cap 3- Mesh 1: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

These results differ from the Strut-and-Tie Model proposed by Dawood and Abdul-Razzaq (2021) (Figure 5.16). As the top length of the pier cap increases, a Strut-and-Tie Model with two trusses connected by a bottom tie becomes inefficient.

Brick Model (Figure 5.17 (a) and (c)) results remain consistent across all mesh refinement levels, with similar strut and tie orientations. However, the medium and fine mesh Membrane Models (Figure 5.17 (b) and (d)) show some variations. In the medium mesh Membrane Model (Figure 5.17 (b)), the additional tensile region appears closer to the top of the structure, forming a single triangle between the tensile

region and the two compression regions. In contrast, in the fine mesh Membrane Model (Figure 5.17 (d)) and the medium and fine mesh Brick Models (Figure 5.17 (a) and (c)), this additional tensile region is located closer to the mid-depth of the structure. Two triangular formations are observed: one formed by the top tensile region, inner tensile region, and outer compression region, and another by the inner tensile region, outer compression region, and inner compression region. These element clusters are highlighted in the figures.

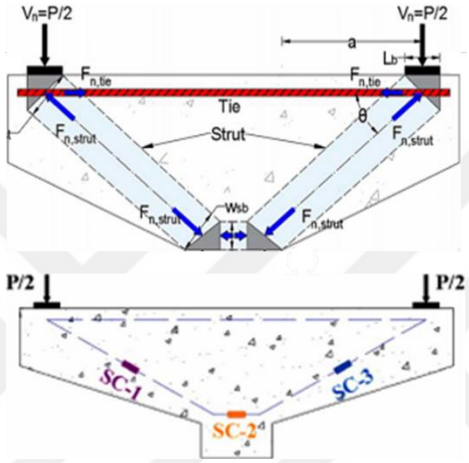


Figure 5.16: Pier Cap 3: STM by Dawood and Abdul-Razzaq (2021)

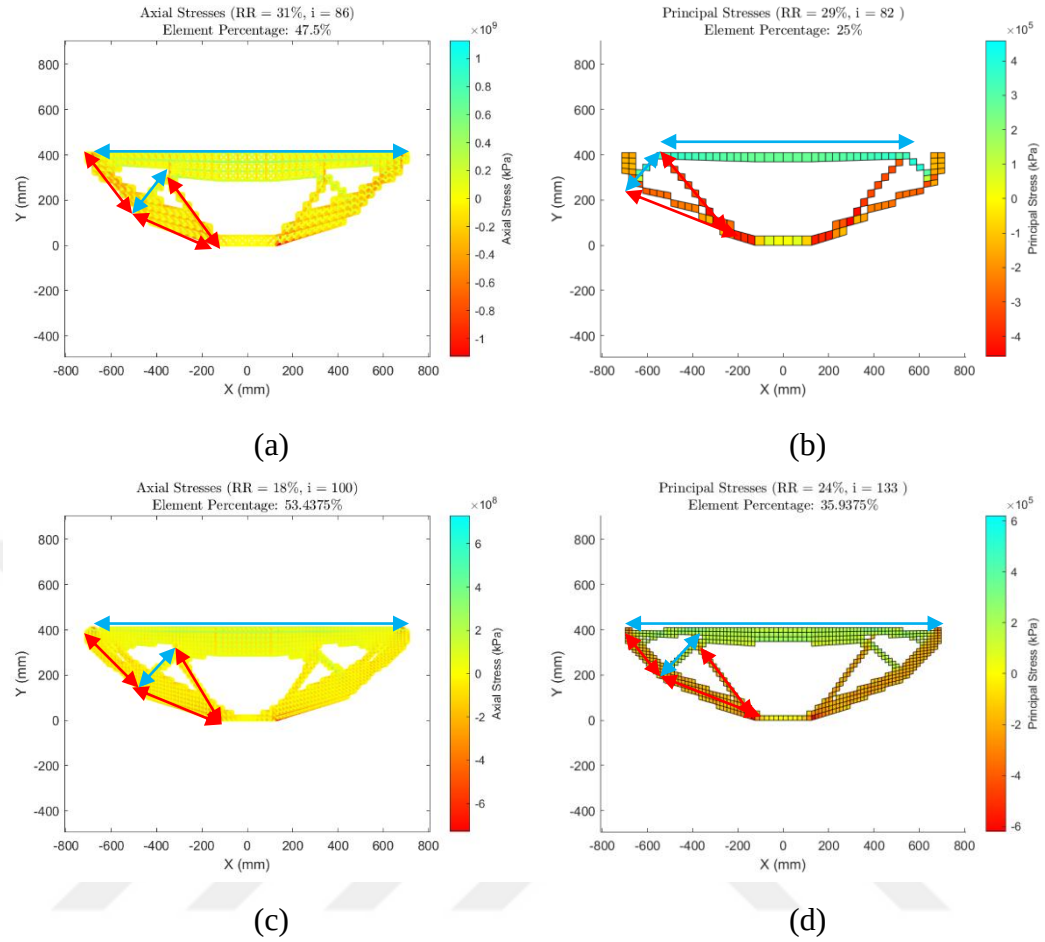


Figure 5.17: Pier Cap 3- Mesh 1 Results: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 3 medium and fine mesh models (Mesh 2) are shown in Figure 5.18. The optimization results for these models are presented in Figure 5.19.

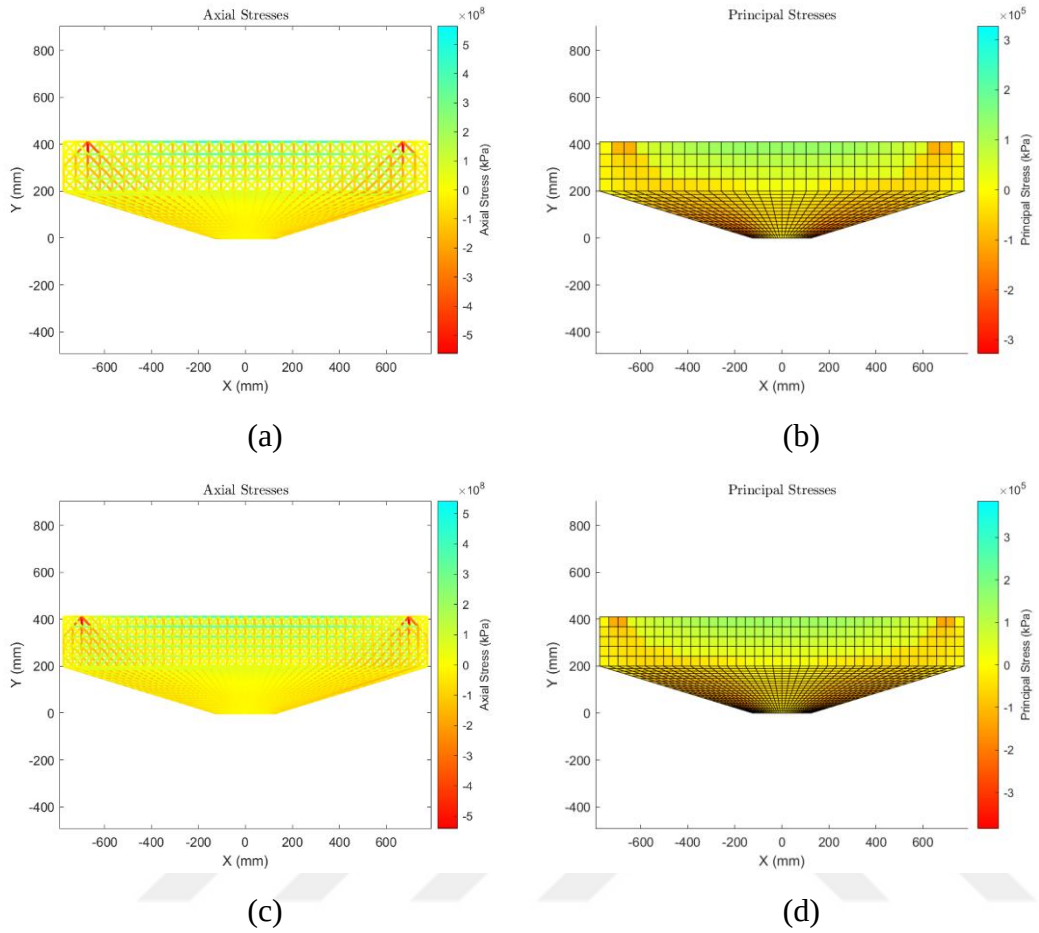


Figure 5.18: Pier Cap 3- Mesh 2: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

Similar to Pier Cap 2 (Figure 5.14 (a) and (c)), the remaining elements in the Brick Model (Figure 5.19 (a) and (c)) cluster toward the top of the structure. However, in this case, the bottom compression regions are more refined compared to the Mesh 1 results. The optimization using Mesh 2 configuration results in fewer remaining elements. Overall, the results align with those of Mesh 1 (Figure 5.17 (a) and (c)), with the main difference being that the compression regions obtained in Mesh 2 models are wider. Other than this variation, the compressive and tensile element configurations remain consistent. The orientation of the compression and tensile element groups is highlighted in the figures.

Unlike the Brick Model, the Membrane Model has not shown a tendency for element accumulation toward the top of the pier cap. However, in this case, the medium mesh model (Figure 5.19 (b)) does not provide enough refinement to clearly distinguish the inner and outer compression regions. The compression and tensile regions marked in the figure are identified with guidance from other results. Once again, the Brick Model demonstrates its ability to produce a clearer internal load path compared to the Membrane Model. Since each brick element consists of six truss elements, the load path can be traced down to the one-dimensional truss elements within a coarser Brick Model mesh.

The fine mesh Membrane Model (Figure 5.19 (d)), however, provides a more refined element distribution than the medium mesh model (Figure 5.19 (b)), offering better guidance for Strut-and-Tie Model design. Although the inner compression element group does not follow a straight line, its results align with the corresponding Mesh 1 model (Figure 5.17 (d)). Despite this irregularity, it still offers valuable insight into designing the Strut-and-Tie Model.

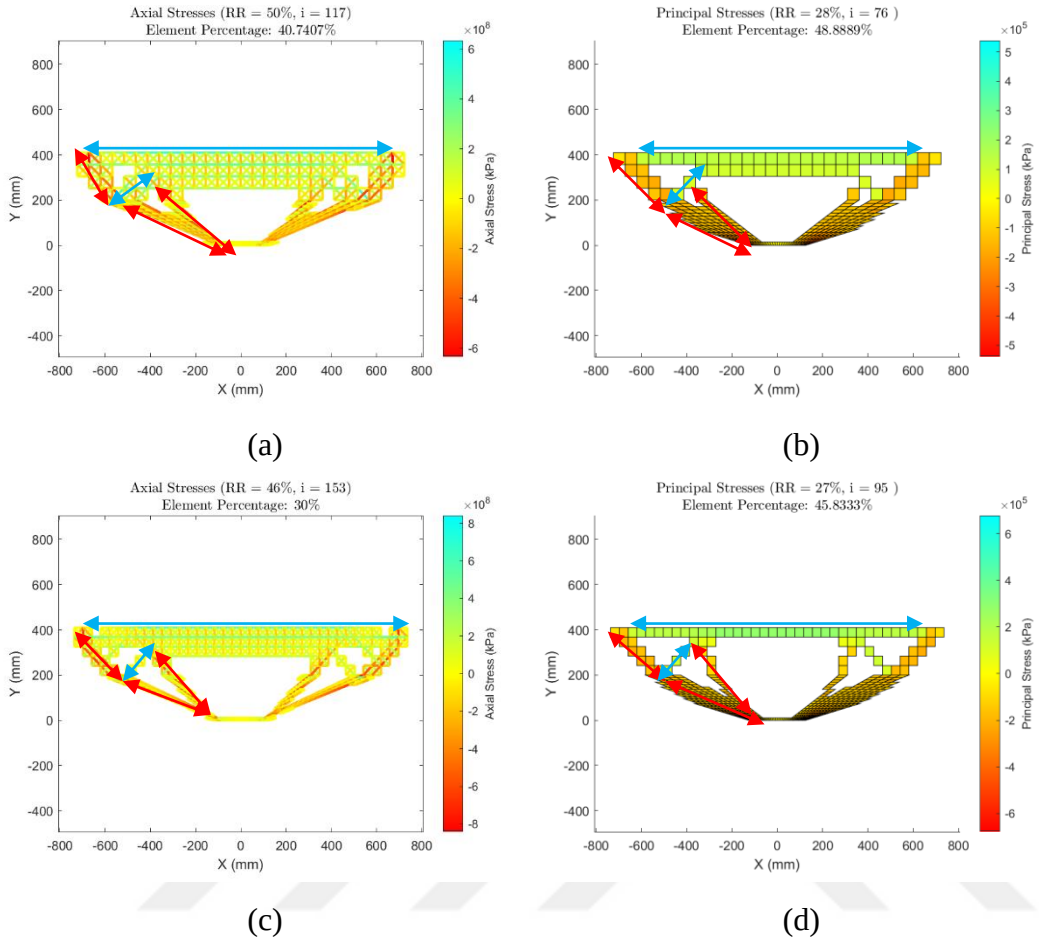


Figure 5.19: Pier Cap 3- Mesh 2 Results: (a) Medium Mesh Brick Model; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

Pier Cap 4

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 4 medium and fine mesh models (Mesh 1) are shown in Figure 5.20. The Strut-and-Tie Model proposed by Bedru (2014) for the design of Pier Cap 4 is presented in Figure 5.21. The optimization results for these models are presented in Figure 5.22.

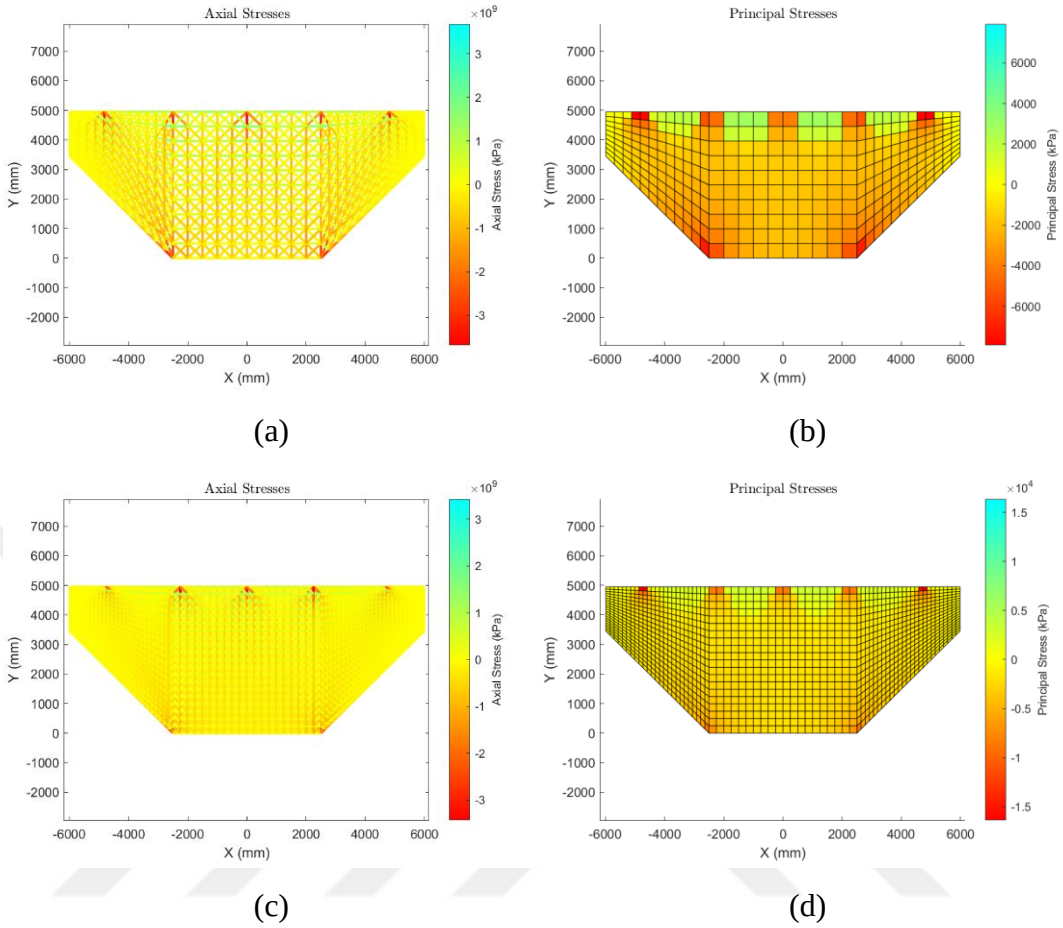


Figure 5.20: Pier Cap 4- Mesh 1: (a) Medium Mesh Brick Model ; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

The medium mesh Brick Model (Figure 5.22 (a)) performs better than the Membrane Model (Figure 5.22 (b)) in several aspects. It achieves a lower number of remaining elements, effectively meeting the minimization objective. Additionally, it maintains a higher percentage of similar stress levels across the structure and provides a refined element distribution that can serve as a precise guide for developing a Strut-and-Tie Model.

While the Membrane Model retains too many elements, this does not always render it ineffective. Previous examples have shown that if the remaining elements are

tightly grouped, they can still indicate internal force flow and assist in identifying a Strut-and-Tie configuration.

However, in this case, the remaining elements in the Membrane Model (Figure 5.22 (b)) are too scattered to clearly define the load path. As a result, it is difficult to extract meaningful insights from its configuration. Only by referring to the Brick Model (Figure 5.22 (a)) can one recognize that the Membrane Model vaguely suggests the presence of strut elements. The Brick Model, on the other hand, distinctly reveals the two struts on both sides and the two central strut elements forming a triangular configuration.

The Brick Model remains consistent across different mesh refinement levels, with only a slight variation in the angle formed by the central compressive elements. However, the fine mesh Membrane Model (Figure 5.22 (d)) differs significantly from the medium mesh Membrane Model (Figure 5.22 (b)) and both of the Brick Models (Figure 5.22 (a) and (c)).

The element configuration obtained from the Brick Model differs from the Strut-and-Tie Model proposed by Bedru (2014) (Figure 5.21). The strut elements at the sides align with the compressive element distribution in both Brick Model results. The outer-most loads are directly transferred to the support at the bottom of the pier cap by single strut elements. However, the three loads at the middle are handled differently.

The Brick Model suggests that all outer loads are directly transferred to the supports via a single strut element. And the strut element configuration at the middle is actually the opposite compared to the proposed model by Bedru (2014). The Brick Model shows the middle load is split into two groups of compressive elements that transfer the force to the C-C-C-C nodes at the bottom of the pier cap.

The Membrane Model (Figure 5.22 (d)) does not exhibit this behavior. While it transfers the outer loads to the supports in a similar manner, no element clusters remain in the middle to support the central load. Instead, a group of tensile elements

forms directly below the central load case. Since all elements in the middle of the pier cap are eliminated, the tensile elements support the top load and connect it to the elements at the sides. However, this configuration does not form a practical Strut-and-Tie Model.

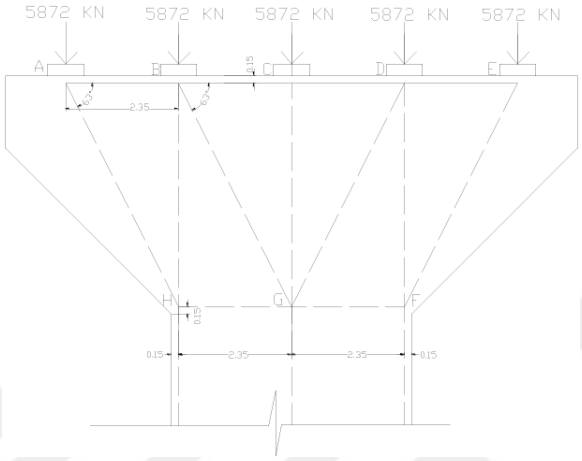


Figure 5.21: Pier Cap 4: STM by Bedru (2014)

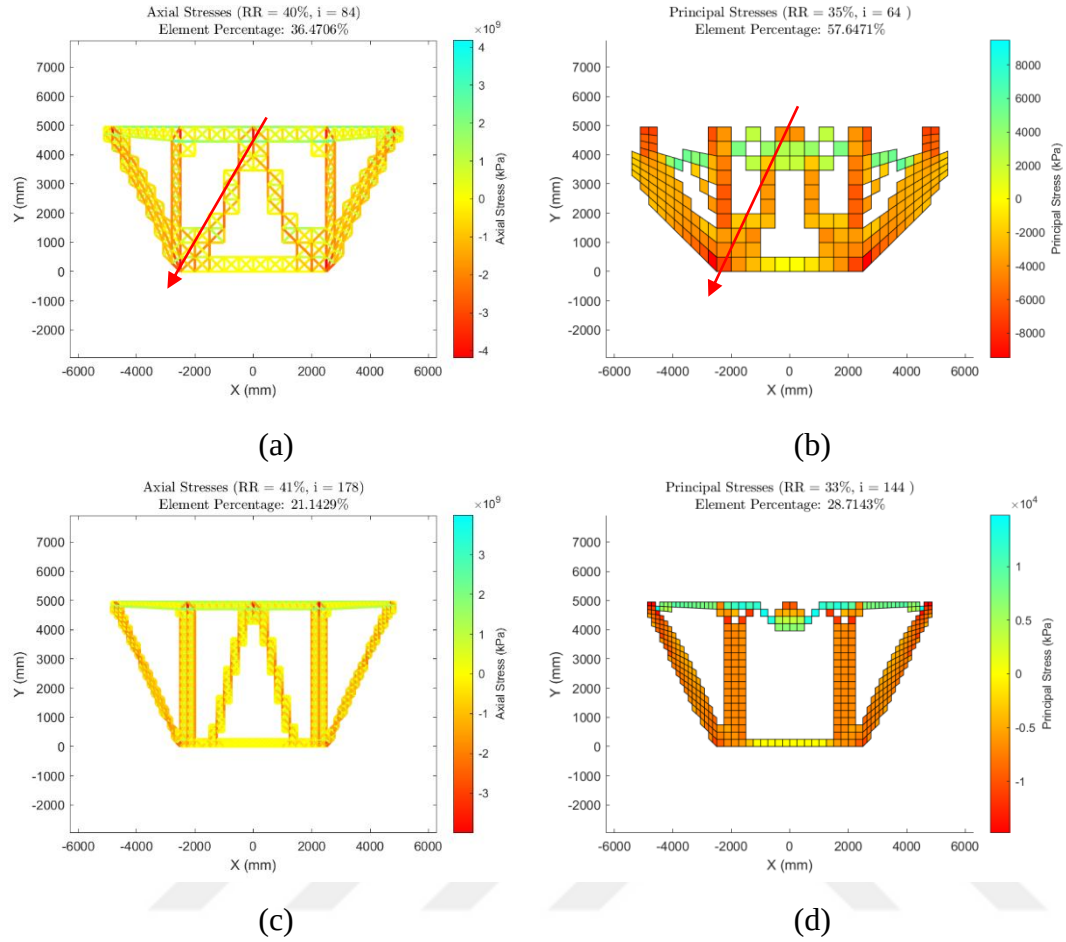


Figure 5.22: Pier Cap 4- Mesh 1 Results: (a) Medium Mesh Brick Model ; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 4 medium and fine mesh models (Mesh 2) are shown in Figure 5.23. The optimization results for these models are presented in Figure 5.24.

The Brick Model again demonstrates its sensitivity to different mesh types in this example. The element distribution remains concentrated near the top of the structure, where quadrilateral elements are rectangular. This behavior is most evident in the medium mesh Brick Model, which features an additional tensile region below the top of the pier cap, highlighted in Figure 5.24 (a). While the compressive elements in the middle form a similar angle to those in the Mesh 1 model (Figure 5.22 (a)),

the adjacent compressive elements are oriented toward the bottom-center rather than connecting directly to the support, as seen in Mesh 1. This deviation in the direction of the compressive elements, highlighted in Figure 5.24 (a), is influenced by the mesh, as the element distribution follows the orientation of the truss elements.

This example shows that the Membrane Model also responds to different mesh types. Both medium and fine meshes differ from their Mesh 1 counterparts. The medium mesh model (Figure 5.24 (b)), in this case, has a more refined element distribution than the Mesh 1 model (Figure 5.22 (b)). Similar to the behavior often observed in the Brick Model, the element distribution is influenced by the mesh. The central compression elements directly connect to the adjacent compressive element groups, as highlighted in Figure 5.24 (b). The fine mesh Membrane Model (Figure 5.24 (d)) produces results similar to the Mesh 1 Brick Models (Figure 5.22 (a) and (c)). The Membrane Model appears less affected by the orientation of quadrilateral elements in the mesh. Unlike the Brick Model, the compressive elements do not strictly follow the geometry of the quadrilateral elements, as shown in Figure 5.25.

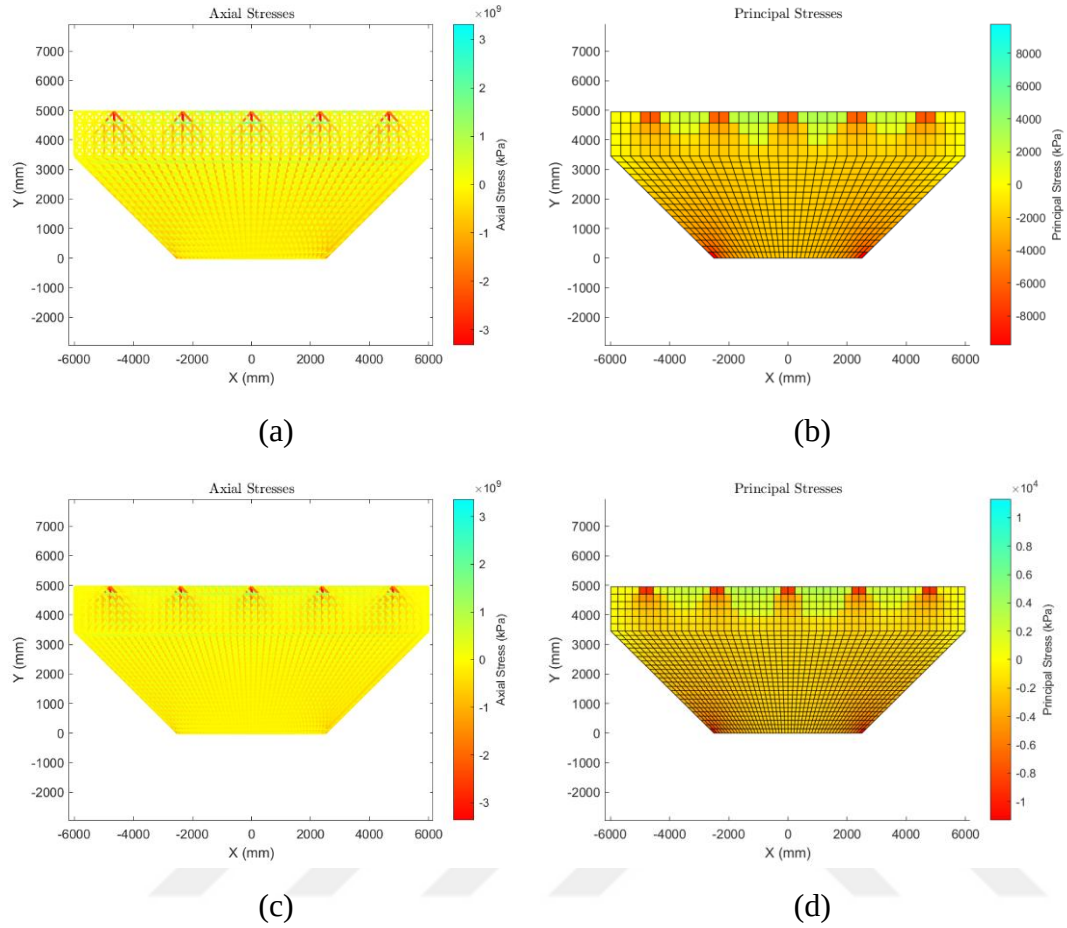
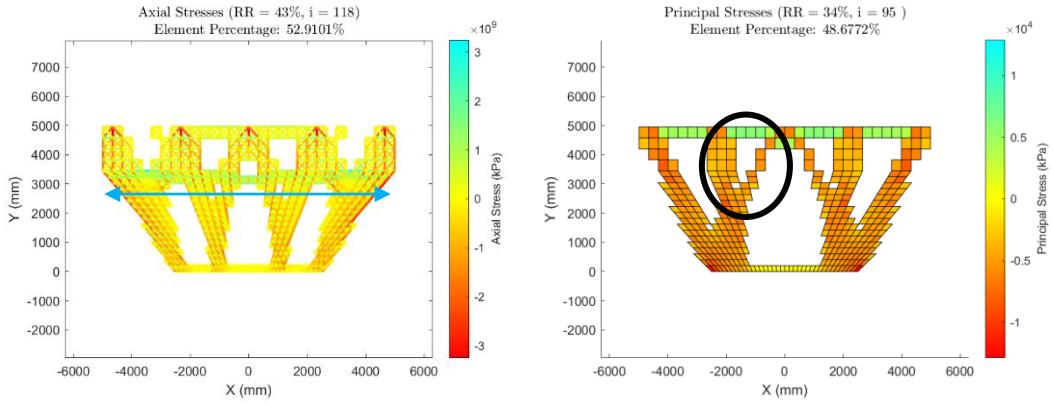
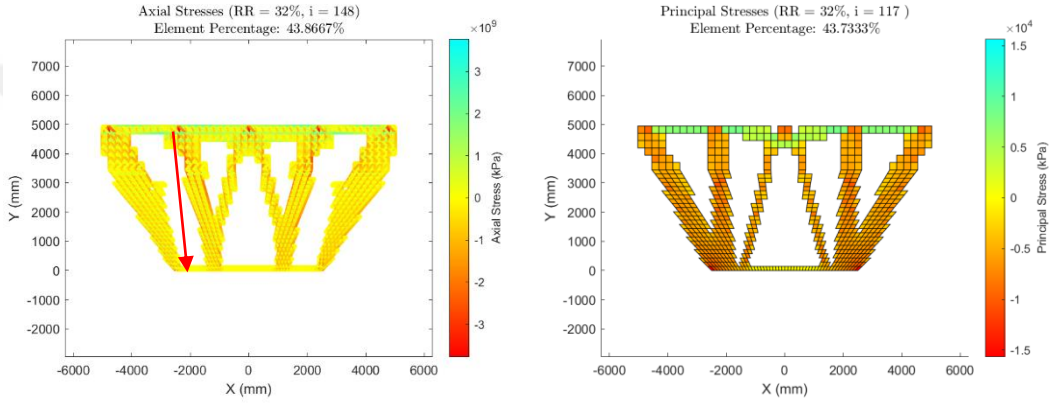


Figure 5.23: Pier Cap 4- Mesh 2: (a) Medium Mesh Brick Model ; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model

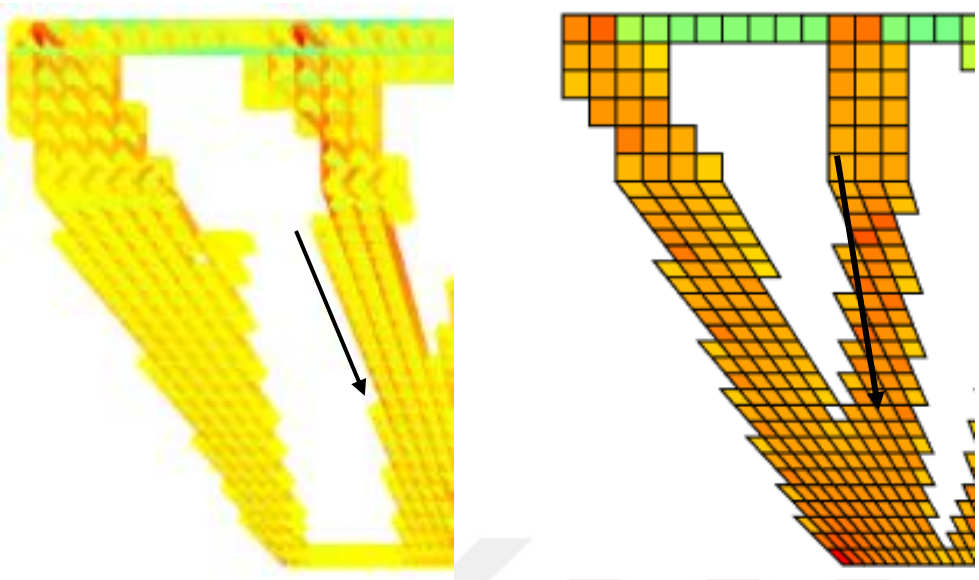


(a) (b)



(c) (d)

Figure 5.24: Pier Cap 4- Mesh 2 Results: (a) Medium Mesh Brick Model ; (b) Medium Mesh Membrane Model; (c) Fine Mesh Brick Model; (d) Fine Mesh Membrane Model



(c) (d)

Figure 5.25: Pier Cap 4- Mesh 2: (a) Fine Mesh Brick Model ; (b) Fine Mesh Membrane Model

Pier Cap 5

The initial axial and principal stress diagrams for the Brick and Membrane Pier Cap 5 coarse mesh models (Mesh 1 and 2) are shown in Figure 5.26. The Strut-and-Tie Model proposed by Mitchell and Collins (2013) for the design of Pier Cap 5 is presented in Figure 5.27. The optimization results for these models are presented in Figure 5.28.

The proposed model features strut elements with steep inclinations. The coarse mesh results (Figure 5.28) show compressive element groups similar to the strut elements in Mitchell and Collins' (2013) model. However, additional tensile element clusters appear between the two supports. These elements contribute to structural stability. Similar to Rectangle Example 4d, slender reinforced concrete structures require inclined tensile elements connecting compressive regions to maintain stability.

Both the Membrane and Brick Models retain over 50% of the elements, making it difficult to achieve a clear element distribution. However, the Brick Models (Mesh 1 and Mesh 2) provide a slightly better representation of internal force flow than the

Membrane Models. The Membrane results also exhibit checkerboarding patterns along the sides, as highlighted in Figure 5.28.

Both stress-based Membrane and Brick Models struggle with slender reinforced concrete members. The resulting topology often lacks a truss-like structure, making it difficult to interpret for Strut-and-Tie Model development. These results further confirm that slender topologies are unsuitable for Strut-and-Tie Modelling, and flexural beam theory provides a more appropriate approach.

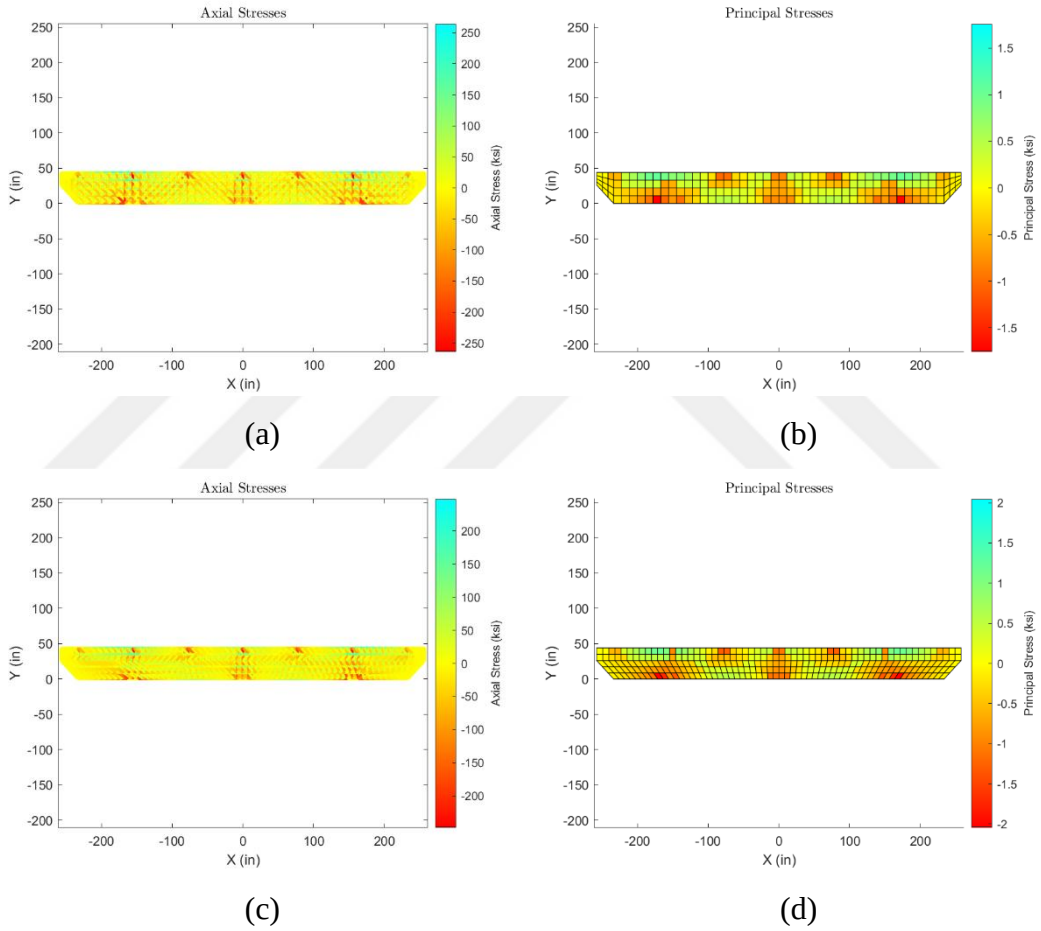


Figure 5.26: Pier Cap 5- Coarse Mesh: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

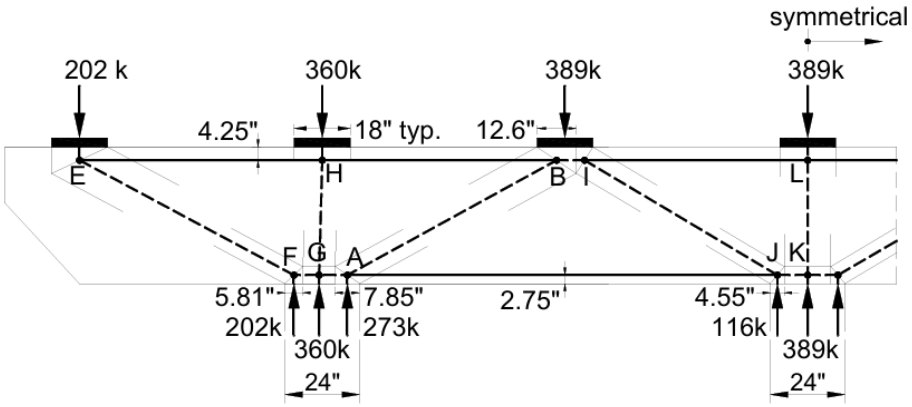


Figure 5.27: The proposed Strut-and-Tie Model of Pier Cap 5 (Mitchell and Collins, 2013)

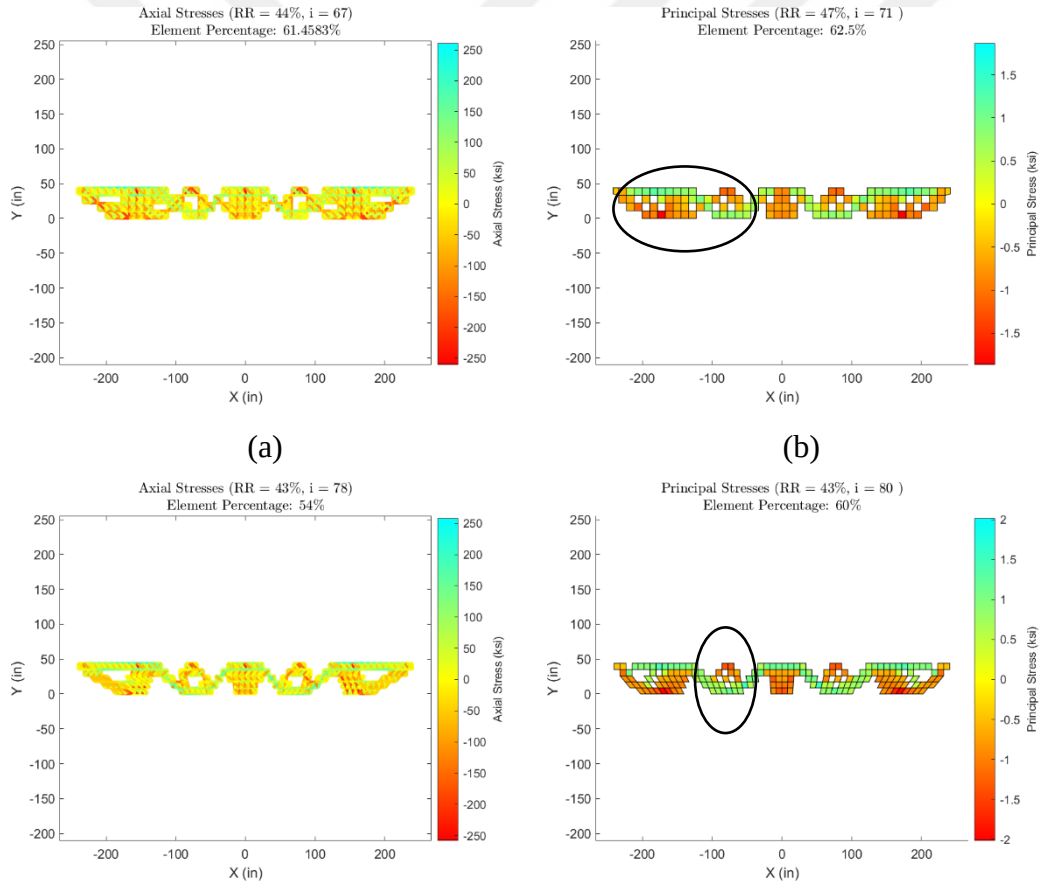


Figure 5.28: Pier Cap 5- Coarse Mesh Results: (a) Brick Model Mesh 1; (b) Membrane Model Mesh 1; (c) Brick Model Mesh 2; (d) Membrane Model Mesh 2

The overall results show that both models are vulnerable to slender reinforced concrete members. As the member becomes slenderer, the linear static analyses are disrupted due to singularities in the stiffness matrix, resulting in increasingly scattered and disorganized element distributions.

However, with reinforced concrete structures who are closer to deep beam geometry, both the Brick and Membrane Models demonstrated the ability to generate organized element distributions even in models with the coarsest mesh refinement. The results often resembles a truss system, which can be used as a guide in determining the locations of the strut and tie elements. Dimensioning these elements are still lay upon the engineer's judgement. Although the resulting element distributions are also successful in giving an insight into the dimensions of the trusses, since some compressive or tensile clusters retain larger numbers of elements than the others, pointing out that they are carrying larger forces.

The stress-based Evolutionary Structural Optimization can produce stable structures for both models without needing an element removal limit per iteration. Large numbers of elements can be deleted at the beginning of the iteration without disrupting the rest of the optimization process. On top of this, and the low number of elements of the initial coarse meshed structure, the optimization procedure was able to produce a critical element distribution within seconds.

Both elements exhibit dependence on mesh configuration. However, the Brick Model is more sensitive to variations in mesh configuration than Membrane Model. However, the results showed consistency across all levels of mesh refinement within the same mesh type. And with increasing mesh refinement, the load path became more detailed and precise. Also, as the mesh refinement increased, the Rejection Ratio of the critical element distribution that can describe the internal flow of forces within the structure decreased. However, it is sometimes observed that with increasing fine mesh refinement, the Brick Model produces an element distribution that is more continuum-like than a truss-like system. These results can be used to examine the internal force flow of the reinforced concrete structure in more detail.

The Brick Model is capable of converging toward a more refined load path than the Membrane Model, even in coarser mesh configurations. Results indicate that the load path achieved using coarse or medium meshes with the Brick Model often requires higher mesh refinement when using the Membrane Model to achieve comparable results. This highlights the efficiency and robustness of the Brick Model in producing clear and refined load paths with less computational effort. Moreover, since a brick element consist of six truss elements, the load path can be traced down to the one-dimensional truss elements within the Brick Model. This makes the load path in a Brick Model generally clearer, even in the coarsest mesh refinement. Another disadvantage of the Membrane Model for obtaining refined compressive and tensile element distribution and detecting the internal force flow is checkerboarding. Membrane Models are prone to produce checkerboarding patterns within the optimization, which causes scattered element distribution. No checkerboarding behavior is observed in the Brick Model.

CHAPTER 6

CONCLUSION

In this study, the primary goal is to evaluate and compare different approaches to optimization problems in various contexts, ultimately determining the analytical element most likely to yield successful results for the development of a strut-and-tie method within an arbitrary framework. Numerous comparisons were made using different algorithm configurations to reach this conclusion.

The tests are categorized based on their objectives. The Preliminary Tests aimed to narrow down the candidates for the final comparison phase. Among these candidates, the truss element—one of the proposed alternative analytical elements—was assessed alongside the brick and membrane elements. An initial set of tests was conducted to determine whether the Truss Model is a viable option. Based on the results, the truss Model was deemed unsuccessful and did not yield results warranting further investigation in this thesis.

Subsequent preliminary test focused on identifying the most effective configurations for the Brick and Membrane Models. Comparisons were made to evaluate whether element stresses or displacements served as the better elimination criterion. The stress-based structural optimization approach was deemed the most suitable method for further analyses aimed at comparing the overall performance of the brick and membrane elements.

The ultimate testing focused on a direct comparison between the axial stress-based Brick Model and the Von Mises stress-based Membrane Model. Various test models were employed to assess the elements' robustness, efficiency, versatility, dependency on mesh refinement, sensitivity to slenderness, and computational speed. The conclusions drawn from these analyses are presented in this final chapter.

- The stress-based Brick Model can successfully produce critical element distribution for pier cap reinforced structures and rectangular beams with or without openings.
- Brick Model results have refined element distribution that show the internal force flow within the reinforced concrete structure in detail and act as a guidance in determining a Strut-and-Tie Model.
- Further improvement is necessary as the Brick Model is not yet able to automatically give a direct Strut-and-Tie Model for a model. The best routine for the model is to run multiple analysis using different mesh refinements to examine the internal force flow and locate the strut and tie elements.

REFERENCES

Abdul-Razzaq, K., & Farhood, M. (2017). Design and behavior of reinforced concrete pile caps: A literature review. *Structural Engineering and Mechanics*.

Almeida, V. S., Simonetti, H. L., & Oliveira Neto, L. (2013). Comparative analysis of strut-and-tie models using Smooth Evolutionary Structural Optimization. *Engineering Structures, 56*, 1665–1675.

American Association of State Highway and Transportation Officials. (2023). *AASHTO LRFD bridge design specifications* (8th ed., Section 5: Concrete Structures). Washington, DC: AASHTO.

Atrek, E. (1989). Shape: a program for shape optimization of continuum structures. *Computer Aided Optimization Design of Structures: Applications*, C.A. Brebbia and S. Hernandez, eds., Computational Mechanics Publications, Southampton, U.K., 135-144.

Baniya, P., & Guner, S. (2019). Specialized strut-and-tie method for rapid strength prediction of bridge pier caps. *Engineering Structures, 198,* Article 109474.

Baumgartner, A. Harzheim, L., and Mattheck, C. (1992). SKO (soft kill option): the biological way to find an optimum structure topology. *Int. J. of Fatigue*, 14(6), 387-393.

Bedru, H. (2014). *Evaluation of hammerhead pier cap bridge design using strut and tie model* (Master's thesis). Addis Ababa University, Addis Ababa, Ethiopia.

Bendsøe, M. P., and Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Comp. Methods in Appl. Mech. and Engrg*., 71, 197-224.

Bendsøe, M. P., Díaz, A. R., Lipton, R., and Taylor, J. E. (1995). Optimal design of material properties and material distribution for multiple loading conditions. *Int. J. Numer. Methods in Engrg*., 35, 1449-1170.

Chu, D. N., Xie, Y. M., Hira, A., & Steven, G. P. (1996). Evolutionary structural optimization for problems with stiffness constraints. *Finite Elements in Analysis and Design, 21*(4), 239–251.

Collins, M. P., and Mitchell, D. (1980). Shear and torsion design of prestressed and non prestressed concrete beams. *PCI J.*, 25(5), 32-100.

Colorito, A. B., Wilson, K. E., Bayrak, O., & Russo, F. M. (2017). *Strut-and-tie modeling (STM) for concrete structures: Design examples* (Report No. FHWA-NHI-17-071). Federal Highway Administration, National Highway Institute.

Dawood, A., & Abdul-Razzaq, K. (2021). Shear friction and strut-and-tie modeling verification for pier caps. *Journal of Bridge Engineering, 26*(9), 1–13.

Dey, S., & Karthik, M. M. (2019). Modelling four-pile cap behaviour using three-dimensional compatibility strut-and-tie method. *Engineering Structures, 198*, Article 109499.

Díaz, A. R., and Bendsøe, M. P. (1992). Shape optimization of structures for multiple loading condition using a homogenization method. *Struct. Optimization*, 4, 17-22.

Díaz, A. R., and Kikuchi, N. (1992). Solution to shape and topology eigenvalue optimization problems using a homogenization method. *Int. J. Numer. Methods in Engrg*., 35, 1487 1502.

Garber, D. B., Gallardo, J. M., Huaco, G. D., Samaras, V. A., & Breen, J. E. (2014). Experimental evaluation of strut-and-tie model of indeterminate deep beam. *ACI Structural Journal*, 111(4).

Geuzaine, C., & Remacle, J.-F. (2024). *Gmsh reference manual: The documentation for Gmsh 4.13.1—A finite element mesh generator with built-in pre- and post-processing facilities.*

Guan, H. (2005). Strut-and-tie model of deep beams with web openings—An optimisation approach. *Structural Engineering and Mechanics, 19*(4), 361–376.

Hemp, W. S. (1973). Michell's structural continua. *In Optimum Structures* (Chap. 4). Clarendon Press.

Krog, L. A., and Olhoff, N. (1999). Optimum topology and reinforcement design of disk and plate structures with multiple stiffness and eigenfrequency objectives. *Comp. and Struct*., 72, 535-563.

Kupfer, H. (1964). Generalization of Mörsch's truss analogy using principle of minimum strain energy. *CEB Bulletin d'information*, No. 40, Comite Ero-International du Beton, No. 40, 44-57.

Kwak, H.-G., & Noh, S.-H. (2006). Determination of strut-and-tie models using evolutionary structural optimization. *Engineering Structures, 28*(10), 1440–1449.

Lampert, P., and Thurlimann, B. (1971). "Ultimate strength and design of reinforced concrete beams in torsion and bending." *Publication* No. 31-I, IABSE, Zürich, 107-131.

Leonhardt, F. (1965). Reducing the shear reinforcement in reinforced concrete beams and slabs. *Mag. of Concrete Res*., 17(53), 187.

Li, Q., Steven, G., & Xie, Y. (2000). Evolutionary structural optimization for stress minimization problems by discrete thickness design. *Computers & Structures, 78*(6), 769–780.

Liang, Q. Q., Xie, Y. M., & Steven, G. P. (1999). Optimal selection of topologies for the minimum-weight design of continuum structures with stress constraints. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 213*(8), 755–762.

Liang, Q. Q., Xie, Y. M., & Steven, G. P. (2000). Optimal topology selection of continuum structures with displacement constraints. *Computers & Structures, 77*(6), 635–644.

Liang, Q., Xie, Y., & Steven, G. (2000). Topology optimization of strut-and-tie models in reinforced concrete structures using an evolutionary procedure. *ACI Structural Journal, 97*(2), 322–332.

Liang, Q., Uy, B., & Steven, G. (2002). Performance-based optimization for strut-tie modeling of structural concrete. *Journal of Structural Engineering, 128*(6), 815–823.

Ma, Z. D., Kikuchi, N., and Cheng, H. C. (1995). Topological design for vibrating structures. *Comp. Methods in Appl. Mech. and Engrg*., 121, 259-280.

Marti, P. (1985). Basic tools of reinforced concrete beam design. *ACI Structural Journal, 82*(1), 46–56.

Mattheck, C. (1997). *Design in nature: Learning from trees.* Springer-Verlag. pp. 218–268.

Mitchell, D., & Collins, M. P. (2013). *Revision of strut-and-tie provisions in the AASHTO LRFD bridge design specifications*. National Cooperative Highway Research Program (NCHRP) Project 20-07, Task 306. Dorion, QC: MDC Research Inc.

Mlejnek, H. P., and Schirrmacher, R. (1993). An engineer's approach to optimal material distribution and shape finding. *Comp. Methods in Appl. Mech. and Engrg*., 106, 1-26.

Mörsch, E. (1909). *Concrete-steel construction*. McGraw-Hill, New York. (English translation by E.P. Goodrich).

Querin, O., Steven, G. P., & Xie, Y. M. (1997). Evolutionary structural optimisation by an additive algorithm. *Finite Elements in Analysis and Design*.

Querin, O., Steven, G., & Xie, Y. (1998). Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations, 15*(8), 1031–1048.

Ramirez, J. A. (1994). "Strut-and-tie design of pretensioned concrete members." *ACI Struct. J.*, 91(4), 572-578.

Ritter, W. (1899). The hennebique design method (Die bauweise henebique). *Schweizerische Bauzeitung (Zürich)*, 33(7), 59-61.

Rodriguez, J., and Seireg, A. (1985). Optimizing the shapes of structures via a rule-based computer program. *Computing in Mech. Engrg*., ASME, 40(1), 20-29.

Rozvany, G. I. N., Zhou, M., and Birker, T. (1992). Generalized shape optimization without homogenization. *Struct. Optimization*, 4, 250-252.

Sanders, D. H., and Breen, J. E. (1997). Post-tensioned anchorage zone with single straight concentric anchorages. *ACI Struct. J*., 94(2), 146-158.

Schlaich, J., Schäfer, K., & Jennewein, M. (1987). Toward a consistent design of structural concrete. *PCI Journal*, *32*(3), 74–150.

Souza, R., Kuchma, D., Park, J., & Bittencourt, T. (2009). Adaptable strut-and-tie model for design and verification of four-pile caps. *ACI Structural Journal*, *106*(2), 142–150.

Suzuki, K., and Kikuchi, N. (1991). A homogenization method for shape and topology optimization. *Comp. Methods in Appl. Mech. and Engrg*., 93, 291-318.

Tenek, L. H., and Hagiwara, I. (1993). Static and vibrational shape and topology optimization using homogenization and mathematical programming. *Comp. Methods in Appl. Mech. and Engrg*., 109, 143-154.

Vicksman, A. S., Williams, C. S., & Howarth, M. A. (2020). *Implementing the strut-and-tie method for the design of bridge components* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2020/01). West Lafayette, IN: Purdue University.

Vinayak, H. K., Senthil, S. N. A. K., & Kumar, P. T. V. (2016). Direct strut-and-tie model for reinforced concrete bridge pier cap. *Mathematical Modelling in Civil Engineering, 12*(1), 1–8.

Wang, X., Liu, Z., Wang, E., & Alsomiri, M. (2024). Structural performance of hammerhead pier under eccentric loads: Strut-and-tie modeling, finite element method, and full-scale experimental study. *Structure and Infrastructure Engineering, 20*(6), 802–818.

Williams, C., Deschenes, D., & Bayrak, O. (2014). A pragmatic approach to strut-and-tie modeling. *Proceedings of the Conference on Accelerated Bridge Construction.*

Xie, Y. M., & Steven, G. P. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures, 49*(5), 885–896.

Xie, Y. M., & Steven, G. P. (1997). *Evolutionary structural optimization.* Springer.

Yang, K. H. (2018). Meshing, element types, and element shape functions. In K.-H. Yang (Ed.), *Basic finite element method as applied to injury biomechanics* (pp. 51–109). Academic Press.

Yang, R. J., and Chuang, C. H. (1994). Optimal topology design using linear programming. *Comp. and Struct*., 52(2), 265-275.

Yang, X. Y., Xie, Y. M., Steven, G. P., & Querin, O. M. (1999). Bidirectional evolutionary method for stiffness optimization. *AIAA Journal, 37*(11), 1483–1488.

Zhao, C., Hornby, P., Steven, G. P., & Xie, Y. M. (1998). A generalized evolutionary method for numerical topology optimization of structures under static loading conditions. *Structural Optimization*, *15*(3–4), 251–260.

Zhong, J. T., Wang, L., Deng, P., & Zhou, M. (2017). A new evaluation procedure for the strut-and-tie models of the disturbed regions of reinforced concrete structures. *Engineering Structures, 148,* 660–672.

APPENDICES

A Pre-processing: Gmsh & Excel VBA Macro

A.1 The Geometry of the Reinforced Concrete Member- Gmsh

To create a mesh in Gmsh, certain properties must be defined. Point objects are created by specifying their spatial coordinates. Line objects are generated using the tags of their starting and ending points. These lines form a closed Curve Loop, which is then used to create a Plane Surface. This sequence of steps defines and visualizes the reinforced concrete member in the graphical user interface of Gmsh. Parametric equations are used to calculate the coordinates of points based on the dimensions of the structure, with three sets of parametric equations written for the three geometry types.

Once the geometry is defined, a Transfinite Surface is created using point tags, marking the area to be meshed. Transfinite Curves are then generated using the line tags, ensuring that opposite lines reference each other to achieve a structured mesh. The number of mesh points along each curve is specified during this process. Different sets of parameters are applied for the three types of geometries to meet their unique meshing requirements.

The dimensions of each shape are input into the .geo text file. These dimensions are then utilized in parametric equations to calculate the Point coordinates. Additionally, the width dimension is specified and stored in the z-coordinate of the Points. This z-coordinate value is later used to define the width of the membrane elements.

Based on their geometrical properties and differences, the example geometries are grouped into three categories: rectangular, trapezoidal hammerhead, and rectangular with openings. The following sections present the specific objects and parametric equations defined for each geometry type.

A.1.1 Rectangular Geometry

The parametric equations used to calculate the Point coordinates of rectangular shapes are presented in Figure A.1 below.

```
// Dimensions of the domain
depth = 2400;
length = 6000;
width = 250;

// Parametric coordinates of the points
// The origin is at the bottom center.
Point(1) = {-length/2, 0, width, lc};
Point(2) = {length/2, 0, width, lc};
Point(3) = {length/2, depth, width, lc};
Point(4) = {-length/2, depth, width, lc};
```

Figure A.1: Parametric Equations for Rectangular Shapes

The geometry of a rectangle shape is created in the Gmsh environment and is shown in Figure A.2.

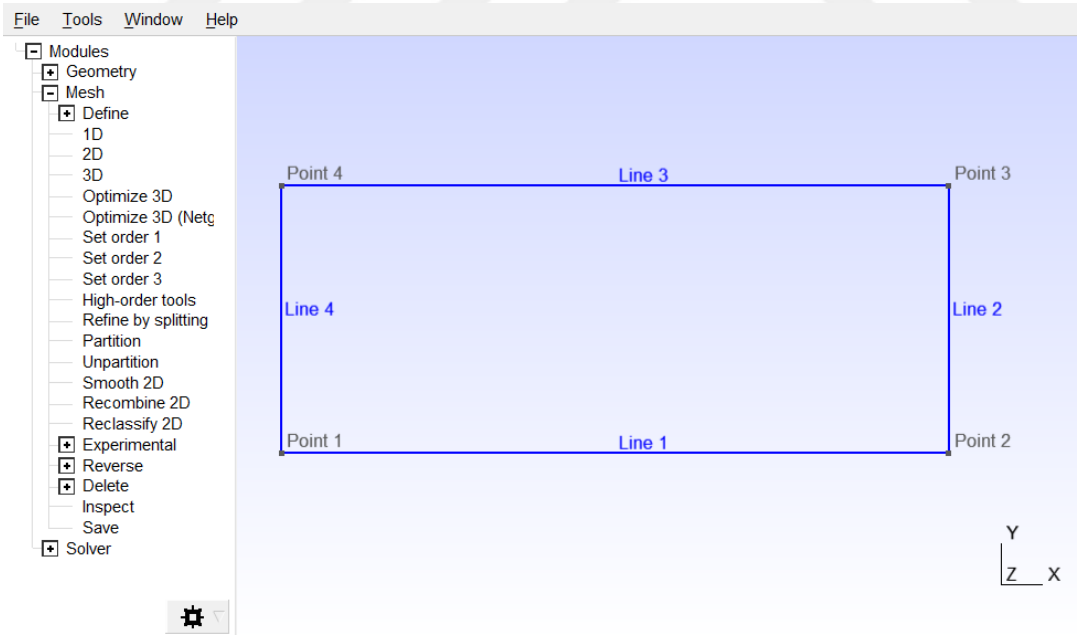


Figure A.2: Rectangle Shape

The opposite side pairings for the rectangular geometry are Line 1 & Line 3, and Line 2 & Line 4. Consequently, the Transfinite Curves are defined by referencing these pairings. A parameter called mesh_size is introduced, representing the dimension of a single quadrilateral element. The number of mesh points along a line is calculated as one plus the number of quadrilateral elements along that line. The equation used to define the number of mesh points for the selected opposite lines, based on the mesh_size parameter, is shown in Figure A.3.

```
// Mesh parameters
mesh_size = 500;

// Number of mesh points
Transfinite Curve {1, 3} = length/mesh_size +1 Using Progression 1;
Transfinite Curve {2, 4} = depth/mesh size +1 Using Progression 1;
```

Figure A.3: The Mesh Points Definition

The mesh of a rectangular shape is shown in Figure A.4.

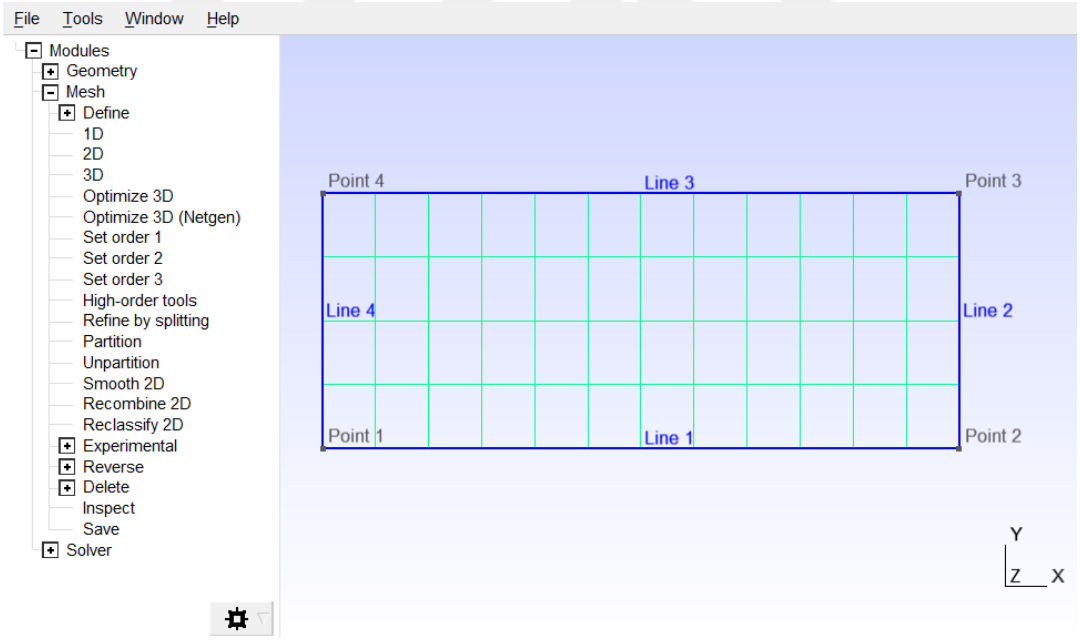


Figure A.4: Structured Mesh of Rectangular Shape

A.1.2 Rectangular Geometry with Openings

In addition to the depth and length of the outer rectangle, the depth of the openings and the coordinates of its center of gravity are entered. These parameters are used to calculate the Point coordinates of the inner rectangular opening shape. The parametric equations for the Point coordinates of the inner rectangular openings are shown in Figure A.5.

```
// Dimensions of the openings
hole = 150;
center1_x = -412.5;
center2_x = 412.5;
center_y = 375;

// Points of the openings
// The first opening
Point(5) = {center1_x-hole/2, center_y-hole/2, width, lc};
Point(6) = {center1_x+hole/2, center_y-hole/2, width, lc};
Point(7) = {center1_x+hole/2, center_y+hole/2, width, lc};
Point(8) = {center1_x-hole/2, center_y+hole/2, width, lc};
// The second opening
Point(9) = {center2_x-hole/2, center_y-hole/2, width, lc};
Point(10) = {center2_x+hole/2, center_y-hole/2, width, lc};
Point(11) = {center2_x+hole/2, center_y+hole/2, width, lc};
Point(12) = {center2_x-hole/2, center_y+hole/2, width, lc};
```

Figure A.5: Parametric Equations of the Inner Rectangular Openings

A rectangular shape with openings is created in the Gmsh environment and can be seen in Figure A.6.

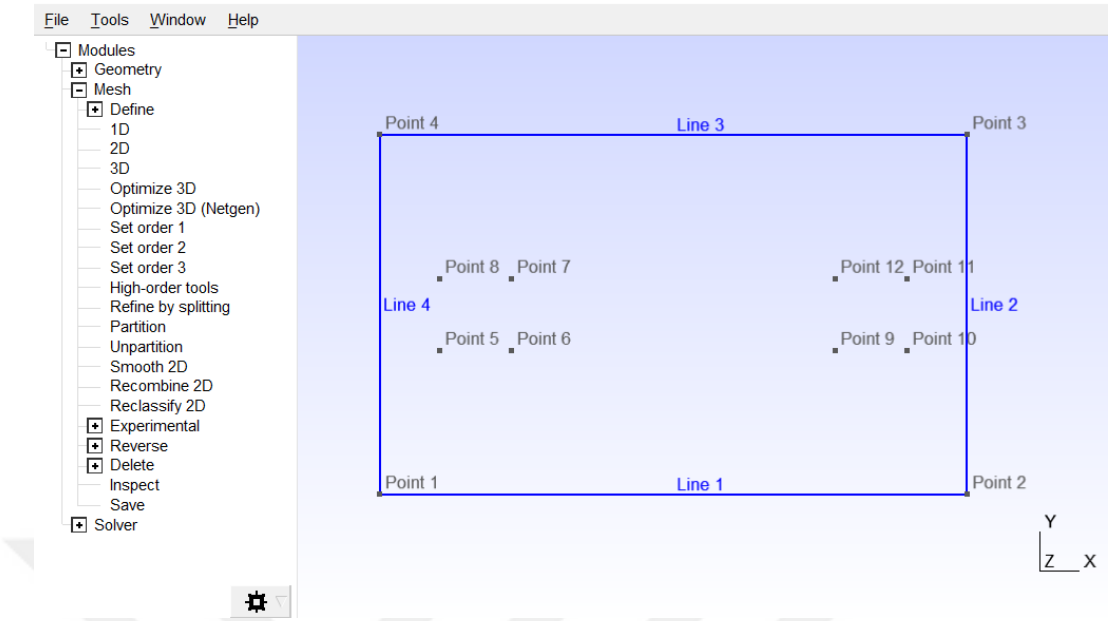


Figure A.6: Rectangular Shape with Openings

Line objects for openings are not generated to prevent Gmsh from meshing the inner rectangles. The opposite side pairings are Line 1 & 3, and Line 2 & 4. Therefore, the Transfinite Curves are defined by referencing these two pairings. The dimensions of the mesh elements must be uniquely calculated to discretize the overall reinforced concrete member into constant-sized quadrilateral elements.

The greatest common divisor of the distances from each side of the openings to each side of the outer rectangular shape is found. The distances from each side of the inner opening to the opposite sides of the outer rectangle are illustrated in Figure A.7. The dimensions are mm.

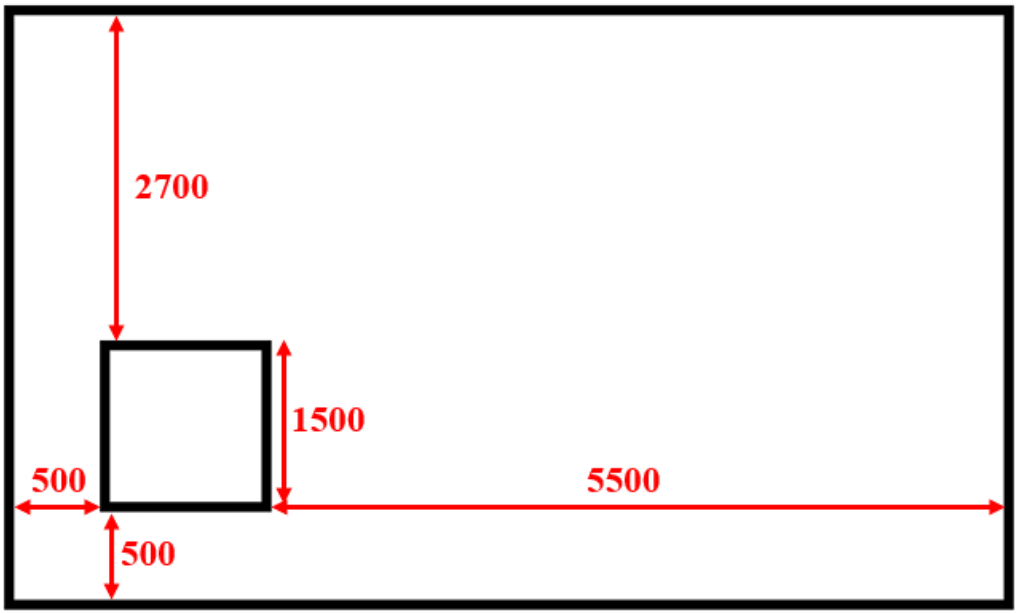


Figure A.7: The Dimensions for the Greatest Common Divisor Calculation

$$\gcd(500,1500,2700,550) = 100 \tag{3.1}$$

Equation 3.1 shows that to generate a structured mesh where all the mesh points coincide with both the inner and the outer rectangles, the coarsest possible mesh density of this domain would have to have quadrilateral elements of size 100. The mesh this rectangular geometry with openings can be seen in Figure A.8.

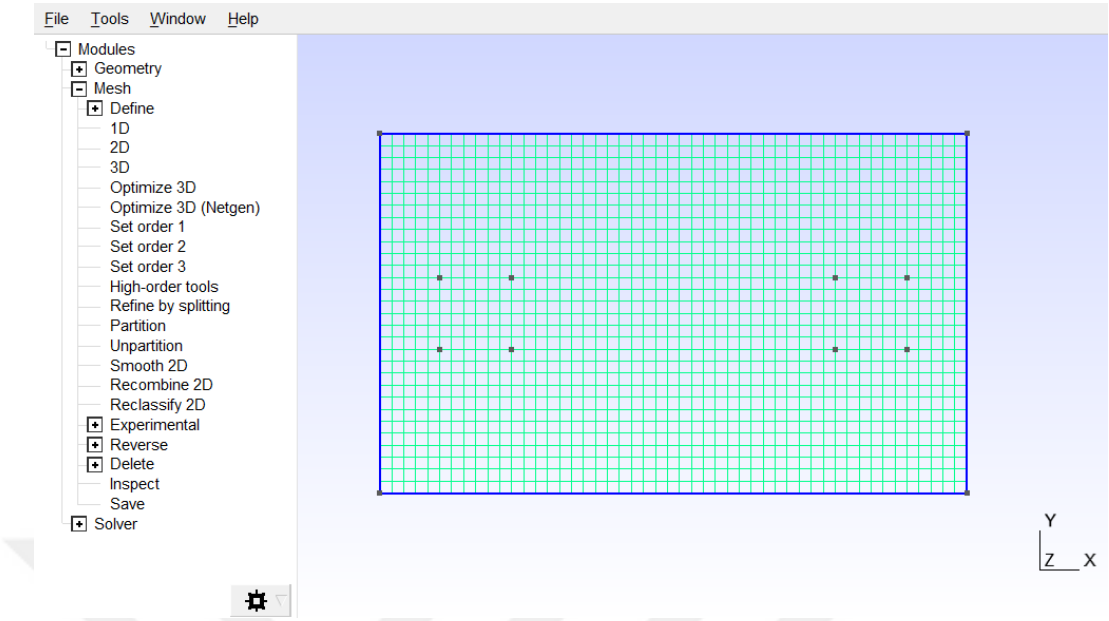


Figure A.8: Structured Mesh of Rectangular Geometry with Openings

The mesh information for both models is exported to a MATLAB script. The connectivity of the mesh elements within the inner opening rectangle should be removed before proceeding with the finite element analysis in MATLAB. The node labels of the quadrilateral elements can be visualized in Gmsh. The connectivity rows corresponding to the inner opening rectangle are removed from the MATLAB script. The nodes in question can be seen in Figure A.9. The connectivity is stored in msh.QUADS. This is also shown in Figure A.10.

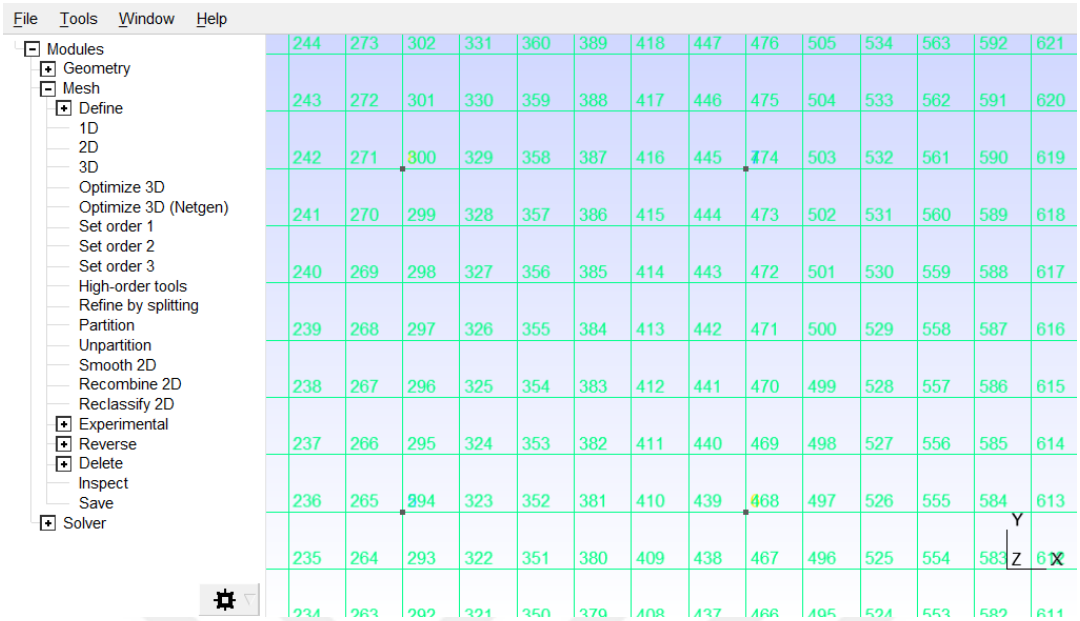


Figure A.9: The Nodes Inside the Opening Rectangle

```
msh.QUADS =[
 1 13 167 166 0
 166 167 168 165 0
 165 168 169 164 0
 164 169 170 163 0
 163 170 171 162 0
 162 171 172 161 0
 161 172 173 160 0
 160 173 174 159 0
 159 174 175 158 0
 158 175 176 157 0
```

Figure A.10: The Connectivity Matrix in Gmsh MATLAB Export File

A.1.3 Trapezoidal Hammerhead Geometry

The dimensions of the trapezoidal hammerhead shape are illustrated in Figure A.11, and the parametric equations for the Point coordinates are shown in Figure A.12.

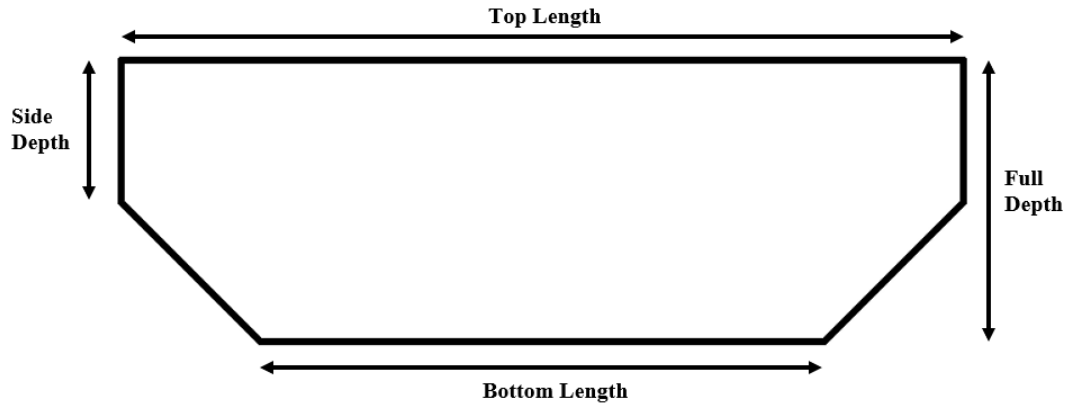


Figure A.11: The Dimensions of the Hammerhead Shape

```
// Dimensions of the domain
top=810;
bottom=250;
depth=410;
side=210;
thickness=125;

// Parametric coordinates of the points
// The origin is at the bottom center.
Point(1) = {-bottom/2, 0, thickness, lc};
Point(2) = {bottom/2, 0, thickness, lc};
Point(3) = {top/2, depth - side, thickness, lc};
Point(4) = {top/2, depth, thickness, lc};
Point(5) = {bottom/2, depth, thickness, lc};
Point(6) = {-bottom/2, depth, thickness, lc};
Point(7) = {-top/2, depth, thickness, lc};
Point(8) = {-top/2, depth - side, thickness, lc};
```

Figure A.12: Parametric Equations of the Hammerhead Shape

Gmsh requires opposite sides to generate a structured mesh within the domain. To accommodate this requirement, additional Lines are defined for two distinct mesh strategies, referred to as "m1" and "m2" for brevity.

A.1.3.1 Hammerhead mesh 1 (m1)

For the first type of mesh, two Lines are drawn from the bottom corners to the top of the hammerhead shape. These Lines are parallel to the sides of the hammerhead shape. The geometry of a trapezoidal hammerhead shape using the m1 mesh strategy is created in the Gmsh environment and is shown in Figure A.13.

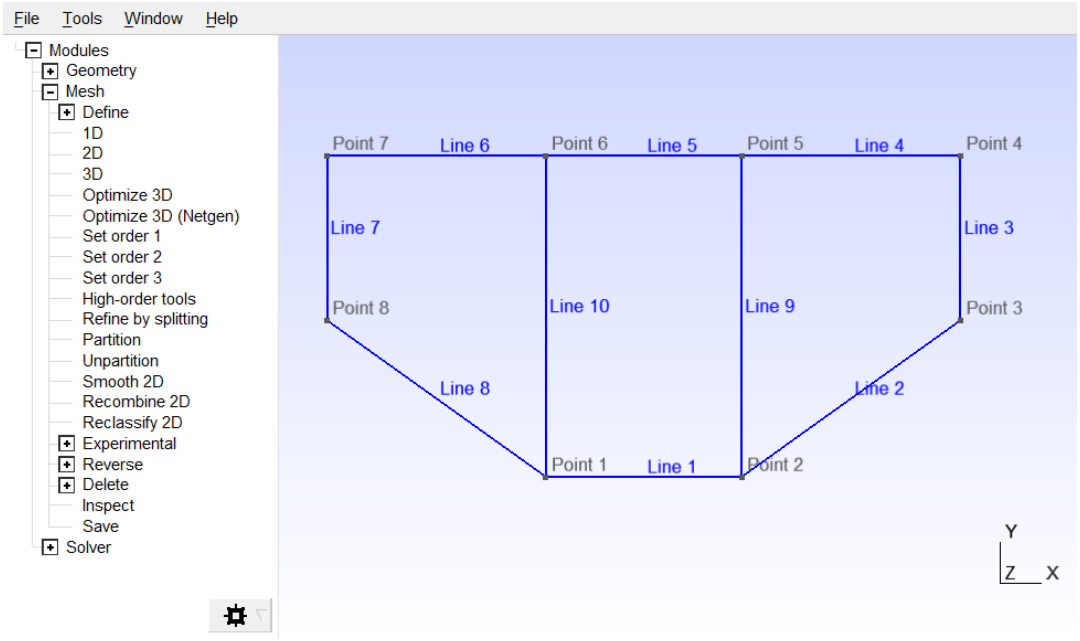


Figure A.13: Trapezoidal Hammerhead (m1)

The discretized trapezoidal hammerhead shape consists of skewed quadrilateral elements due to the domain's geometry. As a result, maintaining constant mesh element dimensions is not feasible for this type of geometry. Instead, the objective is to minimize differences among the quadrilateral mesh elements to achieve more accurate results in the finite element analyses during the post-processing phase.

To achieve this, various mesh parameters are defined and used in equations to calculate the mesh points along the opposite sides of the hammerhead shapes. In mesh 1, three groups of opposite sides are identified: the first group includes Lines 7, 10, 9, and 3; the second group consists of Lines 6, 8, 4, and 2; and the third group

comprises Lines 1 and 5. The mesh parameters and equations for determining the mesh points are presented in Figure A.14.

```
// Mesh parameters
a=(top-bottom)/2;
b=depth-side;
side_width_1=Sqrt(a^2+b^2);
side_width_2=a;
side_width=(side_width_1+side_width_2)/2; // Average width of the surfaces at the sides
side_depth=(depth+side)/2; // Average depth of the surfaces at the sides

// Number of mesh points
m = 0.5; // Mesh factor
Transfinite Curve {7, 10, 9, 3} = 10*m +1 Using Progression 1;
Transfinite Curve {6, 8, 4, 2} = 10*m*side_width/side_depth +1 Using Progression 1;
Transfinite Curve {1, 5} = 10*m*bottom/depth +1 Using Progression 1;
```

Figure A.14: The Mesh Parameters and Mesh Point Equations for Mesh 1

Since the mesh dimension is not constant, the mesh density is controlled by a parameter called the mesh factor (*m*). The mesh of a trapezoidal hammerhead shape according to the first meshing approach (m1) is shown in Figure A.15.

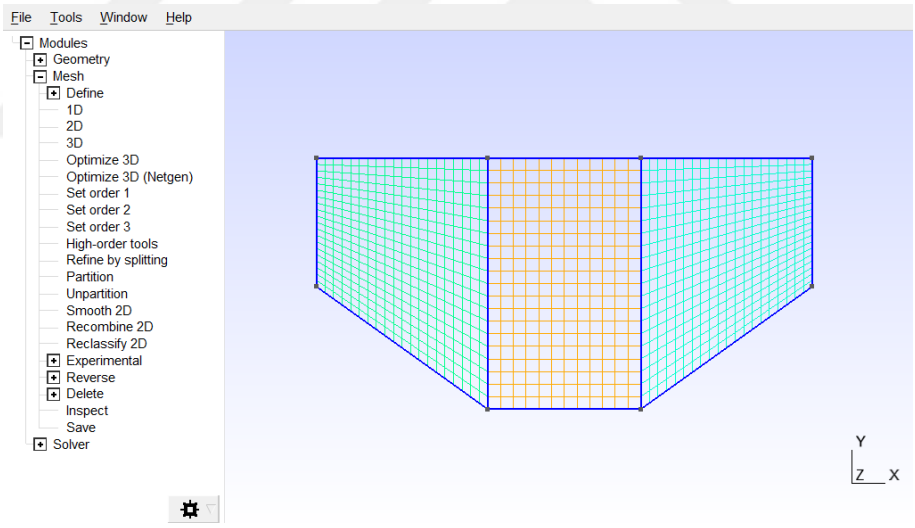


Figure A.15: Trapezoidal Hammerhead Mesh 1

This approach can be preferred for the mesh of trapezoidal hammerhead shapes that have closer dimensions of depth and length.

A.1.3.2 Hammerhead mesh 2 (m2)

For the second type of mesh, a Line is drawn from one side corner to the other, parallel to the top and bottom edges of the trapezoidal hammerhead shape. A trapezoidal hammerhead shape is created in the Gmsh environment and can be seen in Figure A.16.

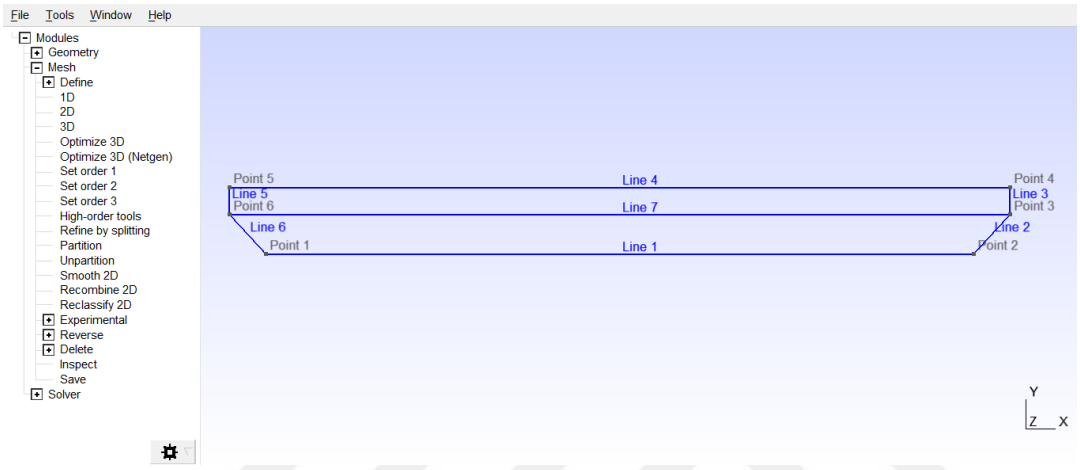


Figure A.16: Trapezoidal Hammerhead Shape in Gmsh

In mesh 2, there are three groups of opposite sides. The first group consists of Lines 1, 7, and 4; the second group consists of Lines 6 and 2; and the third group consists of Lines 5 and 3. The mesh parameters and the mesh point equations can be seen in Figure A.17.

```
// Mesh parameters
a=(top-bottom)/2;
b=depth-side;
lower_side_1=Sqrt(a^2+b^2);
lower_side_2=b;
lower_depth=(lower_side_1+lower_side_2)/2; // Average depth of the lower surface
lower_width=(top+bottom)/2; // Average width of the lower surface

// Number of mesh points
m = 6; // Mesh factor
Transfinite Curve {1, 7, 4} = 10*m +1 Using Progression 1;
Transfinite Curve {6, 2} = 10*m*lower_depth/lower_width +1 Using Progression 1;
Transfinite Curve {5, 3} = 10*m*side/top +1  Using Progression 1;
```

Figure A.17: The Mesh Parameters and Mesh Point Equations for Mesh 2

The mesh of a trapezoidal hammerhead shape according to the second meshing approach (m2) can be seen in Figure A.18.

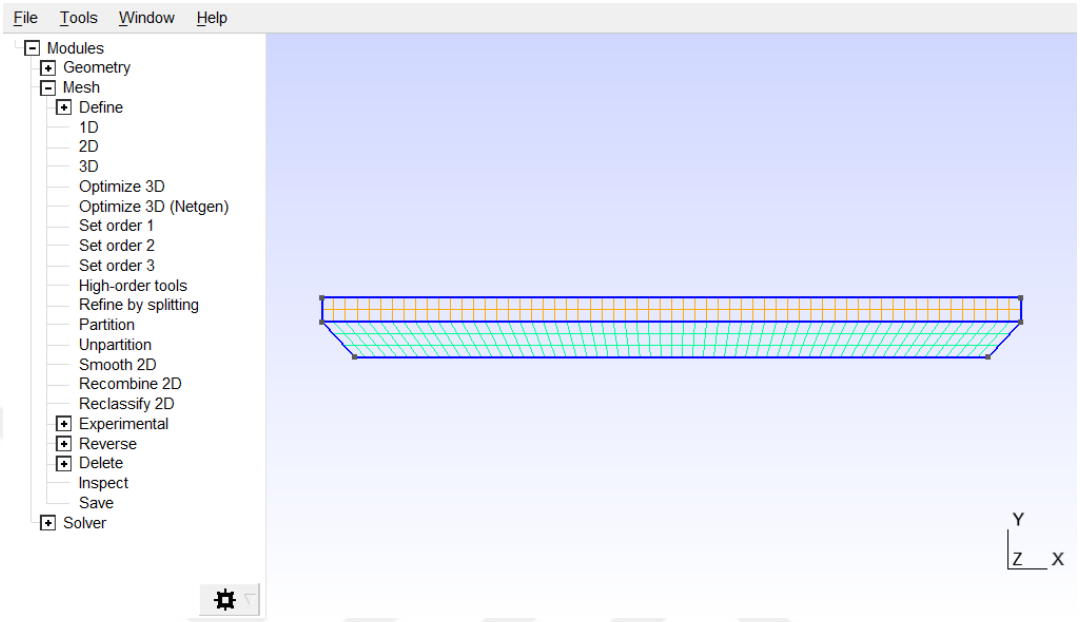


Figure A.18: Trapezoidal Hammerhead Mesh 1

This approach can be preferred for the mesh of trapezoidal hammerhead shapes that have relatively large length over depth ratio.

A.2 The Boundary Conditions and Material Properties- Excel VBA

Three separate MATLAB Projects are created to perform finite element analysis and evolutionary structural optimization using the three different analytical elements. The Excel VBA macro includes three modules, each corresponding to one analytical element. The selected module generates a MATLAB script named *inputScript.m*, containing the input properties for the selected example model. The script file is saved to the appropriate MATLAB project associated with the chosen element. A portion of the *inputData* Excel sheet layout is shown in Figure A.19.

	A	R	S	T	U	V	W
1		gmsh		materialProperties			
2		% Gmsh Matlab file name		s of elasticity of co		% Poisson's ratio	
3	rectangle_example_1	gmshMatlabFile	"rectangle_example_1_coarse.m"	Ec	28567	poissonRatio	0.15
4	rectangle_example_1	gmshMatlabFile	"rectangle_example_1_medium.m"	Ec	28567	poissonRatio	0.15
5	rectangle_example_1	gmshMatlabFile	"rectangle_example_1_fine.m"	Ec	28567	poissonRatio	0.15
6	rectangle_example_1	gmshMatlabFile	"rectangle_example_1_finest.m"	Ec	28567	poissonRatio	0.15
7	rectangle_example_2a	gmshMatlabFile	"rectangle_example_2a_coarse.m"	Ec	28567	poissonRatio	0.15
8	rectangle_example_2a	gmshMatlabFile	"rectangle_example_2a_medium.m"	Ec	28567	poissonRatio	0.15
9	rectangle_example_2a	gmshMatlabFile	"rectangle_example_2a_fine.m"	Ec	28567	poissonRatio	0.15
10	rectangle_example_2b	gmshMatlabFile	"rectangle_example_2b_coarse.m"	Ec	28567	poissonRatio	0.15
11	rectangle_example_2b	gmshMatlabFile	"rectangle_example_2b_medium.m"	Ec	28567	poissonRatio	0.15
12	rectangle_example_2b	gmshMatlabFile	"rectangle_example_2b_fine.m"	Ec	28567	poissonRatio	0.15
13	rectangle_example_2c	gmshMatlabFile	"rectangle_example_2c_coarse.m"	Ec	28567	poissonRatio	0.15
14	rectangle_example_2c	gmshMatlabFile	"rectangle_example_2c_medium.m"	Ec	28567	poissonRatio	0.15
15	rectangle_example_2c	gmshMatlabFile	"rectangle_example_2c_fine.m"	Ec	28567	poissonRatio	0.15
16	rectangle_example_2d	gmshMatlabFile	"rectangle_example_2d_coarse.m"	Ec	28567	poissonRatio	0.15
17	rectangle_example_2d	gmshMatlabFile	"rectangle_example_2d_medium.m"	Ec	28567	poissonRatio	0.15
18	rectangle_example_2d	gmshMatlabFile	"rectangle_example_2d_fine.m"	Ec	28567	poissonRatio	0.15
19	rectangle_example_3	gmshMatlabFile	"rectangle_example_3.m"	Ec	30088	poissonRatio	0.15
20	rectangle_example_4	gmshMatlabFile	"rectangle_example_4.m"	Ec	20820	poissonRatio	0.15

Membrane Brick Truss

Figure A.19: The InputData Excel Sheet Layout

The orange cells in the *inputData* Excel sheet indicate the start of a new example, while the rows below correspond to the different mesh types for the same example. When the Excel VBA macro is executed, it prompts the user to specify a row number. Based on the selected row, the macro generates the *inputScript.m* MATLAB file and saves it to the appropriate MATLAB project associated with the chosen analytical finite element.

The *inputScript.m* file contains a struct named inputData. The first heading in the merged cells corresponds to the field names of the inputData struct. The second heading provides explanatory comments for these fields. The first cell under the headings contains the names of the subfields. The yellow cells in the Excel sheet are used for input. These parameters are entered directly into the yellow cells.

A.2.1 Units

There are five groups of input parameters: units, mesh script, material properties, support locations, and external load location & magnitude. The units group includes the fields inputUnits, resultingUnits, and outputUnits. Each of these unit fields

contains the subfields forceUnit, lengthUnit, and materialUnit. These fields are used within the MATLAB projects to perform necessary unit conversions and ensure that results are displayed correctly. The relevant section of the *inputData.m* script can be seen in Figure A.20.

inputScript.m

```
%% rectangle_example_4

inputData = struct();

% Force unit
inputData.inputUnits.forceUnit = 'kN';
% Length unit
inputData.inputUnits.lengthUnit = 'mm';
% Material unit
inputData.inputUnits.materialUnit = 'MPa';

% Displacement unit
inputData.resultingUnits.displacementUnit = 'm';
% Stress unit
inputData.resultingUnits.stressUnit = 'MPa';

% Force unit
inputData.outputUnits.forceUnit = 'kN';
% Length unit
inputData.outputUnits.lengthUnit = 'mm';
% Material unit
inputData.outputUnits.materialUnit = 'kPa';
```

Figure A.20: The Units

A.2.2 Gmsh

All mesh scripts are stored within the MATLAB project directories. When the node connectivity and node coordinates matrices are required for finite element analysis, the corresponding mesh script must be executed. This information is stored in the Gmsh field of the inputData struct. The relevant section of the *inputData.m* script is shown in Figure A.21.

```
% Gmsh Matlab file name
inputData.gmsh.gmshMatlabFile = "rectangle_example_4.m";
```

Figure A.21: The Gmsh MATLAB File Name of Rectangle Example 4

A.2.3 Material Properties

The modulus of elasticity and Poisson's ratio of the concrete are stored in inputData.materialProperties.Ec and inputData.materialProperties.poissonRatio, respectively. The modulus of elasticity is specified in accordance with inputUnits.materialUnit. The relevant section of the *inputData.m* script is shown in Figure A.22.

```
% Modulus of elasticity of concrete (E)
inputData.materialProperties.Ec = 28567;
% Poisson's ratio
inputData.materialProperties.poissonRatio = 0.15;
```

Figure A.22: The Material Properties

A.2.4 Support Locations

All example models are created in Gmsh such that the midpoint of their bottom sides is positioned at the origin of the global coordinate system. This is done so for the simplification of generating the model geometry. All example models are supported at the bottom. Hence, the y-coordinates of the supports are always zero, leaving just the x-coordinates to be specified.

The x-coordinate ranges for the support regions are entered into an array. For multiple support regions, each x-coordinate range is entered in a separate row. These coordinates are specified with respect to inputUnits.lengthUnit. The relevant section of the *inputData* Excel sheet is shown in Figure A.23.

	A	X	Y
1		supports	
2		% The x-coordinate ranges for the support regions	
3	rectangle_example_1	supportRegions	[-3000,-2400;2400,3000]
4	rectangle_example_1	supportRegions	[-3000,-2400;2400,3000]
5	rectangle_example_1	supportRegions	[-3000,-2400;2400,3000]
6	rectangle_example_1	supportRegions	[-3000,-2400;2400,3000]
7	rectangle_example_2a	supportRegions	[-1000,-750;750,1000]
8	rectangle_example_2a	supportRegions	[-1000,-750;750,1000]
9	rectangle_example_2a	supportRegions	[-1000,-750;750,1000]
10	rectangle_example_2b	supportRegions	[-1500,-1250;1250,1500]
11	rectangle_example_2b	supportRegions	[-1500,-1250;1250,1500]
12	rectangle_example_2b	supportRegions	[-1500,-1250;1250,1500]
13	rectangle_example_2c	supportRegions	[-2000,-1750;1750,2000]
14	rectangle_example_2c	supportRegions	[-2000,-1750;1750,2000]
15	rectangle_example_2c	supportRegions	[-2000,-1750;1750,2000]
16	rectangle_example_2d	supportRegions	[-2500,-2250;2250,2500]
17	rectangle_example_2d	supportRegions	[-2500,-2250;2250,2500]
18	rectangle_example_2d	supportRegions	[-2500,-2250;2250,2500]
19	rectangle_example_3	supportRegions	[-612.5,-487.5;487.5,612.5]
20	rectangle_example_4	supportRegions	[-3750,-3250;3250,3750]

Figure A.23: The Locations of the Supports

A.2.5 External Load Locations & Magnitudes

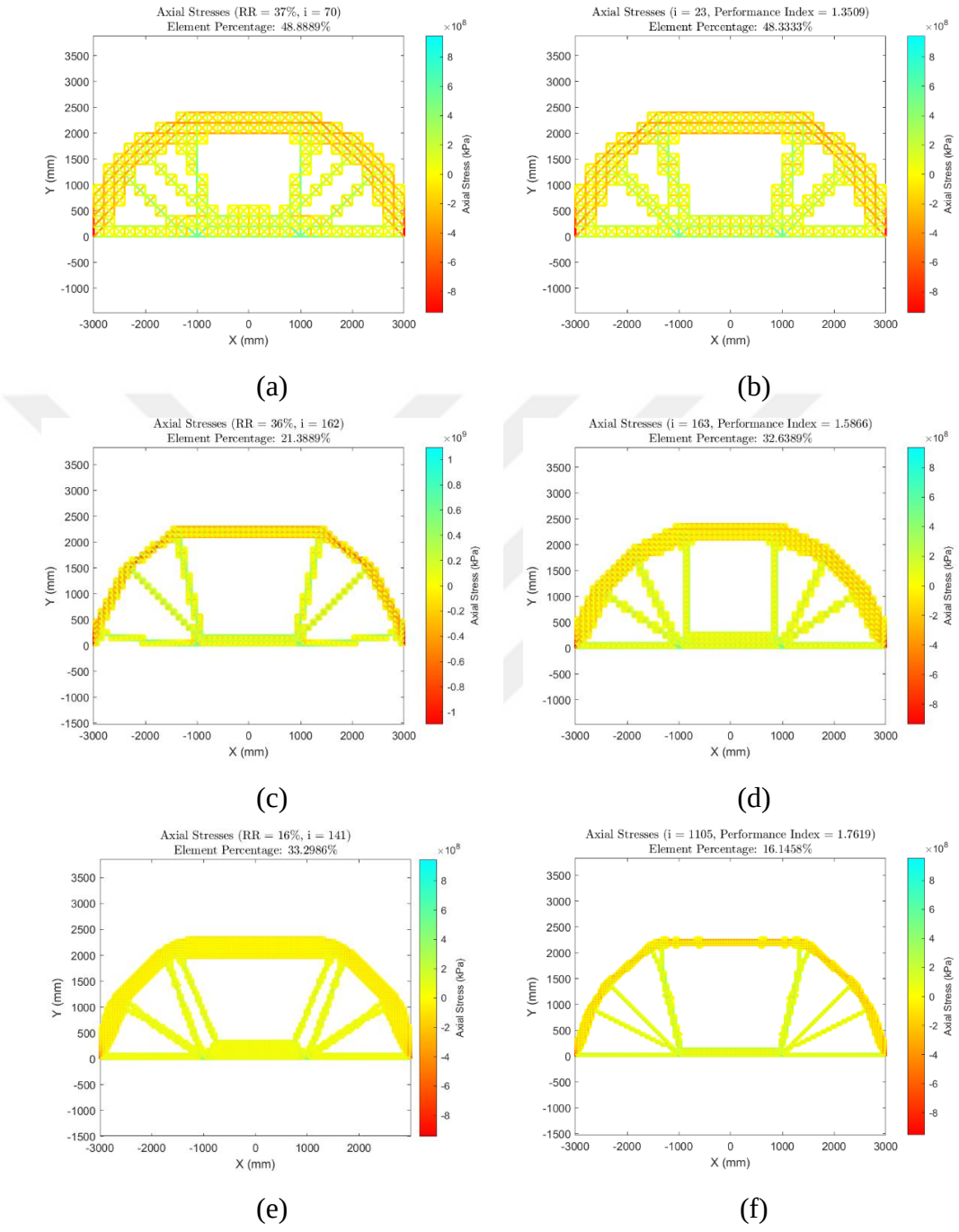
There are three fields used to describe the applied loads on the structure. The loadRegions field stores the x-coordinate regions of the load bearing plates. The loadValues field stores the magnitudes of the external loads. The height field stores the y-coordinate of the external loads.

The units of loads.loadRegions and loads.height are specified with respect to inputUnits.lengthUnit, while the units of loads.loadValues are defined with respect to inputUnits.forceUnit. The x-coordinate ranges and magnitudes of the external loads are stored in an array. For multiple loads, each x-coordinate range and load magnitude are entered in a separate row. The relevant section of the *inputData* Excel sheet is shown in Figure A.24.

	A	Z	AA	AB	AC	AD	AE
1		loads					
2		e x-coordinate ranges for the load reg		value where t		% The load value	
3	rectangle_example_1	loadRegions	[-1500,-500;500,1500]	height	0	loadValues	[-1200;-1200]
4	rectangle_example_1	loadRegions	[-1500,-500;500,1500]	height	0	loadValues	[-1200;-1200]
5	rectangle_example_1	loadRegions	[-1500,-500;500,1500]	height	0	loadValues	[-1200;-1200]
6	rectangle_example_1	loadRegions	[-1500,-500;500,1500]	height	0	loadValues	[-1200;-1200]
7	rectangle_example_2a	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
8	rectangle_example_2a	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
9	rectangle_example_2a	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
10	rectangle_example_2b	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
11	rectangle_example_2b	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
12	rectangle_example_2b	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
13	rectangle_example_2c	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
14	rectangle_example_2c	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
15	rectangle_example_2c	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
16	rectangle_example_2d	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
17	rectangle_example_2d	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
18	rectangle_example_2d	loadRegions	[-250,250]	height	1000	loadValues	[-1200]
19	rectangle_example_3	loadRegions	[-300,-175;175,300]	height	750	loadValues	[-140;-140]
20	rectangle_example_4	loadRegions	[700,1200]	height	4700	loadValues	[-3000]

Figure A.24: The Location and Magnitude of the External Loads

B Brick Model

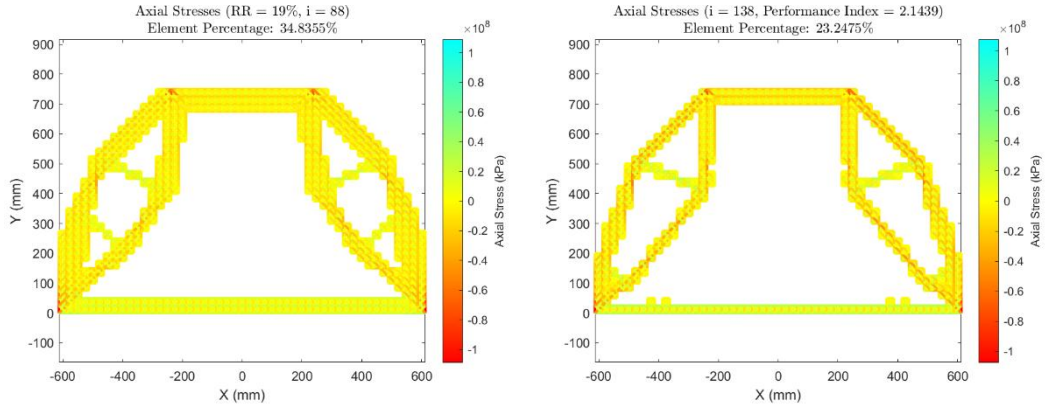


(a) (b)

(c) (d)

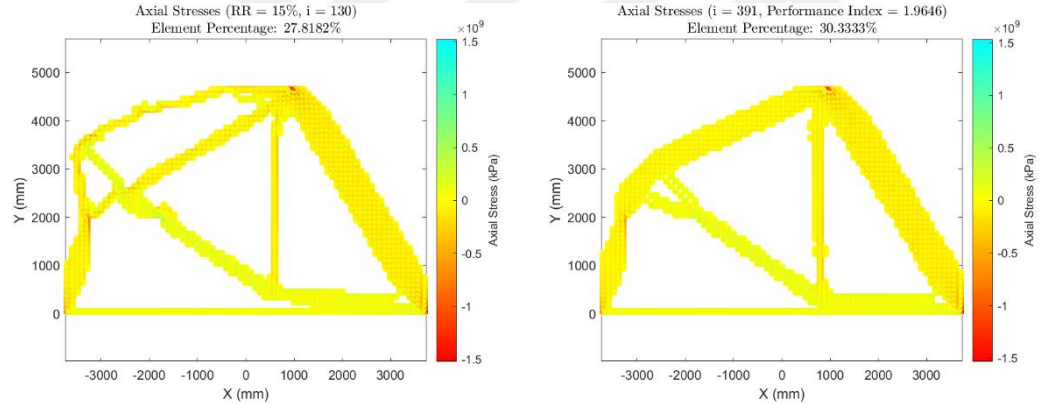
(e) (f)

Figure B.25. Rectangle Example 1: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh



(a) (b)

Figure B.26. Rectangle Example 2: (a) Stress-based; (b) Displacement-based



(a) (b)

Figure B.27. Rectangle Example 3: (a) Stress-based; (b) Displacement-based

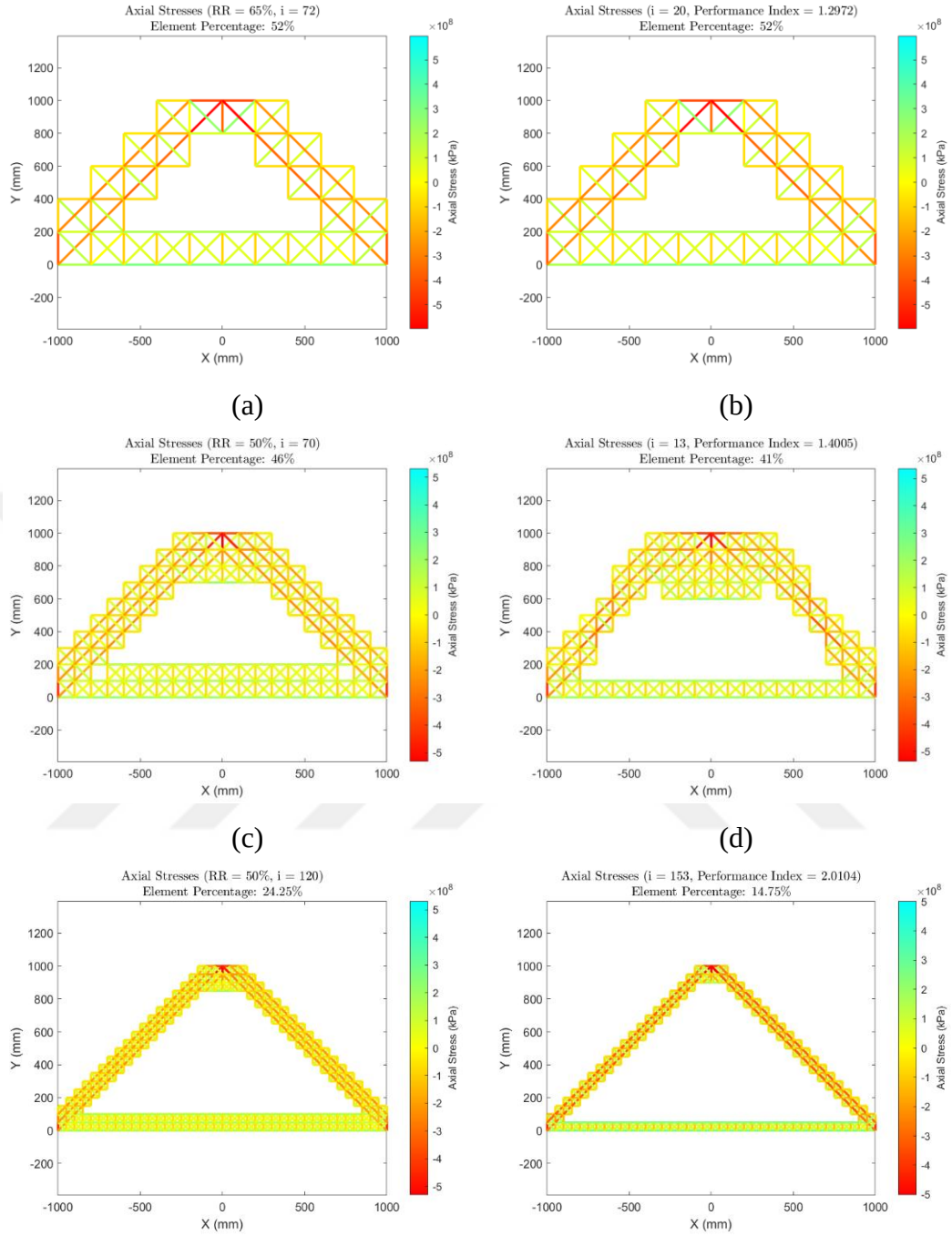
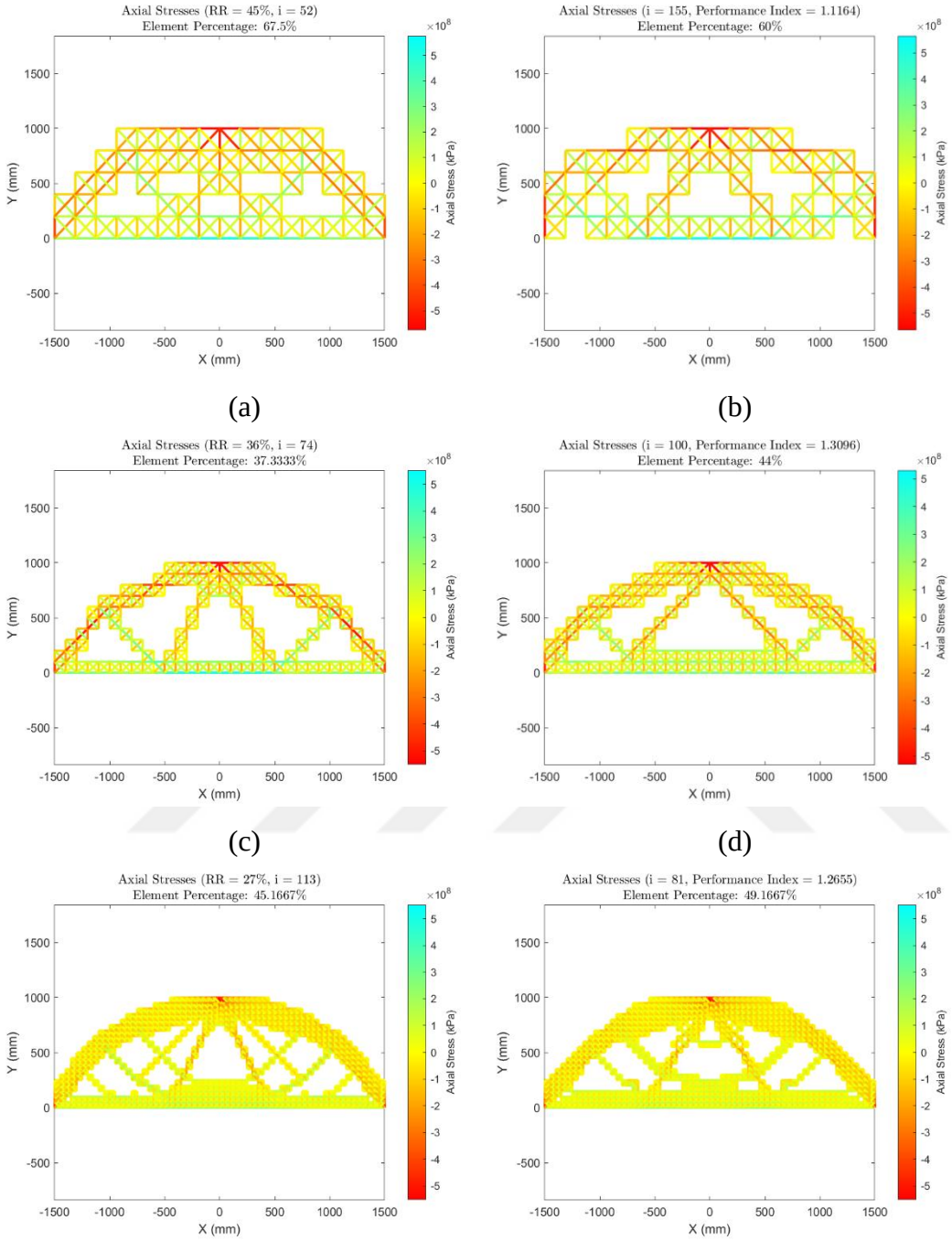


Figure B.28. Rectangle Example 4a: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh



(a) (b)

(c) (d)

(e) (f)

Figure B.29. Rectangle Example 4b: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh

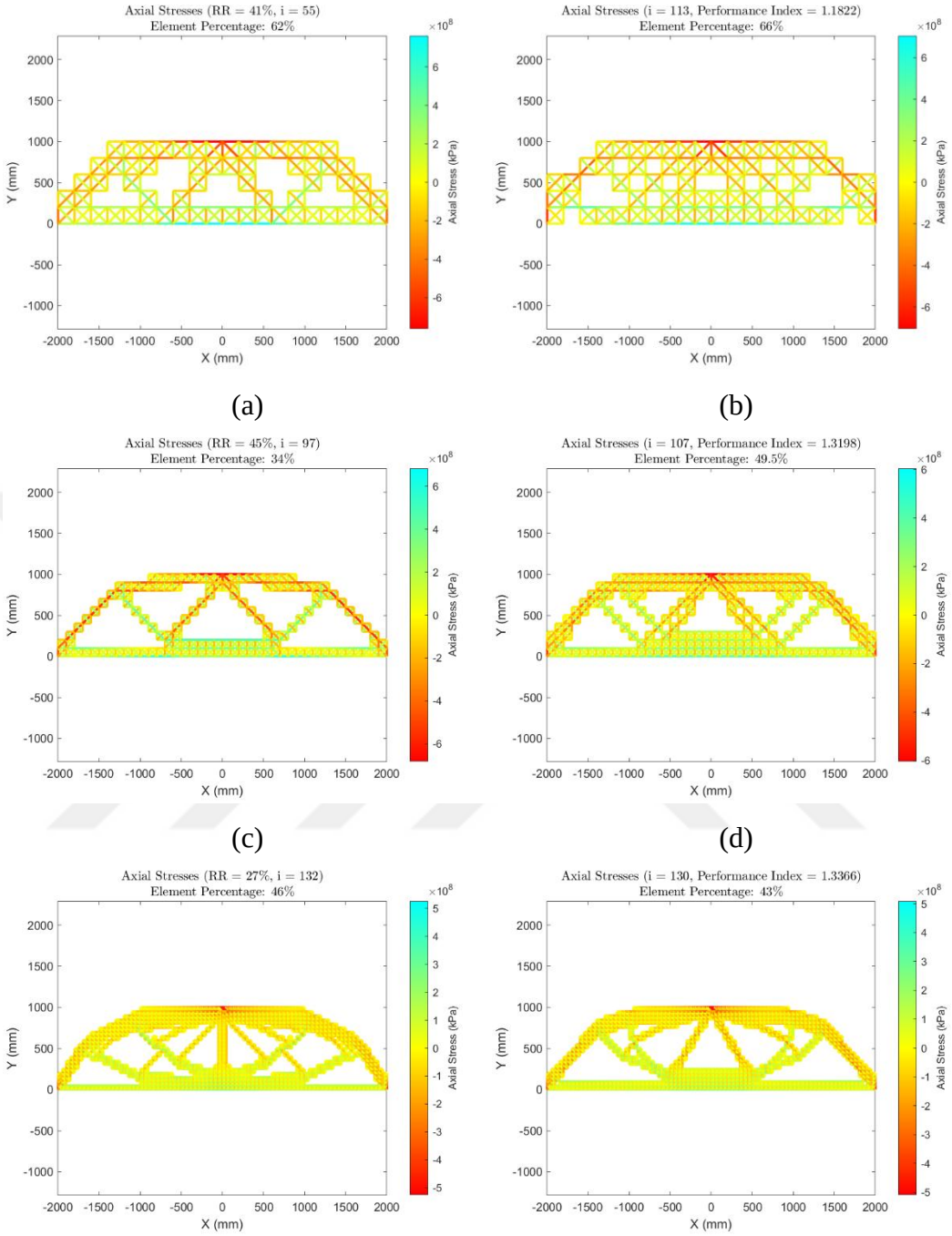


Figure B.30. Rectangle Example 4c: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh

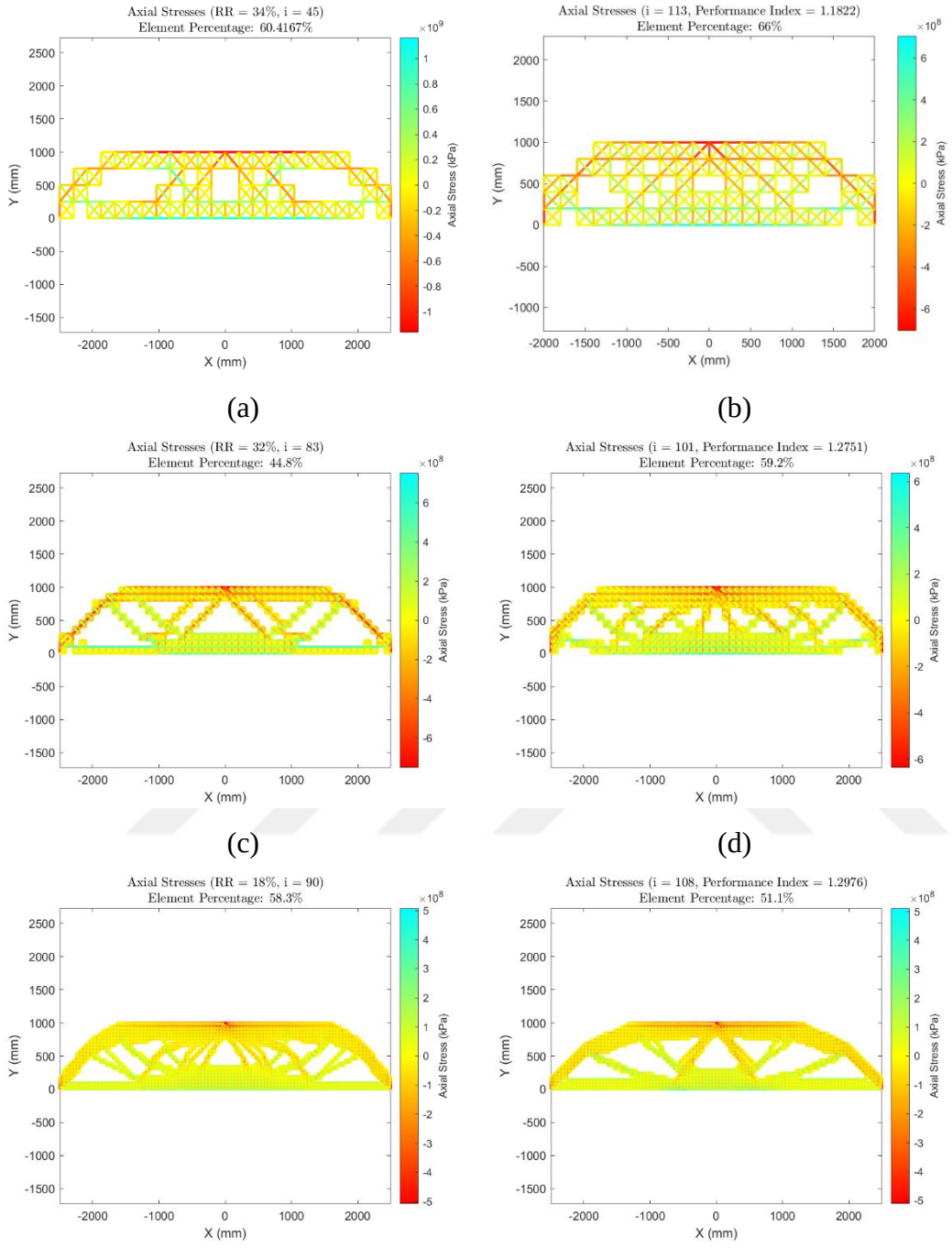
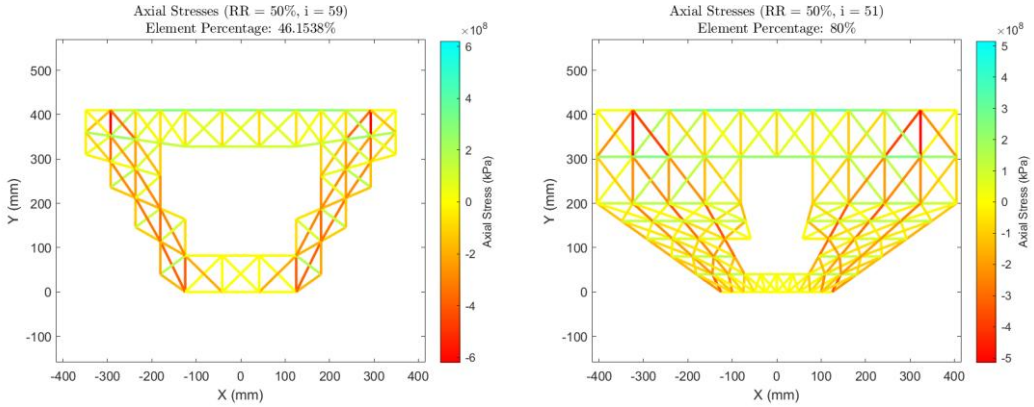
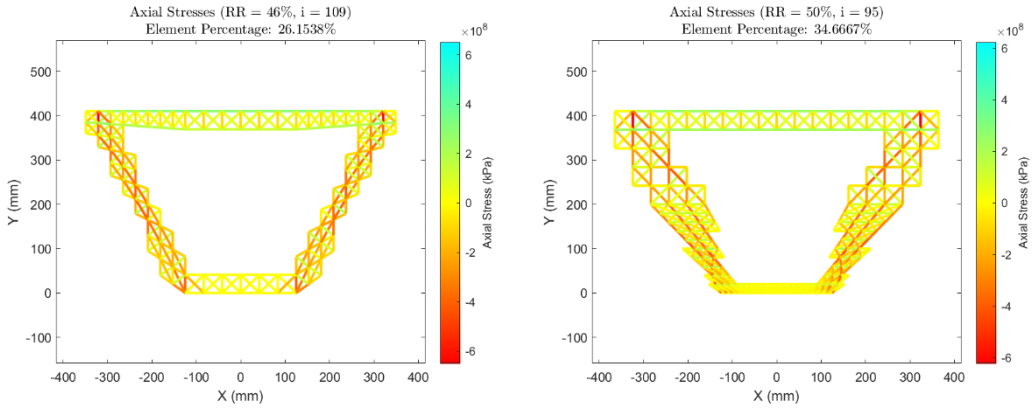


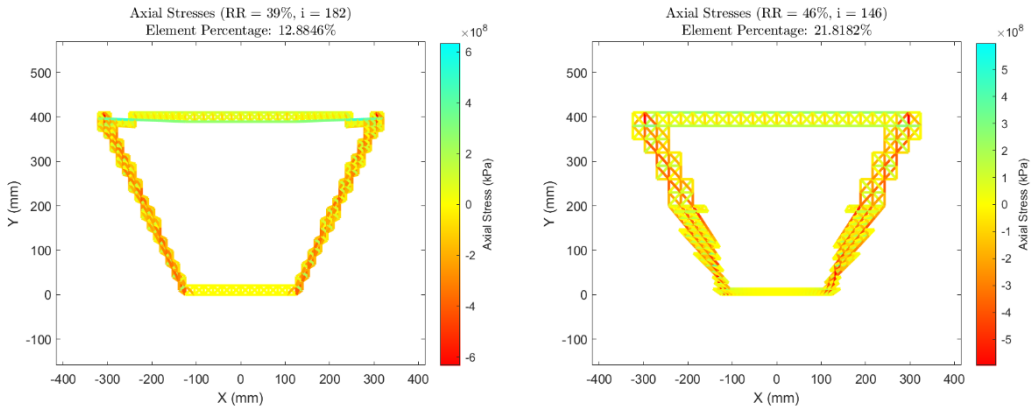
Figure B.31. Rectangle Example 4d: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh



(a) (b)



(c) (d)



(e) (f)

Figure B.32. Pier Cap 1: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

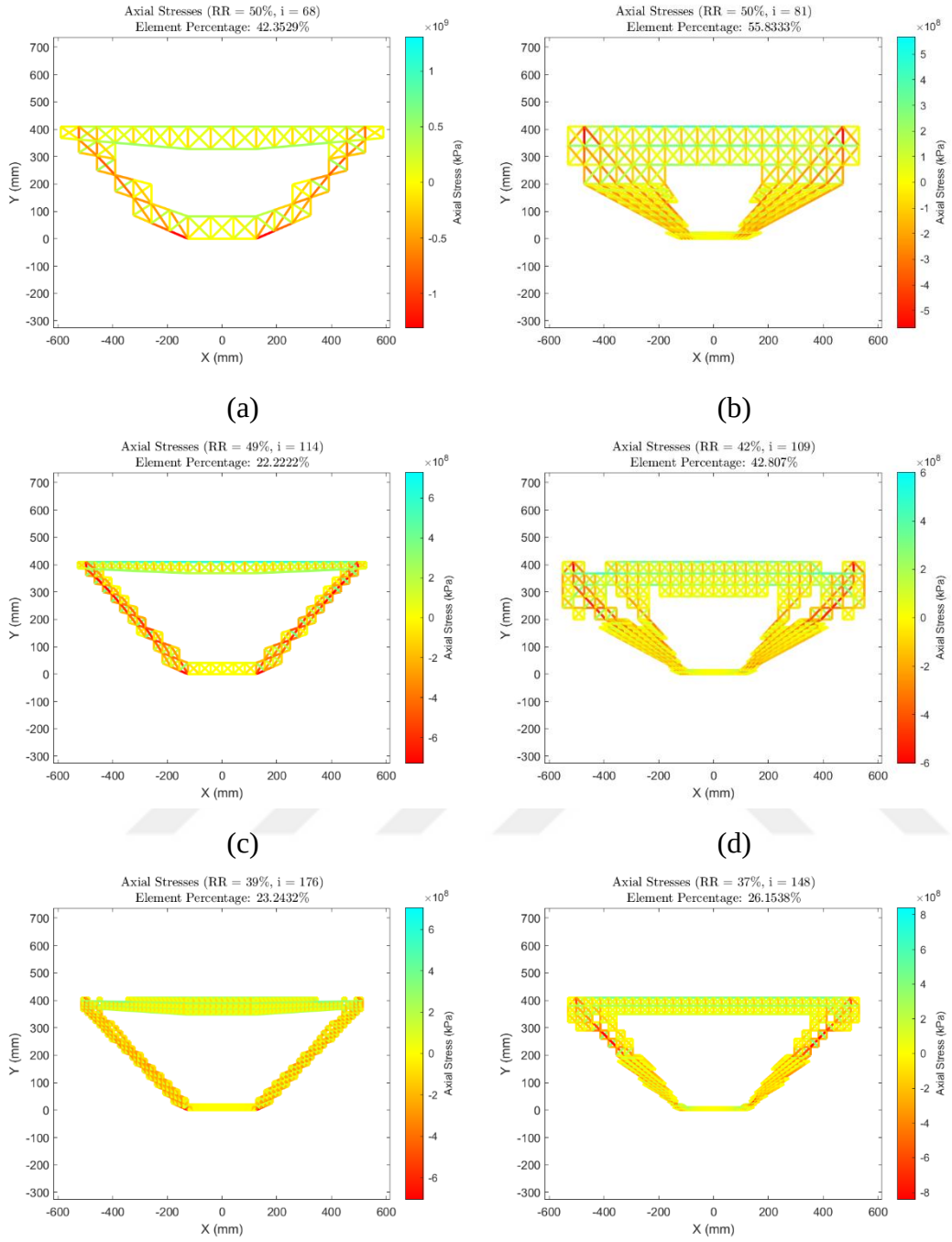


Figure B.33. Pier Cap 2: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

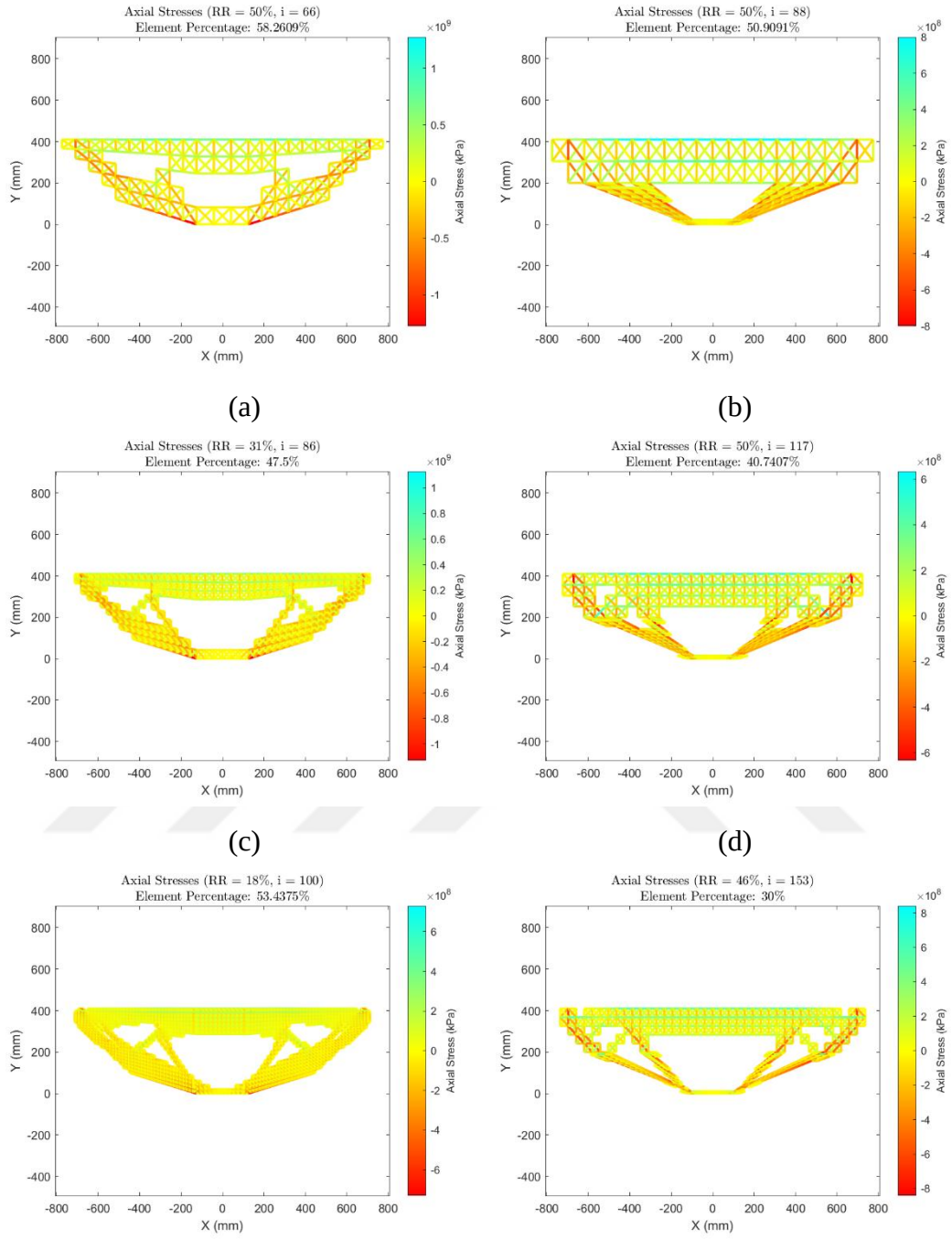
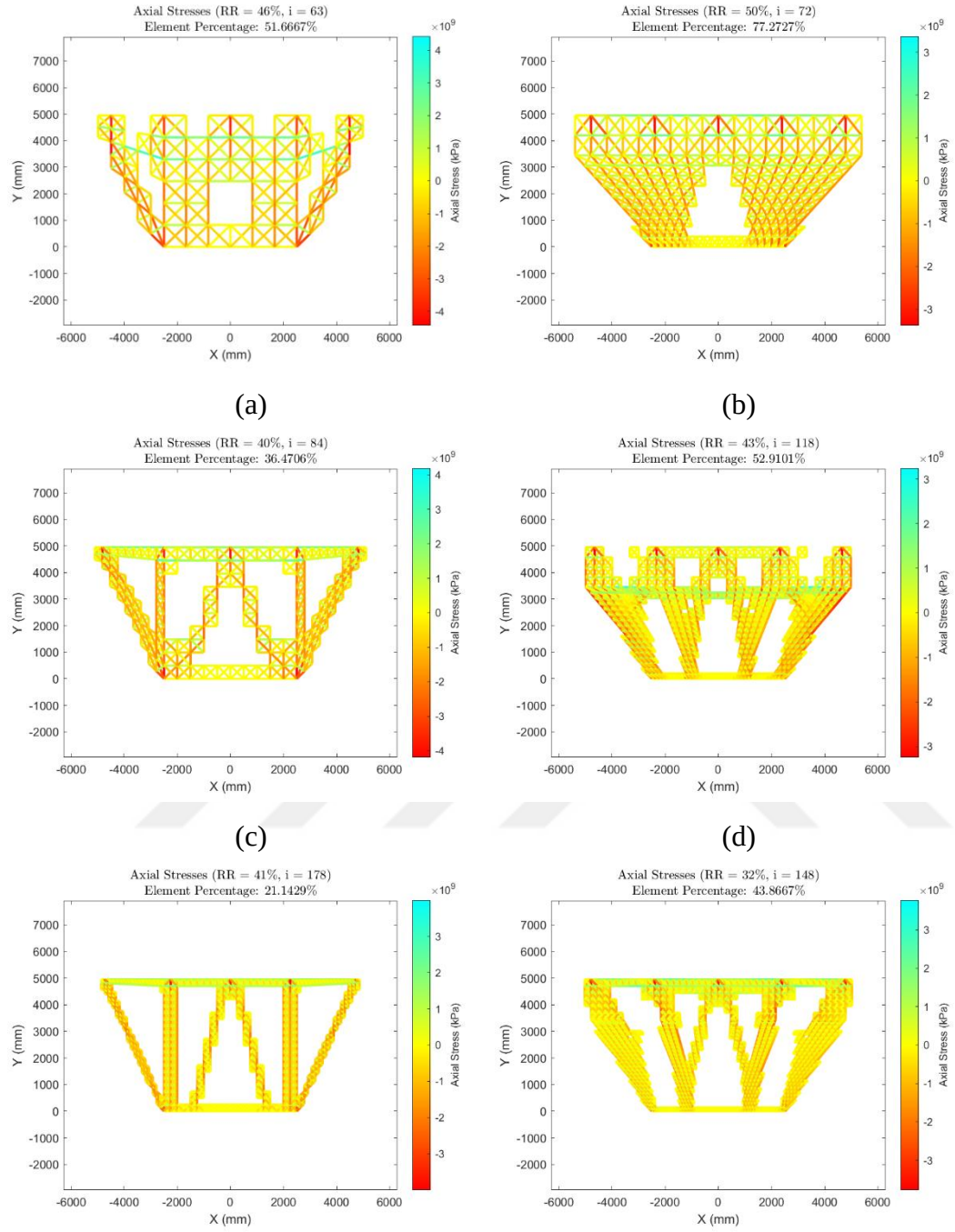


Figure B.34. Pier Cap 3: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2



(a) (b)

(c) (d)

(e) (f)

Figure B.35. Pier Cap 4: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

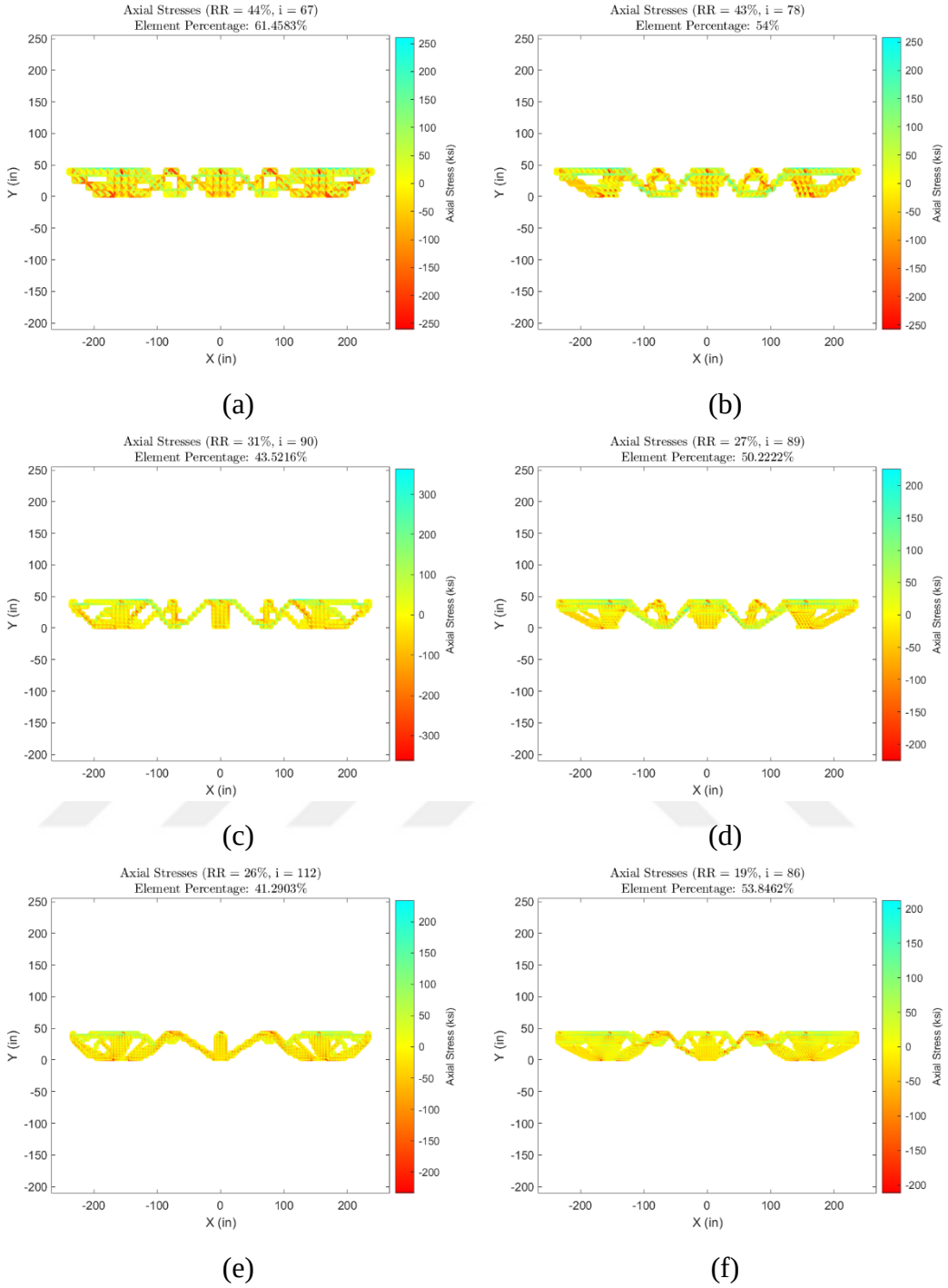
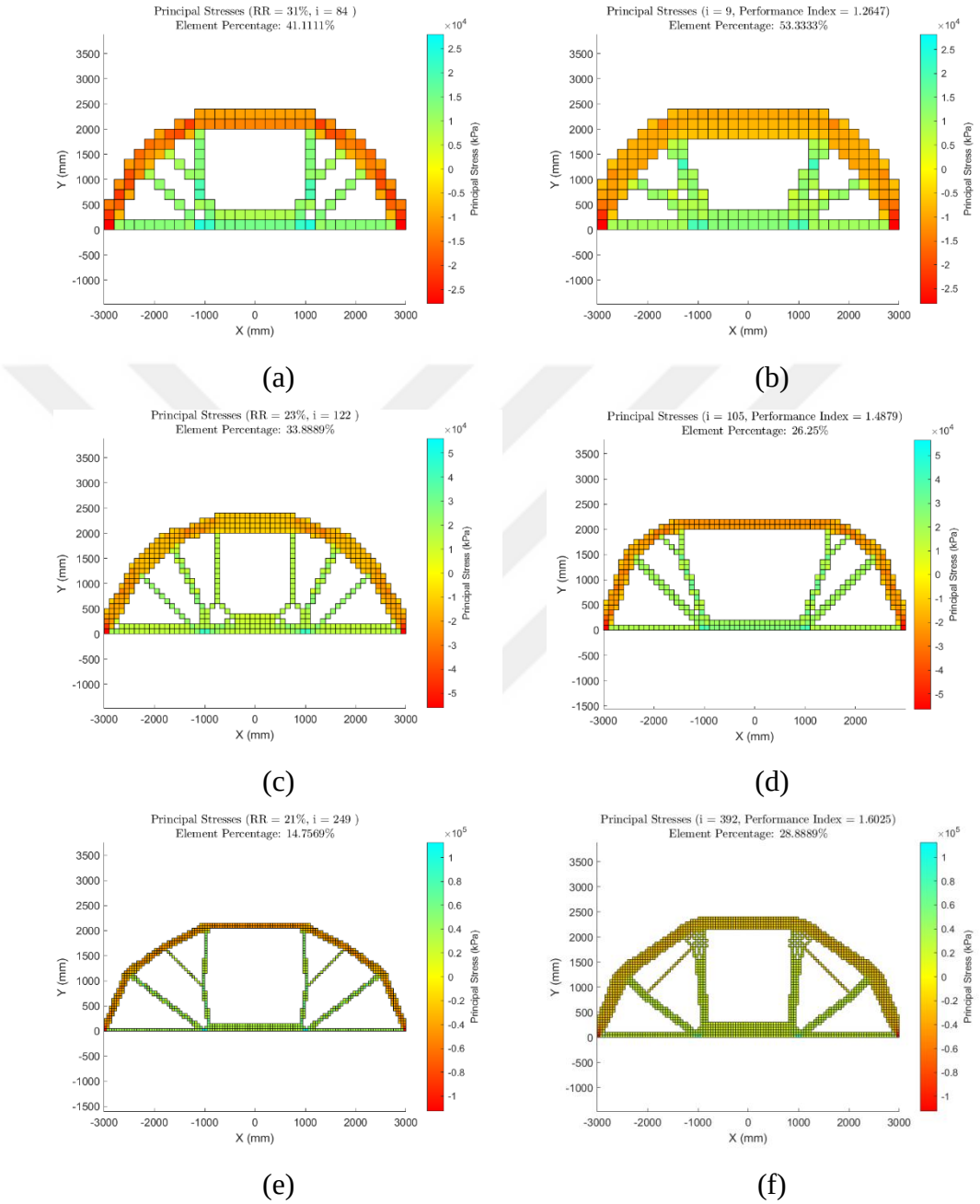


Figure B.36. Pier Cap 5: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

C Membrane Model

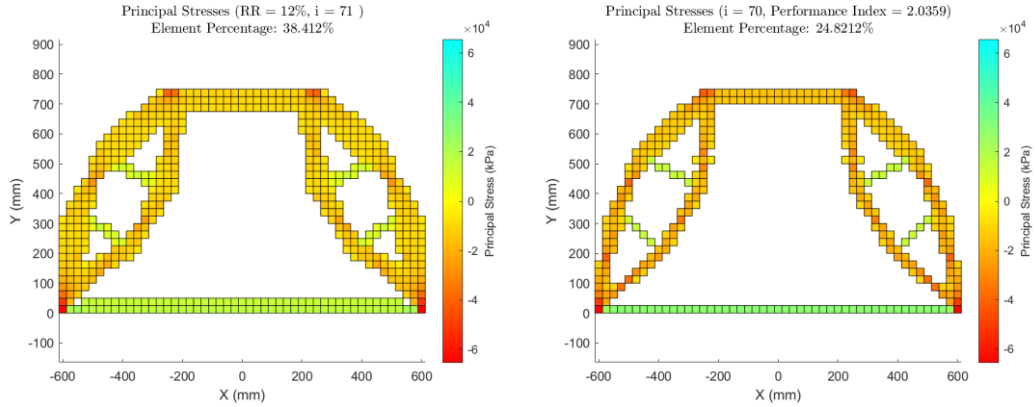


(a) (b)

(c) (d)

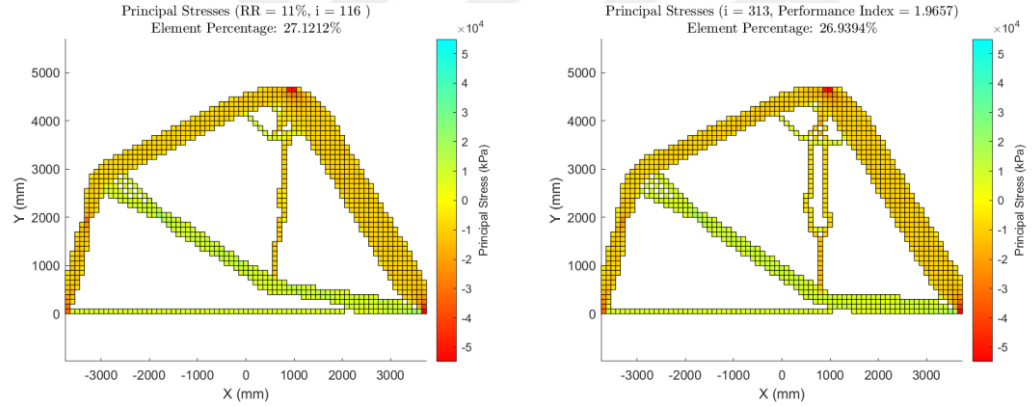
(e) (f)

Figure C.37. Rectangle Example 1: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh



(a) (b)

Figure C.38. Rectangle Example 2: (a) Stress-based; (b) Displacement-based



(a) (b)

Figure C.39. Rectangle Example 3: (a) Stress-based; (b) Displacement-based

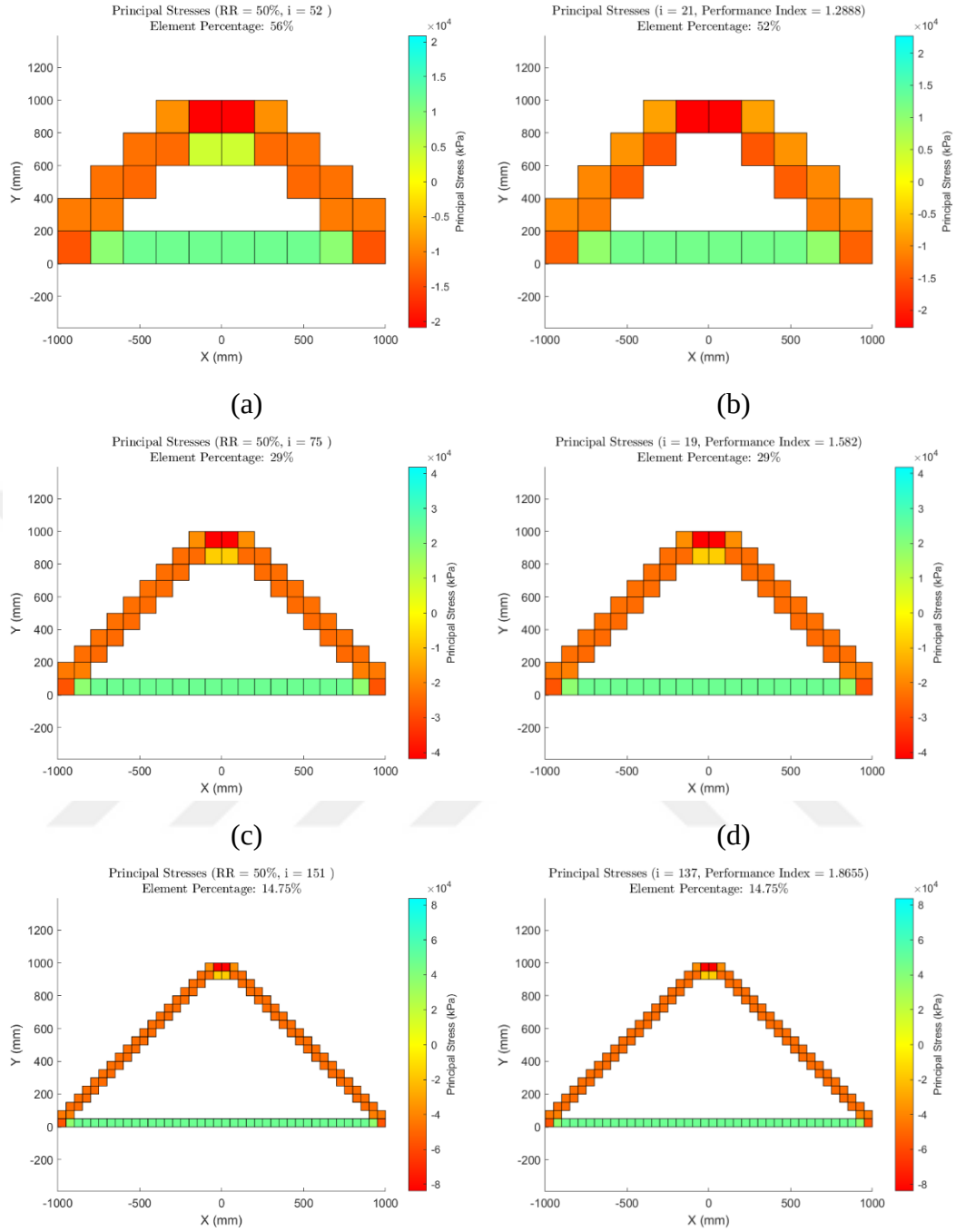
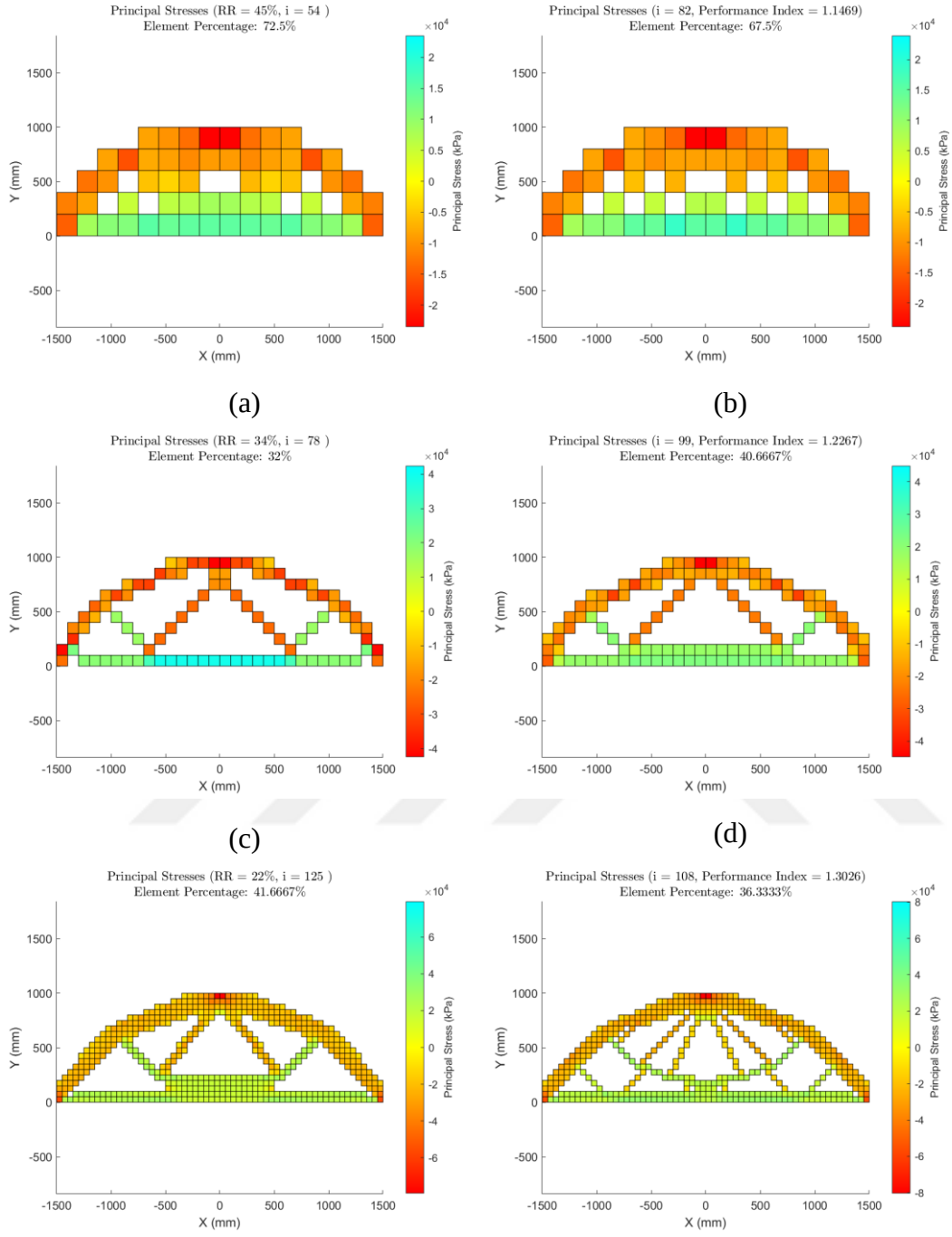


Figure C.40. Rectangle Example 4a: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh

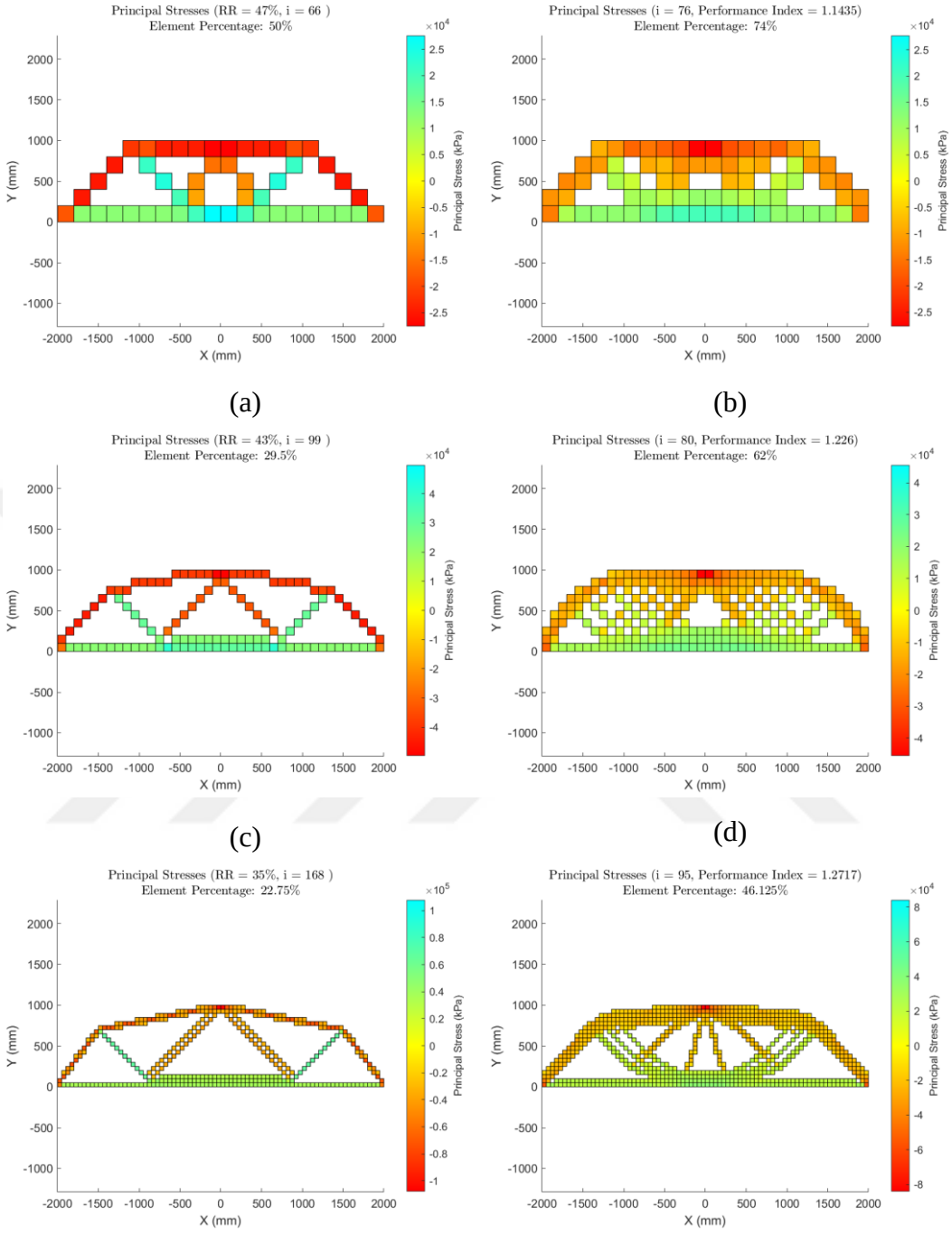


(a) (b)

(c) (d)

(e) (f)

Figure C.41. Rectangle Example 4b: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh

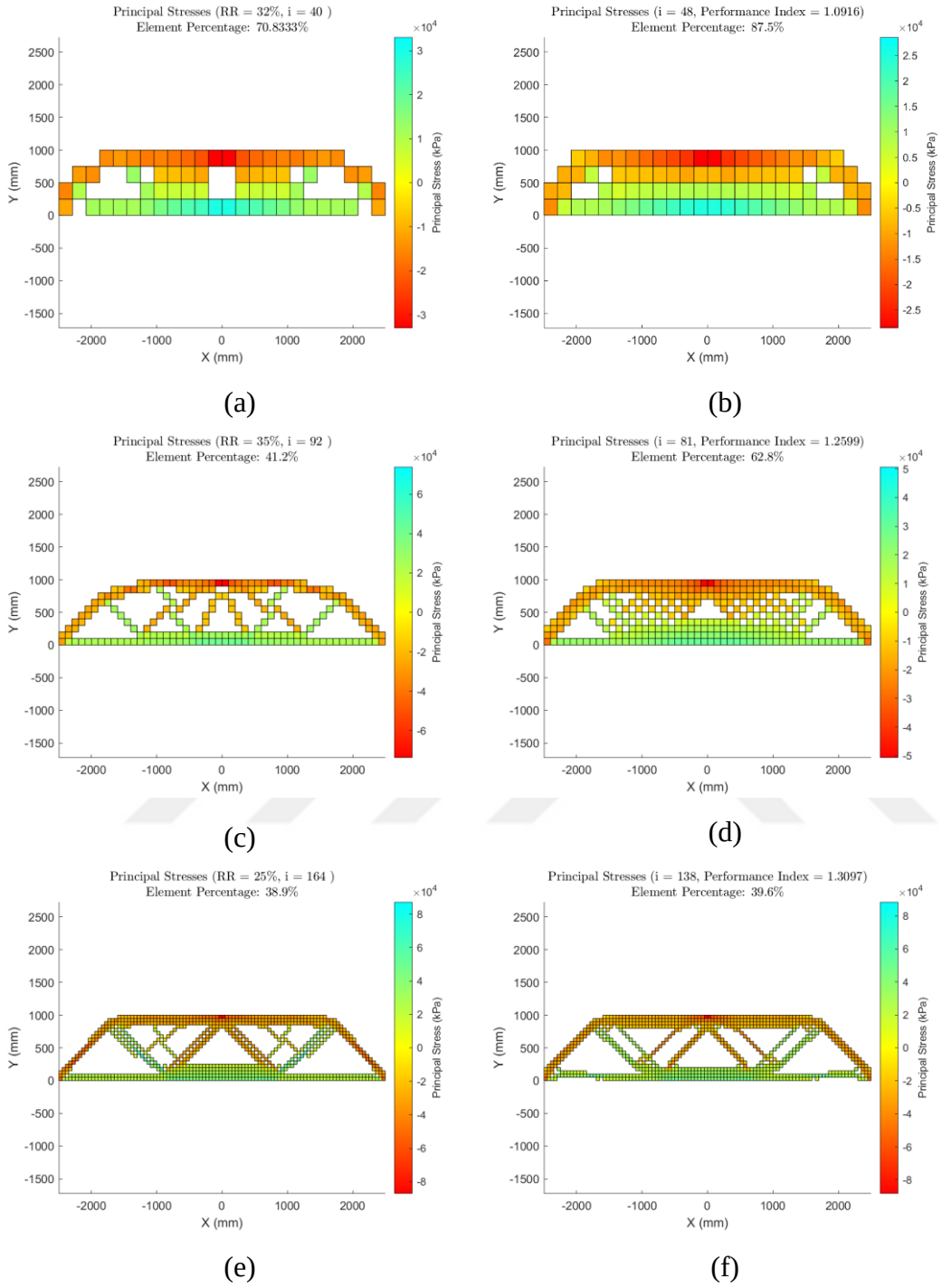


(a) (b)

(c) (d)

(e) (f)

Figure C.42. Rectangle Example 4c: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh



(a) (b)

(c) (d)

(e) (f)

Figure C.43. Rectangle Example 4d: (a) Stress-based Coarse Mesh; (b) Displacement-based Coarse Mesh; (c) Stress-based Medium Mesh; (d) Displacement-based Medium Mesh; (e) Stress-based Fine Mesh; (f) Displacement-based Fine Mesh

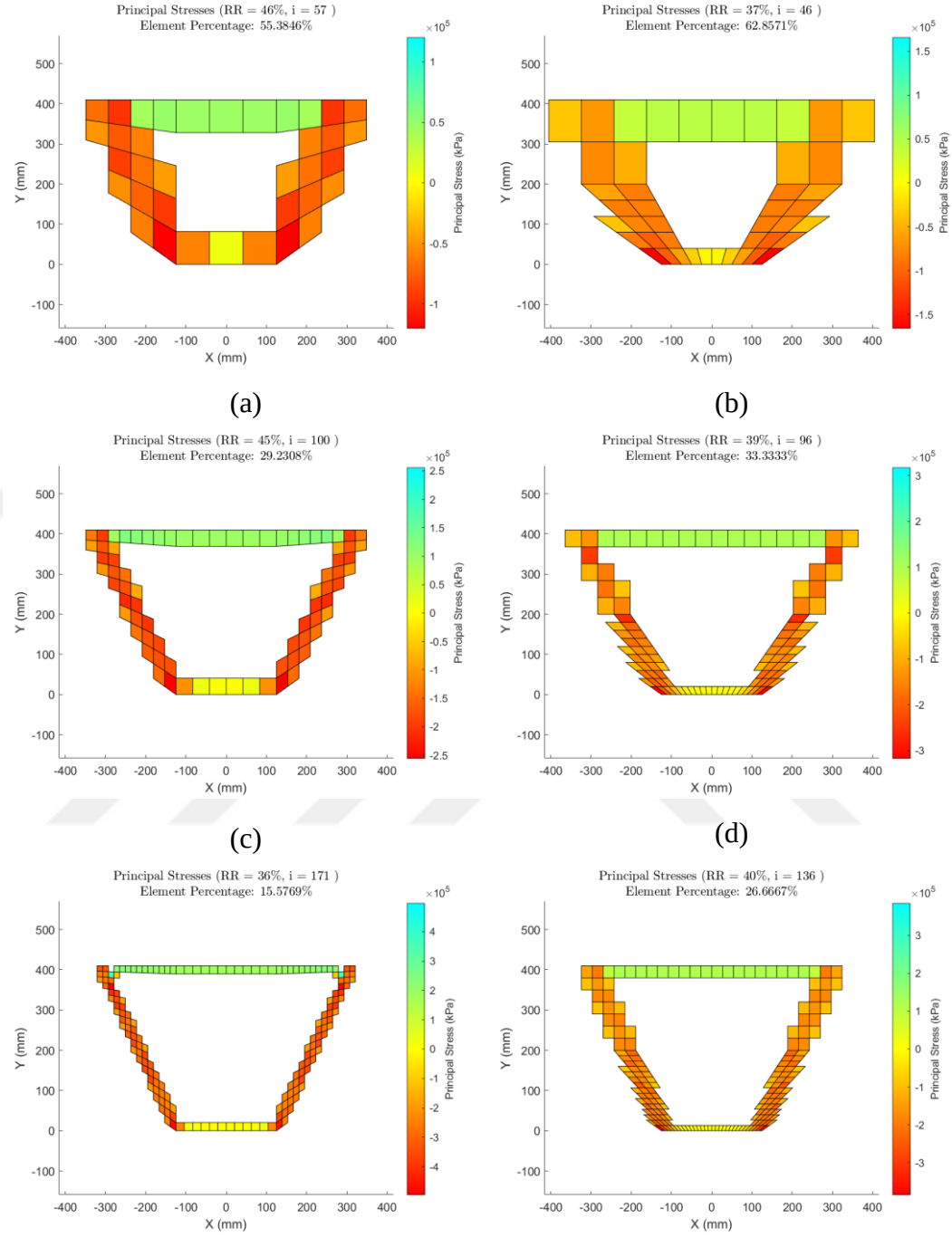


Figure C.44. Pier Cap 1: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

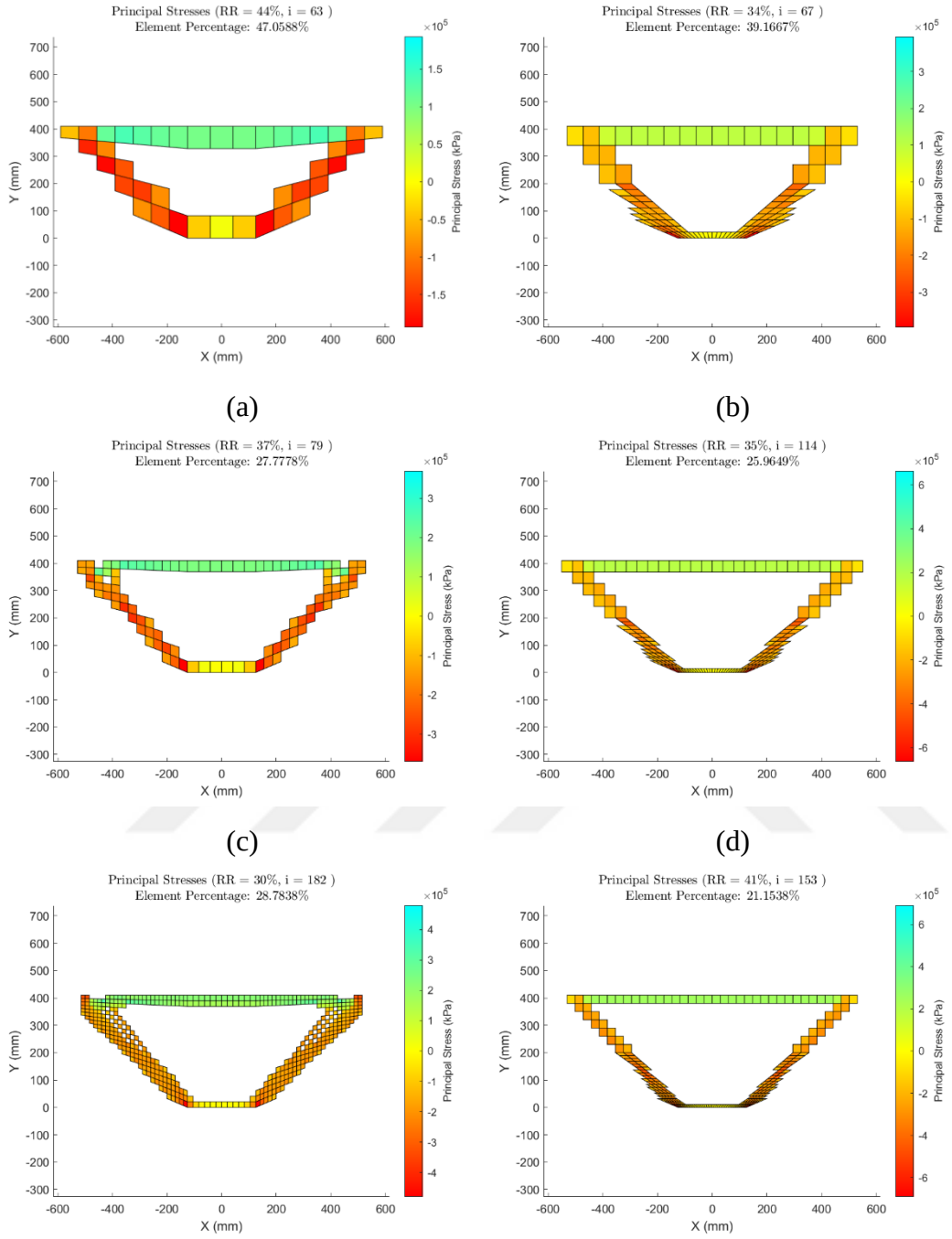
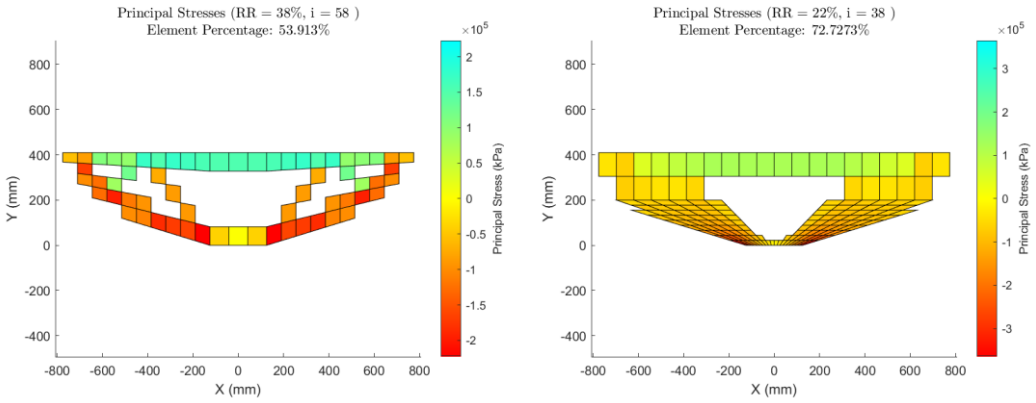
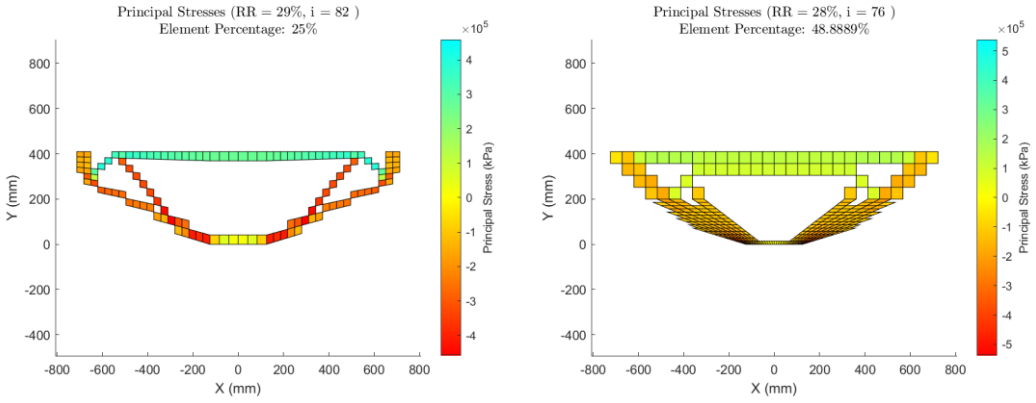


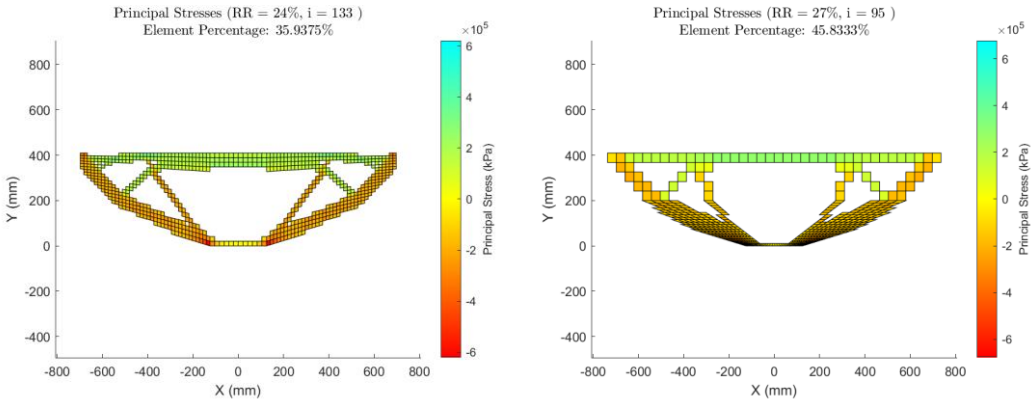
Figure C.45. Pier Cap 2: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2



(a) (b)



(c) (d)



(e) (f)

Figure C.46. Pier Cap 3: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

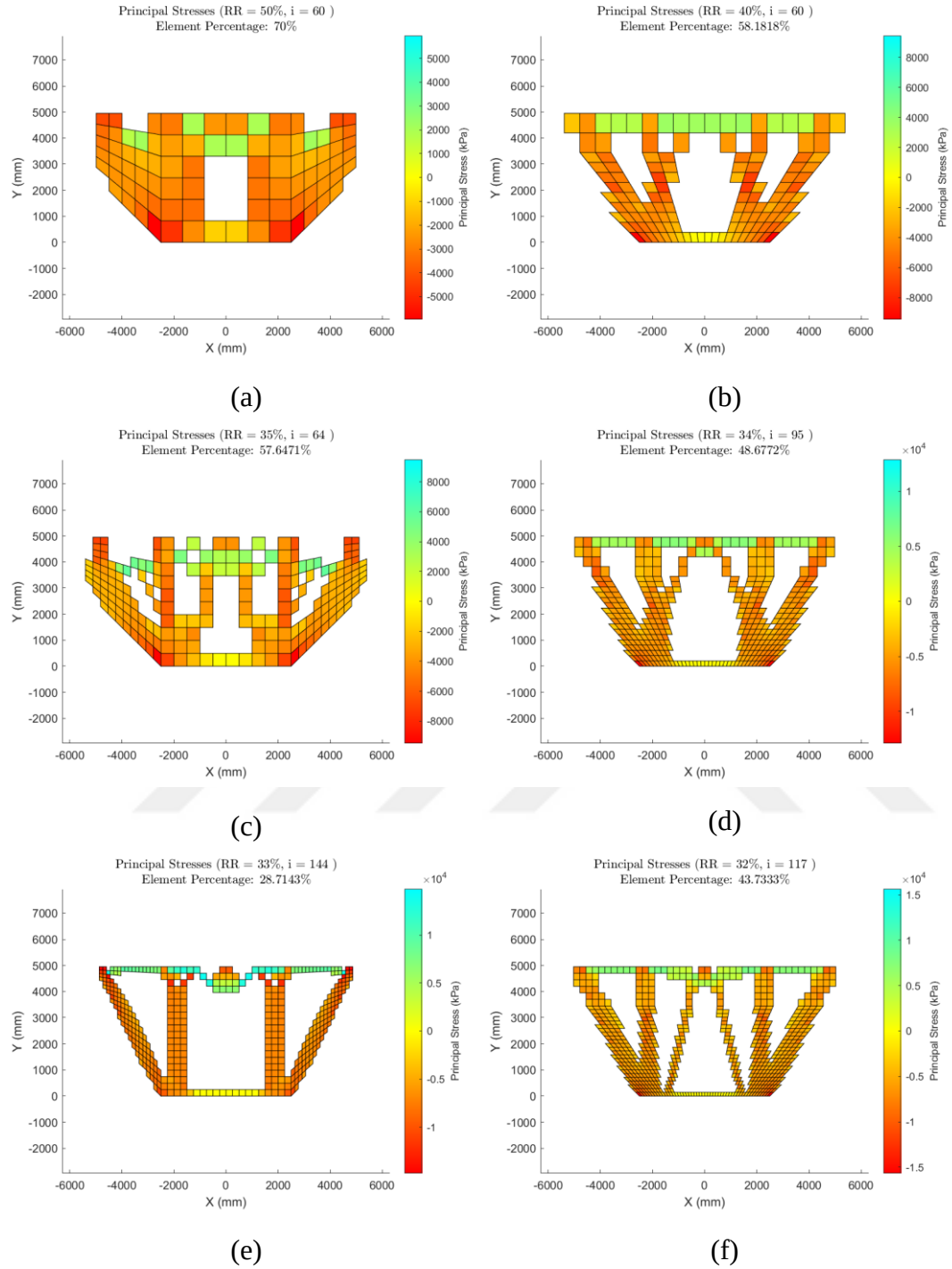
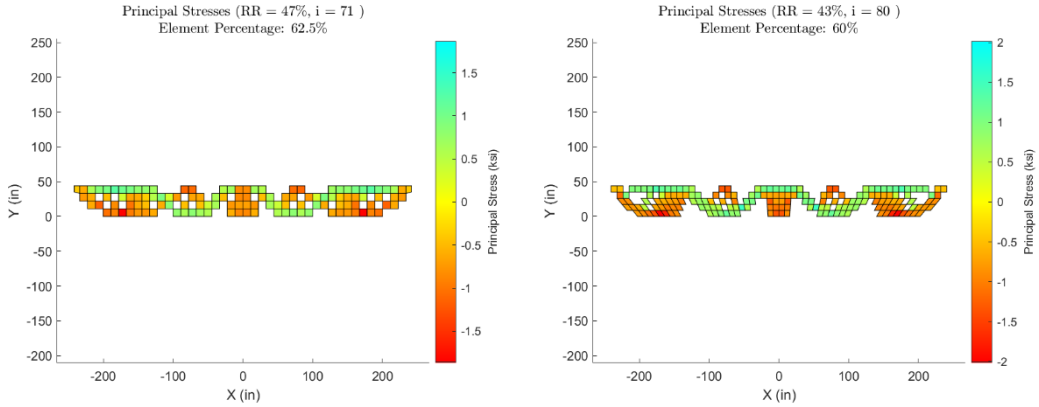
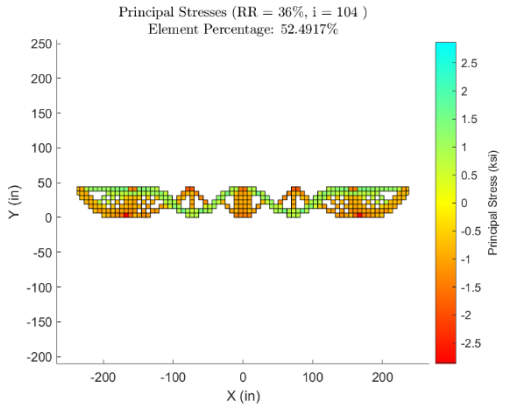


Figure C.47. Pier Cap 4: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2

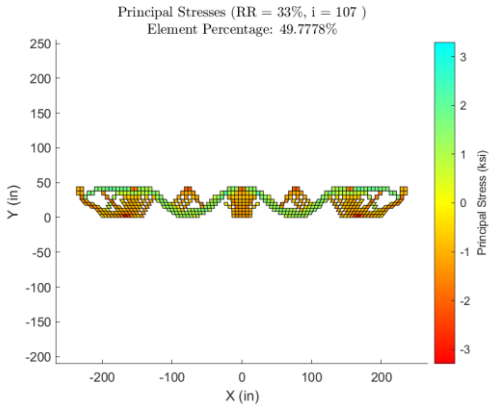


(a)

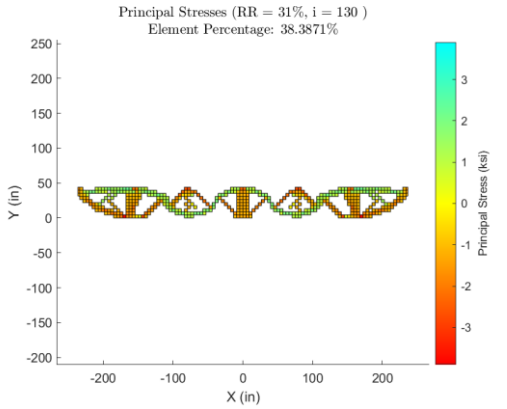
(b)



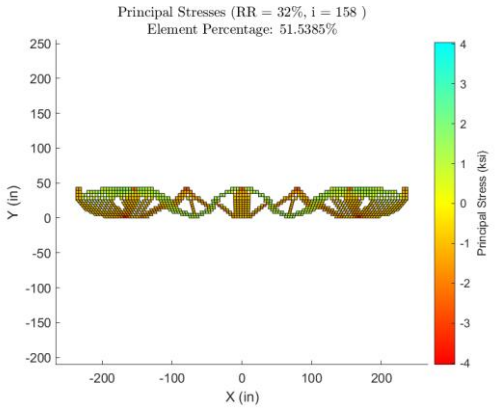
(c)



(d)



(e)



(f)

Figure C.48. Pier Cap 5: (a) Coarse Mesh 1; (b) Coarse Mesh 2; (c) Medium Mesh 1; (d) Medium Mesh 2; (e) Fine Mesh 1; (f) Fine Mesh 2