



YASAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

**A PHYSICS-BASED DESIGN METHOD OF
GRIDSHELL SYSTEMS:**

OPTIMIZATION OF FORM AND CONSTRUCTION COST

ARDA AGIRBAS

THESIS ADVISOR: ASST. PROF. DR. SECKIN KUTUCU

DEPARTMENT OF ARCHITECTURE

PRESENTATION DATE: 22.01.2019

BORNOVA / İZMİR
JANUARY 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

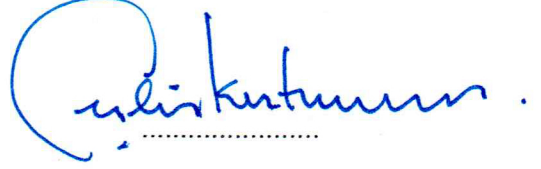
Jury Members:

Asst. Prof. Dr. Seçkin KUTUCU
Yasar University

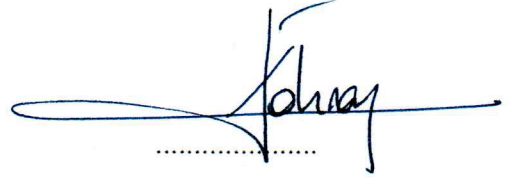
Assoc. Prof. Dr. Ahmet Vefa ORHON
Dokuz Eylul University

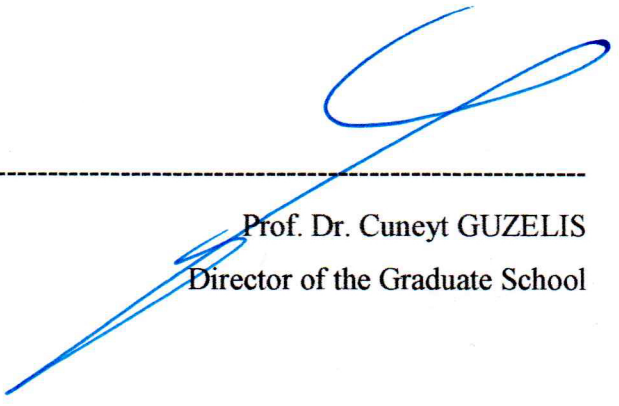
Asst. Prof. Dr. İlker KAHRAMAN
Yasar University

Signature:


.....


.....


.....



Prof. Dr. Cuneyt GUZELIS
Director of the Graduate School

ABSTRACT

A PHYSICS-BASED DESIGN METHOD OF GRIDSHELL SYSTEMS: OPTIMIZATION OF FORM AND CONSTRUCTION COST

AGIRBAS, Arda

Msc in Architecture

Advisor: Asst. Prof. Dr. Seckin Kutucu

January 2019

Structural shell systems have been used by numerous architects for many centuries to cope with different design problems due to their reliable structural behavior and effective material utilization nature. As a kind of traditional shell structures, gridshells have become an essential part of contemporary architecture since they can possess free-form shapes. Despite the numerous advantages, gridshells are not often utilized due to their complex structural nature. The increasing usage of the computer-aided design technologies in the architectural field allows the seamless translation of a design problem to a digital environment in order to overcome the architectural and structural problems simultaneously. This study focuses on creating an adaptive computational method that includes a physics-based form finding procedure, a workflow for structural performance evaluation and an optimization process. This thesis aims to develop a computational and iterative workflow that can be employed in the conceptual idea development phase of different gridshell design problems in order to generate a set of suitable alternative design solutions in accordance with the design-related objectives and constraints. Besides the engineering requirements of a gridshell, the method gives significant importance to the design decisions and limitations in the architectural viewpoint. In order to reach a shape in equilibrium, the “Dynamic Relaxation” form-finding method has been utilized. Afterward, the “Finite Element Analysis” related to the nodal displacements have been performed. Lastly, a multi-objective optimization process has been formulated that aims to minimize the construction cost while creating a discrete transformation of a smooth shell surface as close as possible to the parent shape.

Key Words: Computational Design, Gridshell Systems, Physics-Based Form Finding, Multi-Objective Evolutionary Optimization.



ÖZ

AĞ KABUK SİSTEMLERİ İÇİN FİZİK TABANLI TASARIM YÖNTEMİ: FORM VE İNŞAAT MALİYETİ OPTİMİZASYONU

AĞIRBAŞ, Arda

Yüksek Lisans Tezi, Mimarlık

Danışman: Dr.Öğrt.Üyesi Seçkin KUTUCU

Ocak 2019

Yapısal kabuk sistemleri güvenilir yapısal davranışları ve etkili malzeme kullanım doğası nedeniyle farklı tasarım problemleri ile başa çıkabilmek adına mimarlar tarafından uzun yıllardır kullanılmaktadır. Geleneksel kabuk yapıların bir türevi olan ağ kabuklar, serbest biçimli şekillerde oluşturulabilmeleri sebebiyle çağdaş mimarinin önemli bir parçası haline gelmiştir. Çok sayıda avantajına rağmen ağ kabuklar, karmaşık yapıya sahip olmaları yüzünden tasarım ve uygulama zorlukları sebebiyle tercih edilmemektedirler. Bilgisayar destekli tasarım teknolojilerinin mimarlık alanındaki artan kullanımı, biçimsel ve hesaplama dayalı yapısal özelliklerin bir arada kurgulanabilmesini ve hesaplamalı tasarım yaklaşımı ile entegre çözümler geliştirilebilmesini olanaklı kılmıştır. Bu tezde, içerdiği fizik tabanlı form bulma prosedürü, yapısal performans değerlendirme sistemi ve optimizasyon süreci sayesinde adaptif bir hesaplamalı tasarım metodolojisi geliştirmeye odaklanmaktadır. Tezin amacı ağ kabuk tasarım problemlerinin kavramsal fikir geliştirme aşamalarına dahil edilebilecek bir hesaplamalı tasarım yöntemi geliştirmektir. Söz konusu yöntem, problemlere ilişkin amaçlara ve kısıtlamalara uygun olarak bir dizi alternatif tasarım çözümü oluşturacaktır. Tezde tarif edilen yöntem, ağ kabuk tasarımları ile ilişkili mühendislik gereksinimlerinin yanı sıra, mimari tasarım kararlarına ve sınırlamalara büyük önem vermektedir. Statik olarak dengeli bir şekil bulmak için “Dinamik Gevşeme” form bulma yöntemi kullanılmıştır. Daha sonra, “Sonlu Elemanlar Analiz” yöntemi kullanılarak ağ kabuk yapısındaki deformasyonlar hesaplanmıştır. Son olarak, ana kabuk şeklinin yüzeyine mümkün olduğunca yakın ayrı bir uyarlaması oluşturulurken, inşaat maliyetini en aza indirmeyi amaçlayan çok amaçlı bir optimizasyon süreci formüle edilmiştir.

Anahtar Kelimeler: Hesaplamalı Tasarım, Ağ Kabuk Sistemleri, Fizik Tabanlı Form Bulma, Çok-amaçlı Evrimsel Optimizasyon

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor **Asst. Prof. Dr. Seçkin KUTUCU** for his guidance and patience during this study. I would like to express my gratitude to the jury members **Asst. Prof. Dr. İlker KAHRAMAN** and **Assoc. Prof. Dr. Ahmet Vefa ORHON** for their support and remarkable contribution to this study. I would also like to thank **Asst. Prof. Dr. Feray MADEN**, **Selim ARDALI** and **Orçun Koral İŞERİ** for their support and contribution.

I would like to express my enduring love to my wife **Aybüke AĞIRBAŞ**, who is always supportive, loving and caring to me in every possible way in my life.

Arda AĞIRBAŞ

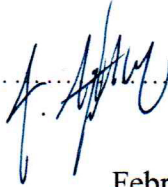
Izmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled “A PHYSICS-BASED DESIGN METHOD OF GRIDSHELL SYSTEMS: OPTIMIZATION OF FORM AND CONSTRUCTION COST” and presented as a Master’s Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Arda Agirbas

Signature

.....


February 16, 2019

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	ix
TEXT OF OATH	xi
TABLE OF CONTENTS	xiii
LIST OF FIGURES	xvii
LIST OF TABLES	xix
SYMBOLS AND ABBREVIATIONS	1
CHAPTER 1 INTRODUCTION	4
1.1. The Scope of the Research	4
1.2. Research Goal and Question	6
1.3. Research Focus and Framework	7
1.4. Method of the Research	9
CHAPTER 2 SHELLS, SPECIAL FEATURES, AND STRUCTURAL BEHAVIOURS ..	13
2.1. Continuous Shell Structures	16
2.2. Gridshells Structures	16
2.3. The Curvature of the Shell Surface	19
2.4. Structural Behavior	20
2.4.1. Flat Plates and Plane Stress	21
2.4.2. Bending Theory and Buckling	23
2.5. Conclusion and Design Goals	24
CHAPTER 3 FORM FINDING METHODS AND STRUCTURAL ANALYSIS	25
3.1. Hooke's Hanging Chain Law	25
3.2. Physical Modeling	26
3.2.1. Scale-Independent Modeling	27
3.2.2. Scale-Dependent Modeling	29
3.3. Computational Form Finding	31
3.3.1. Force Density Method	32
3.3.2. Dynamic Relaxation Method	33

3.3.3. Comparison of the Form Finding Methods	35
3.4. Finite Element Method and Computational Structural Analysis	39
3.5. Conclusion and Design Goals	42
 CHAPTER 4 DEVELOPMENT OF THE COMPUTATIONAL WORKFLOW FOR GRIDSHELL GENERATION AND OPTIMIZATION.....	44
4.1. Design Considerations and Modeling Environment of Generic Gridshell Model	44
4.2. Initial Geometry Creation	49
4.3. Physics-based Form Finding Procedure	53
4.3.1. Creation of the Force Objects	55
4.3.2. Definition of Edge Condition and Support Points.....	56
4.3.3. Definition of Loads and Force Factor	58
4.3.4. Dynamic Relaxation Procedure	59
4.4. Evaluation of the User Defined Constraints	60
4.4.1. Maximum Height Constraint	61
4.4.2. Useful Area Constraint	62
4.5. Assembly of the Structural Model and Structural Performance Evaluation	63
4.6. Optimization of Construction Cost and Shape Approximation	68
4.6.1. Problem Formulation	70
4.6.2. Optimization Procedure	72
4.7. Evaluation of the Set of Alternatives	73
4.7.1. Evaluation of Case 1	75
4.7.2. Evaluation of Case 2.....	79
 CHAPTER 5 CONCLUSIONS AND FUTURE RESEARCH.....	83
5.1. Summary.....	83
5.2. Concluding the Research Questions.....	84
5.2.1. Research Question 1	84
5.2.2. Research Question 2	85
5.2.3. Research Question 3	87
5.3. Research Contribution	89
5.4. Recommendations and Further Research	91

REFERENCES	93
APPENDIX 1 – Values of Objective Functions and Parameters for Case 1.....	97
APPENDIX 2 – Values of Objective Functions and Parameters for Case 2.....	100



LIST OF FIGURES

Figure 1.1. Schematic illustration of the method of the research.....	11
Figure 2.1. Model demonstration of the efficiency of a thin plastic curved element (Chilton & Isler, 2000).	14
Figure 2.2. The Taq-i Kisra Monument and the Roman Pantheon.	15
Figure 2.3. Interior and exterior view of the Fronton Recoletos (Lozano-Galant & Paya-Zaforteza, 2011).	17
Figure 2.4. Schwedler Cupola (Berlin, 1863) and Exhibit Pavilion during construction (Vyksa, 1897).	17
Figure 2.5. The Biosphere by Buckminster Fuller, Montreal, Canada.	18
Figure 2.6. Gaussian curvature examples.	20
Figure 2.7. Structural behaviors of shell and gridshell elements.	21
Figure 2.8. Plane stress and plate bending (Williams, 2014).	22
Figure 3.1. Poleni’s drawing of Hooke’s analogy between an arch and a hanging chain, and his analysis of the Dome of St. Peter’s in Rome (1748).	26
Figure 3.2. Stevin’s funicular form diagrams (1586).	27
Figure 3.3. Proposed section of St. Paul’s Cathedral.	28
Figure 3.4. Gaudi’s hanging chain model reproduction (Asmaljee, 2014).	29
Figure 3.5. 1:60 scale model of Sydney Opera House’s thin-shell roof.....	30
Figure 3.6. 1:300 scale and 1:98.9 scale models of the Multihalle.	31
Figure 3.7. Cable Net Examples by Fresl and Vrancic (2013).	33
Figure 3.8. A catenary model, deformation caused by the applied loads to the nodes.	34
Figure 3.9. The categorization of form-finding methods (Diederik Veenendaal & Philippe Block, 2012).	36
Figure 3.10. The Cycle of Structural Analysis and Design of a Structure (Kaveh, 2014)	39
Figure 4.1. Line creation by using Grasshopper 3D.	46
Figure 4.2. Flowchart of the Grasshopper workflow.	48
Figure 4.3. Initial 2D geometry definition flowchart.	49

Figure 4.4. Two shell surfaces generated from a trapezoid and a rectangle, respectively.	Error! Bookmark not defined.
Figure 4.5. Two identical cylindrical surfaces divided into quadrilateral sub-surfaces.....	51
Figure 4.6. Different surface tessellation types.....	52
Figure 4.7. The flowchart of the physics-based form finding the procedure.....	54
Figure 4.8. Numbering procedure of geometry's edges and Edge Condition Component defined in Grasshopper.	56
Figure 4.9. Edge Condition values and their effect on shell geometry.....	57
Figure 4.10. Dynamic Relaxation process of a flat rectangular surface.	60
Figure 4.11. Flowchart of Constraint Evaluation in Grasshopper workflow.....	61
Figure 4.12. Interpretation of Useful Height input and Useful Area constrain.	62
Figure 4.13. Flowchart of Assembly of structural model and evaluation in Grasshopper workflow.....	64
Figure 4.14. Loads and support points of the <i>Finite Element Model</i> in <i>Karamba</i>	67
Figure 4.15. Displacement Analysis in <i>Karamba</i>	67
Figure 4.16. Discretization of a smooth sphere (Pottmann et al., 2007).	71
Figure 4.17. Workflow diagram of the optimization procedure.	72
Figure 4.18. The input geometries for case 1 and case 2, respectively.	73
Figure 4.19. Edge numbers of input geometries in case 1 and case 2, respectively.	74
Figure 4.20. Pareto Chart of Non-Dominated Solutions at the 200 th generation for case 1 with selected alternatives.	75
Figure 4.21. Alternatives selected from the non-dominated solutions for case 1.....	76
Figure 4.22. Alternatives selected from the non-dominated solutions for case 1.....	77
Figure 4.23. Pareto Chart of Non-Dominated Solutions at the 200 th generation for case 2 with selected alternatives.	79
Figure 4.24. Alternatives selected from the non-dominated solutions for case 2.....	80
Figure 4.25. Alternatives selected from the non-dominated solutions for case 2.....	81

LIST OF TABLES

Table 2.1. Classification of the surfaces by their shapes.	19
Table 3.1. The needed values to be defined by the user for each method (Veenendaal and Block, 2014).	38
Table 4.1. Classification of the wooden beams by their dimensions.	58
Table 4.2. Properties of the pre-defined wood material.	65
Table 4.3. Decision Variables.	68
Table 4.4. Octopus settings.	74
Table 4.5. Values of decision variables and objective functions for selected alternatives for case 1.	76
Table 4.6. Values of decision variables and objective functions for selected alternatives for case 1.	77
Table 4.7. Values of decision variables and objective functions for selected alternatives for case 2.	80
Table 4.8. Values of decision variables and objective functions for selected alternatives for case 2.	81

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

FDM Force Density Method

DRM Dynamic Relaxation Method

FEM Finite Element Method

FEA Finite Element Analysis

NURBS Non-Uniform Rational Basis Splines

CAD Computer-Aided Drawing

MHC Maximum Height Constraint

UH Useful Height

SYMBOLS:

A_s	Gridshell surface area.
A_{shape}	Shape approximation.
A_x	Cross-section area.
A_{x-sel}	Cross-section selection.
C_b	Total cost of the wooden beams.
C_{dir}	Directional factor of the wind.
C_g	Total construction cost of the gridshell.
C_j	Total cost of the joints.
C_n	Unit cost of the joint elements.
C_{season}	Season factor of the wind.
d_w	Density of the wood material.
E	Modulus of elasticity.
E_{con}	Edge condition.
F_n	Force in each node.
F_x	Force factor.
F_w	The wind force.
H_{max}	Maximum height constraint.
k	Spring stiffness.
l_{max}	Largest span of the structure.
L_s	Initial length of the spring.
L_T	Total length of the discrete structural elements.
N_j	Total joint count.
N_{u-div}	Surface division count in U - Direction.
N_{v-div}	Surface division count in V - Direction.
q_b	Dynamic pressure.

- R_{UA} Useful Area Constraint.
- S_t Surface triangulation.
- v_b Basic wind velocity.
- $v_{b,0}$ Fundamental value of the basic wind velocity.



CHAPTER 1

INTRODUCTION

1.1. The Scope of the Research

The architectural design procedure is often described as a complicated process since it requires being able to answer people's different needs at the same time. In addition, the design process involves making decisions on many parameters simultaneously, in order to develop a required output, while satisfying design constraints. The decisions in this process significantly affect the finalized solutions.

Throughout history, the architects used shells to overcome various design problems due to their structural stability and efficient material usage. A shell structure can be defined as a system that not only provides a covering skin but also is capable of carrying itself and relative loads without needing any additional supports. In most of the cases, the high structural performance of the shell is the result of its geometrical properties which allows the redirection of the forces to a support structure. A similar principle can be seen in the examples in nature such as bird eggs, seashells, and soap bubbles. A well-design shell system can work effectively in relatively large spans; therefore, in the design problems which contains large areas without any vertical supports, a shell can be considered among the most effective choices. Also, a shell system can provide the required freedom to the designer in the formal exploration process of the structure since it can offer great flexibility regarding shape. Shells may offer an extended range of choices in terms of shape, from simple symmetrical vault or dome shape to more complex asymmetrical freeform shapes.

Gridshells can be defined as a kind of updated traditional shell structures such as a vault or a dome. Instead of one continuous surface element that acts as both cover and support in shells, gridshells consist of discrete structural elements with space in-between them. In other words, gridshells are shells where material has been removed in order to create a balanced pattern for the stress. This material relief affects the structural behavior of the gridshell and makes it a little different than a continuous shell. In gridshells, the loads are carried by discrete structural elements instead of a continuous surface. Therefore, the internal forces in a gridshell can follow a limited number of paths whereas a continuous shell contains an infinite number of possible load paths. Even though, both shell structures resist the loads through axial stress, the

cross-sections of slender gridshell elements contribute the resistance to the bending forces that may occur as a result of a non-optimized form. Besides these structural features, gridshells are considered very similar to continuous shells concerning overall structural behavior and the possibility of adopting a range of different geometries. Gridshells can be built in various forms both symmetrical and asymmetrical in order to cross large spans with well-adjusted material usage. Unlike a continuous shell, the in-between spaces can be utilized as openings on the structure's surface. They can be made of numerous different materials such as steel, structural aluminum or wood. Also, the pattern that is utilized in the tessellation of the surface may be considered as a decision variable in order to contribute to the overall appearance of the structure.

Despite the advantages, gridshells are not often utilized due to their complex design process and hard engineering structural calculation requirements. One of the most significant problems in the gridshell design is to decide geometrical features of the structure since the geometry of the structure should satisfy the structural requirements while fulfilling the functional, aesthetical and programmatic concerns. Therefore, the form finding process is a critical decision-making process which must include the synthesis of both architectural and engineering design decisions related to the problem. Nonetheless, the use of algorithms in the computer technologies and the increasing usage of the computer-aided drawing programs in the architectural field make the design problem's translation to the digital environment possible in order to coop with both architectural and engineering aspects simultaneously. By using algorithmic modeling programs, the elements of a gridshell can be generated and controlled parametrically. Moreover, the initial formal exploration process can be richened with the integration of the form-finding methods. As a result of the rule-based computational process, the initial form will be statically stable. Also, the structural verification of the elements can be done throughout the design process.

For this reason, the study focuses on developing a generic computational model which can be utilized in the gridshell design process by the designers. The model includes the generation of gridshell, its structural verification, and optimization of the form and construction cost. As a result of the optimization process, within the defined decision variables, design constraints, and design objectives a set of multiple solutions in a specific design problem can be generated.

In this thesis, an iterative, computational design method for gridshell design is developed. The method does not only include engineering features but also gives high importance to the design decisions from the architectural perspective. The designer defines the decision variables and design constraints, and they have a significant impact on the generated set of alternatives.

1.2. Research Goal and Question

The primary objective of this thesis is to develop a computational design method that is capable of generating gridshell structures with different geometrical and structural properties. Within the scope of the user-defined inputs and limitations, the results of the created workflow vary. Also, the developed method includes an optimization process between the geometry of the gridshell and initial construction cost subject to the user-defined design constraints such as the maximum desired height and the ratio of the useful floor area to the total floor area. Besides the architectural constraints, the method includes structural constraints related to the maximum allowable deflection. The user should define the inputs such as the geometrical boundary of the floor which is intended to be enclosed with the structure, the placement of support elements and their relationship with the overall structure.

Concerning the primary goal, the research questions are also as follows:

- What are the geometrical features and structural principles of shell systems?
- What are the methods related to form-finding and structural analysis of a shell and can we determine a design method in the utilization of a gridshell?
- Can we develop a computational workflow for a generic gridshell design process that includes the form-finding process as well as an optimization between geometrical features and construction cost?

1.3. Research Focus and Framework

The thesis focuses on developing an iterative design method that can be included in the design phase of gridshell structures. The method concerns about the architectural decision and their effects on the shape as well as the required engineering equilibrium calculations related to a gridshell design. In addition, the method includes an optimization process for minimizing the construction cost while creating a discrete gridshell geometry as close as possible to the initial smooth shell surface. In this context, literature reviews on gridshells' structural behavior, form-finding methods and structural performance verification are done. Thus, a framework is determined for the computational gridshell design method.

The computational design approach of the study is essential in order to create a design environment that allows the examination of the outcome of architectural decisions related with a gridshell design problem as well as their effect on the structural performance, simultaneously. Moreover, including an optimization module to a computational design environment is relatively easy since the digital model which is created in this environment consists of multiple parameters that can be translated into decision variables and constraints.

In order to control the architectural aspects of the gridshell, the architectural decisions related to the overall design are defined as discrete parameters, inputs, and constraints in the context of the computational design method. Due to the parametric nature of the design method, it can be adapted to different site conditions and limitations related to a gridshell design problem. Also, constraints that are related to the architectural aspects of the gridshell are included in the workflow. The reason for that is to control physical properties of the structure as well as the ratio of the useful floor area to the total floor area in order to minimize the loss of area due to the unsuitable ceiling height related with the usage of the space.

By introducing architectural and structural constraints, the optimization process results with a suitable set of alternative gridshell designs that satisfies the required criteria. The iterative character of the method takes its power from the optimization module which is included in the context of this study. The objectives of the optimization process are defined as the minimization of the construction cost and maximization of the shape approximation of the discrete gridshell geometry to the smooth shell surface

while satisfying the constraints related with architectural decisions and structural performance.

The construction cost is calculated by considering the volume of the discrete gridshell elements and the number of the joint elements in accordance with their relative unit prices. The costs related to workmanship and maintenance is not included in the cost calculation. Even though glass panels with aluminum structural frames are included in the structural calculations as cover material for gridshell, they are not included in the price calculations. The reason for that decision is, the force on the gridshell due to the self-weight of this material is relatively large. Therefore, a structure that is capable of carrying this material can carry most of the different construction materials. However, the price of the glass paneling system is significantly higher than the price of the wooden structural elements of gridshell; therefore, the price calculations result with the domination of the price of glass paneling system dominated when it is included.

The approximation of the smooth shell surface is defined within the context of the discrete transformation of the initial surface geometry. In cases that a gridshell consists of discrete linear elements and planar faces, it is not possible to create a smooth gridshell surface. However, by refining the surface division and tessellation, it is possible to control the generating lines in the discrete transformation process. In other words, the more the generating lines and discrete transformation are refined the better the discrete version will approximate the initial smooth surface geometry. Therefore, the maximization of the number of sub-surfaces as a result of the discrete transformation is considered as an objective in the optimization process in order to create the discrete version as close as possible to the initial surface.

Regarding the optimization algorithm, the evolutionary algorithm named *HypE* is used in this study since it is a predefined algorithm within the digital tools which are included in the computational design method. Due to the usage of a single algorithm in the optimization process, the performance evaluation of the optimization algorithm is excluded from the scope of the thesis.

A generic wood material, which is a predefined material within the context of the tools that are included in the workflow, is utilized for the gridshell elements in the scope of the study. Three different rectangular cross-sections for the linear wooden elements are employed. The form-finding approach which is included in the design method is

considered as material-oriented rather than a geometry-oriented procedure since the properties of the material are known, therefore, the *Dynamic Relaxation Method* is chosen in order to utilize in the physics-based form-finding process of the gridshell model.

Structural performance evaluation of the gridshell alternatives is done by using “*Finite Element Method (FEM)*,” therefore, the design method includes the assembly of the *Finite Element Model* of the gridshell. Concerning the loads, the gridshell structure’s self-weight, the weight of the cover material and wind loads are taken into account. In the FEM analysis module, the maximum displacement of the structure, as well as the nodal displacements of the relative elements, are calculated and evaluated. Material properties of the generic wood material which is included by the FEM module have been used to perform the analysis.

1.4. Method of the Research

The design of a gridshell is often considered as a complex process because the architectural decisions related with the geometry have a direct relationship with the structural performance since the elements of gridshell are load-bearing besides the aesthetical features. Also, the engineering calculations related to finding the shape for a gridshell which is structurally in equilibrium can be considered as complicated. According to the research, it can be stated that thanks to the recent development in the computational design programs, the gridshell design problems can be translated into a digital design environment in order to coop with architectural and structural aspects simultaneously. Within this regard “*shells and their structural behaviors*,” “*form-finding methods*” and “*computational workflow for gridshell generation and optimization*” are the chapters of the method to follow. The method of the research has been expressed as in the followings;

Shells, Special Features and Structural Behaviors: This chapter of the study aims to explore the difference between structural shell types and their structural behaviors. The historical examples of continuous shell systems and the development of the gridshell structural system are examined. The examination of their geometrical properties and the structural principles helps to define multiple criteria for gridshell design and structural evaluation.

Form Finding Methods and Structural Analysis: In this chapter, the form-finding methods, as well as the structural analysis procedures related to shell systems, are studied. The evolution of form-finding methods from the traditional physical form-finding methods to computational form-finding methods is examined. Moreover, a comparison is made between the different types of computational form-finding methods. Furthermore, the application of *Finite Element Method* in the structural analysis field is examined. The information gained from this part of the study is utilized in the form-finding process and the structural performance verification part of the computational gridshell design method.

Development of the Computational Workflow for Gridshell Generation and Optimization: This chapter aims to develop a computational workflow that includes generation of a gridshell as well as the optimization process considering related objectives and constraints. In order to create the workflow, *Rhinoceros 3D a Computer Aided Drawing (CAD)* software and its visual programming extension *Grasshopper* is used. In this digital design environment, an algorithmic design method is created that consists of consecutive steps. The adaptive nature of the workflow allows the creation of different gridshell design alternatives as a response to a given design problem. Following the initial definition of the user inputs, the physics-based form-finding procedure which is included in the workflow helps to create an initial shell geometry by utilizing real-world physics simulations. For the simulations, *Kangaroo* a physics engine plug-in for *Grasshopper* is used. Following the acquisition of shell geometry, a discrete version of this geometry is creating by dividing its surface considering the division count and tessellation type. To assemble a *Finite Element Model*, *Karamba* plug-in for *Grasshopper* is included in the workflow. With the help of this plug-in deformation analysis are made to evaluate the structural performance of the gridshell model. In order to create a set of gridshell design alternatives *Octopus* which is a *Multi-Objective Evolutionary Optimization* plug-in for *Grasshopper* is used. The gridshell design problem is translated into a multi-objective optimization problem that includes the minimization of the construction cost and the maximization of the approximation of smooth shell surface related to the discrete transformation of the shell surface. Lastly, the method is applied to two different gridshell design problems, and the outputs of these two cases are evaluated.

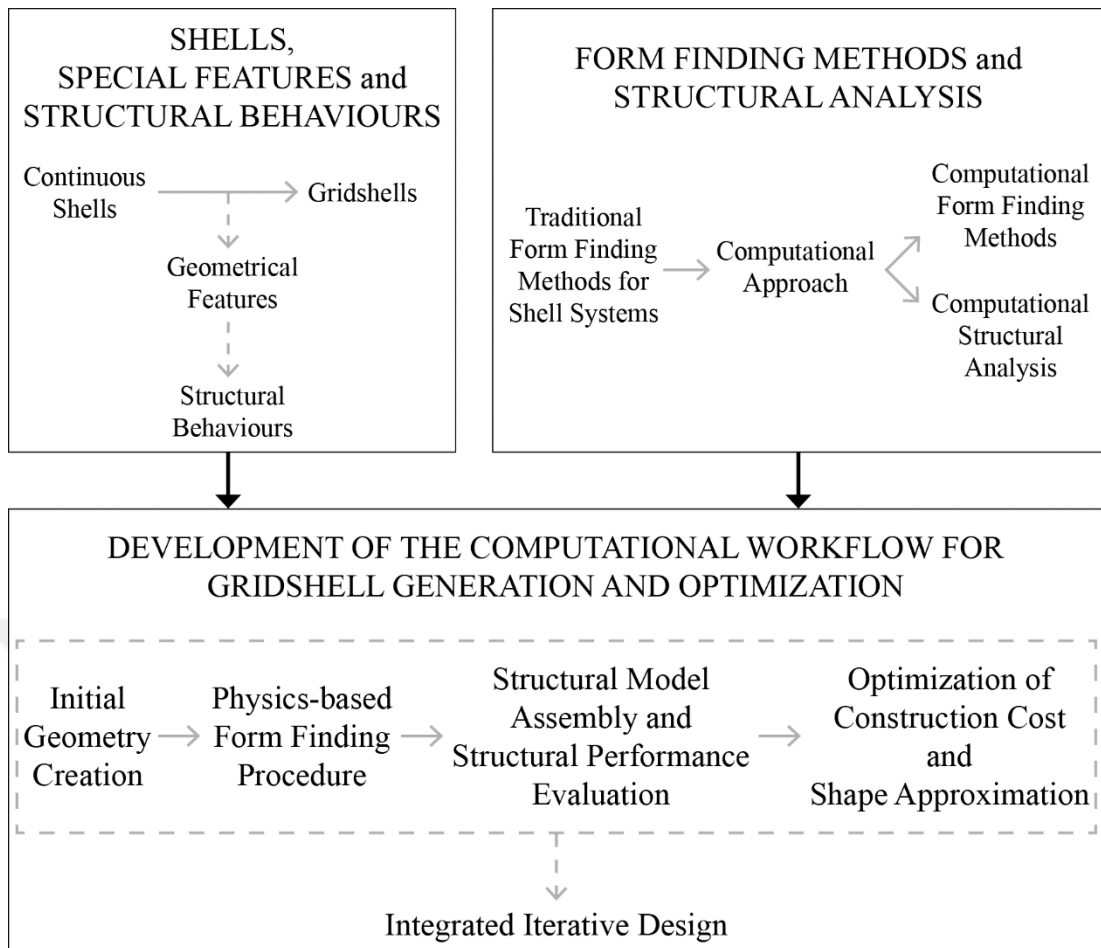


Figure 1.1. Schematic illustration of the method of the research.

CHAPTER 2

SHELLS, SPECIAL FEATURES, AND STRUCTURAL BEHAVIOURS

Structures can be categorized by following different approaches in terms of their form, their functionality and the materials that included the structures, etc. (Engel & Bandel, 1967). Considering the shape of the elements which are created throughout the history of structural design, the shapes can be roughly categorized as linear and surface elements (Dimcic, 2011). If one dimension of the element is significantly larger than the other two dimensions, the element can be categorized as a linear element such as columns and beams whereas, if the two dimensions of the element are relatively proportional and significantly bigger than the other dimension, the element can be named as surface elements such as walls, slabs, etc.

The orientation of the elements in 3D space can be different ways, and it is one of the most critical factors that define the character of the element. In other words, a linear element could be utilized vertical or horizontal which named different in each approach as a column or a beam, respectively. The same rule applies to the surface elements; they also can be named as a slab for horizontal utilization and a wall for vertical utilization. Even though elements that have a certain slope are used generally for roof structures, therefore, name differently. However, they still have the same principle regarding the load distribution over one or two dimensions. Different combination of both systems can be seen throughout the history following the development of the technology and discovery of new materials.

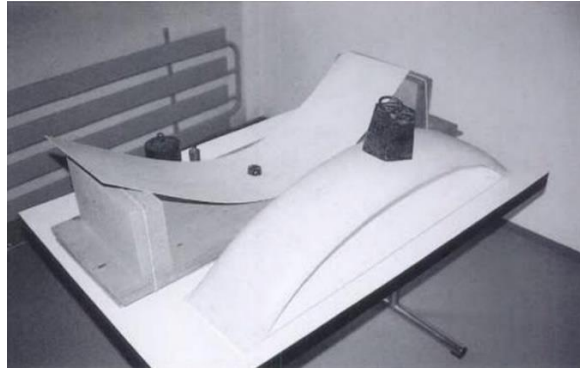


Figure 2.1. Model demonstration of the efficiency of a thin plastic curved element (Chilton & Isler, 2000).

However, the straight and flat elements can be considered as inefficient in comparison to the curved ones. Heinz Isler who is a Swiss engineer created a simple experiment model (shown in Fig. 2.1) to demonstrate the strength of a curved plastic element in which can resist more than 30 times greater load than an equivalent flat element (Chilton & Isler, 2000). Throughout history, the arched structures similar to the Isler's model were made by using materials that are strong in terms of compression but relatively weak in tension. One of the early examples of the arched structures is the Persian monument Taq-i Kiswa which was built in 540 A.D. The structure has 37 meters high at the peak point of the arch which is made out bricks, and quick drying cement spans 26 meters distance (Scarre, 2002). As a better solution than laying bricks, in the Ancient Rome, a cast material which can be considered as an early version of concrete consists of mortar, volcanic sand, water, and little stones is developed (Kleiner & Mamiya, 2005). The Roman Pantheon which is considered as the largest unreinforced concrete dome ever built spans 43.3 meters with a thickness that demonstrates only 2.8% of its diameter. Cotterell and Kamminga (1992) have examined the dome of Pantheon alongside some of the other examples such as Hagia Sophia and Augustan Temple of Mercury and concluded that a hemispherical dome could establish stability if it is thicker than 2.1% of the dome's radius (Cotterell & Kamminga, 1992). The pictures of Taq-i Kiswa which was built in 540 A.D. and the Roman Pantheon which was built in 126 A.D. is shown in Fig. 2.2, respectively.

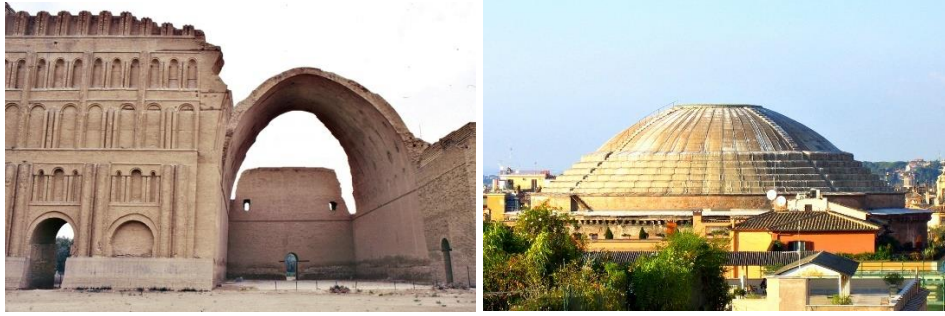


Figure 2.2. The Taq-i Kisra Monument and the Roman Pantheon.

Curved structures, shells, are developed following the introduction of reinforced concrete systems in the 19th century which can be considered as a special kind of material that can work efficiently in both compression and tension. Since the reinforced concrete is a homogeneous material needed to be cast, it allows even distribution of force. Therefore, it makes the transformation of the flat slab plates into curved shell structures possible. Structural shells can be described as a system of uniform or composite structure that defines a supporting system as well as coverage. Due to the structural stability of its geometry and efficient usage of material, shell structures often preferred in architecture. Shells are unique structural elements because of their span to thickness ratio, usually smaller than 500 to 1 (Chilton & Isler, 2000). Shells can be examined in two groups regarding the material usage and the overall appearance mainly named as continuous shells and gridshells. The utilization of the structural members is different in these two systems, in continuous shells, as the name suggests, the continuous envelope also acts as the structural element the loads are diverted on one large surface.

On the other hand, the gridshell system consists of discrete structural elements. Therefore, the paths of the loads are limited. Even though they have similar structural principles as they both carry the loads mainly through axial stress, the utilization of discrete elements makes gridshells more resistant to the buckling. In terms of geometry, both of them can possess double curvature which may result in symmetrical and asymmetrical forms.

2.1. Continuous Shell Structures

Continuous shells are 3D surface-like solid structures which most of the cases possess a specific curvature. The geometry of a shell may have a single or double curvature; this situation creates a significant diversity in form finding. Besides the variety of the forms can be created, the surface curvature of the shell helps to divert gravitational force efficiently to shell's supports via in-plane stresses. This situation allows the creation of lightweight structures which can cover vast distances without the requirement of any additional vertical structural elements.

Continuous shell systems do not consist of vertical and horizontal discrete structural elements; instead, they carry the loads through their geometry. The stiffness which is required to resist the loads is a natural conclusion of their 3D forms. Continuous shells possess membrane action which creates uniformly distributed in-plane stress fields which consist of normal and shear stresses. In order to create a required resistant to the applied loads, the membrane forces such as tension, compression, and shear forces are employed by the shell structures. Because these membrane forces are the only forces when considering shell's structural behavior, loads are not resisted by bending resistance. Therefore, since the thick surfaces and high moments of inertia are out of the concern, thin sections can be employed in shell structures. However, the shells that made of thin sections may suffer out-of-plane buckling regarding structural stability.

2.2. Gridshells Structures

Gridshell systems can be considered as derivations of continuous shell systems. The simple idea was including the linear systems by merging a tessellated surface with a continuous shell structure. The resulted structure is a space-frame structure of connected discrete structural elements. This structure type allows the utilization of materials such as steel or wood in order to coop with pressure and tension forces with the ingenious surface division and connection node design. Milos (2011) described gridshells as *"It was a mix between a grid structure and a continuum shell, an evolution of truss structures into spatially curved grids."* One of the best examples which utilize the merging idea of a grid and a shell is the Torroja's *Fronton Recoletos* which was built in Madrid in 1935 (Lozano-Galant & Payá-Zaforteza, 2011). The structure' roof which consists of both triangular gridshell and concrete shell combined is shown in Fig. 2.3.

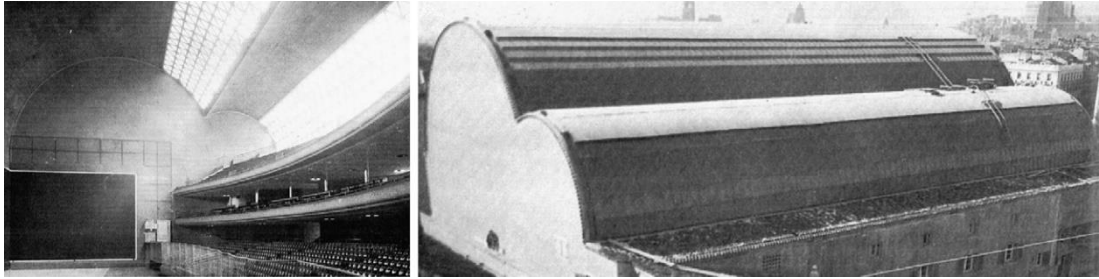


Figure 2.3. Interior and exterior view of the Fronton Recoletos (Lozano-Galant & Paya-Zaforteza, 2011).

As a material choice, iron and steel often used in gridshell systems due to the possibilities of prefabrication. Johann Schwedler makes one of the first successful application of steel gridshell system. *Schwedler Cupola* which was developed by Schwedler was managed to work on span distances of 25-45 meters (Knippers, 2000). Its first utilization was a steel roof element for the gas holder of the *Imperial Continental Gas Association* in 1863. Following the work of Schwedler, Vladimir Shukhov, a Russian engineer was able to build a double-curved gridshell structure which the surface tessellated into quadrangular elements for the roof of exhibition pavilions at the *All-Russia Industrial and Art Exhibition* in 1897. The pictures of *Schwedler Cupola* and *Exhibit Pavilion* during construction are shown in Fig. 2.2, respectively.



Figure 2.4. Schwedler Cupola (Berlin, 1863) and Exhibit Pavilion during construction (Vyksa, 1897).

In terms of shape, a sphere can be considered as a geometric shape that is capable of closing a volume with a minimum surface. Buckminster Fuller, who is considered one of the pioneers in the field of architecture and engineering employed this idea and translated it to the grid shell structures. The spherical gridshell structure which consists of triangulated tessellation is capable of working in large spans made of lightweight structural elements. Fuller (1982) described his vision as; “*Geodesic structures opened up the ability of humans to build unlimited-diameter clear-span spherical structures.*”

*By 1958 I had built a clear-span geodesic hemispherical dome of 117-meters diameter. Since then they have gone to 213 meters in diameter, and they will keep on growing in clear-span size at an ever faster rate until we enclose whole cities". Fuller's work showed that the shape of a structure with the geometric disposition of the assembled elements, has a significant effect on structural design and ingenious prefabrication process can be adopted in order to create light-weighted and stable structures. Fuller's most famous work *The Biosphere* in Montreal, Canada which was built in 1967 and manage to achieve a working span of 76 meters is shown in Fig. 2.3.*



Figure 2.5. The Biosphere by Buckminster Fuller, Montreal, Canada.

The nature of gridshell structure allows to have symmetrical or asymmetrical forms as well as to have double curvature geometry similar to the continuous shell structures. Although the gridshells have similarities with shell systems in terms of formal variety, their structural behavior is different from continuous shells. Gridshells consist of discrete structural members which form the shell geometry. The non-continuous nature of the surface of gridshell systems and the high number of structural elements often results with structural complexity because of the high amount of time which needed to conclude the structural calculations. This is one of the main reasons that there are significantly fewer gridshell examples than continuous gridshells that have been built. However, because of the efficient material usage and the utilization possibility of in-between spaces between the structural members in different ways, gridshell structures can contribute to the economic and formal aspects of many different design problems. Gridshell structures consist of discrete structural members as well as joint elements on connection points. Different purposed panels can be placed on the intermediate spaces between the structural elements. According to their utilization, the in-between spaces can be transformed into opaque, transparent or translucent surfaces. The grid layout of

the structural elements directly affects the panel shapes, structural behavior of the structure and aesthetical aspects of the structure. Concerning their structural complexity, gridshell structures make good cases for *Finite Element Analysis* in which the values and equations such as the displacement, stress and assembled elements stability can be calculated (Vejrum, 2013).

2.3. The Curvature of the Shell Surface

One of the most important geometrical features of the shell systems is they can possess double-curvature through their geometries. According to (Stevens, 1981) and others, there are two principal curvatures named as K_1 and K_2 that can be used in order to create the local shape of a surface. These represent the maximum and minimum local curvatures of the planar curvatures which are created by employing the intersections of the relative surface and normal planes. If none of the curvatures in a surface is equal to zero, the resulting surface is considered as it possesses a double-curvature. In Tab 2.2, the six possible surface shape classes and the relationship with the two curvature categories are shown.

Table 2.1. Classification of the surfaces by their shapes.

	$K_1 < 0$	$K_1 = 0$	$K_1 > 0$
$K_2 < 0$	Ellipsoid	Cylinder	Hyperboloid Surface
$K_2 = 0$	Cylinder	Plane	Cylinder
$K_2 > 0$	Hyperboloid Surface	Cylinder	Ellipsoid

The Gaussian curvature is a result of these two principal curvatures $K_G = K_1 \cdot K_2$ (Hoefakker, 1900). When one of the main curvatures is equal to zero $K_G = K_1$ the resulting surface will be a cylindrical surface. When the two main curvatures are negative to each other ($K_G < 0$) the resulting surface will be a hyperboloid surface and the negative gaussian curvature is also called anticlastic. When the two main curvatures are both positive ($K_G > 0$) the resulting surface will be an ellipsoid and the positive gaussian curvature is also called synclastic.

In Fig 2.8, the three types of surfaces that have different gaussian curvatures cylindrical, hyperboloid and ellipsoid are shown respectively.

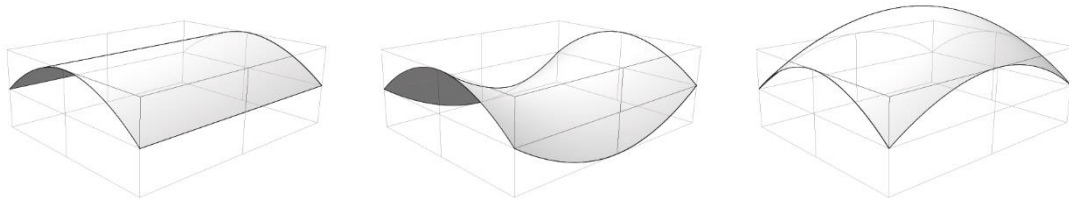


Figure 2.6. Gaussian curvature examples.

2.4. Structural Behavior

Structural elements which have a significantly greater dimension than their other two dimensions (also can be named as linear structural elements such as cables, structural beams, and arches) or structural elements which have a significantly smaller dimensions than the other two (such as surfaces) have the same principle; they tend to be bent easily rather than be stretched. If an amount of tension is applied to a cable, the cable will stretch as the opposite way of the stretching (negative stretch since its length change in a negative way) of a column under an amount of compression. However, if the amount of applied load is increased, the column will buckle, and its length decrease through bending in contrast to axial strain.

A cable is capable of carrying a uniformly distributed vertical load per unit length by using only axial compression. The normal vector of the load on the cable is balanced by the axial force multiplied by the curvature ($1/\text{radius of the curvature}$). Also, other loads will result with deflection of the cable.

Since a continuous shell consists of a continuous material all along the structure, the loads are distributed as in-plane normal as well as the shear stress distributed equally on shell's structural material. This structural aspect effects the elements' behavior, they act as if they are locked in their places (Toussaint, 2007). The gridshell, on the other hand, is a system that consists of structural elements that are connected in joints. Due to the existence of the separate linear structural elements in the system, the elements transmit the forces in a relationship with the elements' directionality and gain resistance to the out of plane bending. The bending stiffness of an element in a gridshell prevents any inconvenience in the membrane behavior. In order to transmit the shear forces between the elements and lock the system, diagonal stiffness is required. In

order to achieve diagonal stiffness, linking structural elements with rigid connections, creating an external layer over the members, employing cross cables or bracing could be the possible ways.

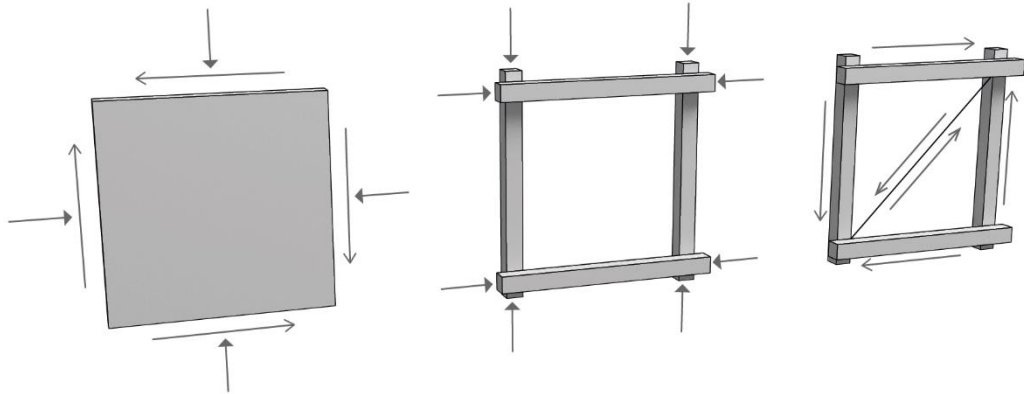


Figure 2.7. Structural behaviors of shell and gridshell elements.

In Fig. 2.3, some of the structural behaviors of the shell and gridshell elements are shown. A continuous structural shell element which is shown on the left is able to carry the applied loads through the equally distributed in-plane stress fields over its thickness. When there is no diagonal reinforcement on a gridshell element which can be seen in the middle of Fig. 2.3 the elements' resistance to the applied loads through axial forces and out-of-plane bending resulted in higher deformation than a continuous shell element. Lastly, the representation on the right shows that when a diagonal reinforcement applied to a gridshell system, the shear forces can be transmitted between the edges of the structural elements.

2.4.1. Flat Plates and Plane Stress

The applied forces on a flat plate can be examined in two main categories; forces on its own plane (in-plane stress) and forces out of its plane as shown in Fig. 2.3. The concept of “plane stress” includes in-plane loading that presents itself in most of the situations such as the bending of an I-beam. Even though the loading of the beam is considered perpendicular to its relative axis, the significant part of the stress appears in the web and flanges are in fact in the plane of the steel plates. Out of plane loading of a plate cause the plate to be bent.

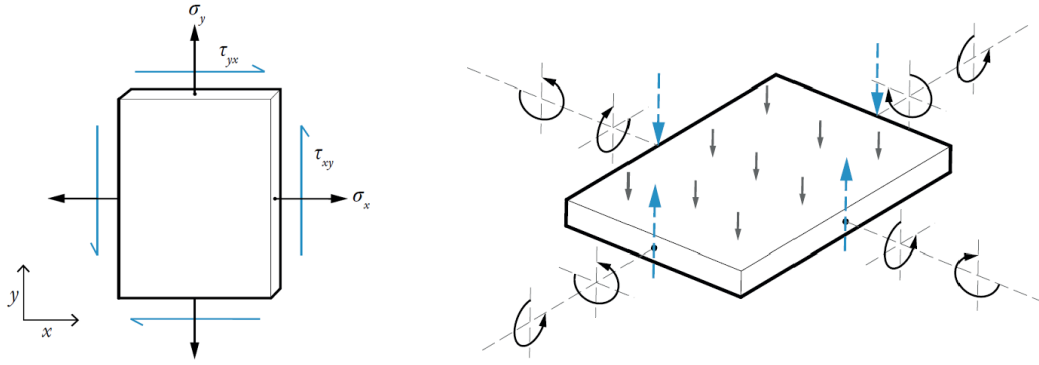


Figure 2.8. Plane stress and plate bending (Williams, 2014).

In Fig. 2.3, the elements of membrane stress are shown which are normal stress in the x-direction and y-direction σ_x and σ_y , respectively. In addition, shear stress perpendicular to the X-Axis in the Y-Axis is shown as τ_{xy} whereas the shear stress perpendicular to the Y-Axis in the X-axis is shown as τ_{yx} . In-plane membrane stress which can be correlated with the axial stress that appears in an arch in contrast to the bending stress. Membrane stress is often defined as the force effect on per unit length of an imaginary cut of a surface instead of force per unit area of the surface. According to the equilibrium of moments, one may state that $\tau_{xy} = \tau_{yx}$. Therefore, by using σ_x , σ_y and $\tau_{xy} = \tau_{yx}$ following equations of equilibrium, Eq. 2.1 and Eq. 2.2 can be derived in the X-Axis and Y-Axis, respectively.

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} = q_x \quad (2.1)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} = q_y \quad (2.2)$$

The applied loads on per unit area of the relative plates are notated as q_x and q_y , respectively. Self-weight of the structure can be given as an example to these loads. The plane stress can be considered as statically indeterminate since there are three unknown stress whereas two equations of equilibrium. If q_x and q_y are both considered as zero, the relative stress can be translated into an airy stress function (ϕ) in order to satisfy the equilibrium equations as follows:

$$\sigma_x = \frac{\partial^2 \phi}{\partial y^2} \quad (2.3)$$

$$\sigma_y = \frac{\partial^2 \phi}{\partial x^2} \quad (2.4)$$

$$\tau_{xy} = \tau_{yx} = -\frac{\partial^2 \phi}{\partial x \partial y} \quad (2.5)$$

In cases which include elastic plates, the stresses can be solved by employing the stress-strain relationships which includes Young's modulus (E) and Poisson's ratio (ν) with compatibility equation. As a result of two components of displacement relative to the X-Axis and Y-Axis, strains will occur. After the elimination of strains by differentiating them twice and subtracting, a biharmonic equation can be found (Williams, 2014). The equation behaves well and not difficult to solve according to Timoshenko and Goodier (Timoshenko & Goodier, 1970).

2.4.2. Bending Theory and Buckling

In a case of deformation, a structural shell relies on bending stiffness to carry the relative loads alongside the membrane action. To have a bending stiffness contributes significantly to prevent buckling in the cases which include the compressive membrane stresses. Since the shells are very efficient regarding their structural performance, shell buckling is a crucial factor which should be considered carefully because it can cause a total collapse even if there are no visible deflections. The behavior of bending action can be considered as in a contrast relationship with the shell's efficiency in terms of form, tessellation and the support. In other words, if a shell's efficiency decreases, it starts to behave better in buckling since bending action requires more deflection than membrane action.

Williams (2014) stated that; *“Experiments show that a properly supported shell working primarily by membrane action can never support anything like the theoretical ‘eigenvalue’ buckling load or ‘linear’ buckling load even when the utmost care is taken to eliminate initial imperfections.”* Unfortunately, the creation of the analysis concerning the shell buckling by hand calculations is extremely hard, event eigenvalue analysis for a spherical shell is complicated. This situation requires the translation of these analyses into the computer environment, however, since this is a relatively new approach, the results should be treated with caution.

2.5. Conclusion and Design Goals

In this chapter, the two structural shell types their features and the structural behaviors are presented. According to the current literature, structural shell systems can be categorized into two by considering their appearance and different material and structural member utilization. In continuous shell systems, the structure's envelope also has the load-bearing characteristic. The loads are carried by one large surface that also creates the general appearance of the structure. Alternatively, gridshell systems consist of multiple discrete structural elements and connection nodes. The tessellation pattern on the gridshell geometry defines the way that the discrete structural members connected. In addition, the geometrical characteristics of a tessellation pattern affect the appearance of the structure as well as the structural performance. Both of the shell systems have the same principle in terms of diverting the applied loads, they carry them through axial stress. However, the utilization of discrete elements in gridshell, increase the structure's resistance to out of plane bending. Furthermore, a shell's ability to possess double curvature in its geometry leads to membrane action that effectively diverts the forces to shell's supports via in-plane stresses.

In light of the reviewed literature in this chapter, design goals for the computational gridshell design method are considered as:

- The method should be able to deal with complex gridshell geometry and make it easier to control.
- The method should contain different tessellation types related to the gridshell's surface to create a variation in the generated alternative solutions.
- Since the shell systems behave very well when possessing double curvature in their geometry, the method should be able to generate gridshell alternatives by employing this principle.

CHAPTER 3 FORM FINDING METHODS AND STRUCTURAL ANALYSIS

Form-finding of a structure is one of the most critical decision-making processes since it is directly related to the decision of the overall shape. In the case of gridshell structures, the decision process is different from most of the architectural design processes because the structural elements are also the elements that create an overall appearance. Rather than designing a structural system after the overall architecture is decided, in a gridshell design process, the appearance and structure of the system should be decided simultaneously. Therefore, the architectural decision variables and the structural requirements of a gridshell must be considered as the same elements that can fulfill both. The primary purpose of the form-finding process is to find a geometry which can be efficiently carry required loads as well as satisfy the architectural considerations. The form-finding process is also essential because of the economic reasons since the amount of the material required to create the gridshell is the direct result of this process.

3.1. Hooke's Hanging Chain Law

One of the primary aspects of shell design is to find a form which is capable of carrying the required loads in axial compression without bending outside the optimum margins. Robert Hooke who is an engineer and scientist published one of the first examples of structural form finding of an arch. In 1676, Hooke published his ideas which one of them is the most known *Hooke's Law of Elasticity*. The idea behind this invention was very straightforward; if a chain is hanged horizontally within a shorter distance of its length, it will collapse and behave in pure tension with no bending. The equivalent negative of this chain's form is equal to an arch that works on pure compression. A chain that made of homogenous material would have a constant weight per unit length, and because of this weight, the chain deforms under its self-weight creates a catenary. However, the form of an arch greatly depends on the amplitude of the applied loads. If the load distribution on the horizontal plane is uniform, the arch tends to take a form that is similar to a parabola. Also, the span to rise ratio is not a constant and can be changed in a range of 2 to 10 (Adriaenssens et al., 2014) as most of the shell structures have. Because of this situation, the formal exploration of even a simple arch that acts

on a pure compression because of its weight has many possibilities due to the load distribution and the rise of the arch.

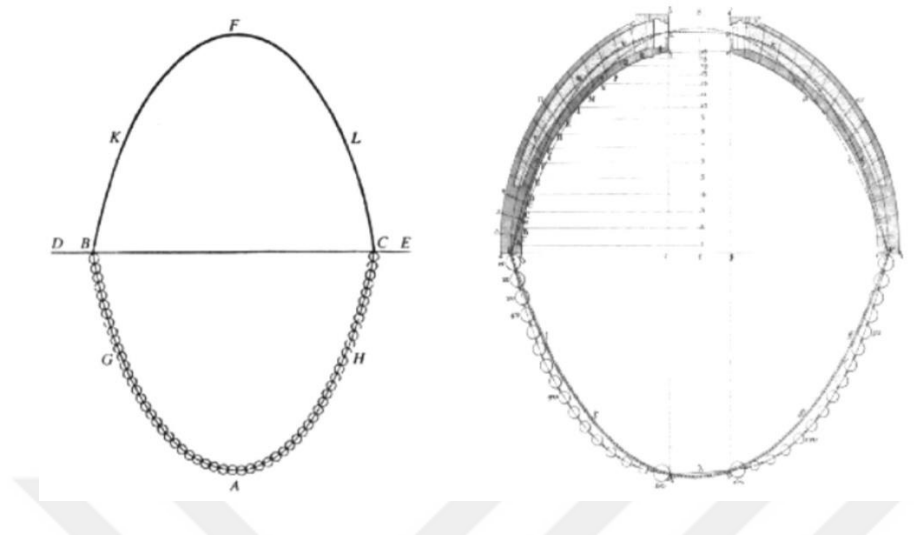


Figure 3.1. Poleni's drawing of Hooke's analogy between an arch and a hanging chain, and his analysis of the Dome of St. Peter's in Rome (1748).

Hooke's Law can be considered for shell structures concerning different geometrical approaches rather than a single arch. Shell systems are significantly more complicated than a single arch because of the possibilities of multiple load directions which may introduce bending. In order to apply Hooke's principles to a shell system, a 3D chain model which consists of intersecting catenaries in specific support points could be created. The model could also be converted as a discrete shell design that its surface elements are continuously shaped according to the connection nodes. By adjusting each chain's length that creates this network of chains which acts in tension, a variety of form derived from self-weight of the catenaries can be found. The inversion of this networks' form represents a discrete compression only shell form.

3.2. Physical Modeling

The use of physical models in order to solve a specific design problem is an old and relatively effective method which designers still could rely on today (Addis, 2014). Throughout history, many designers have been using this procedure in many different design problems to create even the most complex forms with simple geometries such as circles, squares or triangles. Even though the decision on a specific form could be a conclusion of a unique design method that greatly depends on the architect's preferences, creation process of an architectural form brings similar issues such as its

projection on the plan, the required height, the number of loads, structural effectivity and the appearance of the final shape.

Until the discovery of the computational computer programs, the structural analysis of the shell structures was based on mathematical equations and models that structural engineers could create manually. This situation has affected the shell design methods throughout history since it is easy to calculate required dimensions when the simpler forms employed such as spheres, cones, ellipsoids and so on. Creation of small-scale physical models alongside the mathematical models of the structures helps to both the engineers and the architects to prove the structure's viability. In addition, rather than building a full-sized prototype, building a small-scale model is cost effective and may give an idea to the structural behavior of the full-sized structure when it is constructed. However, for some structures, this is not the case. Therefore, the physical models can be considered in two categories in terms of structural behavior which are independent on the scale and dependent of scale.

3.2.1. Scale-Independent Modeling

There are cases that structural behavior can be scaled up linearly in order to give an idea about how the full-size structure will behave. Catenary models with chains and nets of funicular arches and vaults and models of the compression structures such as masonry arches and vaults can be considered as examples to these cases. Using small-scale models to predict these structural behaviors are often reliable.

In a hanging chain model, the applied forces can be considered as in pure tension with no bending movement. Simon Stevin who is a civil engineer developed forces' mathematical representation using vectors (Stevin, 1586). In his works, he created different examples by attaching weights on a string in order to create 2D and 3D funicular shapes, in Fig 3.2 the funicular form diagrams of Stevin are shown.

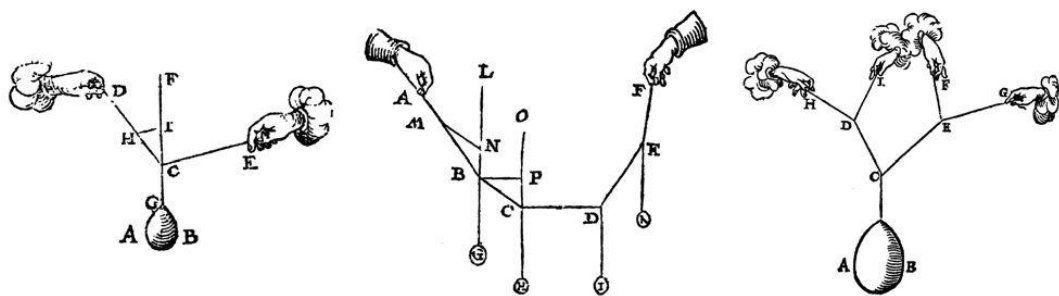


Figure 3.2. Stevin's funicular form diagrams (1586).

Robert Hooke used his approach to the inverted catenary during the design phase of St Paul's Cathedral in London alongside Christopher Wren. When designing the dome of the cathedral, they employed Hooke's law of inversion in order to help the structural calculations and design. For designing the dome which has a 33m diameter, Wren used a simple chain model which can be seen on his cross-section sketches of the building in Fig 3.3. This is not only raised the Hooke's and Wren's confidence to the structure but also it is one of the early examples of employing a physical model for a compression structure in order to determine its final shape of the structure.

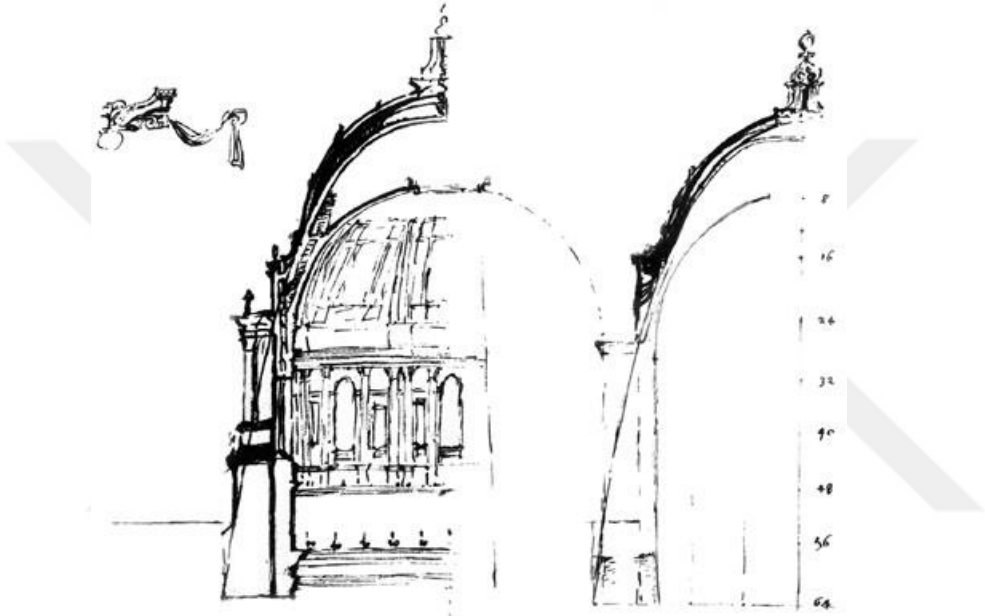


Figure 3.3. Proposed section of St. Paul's Cathedral.

The work demonstrated a catenary arch on top of the masonry in order to achieve the structural stability for the dome. In this work, a uniform chain with weights is used to represent the structure consists of the same sized elements. The chain models that were used in the design were 2D since there is no evidence to support a 3D chain model was used in the process.

The well-known Catalan architect Antoni Gaudi used the hanging chain models often during the design phase of his works. The models that he created were both 2D and 3D hanging chain models made of strings and different kind of weight. Also, the idea of using 3D hanging chain models to represent a space most likely developed by Gaudi (Huerta, 2006). These models helped him to decide the final shape of the arches and the vaults that he included his several masonry building designs. One of the examples to his work is the Crypt of Colonia Guell (Tomlow, 2011), in which Gaudi use a rather

complex hanging chain model to create the final forms of the inclined columns and large vaults as well as testing the structural behavior in the design process. A photograph of the Gaudi's 3D hanging chain model reproduction for the Crypt of Colonia Guell is shown in Fig. 3.4.



Figure 3.4. Gaudi's hanging chain model reproduction (Asmaljee, 2014).

3.2.2. Scale-Dependent Modeling

The scale-dependent models to represent the structural behavior can be considered as two groups which are: the models that are created to examine the shells' elastic behavior under the loads can cause the out-of-plane bending and the models to demonstrate and examine the buckling failure because of the high in-plane stresses.

Jørn Utzon proposed a building design for the Sydney Opera house that includes a roof made of thin concrete shells which have a rather unusual geometry to represent the sails of yachts. In the process of formal exploration for this shell roof, engineers Ove Arup & Partners experimented by employing a variety of different shapes in order to achieve structural stability by using of a 1:60 scale model of the building which can be seen in Fig 3.5.

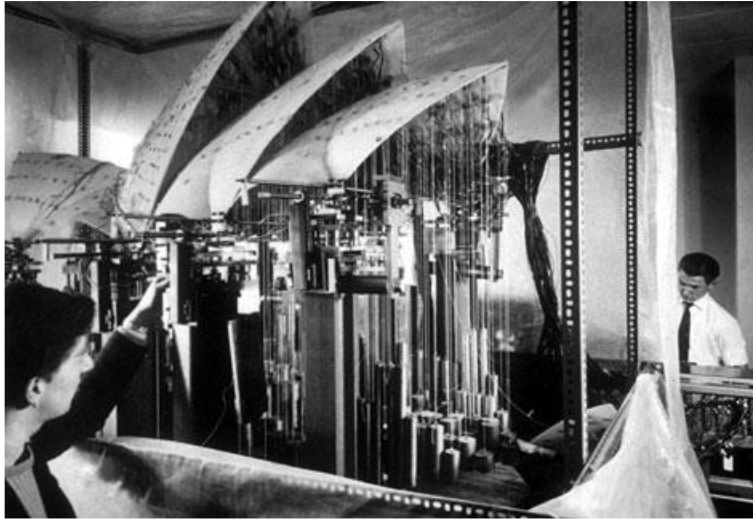


Figure 3.5. 1:60 scale model of Sydney Opera House's thin-shell roof.

The models proved that the proposed roof design could not carry the acting loads due to insufficient stiffness and strength of the thin shell concrete elements. After a six-year design process that many alternative forms are considered during this time to achieve structural stability, a decision was made to use a spherical surface which can simplify the construction process of the pre-cast concrete ribbed arch elements and ceramic-tile application on top of the concrete structure.

During the design phase of the Bundesgartenschau Multihalle in Mannheim, the small-scale models have also used in order to help the decision-making process. The challenge was to cover approximately 7.400 sqm irregular shaped area by using a post-formed timber gridshell structural system. The proposed gridshell system consisted of timber structural members, and the total structure was 160 m long with the largest span of 70 m x 60 m. Since there is not an example of a structure with this size and complexity before, physical models were used in order to discover the ideal shape of the structure, alongside the manual and computer-aided structural analysis process (Happold & Liddell, 1975). During the design process, several different structural models were created to decide the overall shape, to evaluate the relative grid regarding stress distribution and the best way to construct the gridshell structure (Bächer & Burkhardt, 1978). At the beginning of the design phase, a 1:300 scale structural model was created by using wires to evaluate the basic shape of the structure that includes two main halls and a tunnel in between them. Following this process, a hanging chain model with 1:98.9 scale was built to help the decision-making process of shaping the boundary support elements. The network of wires that consists of 15mm long strings

was suspended from 80 support points to make final adjustments in order to create the ideal funicular form. In Fig 3.6, the 1:300 scale wire mesh model and the 1:98.9 scale 3D hanging chain model were shown.

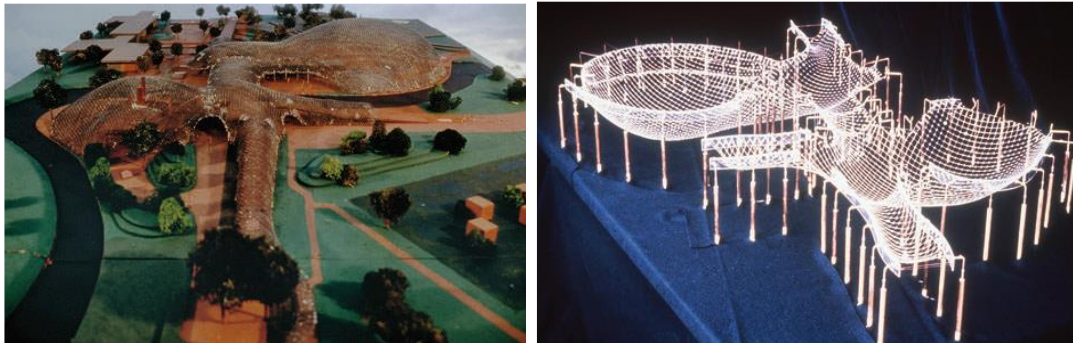


Figure 3.6. 1:300 scale and 1:98.9 scale models of the Multihalle.

3.3. Computational Form Finding

Today, due to significant developments in computer technologies, employing computer-aided methods in order to determine a structure's overall form is possible. The increased interest in using computational design tools in the architectural field, the formal exploration process can be richened (Fleischmann et al., 2012). Moreover, by using computational tools, the form finding process of a structure can be shortened dramatically in comparison to the creation of actual physical models alongside the development of relative calculations. In addition to that, the required calculations to find a structurally stable form which has been done manually by the engineers for decades can be translated into the computer environment to create digital computational models and simulations.

Computational form finding process includes the variables that can be controlled via computer programs in order to achieve a form that can be considered as in statically equilibrium. The output geometry is a direct result of the required calculations. There are different form-finding methods available which can be translated into a computational design environment. All of these methods are different theoretical approaches which have different usability, required computation time and complexity. The required parameters for these methods are similar such as the boundary conditions, support or anchor definition, axial loads, forces and their interaction with the geometry. As (Cevizci & Kutucu, 2017) pointed out form-finding methods can be discussed in two separate groups which are geometry-oriented form-finding methods and material-

oriented form-finding methods. The main difference between these two types of methods is in geometry-oriented form-finding methods such as *Force Density Method*; material properties are not considered. The structural equilibrium is solved by considering the geometrical features of the given problem. On the other hand, in material-oriented form-finding methods such as *Dynamic Relaxation Method*, the material properties translated into spring stiffness in order to solve dynamic equilibrium problems.

3.3.1. Force Density Method

The Force Density Method (FDM) is a well-known form finding method which often used in the engineering field, first introduced by Schek (Schek, 1974). FDM is developed in response to the computational modeling of the structures for the Munich Olympic complex (Lewis, 2003). Lewis (2003) defined the operation of the FDM as “... uses a linear system of equations to a model static equilibrium of a pre-tensioned cable net under prescribed force/length ratios.” The method mainly used for finding the equilibrium shape of a structure that contains a network of cables with different elasticity properties against applied stress (Southern, 2011). In addition, the formal exploration process of a cable network can be defined as the determination of the architectural form (Gidak & Fresl, 2012). Therefore, a form that is the result of FDM is not only an aesthetic concept but also can satisfy the engineering aspects such as load transfer and structural performance requirements.

FDM is developed to prevent the problems related to the computerization of inverse hanging chain approach. The form-finding process of the method consists of two parts. The first part includes the creation of the physical model of the geometry by using materials such as soap, elastic fabrics, and threads in a relationship with the geometry's boundary conditions. For the second part, following the creation of the shape which is aesthetically determined, a numerical model is developed.

Some of the properties of FDM which pointed out by Southern (2011) are; “... depending only on the force density of the edges and the topology of the network, and the system is sparse, symmetric and positive definite, quickly solved using the conjugate gradient method”. Besides, FDM does not require any material information. The non-materialized equilibrium shapes are independent of the existed material laws (Linkwitz, 2014). Therefore, the decisions about the material can be defined after the

determination of the equilibrium shape. According to Linkwitz (2014) the two important potentials of an unlimited possibility of material properties are; *“First, resulting design can be materialized arbitrary, giving the initial lengths of the network in the un-deformed state without affecting the final shape. Second, one can multiply the loads to any realistic value and then calculate the internal force distribution, again without changing geometry”*.

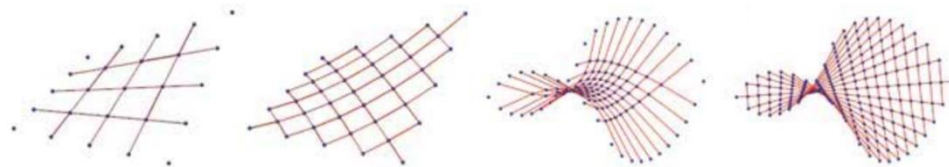


Figure 3.7. Cable Net Examples by Fresl and Vrancic (2013).

Even though FDM was introduced many years ago, the method still in use and one of the preferred methods in the cases that require form-finding and calculation of the equilibrium state of tensile structures (Cevizci & Kutucu, 2017).

3.3.2. Dynamic Relaxation Method

The Dynamic Relaxation Method (DRM) which is invented by Alistair Day is a tool for examining the behavior of geometrically variable systems. DRM is an iterative process that can be included in the structural form-finding process and to achieve a static solution (Hüttner et al., 2014). The method is a representation of structural elements as a particle-spring system. This system consists of a grid which is a representation of spring elements connected by nodes. The mass of these elements is also considered as a force on these nodes. When a specific force applied to these nodes, it causes a movement on the system until it finds the equilibrium. After the nodal movement of the system which is also called as damping is finished, the equilibrium state of the system is achieved. The dynamic relaxation method is derived from Newton's second law of motion; it is a step by step solution for a small-time increment. In each time frames, the axial movement of the nodes due to the applied force is calculated, and nodes move accordingly for a specific period.

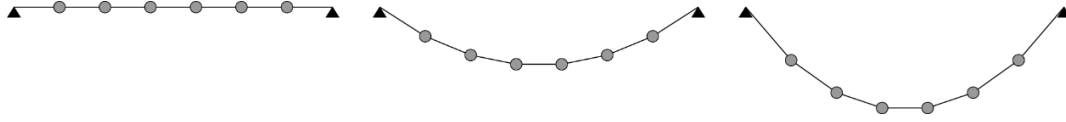


Figure 3.8. A catenary model, deformation caused by the applied loads to the nodes.

According to Newton’s first law of motion, an object will accelerate when there is an unbalanced force acting upon it which will result in changing this object’s speed, direction or both. In addition to that, Newton’s second law of motion states that the acceleration depends on two variables which are the sum of the acting forces on an object and object’s mass.

$$Force (F) = Mass (m) \times Acceleration (a) \quad (3.1)$$

For the forces and movement on the X- direction for a given time, the law can be translated as

$$F_x(t) = m \times a_x(t) \quad (3.2)$$

The translation of the equation for Y- and Z- directions which are used in Dynamic Relaxation are on the same logic. Besides, Hooke’s law stated that the deformation of a spring is proportional to the force that is required to move a spring. The ratio between the applied force and caused deformation is a constant ratio and defined as the spring stiffness.

In other words, the dynamic relaxation method includes the movement of intersection nodes of the system which is also defined as springs, throughout the small-time increments. In addition, all grid lines of the system are assigned values for their axial stiffness (EA) and bending stiffness (EI) where E is Young’s modulus, A is the area of their cross-section, and I is the second-moment area. The applied gravity loads on springs will cause motion on -Z direction and create a hanging tension form. If a negative gravity load is applied, as the Hooke’s law suggests, the spring movement will be upwards which results with a form that works on compression.

Dynamic Relaxation Method is mainly used to find the form of cable and fabric structure. However, as Garcia mentioned DRM could be defined as “... a numerical method that can be utilized in the form-finding of all kind of structures that consists in considering that the mass of the system is discretized and lumped in the nodes; these nodes oscillate about the equilibrium position, and by introducing artificial inertia and

damping, the nodes come to rest in the static equilibrium position” (Garcia, 2011). Therefore, the method can be considered as a valid approach that can be included in the form finding process of a shell structure.

3.3.3. Comparison of the Form Finding Methods

Thanks to the technological developments in the last few decades, the design of the complex geometries and the relative calculation of their static equilibrium shapes can be translated into the digital environment. Therefore, In the field of architecture, computational interactive form-finding methods can be included to the formal design process of the shell structures in order to evaluate the result of the design decisions concerning appearance and structural stability. One of the most critical decisions in the formal exploration process of a gridshell structure is to determine a suitable method for form generation process by considering the parameters and decision variables such as structural properties and materials.

In this chapter, the two form-finding methods which are mentioned above are compared with each other in order to discuss two questions;

- What are the main differences between these methods?
- Are they suitable for all types of structural problems regarding structural approach and material properties?

There are a limited number of researches that include the form-finding methods of the shell structures. Books and articles that are prepared by Veenendaal and Block are reviewed in order to make a comparison.

In Fig. 3.9 the categorization of the form-finding methods are shown with respect to the related time-period by Veenendaal and Block (2012). As Fig. X. shows, in the works of Veenendaal and Block (2012), the force density method has been defined as a geometric stiffness method whereas Dynamic Relaxation Method has been defined as a dynamic equilibrium method.

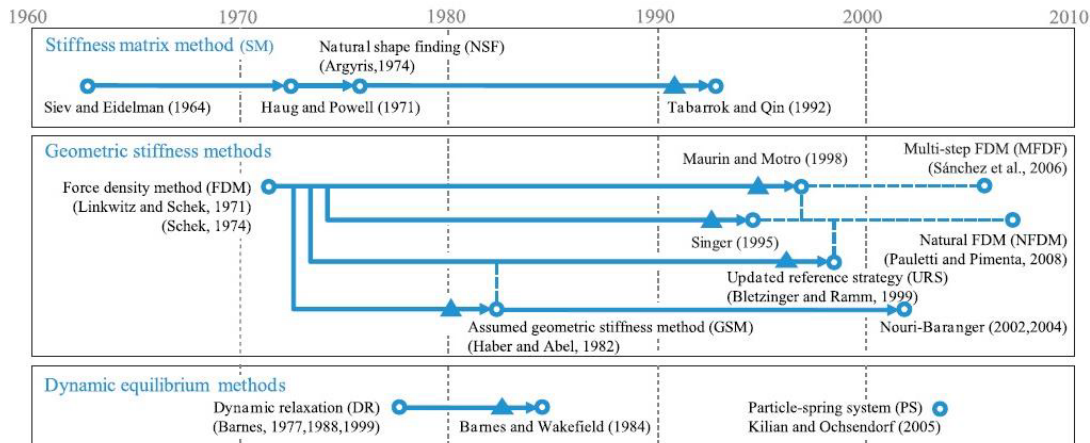


Figure 3.9.The categorization of form-finding methods (Diederik Veenendaal & Philippe Block, 2012).

Geometric stiffness methods are considered material-independent methods that include geometric stiffness (Block & Veenendaal, 2014). Furthermore, Block and Veenendaal (2014) state that “*In several cases, starting with the Force Density Method, the ratio of force to length is a central unit in the mathematics*”. On the other hand, dynamic stiffness methods are considered as a validation of the shape equilibrium in order to achieve a stable state related to the static equilibrium. Dynamic Relaxation Method is categorized under this group.

Veenendaal and Block (2014) employed both FDM and DRM as a part of their comparison study and applied them to the same data structure which is resulted in a shape of a simple shell. According to the method of their study, the minimum requirements of each form-finding methods are;

- A discretization to represent the geometry of the shell. The relative discretization process can include the line elements or surface tessellation elements such as triangles and quadrilaterals.
- A data tree consists of the information which describes the shape, connection of the relative discrete elements and relative forces to the shell.
- Equations of equilibrium that describe the correlation between the internal and external forces. Since the result of the addition of the internal and external forces should be equal to zero, a resulting shape of the form-finding process can be considered as in the state of static equilibrium.

- A solver that defines the method of solving the equilibrium equations. In the cases which the system of equations is considered nonlinear, the related solver employs an incremental approach to solve the system. In order to evaluate the system's convergence, the solver contains data which defines the stopping criteria. Solving methods may differ from each other regarding the time which requires to converge and the working stability. However, assuming that they are capable of converge eventually, the results should be the same for the same problem considering that the problem and the boundary conditions are alike.

In their works, Veenendaal and Block (2014) defines the data about the properties which should be provided as;

1. Coordinates that describe the location of the support elements in 3D space,
2. Surface topology and the connection nodes of the networks,
3. Related loads,
4. A threshold for convergence in iterative approaches.

Following the conducted experiments and applications by Veenendaal and Block (2014), the input for the Force Density Method was reduced to a bare minimum. This situation may be considered as an advantage since force densities are not eloquent concerning physics; therefore, it is not easy to control. Nevertheless, the situation is different for dynamic methods as the drawback of these methods is stated as “... *Dynamic methods such as Dynamic Relaxation are the much larger number of parameters necessary for their control. However, in DRM the parameters such as stiffness, bending stiffness, initial coordinates, or lengths are either fictitious values, chosen for their effect on convergence or on the resulting shape, or they are related to the material and physical properties of the structure*”. The quantities which should be defined by the user are shown for each method in Tab. 3.1.

Table 3.1. The needed values to be defined by the user for each method (Veenendaal and Block, 2014).

Method	User-defined quantities
FDM	Force densities
DRM	Axial stiffness
	Bending stiffness (for splines)
	Initial coordinates, or lengths
	Damping factor (for viscous damping)
	Time step

It is stated that, following the achievement of an equilibrium state, material or physical properties can be changed without causing any disturbance to the shape of equilibrium. Veenendaal and Block (2014) mentioned this as; “... *combined with the ability to manipulate the internal forces (through force density, elastic stiffness or spring stiffness, as well as loading), suggests that these methods are theoretically interchangeable*”.

Moreover, for the problems, in which a shape of equilibrium that works mainly on compression can be accepted as a solution, the utilization of geometric methods such as FDM can achieve a solution quickly. However, in cases that initial shape and deformation are eloquent and the relative material properties are known, the *Dynamic Relaxation Method* is more precise and more suitable. Besides, Cevizci and Kutucu (2017) stated that “*The bare integration schemes in Dynamic Relaxation Method do not need matrix algebra, which may be an advantage concerning a simple implementation.*”

3.4. Finite Element Method and Computational Structural Analysis

Following the implementation of scientific decisions to the design phase of a structure, the design practice has been considerably developed regarding the methodology. In ancient times, the builders did not require any computation or theory related to the construction of a building. Although there are a variety of historical structures that were managed to exist on this day, master builders, architects and engineers have developed different methods to examine and understand the nature of the structures. Thanks to the experience which has gained throughout the centuries, the improved understanding of a building in terms of architecture and engineering create a possibility to build more complex structures. The methods which are discovered to analyze the structural properties and material properties of a building played a significant role in the development of building techniques.

Kaveh (2014) defines structural design and analysis as; “... *the determination of the response of a structure to external effects such as loading, temperature changes, and support settlements. Structural design is the selection of a suitable arrangement of members, and a selection of materials and member sections, to withstand the stress resultants (internal forces) by a specified set of loads and satisfy the stress and displacement constraints, and other requirements specified by the utilized code of practice*”. He defined structural decision-making process as a cycle which requires an iterative application of structural analysis and design in order to find a suitable solution which satisfies the design objectives such as the weight of the structure or the construction cost. In Fig. 3.10, the cycle of structural analysis and design which is defined by Kaveh (2014) is shown.

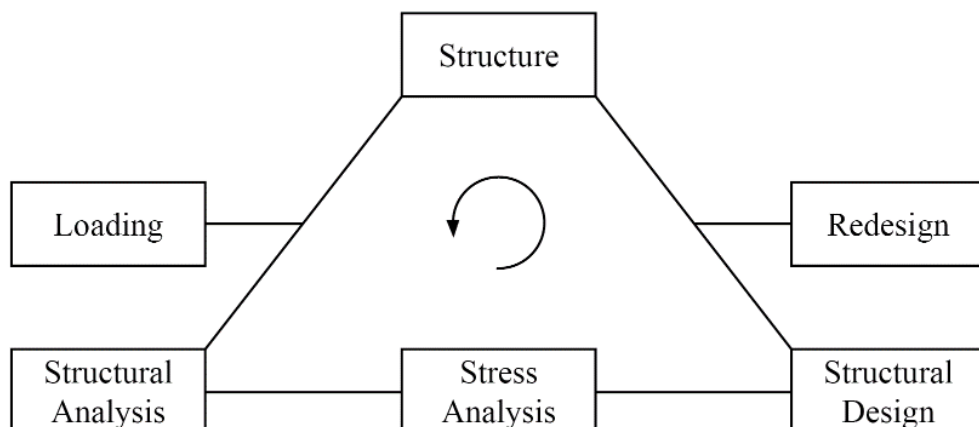


Figure 3.10.The Cycle of Structural Analysis and Design of a Structure (Kaveh, 2014)

The structural equilibrium state is defined as the state in which the external loads applied to the structure as well as the internal loads are in equilibrium for each node that is subjected to the analysis. In order to achieve this state of equilibrium, the nodes must be deformed to fit each other within a specific range since the internal loads and deformation should satisfy the correlation of stress and deformation of the nodes. There are two complementary matrix methods of formulation which can be employed for any structural problem named as *Displacement Method* and *Force Method* (Przemieniecki, 1985). In both methods, the structural analysis is considered as a systematic combination of each structural elements into an assembled system that satisfies the objectives of equilibrium and compatibility.

In the *Displacement Method* also known as the *Stiffness Method*, the displacements of the connection nodes which describes the structure's deformed state are considered as unknowns. The calculations related to the nodal deformations contains the movement of the nodes. In order to include the internal forces to the calculations, the correlation between stress and strain is taken into account. In the end, by solving the linear equations which concern about finding the equilibrium state of each node, solutions for unknown displacements are achieved.

On the other hand, in the *Force Method* also known as the *Flexibility Method*, as the name suggests, the forces are chosen as unknowns. The deformations of the discretized members, as the result of external and internal forces, are defined considering the stress-strain relations. In order to calculate the unknown forces, the linear equations are taken into account by considering the suitable conditions in which the deformed members can be fitted together.

The finite element procedure has its origin in the work of Turner (1956). In Turner's work, the difference from the previous concepts was the establishment of arbitrarily shaped elements as pieces of the original domain. The process considered a specific stress distribution for each element in terms of nodal force equilibriums. The fundamental concepts of the *Finite Element Method*, as it is known today, were established in the approaches in structural extension and classifications of the matrix method suggested by Argyris (1960) (Zienkiewicz, 2000). By mid-1960s, most of the fundamental ideas of permitted element forms for self-adjoint problems were included in the literature, and finite element procedure was fully established (Zienkiewicz & Cheung, 1967) and it was defined as "*a method of analysis for highly redundant*

structures which is particularly suited to the use of high-speed digital computing machines”.

Finite Element Method (FEM) categorizes a set of techniques that share similar approaches in the field of engineering (Owen & Feng, 2012). Following the translation of this method to the computers, structural analysis in which include the computational power of the computers can be done. Pedron (2006) evaluates the performance of the FEM when translated to the digital environment as “... *non-trivial calculations concerning dynamics, collapse mechanisms, materials, and geometrical non-linearity as well as ultimate loads could also be routinely performed*”.

FEM is able to coop with structural problems that concern the equilibrium equation which include the loads and the nodal displacements. *Finite Element Method* can be used to analyze the physical systems and also known as *Finite Element Analysis*. Following the development of FEM, Pedron (2006) states the impact of this method to the structural analysis field as “*Until the mid-20th century, despite the use of simplified calculation methods like the force method, the displacement method, and the Hardy-Cross method, it took a long time to analyze structures even of medium complexity, mainly due to the difficulty of solving linear equation systems. In the late 1950s, the advent of computers and the development of the Finite Element Method completely revolutionized structural analyses*”. The computer programs which include FEM such as Karamba can be found easily (Preisinger & Heimrath, 2014). The programs can calculate the internal forces and stress on a specific digital computer model following the definition of external forces by the user. Until the satisfied criteria are achieved, the parameters of the model can be changed.

Gridshell structures make good cases for *Finite Element Analysis (FEA)* due to their complex structural nature. FEA can be employed in order to calculate nodal displacements, stress and the stability of the structural elements. Gridshells often consist of constrained joints rather than pin-joints. This connection type choice on connection points allows the movement of the connection nodes that have six degrees of freedom when evaluated with FEA (Dimcic, 2011). The six degrees of freedom includes the axial forces and moments for X-, Y- and Z-Axis. The members of gridshell resist these forces through the area of their cross-section for axial forces and their moment of inertia for bending.

3.5. Conclusion and Design Goals

In the conceptual idea development process of a gridshell, any decision related to the architectural aspect of the structure directly correlated with the structural performance. This is mainly because due to the nature of gridshell systems, the architectural elements are also the load-bearing elements of the structure. This situation makes the form-finding process a crucial phase of any gridshell design problem. In this chapter, the traditional and computational form-finding methods are reviewed. Besides, an evaluation method for the structural performance of generated gridshell alternatives is studied.

According to the current literature, there are different methods that can be employed in the context of gridshell design problems in order to find an equilibrium shape. As one of the pioneers in this field, Robert Hooke developed a method to find the form of an arch that works on pure compression by utilizing a simple reverse catenary model. Following the development of Hooke's law, the application of the method extended to the 3D shell problems by considering multiple catenaries as a network of forces. This approach forms the basis of different physical model-making techniques and computational form-finding methods.

In light of the current literature, it can be stated that making physical models is a technique that has been used by numerous architects throughout history. The purpose of this is to visualize the output of the specific design problem beforehand. By adopting the principles of Hooke's Hanging Chain Law, architects have been using small-sized scale-dependent and scale-independent physical models in gridshell design problems to evaluate their design's structural performance. Thanks to the developments in computer technologies, the experience gained from various researches and experiments about form-finding principles can be translated into the digital environment. By using digital computational form-finding methods, the form-finding process of a structure can be shortened significantly. This is mainly because in the computational environment the architectural models can be created in relation to the required structural calculations. In the current literature, the computational form-finding methods are categorized into two groups: Geometry-Oriented Methods and Material-Oriented Methods. The main difference between these methods is in geometry-oriented methods, the geometrical features related with a specific problem are taken into account whereas, in the material-oriented methods, the material

properties are also included to the context of the method. One of the most-used methods for each category is reviewed and compared to each other. In the reviewed literature it is stated that as a geometry-oriented method Force Density Method is capable of achieving a solution efficiently in the cases that the equilibrium shape considered as working on compression. However, when the relative material properties are known, the Dynamic Relaxation Method as a material-oriented method is a more suitable choice to include the form-finding process of a design problem.

Alongside the development of computational form-finding procedures, the computational power of the computers can be used to evaluate the structural performance of an architectural design. In the reviewed literature, it is seen that the Finite Element Method can be employed to analyze the structural behaviors of the physical systems. The Finite Element Analysis process creates a specific stress distribution for each element to achieve nodal force equilibriums. The complex structural nature of gridshells allows the utilization of Finite Element Analysis to calculate the nodal displacements and stability of the structural elements.

In the light of the reviewed literature in this chapter, design goals for the computational gridshell design method are considered as:

- The method should be able to visualize the results of the architectural decisions as well as the structural performance.
- The material-oriented form-finding procedure Dynamic Relaxation Method should be included in the form-finding process of the computational method since the material properties are known.
- The method should contain the Finite Element Method to calculate the nodal displacement in order to evaluate the structural performance of the gridshell alternatives.

CHAPTER 4 DEVELOPMENT OF THE COMPUTATIONAL WORKFLOW FOR GRIDSHELL GENERATION AND OPTIMIZATION

In the previous chapters, the main principles and features of shell and gridshell systems, as well as the different types of the form-finding procedures, are depicted. As mentioned before, the translation of a gridshell problem into the computer environment allows adopting computational methods in order to overcome the complex problems related to designing a gridshell. The form-finding process of a gridshell should include both architectural decisions and engineering calculations in order to fulfill the aesthetical, functional and structural requirements. Using an iterative computational method allows the user to examine the correlation between the architectural and engineering aspects simultaneously.

In this chapter, the development of the intuitive gridshell generation method is described. The method includes a computational model that is defined in an algorithmic design environment and can be controlled parametrically. Moreover, the integrated form-finding procedure and structural evaluation are included in the method which responds to any change in parameters.

4.1. Design Considerations and Modeling Environment of Generic Gridshell Model

In contemporary architecture, the free-form structures are often used since they can offer more flexibility concerning shape variation in comparison to the traditional architectural approach. However, in most cases, free-form shapes are considered as complex geometric shapes. Therefore, the translation problem of these shapes to the architectural design process requires innovative and unique solutions. Following the developments in the aeronautical and shipbuilding industries, free-form shapes take place in the architectural field. As an example, in the construction phase of Frank Gehry's Guggenheim Museum in Bilbao which is a famous example for free-form building, the manufacturing methods and experience gained from the ship, aeronautical and car industries are employed in order to construct the building which consists of free-form geometries. Moreover, the software which can demonstrate free-

form surfaces, with mathematical precision, was developed to cope with the problems related to the ship and car design.

Pierre Bezier and Paul de Casteljaou who are the two employees of two firms in the car industry, Renault and Citroen respectively, can be considered as the pioneers of free-form design principles using a polynomial representation of curves in the 1950s. Later, the problem was generalized from *Bezier Splines* to *Non-Uniform Rational Basis Splines (NURBS)* and developed into *NURBS Surfaces* (Piegl & Tiller, 1987). Since theoretically any shape can be demonstrated by using NURBS, it became a part of the *Computer Aided Drawing (CAD)* environment in the '80s. Even today, NURBS are often used in all design fields in order to create geometrical representations of free forms.

Rhinoceros 3D is a NURBS based 3D modeling tool, which is developed by Robert McNeel and Associates and first introduced in 1993. Even though the software is created for mainly industrial designers, the software's capability of mathematical surface representation allows the use of software in different design fields. In order to represent the free-form gridshell geometries, *Rhinoceros 3D* is used in this study.

Furthermore, alongside the *Rhinoceros 3D*, the study requires an environment in which the elements can be defined by using discrete parameters and allows the use of specific algorithms. For this reason, a visual programming extension of *Rhinoceros 3D* named *Grasshopper 3D* is also used in this study. *Grasshopper 3D* is a visual programming environment that works within the *Rhinoceros 3D* software, developed by David Rutten at Robert McNeel and Associates (Rutten, 2007). *Grasshopper 3D* environment allows to control and define design elements parametrically as well as the usage of algorithms.

Considering the generative process of *Grasshopper*, one may state that it is a tool for creating a design method rather than creating just one model since it consists of sequential operations in which the output highly depends on the input definitions even in the case of using same consecutive operations. As an example, in Fig. 4.1, a simple line creation process in *Rhinoceros 3D* by using *Grasshopper* environment is shown.

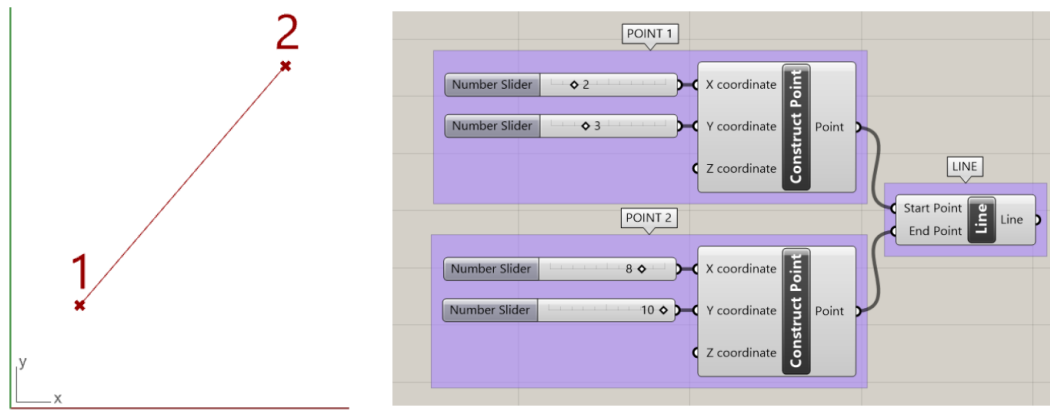


Figure 4.1. Line creation by using Grasshopper 3D.

Line definition requires two inputs, the coordinates of the points which the line pass through. To create points in the XY Plane, the required inputs are defined as (2,3) for Point 1 and (8,10) for Point 2 as shown in Fig. 4.1. Any change in the points coordinates which in this case the parameters of the line, result with the change in the line's length, position or angle with X and Y axis. If multiple lines are created with the same principle to define a closed geometry, any change in coordinates of the points will affect the line; thus, the overall geometry will change. This aspect of Grasshopper allows the user interaction in each step of a sequential workflow.

Thanks to the visual programming interface of the Grasshopper environment, no prior knowledge about text-based programming languages are required in order to create an algorithm. However, algorithms that are created in text-based languages such as C# and Python can also be defined as a component in the Grasshopper environment. Besides, since Grasshopper is free of charge software, there are a lot of plug-ins developed by Grasshopper community that enhance the software's capabilities and create possibilities to use the software in different fields. One of the most used plug-ins is Kangaroo which allows the creation of physics-based simulations in Grasshopper environment. Kangaroo is developed by Daniel Piker who works with the *Specialist Modelling Group* at Foster and Partners (Piker, 2013). Kangaroo can be described as a collection of algorithms related to the simulation of behaviors of the materials and objects in the real world, in other words, a *Physics Engine* (Hecker, 1996). In order to create an initial geometric form of the generic gridshell surface, the *Dynamic Relaxation* algorithm of Kangaroo is included in this study.

Following the development of generative design components in the architectural field, the next logical step can be considered as the evaluation of the design's real-world application regarding structure. For this reason, an application that allows evaluating the design's structural aspects is introduced by Preisinger and Heimrath (2014) named *Karamba* which is a *Parametric Structural Design* application that can be integrated seamlessly with the Grasshopper environment. As mentioned in the previous chapters, the decision-making process in a gridshell design should include both architectural decisions and engineering requirements. Since *Karamba* as a *Parametric Finite Element Toolkit* allows to evaluate the generic parametric gridshell model structurally, it is included in this study to extend the Grasshopper's structural evaluation aspect.

The generative design approach of Grasshopper which takes its power from parametrically defined design objects in its context can easily be turned into an iterative design tool to create a set of different outputs since a slight change in the defined parameters result with a completely different output. In order to automate these iterations based on changing design parameters to create a set of solutions which satisfied the required criteria, a loop can be included in the workflow. *Octopus* is a plug-in for Grasshopper developed to find a set of optimum solutions by employing evolutionary algorithms (Vierlinger & Schneider, 2014). *Octopus* is described as *Multi-Objective Evolutionary Solver* which applies evolutionary principles to the parametric design environment. It is developed based on the works of David Rutten (2013) on *Galapagos* generic solver. Even though *Octopus* includes the same algorithms which are named as *SPEA-2* and *HypE* as the *Galapagos* plug-in, *Octopus* allows employing multiple fitness values in the optimization procedure whereas in *Galapagos* a single fitness value can be included in the optimization. In order to create a set of alternative gridshells which satisfies the multiple design objectives and constraints, *Octopus* is included in this study.

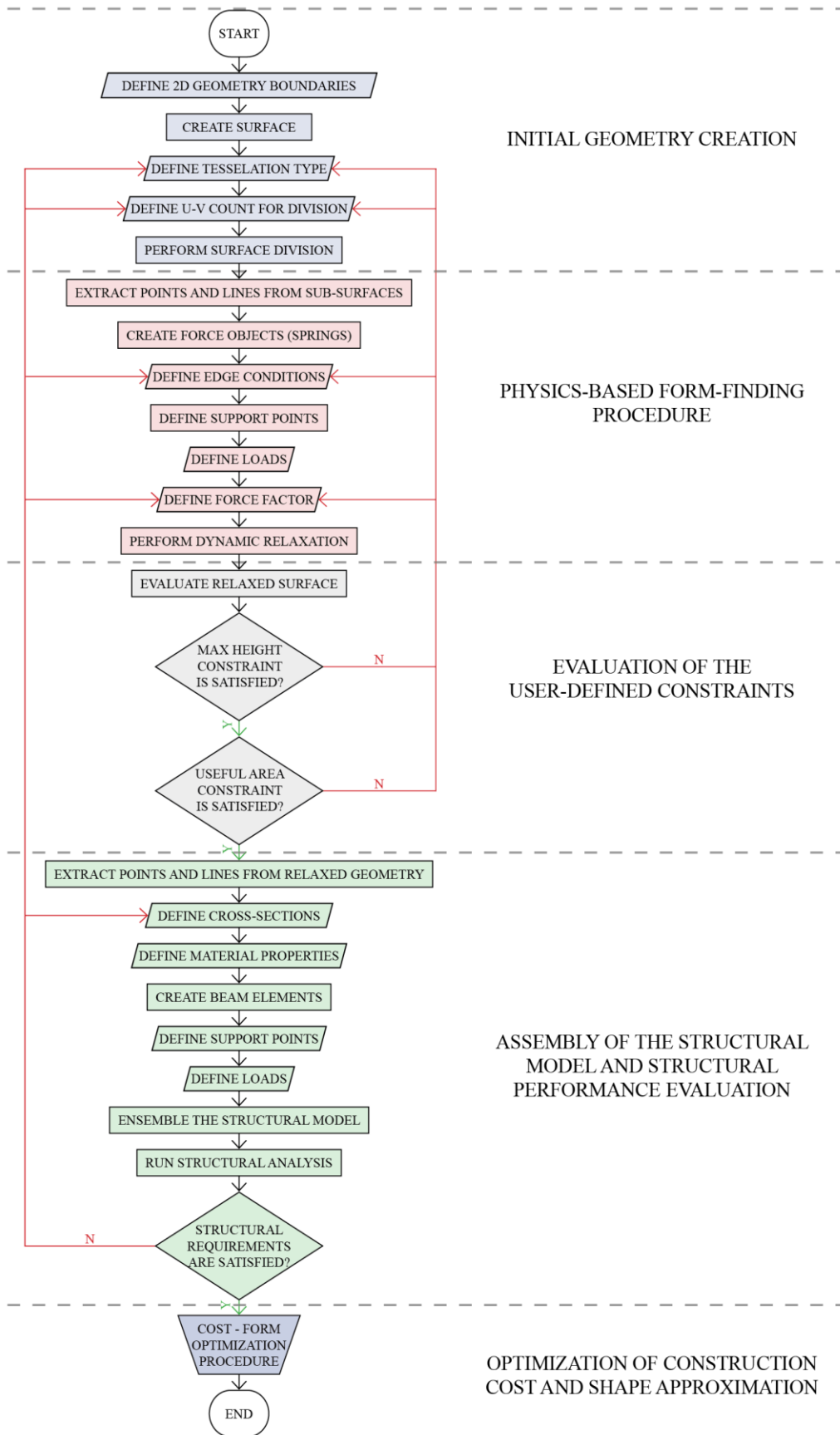


Figure 4.2. Flowchart of the Grasshopper workflow.

In the light of the examinations done on the previous sections, the Grasshopper workflow for generating a gridshell model should include the initial 2D geometry creation, the physics-based form finding the procedure, assembly of the structural model, the structural performance verification and the optimization related with the design objectives in order to create a set of alternatives. The complete flowchart of the workflow is presented in Fig. 4.2.

4.2. Initial Geometry Creation

The flowchart of the initial 2D geometry definition is shown in Fig. 4.3. The user-defined inputs are represented with parallelograms in the flowchart.

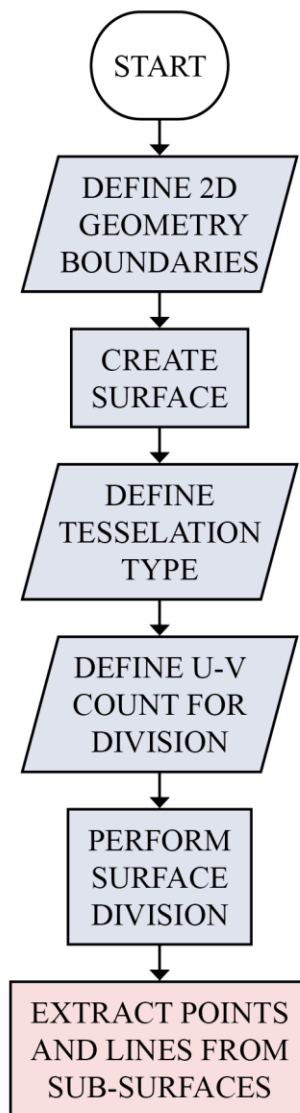


Figure 4.3. Initial 2D geometry definition flowchart.

For the first step, the area boundaries should be determined. The boundaries also represent the gridshell structure's dimensions on the XY Plane if the distance in the +Z direction is considered as the height of the structure. The definition of the boundary may include lines to represent the enclosed geometry or points that are eventually connected with lines in order to define the edges of the geometry. Since Grasshopper is defined as the primary tool to create the generative workflow, the shape of the initial 2D boundary of the structure will significantly affect the output. This is mainly because of the adopted physics-based form finding procedure Dynamic Relaxation which will be discussed in detail in the following sections. The form-finding procedure starts with a flat surface and deforms the surface eventually by considering the applied loads and force objects. If the initial 2D geometry is symmetrical, the resulted shell surface will also be symmetrical. The same principle applies if the initial geometry is asymmetrical. For the sake of the explanation, two shell surfaces that are initially generated from a trapezoid and a rectangle are shown as an example in Fig. 4.4, respectively.

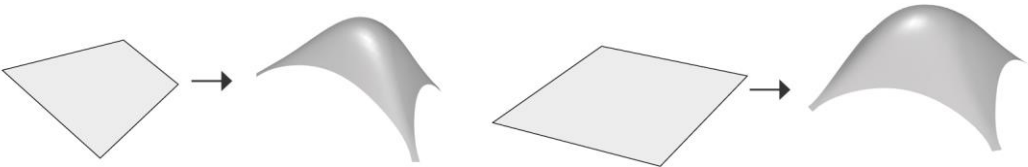


Figure 4.4. Two shell surfaces generated from a trapezoid and a rectangle, respectively.

Following the definition of 2D geometry that represents shell's boundary on the horizontal plane, next step is to define a tessellation method for the surface division and a division count in order to extract points and lines from the surface. Since Rhinoceros 3D allows the creation of NURBS surfaces, every surface has a UV coordinate system. This is a translation of the world coordinate system to each surface; therefore, each NURBS surface can be manipulated separately using the related coordinate system. For the sake of the explanation, two identical simple cylindrical surfaces are created and divided into quadrilateral sub-surfaces by using two different U-V values as shown in Fig. 4.5.

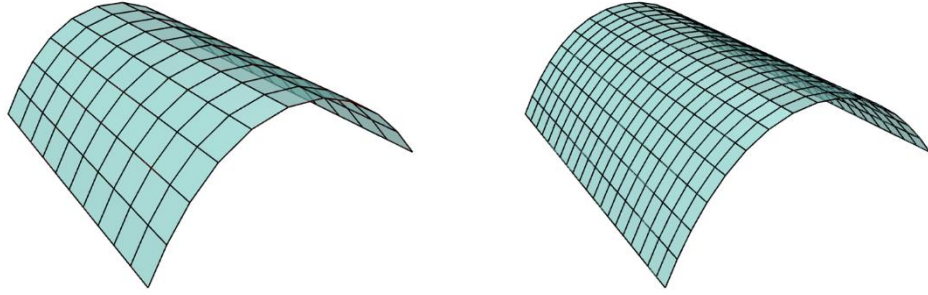


Figure 4.5. Two identical cylindrical surfaces divided into quadrilateral sub-surfaces.

On Fig. 4.5, for the surface on the left division values defined as 10 for U-direction and 10 for V-direction whereas for the surface on the right the values are 15 for U-direction and 20 for V-direction. Since the U and V values determine the number of sub-surfaces after the division, an increase in U and V values results with an increased sub-surface count. For example, the surface on the left is divided by 10 in each direction this results with 100 sub-surfaces (10 multiplied by 10) whereas, the surface on the right is divided by 15 and 20 in the U and V directions, respectively, this results with 300 sub-surfaces (15 multiplied by 20). Considering gridshells, the surface division definition directly related to the discrete element count and their length, since the lines will transform the beam elements eventually. In the Grasshopper workflow, the UV count of the surface division is defined as a decision variable which can be defined by the user or the optimization algorithm at the end.

Considering the surface division, tessellation type is also needed to be defined. In the surface examples shown in Fig. 4.5, the surface tessellation type is defined as a quadrilateral division. Also, there are different tessellation algorithms for the surface division as shown in Fig. 4.6. Quadrilateral tessellation can be considered as a common approach in gridshell structures since the production phase related to the quadrilateral sub-surfaces is more straightforward than the other sub-surface types. This is mainly because each connection point connects only four discrete elements. However, the quadrilateral sub-surfaces, as well as the diamond sub-surfaces, may not be as stable as triangulated surfaces. This is due to the lack of diagonal bracing and may result in buckling.

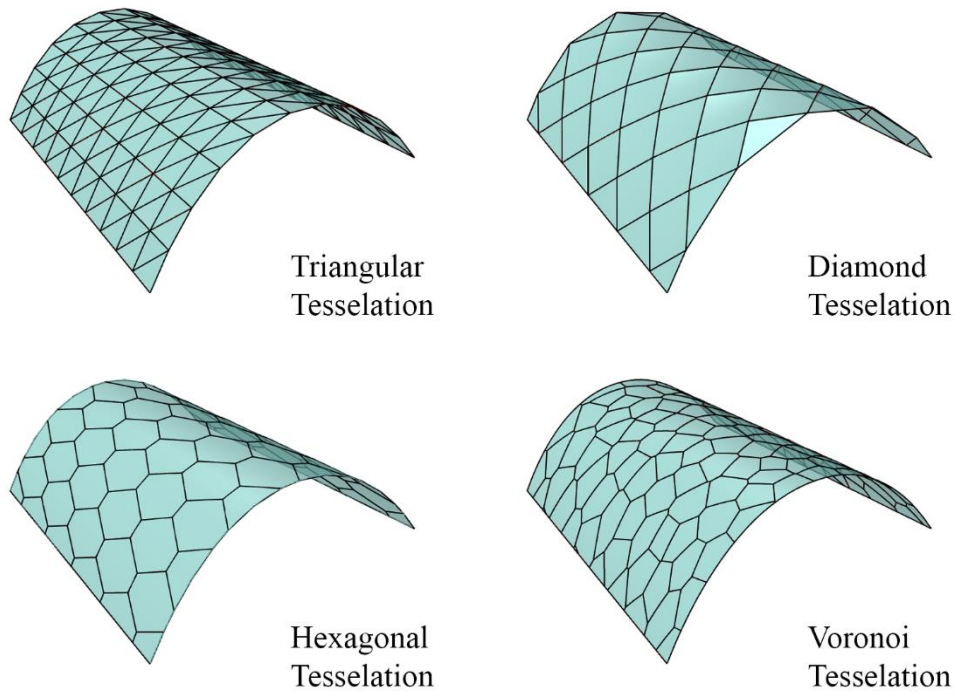


Figure 4.6. Different surface tessellation types.

Triangular tessellation, on the other hand, results with geometrically locked and rigid structures. Due to the geometric nature of a triangle, panels that are used to cover the structure are always planar. However, the joint connections in triangulated sub-surfaces are not as simple as the other sub-surface types. For each connection point, there are six intersecting discrete elements; therefore, this results with complex connections.

Hexagonal and Voronoi tessellation procedures are the two types of tessellation methods which can be employed. The hexagonal tessellation results with a structure similar to a honeycomb whereas, the Voronoi tessellation results with a more organic appearance. The advantage of these structure types is there are three discrete elements in each connection point. This results with a simpler joint connection, however, they behave very poorly regarding structural performance.

The tessellation types which are included in this study are the quadrilateral tessellation and triangular tessellation due to their structural stability in comparison to the other tessellation types. In the Grasshopper workflow, there is a decision variable related to the selection of surface tessellation type. The selection of the tessellation method is both architectural and structural decision. Therefore, this input can be defined by the user or the optimization algorithm during the optimization process. If the tessellation

procedure is considered as an important decision by the user since it can directly affect the overall appearance of the gridshell, the input can be defined by the user; as a result, it will not be a decision variable for the optimization process.

4.3. Physics-based Form Finding Procedure

As mentioned before, in the gridshell design process there is a clear correlation between architectural and structural decisions since, a gridshell is not only an aesthetic element but also, it should satisfy the structural requirements. Because of this situation, considering only the aesthetic features in the design phase of a gridshell may result in structurally unstable output. Therefore, in this study, a physics-based form finding procedure is employed in order to find a shape that in static equilibrium. After the initial 2D geometry creation and division into sub-surfaces, a dynamic relaxation module is used to find a shape that can be considered as in static equilibrium. Kangaroo plug-in for Grasshopper is used in the workflow to create Dynamic Relaxation simulation. The flowchart of the physics-based form finding procedure is shown in Fig. 4.7.

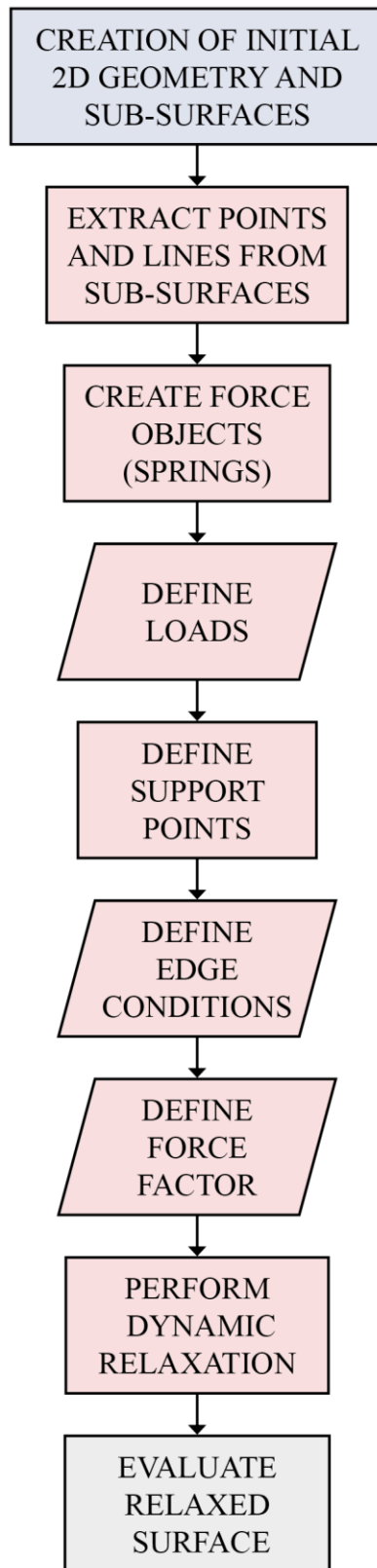


Figure 4.7. The flowchart of the physics-based form finding the procedure.

4.3.1. Creation of the Force Objects

Following the creation of sub-surfaces, the vertices (corner points) and the edges of the surfaces are extracted in order to include them to the spring definition. Kangaroo plug-in contains a component that is able to create Hooke's law springs. The component needs the following inputs:

- Lines that connect two relative vertices to define springs' in-between action,
- Rest Lengths, the length that the relative spring try to reach,
- Spring Stiffness.

For the first input, the surface edges and relative vertices are defined. The input required to define the in-between action of the relative springs. The second required input, *Rest Length* represent the length that the relative spring tries to reach. In other words, if the rest length is equal to the length of the relative line, the spring starts out in a state of no stress. If the *Rest Length* is smaller than the relative line's length, this corresponds to pre-tension. In this study, the Rest Lengths are defined for each spring separately by using the relative lines lengths. Therefore, there all springs start simulation in a non-stressed state.

The third required input is *Spring Stiffness*. Piker (2013) the developer of the *Kangaroo* plug-in, pointed out the difference between Rest Length and Spring Stiffness as: "*Rest Length is the length the spring tries to reach, while Stiffness is how strongly it tries to get there.*" As Piker (2013) mentioned, the *Spring Stiffness* can be defined as *Axial Stiffness* as follows:

$$k = \frac{A_x \times E}{L_s} \quad (4.1)$$

where k represents the spring stiffness, A_x represents the cross-section area of the relative element, E represents the Modulus of Elasticity of the relative material and L_s represents the initial length of the relative spring.

As mentioned in the previous chapters, Dynamic Relaxation is a material-oriented form finding method; thus, it requires the value of *Modulus of Elasticity* of the materials included in the simulation. For this study, a generic wood material is selected and utilized in the creation of beams which has the E value of 1050 kN/cm². This wood

material is a predefined material in Karamba plug-in that is used in the structural model assembly and verification part of this study which is explained in Section 4.5.

4.3.2. Definition of Edge Condition and Support Points

Following the spring creation, simulation requires the support (anchor) points. These points can be considered as fixed points that will not be allowed to move in the simulation. In order to define the support points, there is an input in the Grasshopper workflow that needs to be defined by the user, by considering the architectural and functional needs, is related with the shell surface's relationship with the geometry's numbered edges, named as Edge Condition.

In order to define Edge Condition separately to each edge of the initial surface, the edges are numbered in order to control each edge by using an assigned number to each edge in the workflow. Numbering procedure of geometry's edges and their relationship with the Edge Condition component in Grasshopper is shown in Fig. 4.8. This is an integrated feature of Grasshopper which allows managing large data sets efficiently by using lists in which all the elements have specific index numbers. In the Grasshopper workflow, Edge Condition is an integer value for each edge of the geometry within the range of [0,2].

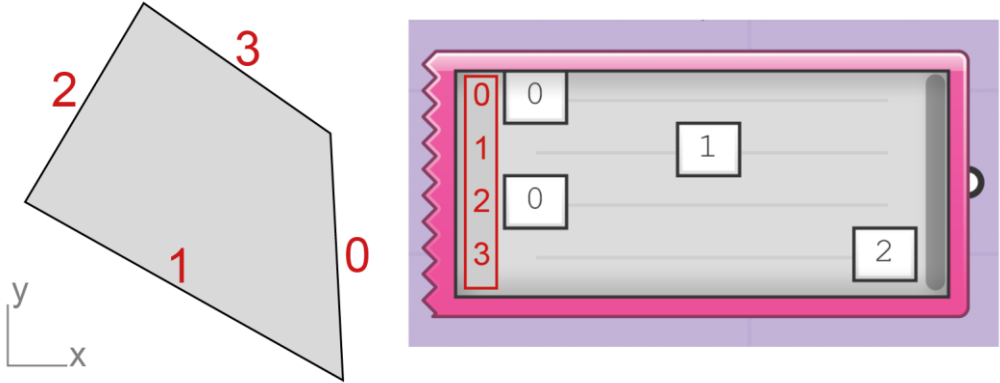


Figure 4.8. Numbering procedure of geometry's edges and Edge Condition Component defined in Grasshopper.

Edge Condition component is used to define the relationship between the edges of the geometry and the shell surfaces that will be generated after the Dynamic Relaxation procedure. In other words, the component eventually defines the support points of the shell geometry. For each edge, a value should be defined. In Fig. 4.9, the defined Edge Condition values and their effect on shell geometry are shown.

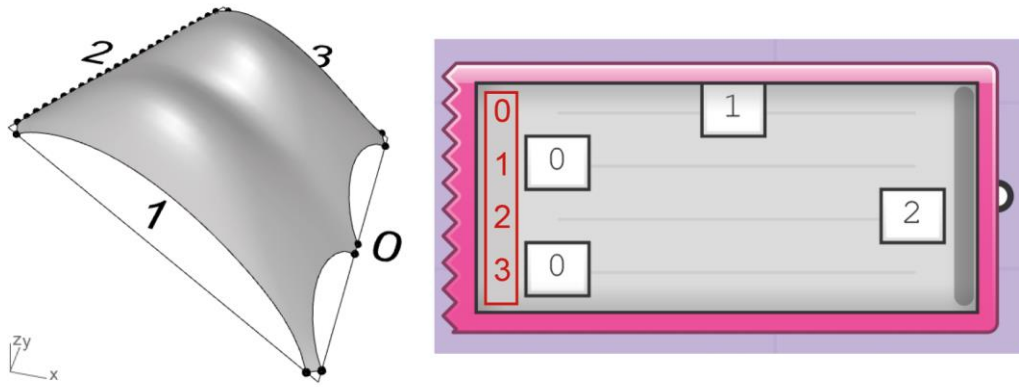


Figure 4.9. Edge Condition values and their effect on shell geometry.

In the example shell surface in Fig. 4.9, the resulting shape of three possible Edge Condition values can be seen. For the edges numbered as 1 and 3 the Edge Condition value defined as “0”, this result with the definition of support points on the start point and end point of the edge. There are no in-between points defined as support points. Therefore, the translation of the 2D geometry edge to the shell surfaces edge result with a continuous curve with a large span. According to architectural needs, this result may be wanted to create large permeable face to the shell structure without any division. However, because of the large span without any additional supports, the resulting shell may behave poorly concerning structure. The Edge Condition value “1” which is defined for the edge of the geometry which is numbered as 0 on the example, allows to make a selection between one continuous opening and creating additional support point. Edge Condition value 1 can be selected by the user in cases that the permeability from this side of the structure is not an important issue. Following this architectural decision, the definition of Edge Condition value as “1” creates a decision variable that included to the optimization process in the later stage which results with different shell and geometry relation related to this edge. This allows to a selection between a continuous vertical opening on the shell or multiple openings divided by additional supports on the edges. Lastly, when Edge Condition defined as “2”, the shell surface’s edge adopts the edge of the 2D geometrical boundary and creates continuous support points all along the edge. According to the architectural needs, the resulted face of the shell may be utilized for different purposes as well as to create a non-permeable face to the shell structure.

4.3.3. Definition of Loads and Force Factor

Following the creation of springs and the definition of support points, Dynamic Relaxation component requires a force in order to calculate the displacement for each spring element. For this reason, in the Grasshopper workflow, the forces are calculated and defined on each connection point as vectors. The force which is included in the physics-based form finding procedure is the structure's own weight. The force for each node is calculated as follows:

$$F_n = \frac{(d_w \times L_T \times A_x)}{N_j} \quad (4.2)$$

where d_w is the density of the generic wood material, L_T is the total length of the lines (sub-surface edges), A_x is the cross-section area of the beams that are going to be created, and N_j is the total count of the connection points which is a direct result of the surface division mentioned in Chapter 4.2.

Even though the beams are not created yet and Dynamic Relaxation procedure includes lines as the spring elements, the weight of the beams is calculated in order to achieve a realistic weight calculation. The selection of the beam dimensions is not a user-defined decision variable. Instead, the optimization algorithm can do the selection in order to achieve the design objectives while satisfying the constraints mentioned in Chapter 4.6. The dimensions of the beams that are included in this study are shown in Tab. 4.1.

Table 4.1. Classification of the wooden beams by their dimensions.

Beam No	Width	Height
1	5 cm	10 cm
2	5 cm	15 cm
3	10 cm	15 cm

After Dynamic Relaxation procedure concludes, the initial surface will collapse by its own weight and creates a shape similar to a weighted fabric hanged from support points. Similar to Hooke's hanging chain principle, if the direction of gravity is reversed, the resulting shape will be a shell surface that has the minimum height in order to create a structural equilibrium. As a rule of thumb, any shell geometry that has a height above this threshold can be considered as suitable.

In order to increase the output variation, a decision variable is defined in the Grasshopper workflow named Force Factor which is a numeric value. Force Factor (F_x) is a floating-point value which has a range of [1.0, 5.0] can be multiplied by the value of the force for each node (F_n). As an example, if the F_x is defined as 1.0, this means resulting shell surface will have the minimum required height which achieves structural stability whereas, if it is defined as 5.0 the load in the calculation is multiplied by 5 therefore, the height of the resulting surface will be higher.

4.3.4. Dynamic Relaxation Procedure

The integrated Dynamic Relaxation component in Kangaroo plug-in is included in the Grasshopper workflow. It requires the definition of springs, loads and support points in order to perform a simulation. As mentioned in Chapter 3.3.2, the force that required to move a spring is directly proportional to the displacement of the spring:

$$F = k \times \Delta x \quad (4.3)$$

F represents the force, k represents the spring stiffness and Δx represents the total displacement of the spring. Since Dynamic Relaxation process is an iterative process that evaluates each spring is specific time intervals, it will continue until the Kinetic Energy of the system reaches zero. When Kinetic Energy of the system is zero, the system is considered as in equilibrium. An example of the Dynamic Relaxation process is shown in Fig. 4.10.

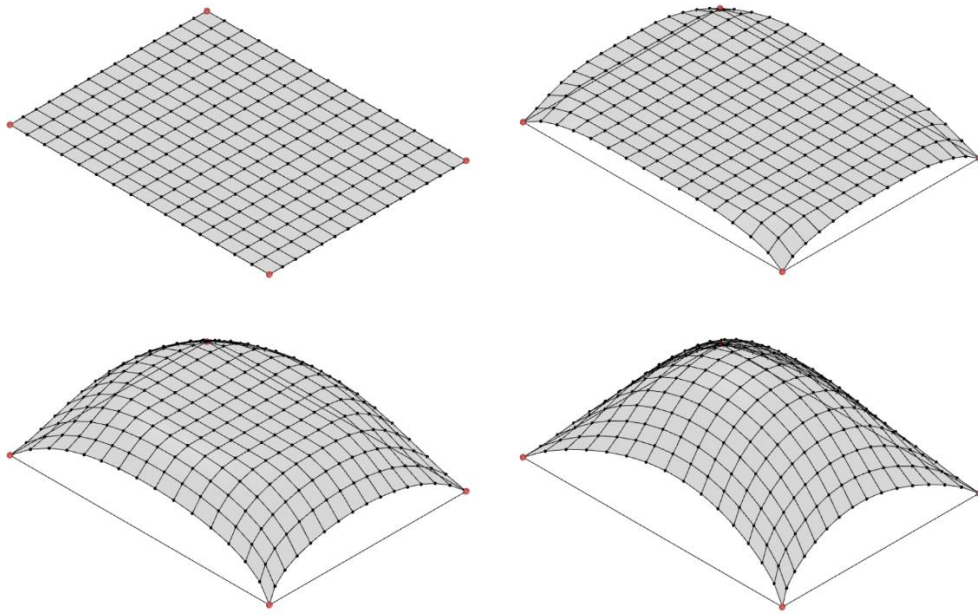


Figure 4.10. Dynamic Relaxation process of a flat rectangular surface.

In Fig. 4.10, a flat rectangular surface is created as an example. The points colored as red are the corner points (vertices) of the surface which defined as the support points. In the example, spring stiffness is a generic value that is defined in the Kangaroo's default settings and the applied load is the gravity load in reverse direction. Therefore, the surface appears as if it is inflated instead of collapsing due to its own weight.

4.4. Evaluation of the User Defined Constraints

Following the creation of the relaxed surface, there are some constraints in the Grasshopper workflow that needed to be defined related to the architectural aspects of the shell. These constraints defined as hard constraints which means if the constraints are not satisfied, the output is considered as not feasible, therefore, the output will not be included to the set of alternative shell surfaces at the end. A flowchart of the constraint evaluation process in Grasshopper workflow is shown in Fig. 4.11.

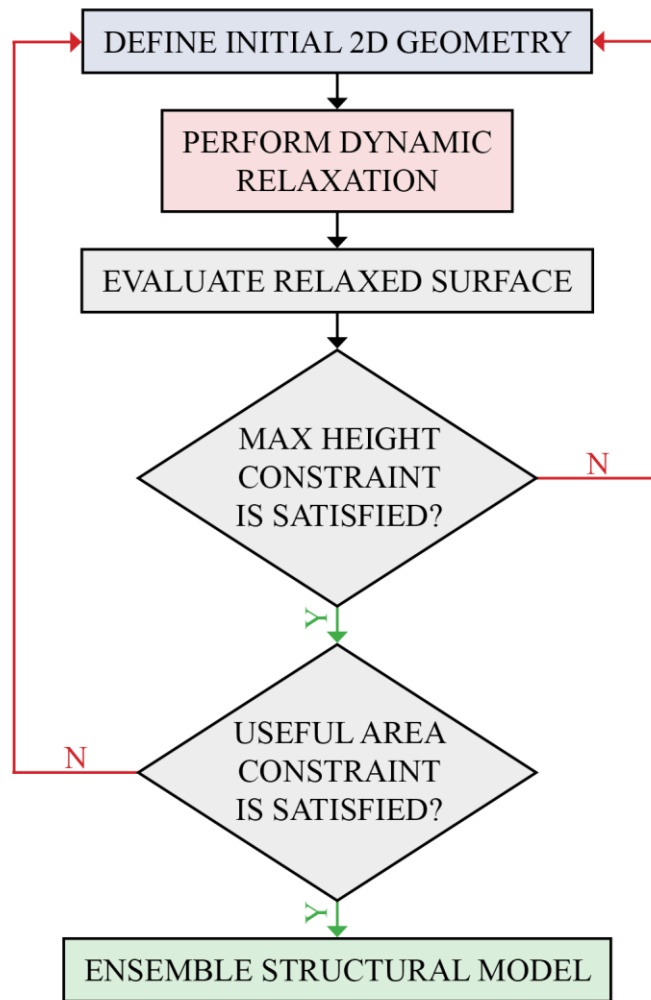


Figure 4.11. Flowchart of Constraint Evaluation in Grasshopper workflow.

4.4.1. Maximum Height Constraint

One of these constraints is related to the structure's maximum height. The shell surface's height is not an input that can be defined in the workflow since it is a result of the Dynamic Relaxation form-finding process which is explained in the previous section. DR procedure helps to determine the minimum height which is required to create a structurally stable shell surface. However, because of the limitations which are depicted in regulations the user may want to limit the maximum height of the structure by defining an upper limit as a design constraint. Therefore, a hard constraint named as Maximum Height Constraint (MHC) is introduced to the workflow. Since MHC is defined as a hard constraint in the Grasshopper workflow, if the output's height is larger than the allowed height for the structure, the output will not be considered as suitable for the given design problem.

4.4.2. Useful Area Constraint

The second user-defined design constraint is named as Useful Area Constraint (R_{UA}). It directly corresponds to the Useful Height (UH) numeric input which is an input parameter that needs to be defined by the user considering architectural aspects of the space. Since the utilization of the area bounded by a gridshell may differ from an area surrounded with vertical walls due to the slope of the shell surface. Having a slope results with spaces that may not be used in specific functions due to the height of the ceiling.

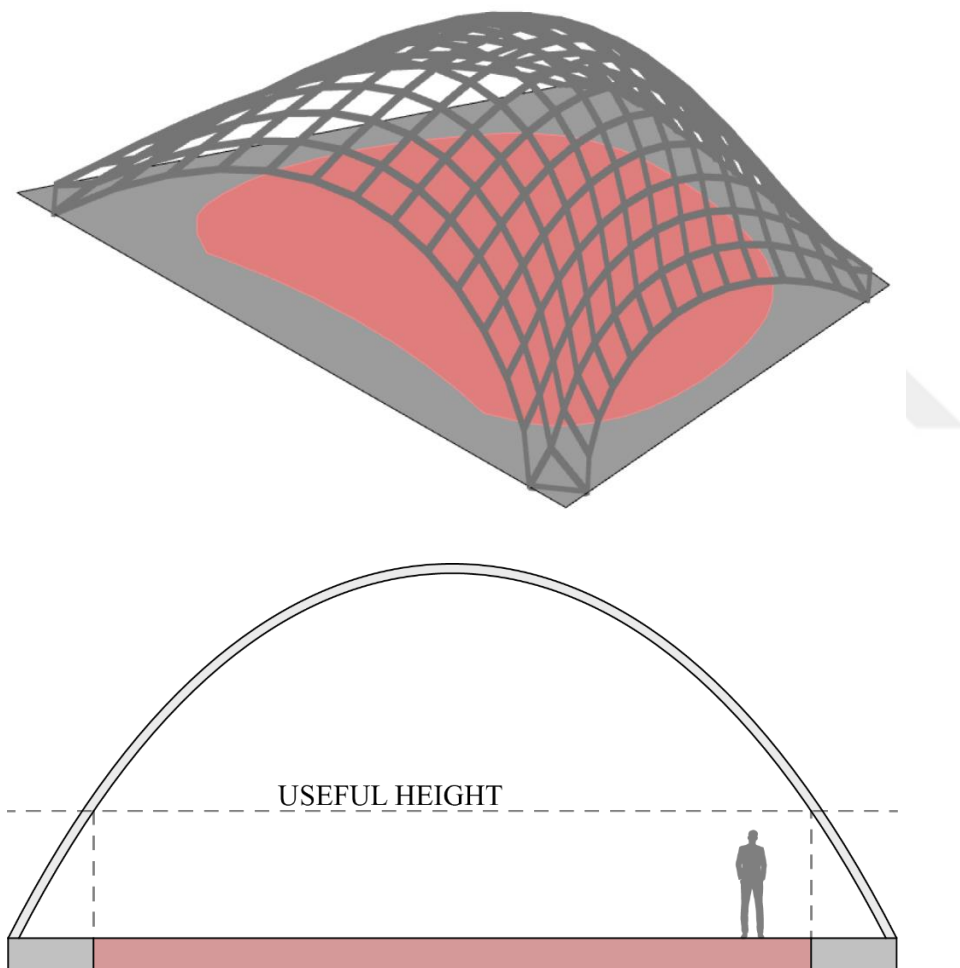


Figure 4.12. Interpretation of Useful Height input and Useful Area constrain.

For the sake of the explanation, a gridshell example is shown in Fig. 4.12. In the example, a specific UH value is defined and according to this value, the area separated into two regions. The gray region is considered as not suitable for specific functions because the ceiling height in this region is lower than the UH value. On the other hand,

the red region represents the floor which has an acceptable ceiling height for specific functions. Considering the architectural needs and program the UH could vary; therefore, it is included in the Grasshopper workflow as a User Defined input that directly affects the calculations related with R_{UA} .

The UAC is the ratio of useful floor area to the total floor area of the initial 2D geometry as percentages. As an example, if the resulted percentage is 50%, this means the half of the floor is not feasible considering the minimum height defined as UH input. In order to increase of the Useful Floor Area (UFA) by employing a specific UH limitation, the gridshell's total height should be increased to create a steep slope in shell surface thus, more area on the floor that can be considered as useful. However, an increase in the gridshell's height may result in higher construction costs as well as it can affect the decisions related to the form. Therefore, the introduction of the R_{UA} creates a good case for an optimization procedure.

4.5. Assembly of the Structural Model and Structural Performance Evaluation

Following the creation of relaxed geometry and constraint evaluation, a structural model is ensembled in Grasshopper workflow, in order to evaluate the structural performance of the model. The flowchart of the structural model's assembly process is shown in Fig. 4.13.

Karamba plug-in is included in the Grasshopper workflow to create a structural model. As mentioned in Chapter 4.1, *Karamba* is a Finite Element program which includes all the required functionality of commercial structural analysis software. The main difference of *Karamba* from other available programs is it can be integrated into the Grasshopper environment. Thanks to the plug-in, the advantages of creating a parametric workflow can be carried all the way into the structural analysis. In other words, any change in the parameters of the geometry will change the structural model simultaneously. Therefore, thanks to the instant feedback related to the structural performance, during the conceptual design process of the gridshell the structural results of the architectural decisions can be seen.

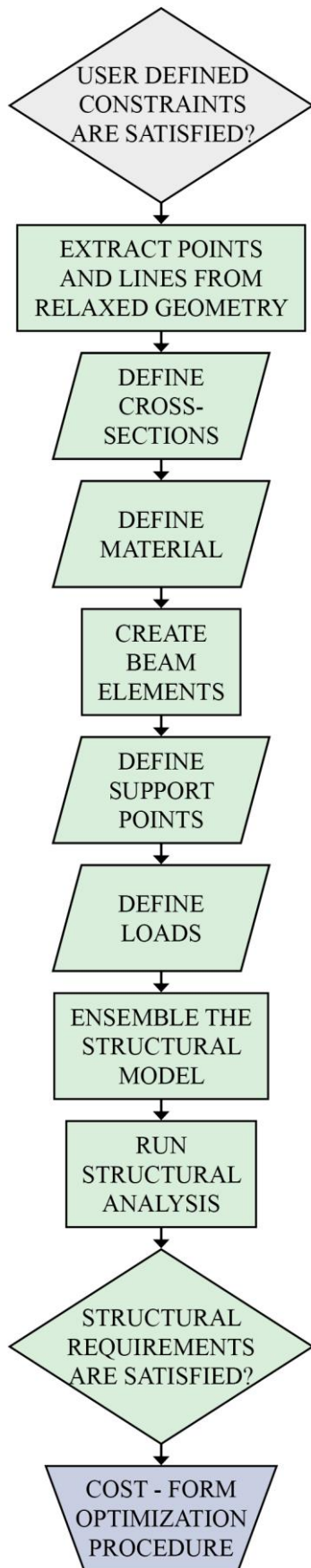


Figure 4.13. Flowchart of Assembly of structural model and evaluation in Grasshopper workflow.

Karamba component in the workflow uses the lines and points which are extracted from relaxed geometry. This way, following the end of the Dynamic Relaxation procedure, the structural model has created automatically. The component related to the assembly of the structural model requires the following inputs:

- Input nodes, which is translated to the intersection points of the beams,
- Beam elements,
- Cross-sections and material properties of the elements.
- Support points,
- Applied loads,

Following the extraction of points and lines, they are used to create beam elements. Beam elements also require information about cross-sections and material properties. As mentioned in Chapter 4.3.3, Karamba includes the same beam cross-sections and materials as the Dynamic Relaxation process which is shown in Tab. 4.1. The selection of beam cross-section is a decision variable that affects both the Dynamic Relaxation procedure and assembly of the structural model.

Regarding material selection, a generic wood material is utilized in the creation of beams. As mentioned before, the selected wood material is a pre-defined material in Karamba, the specific properties of the material are shown in Tab. 4.2.

Table 4.2. Properties of the pre-defined wood material.

Family	Wood
Name	Wood
Young's Modulus	1050 kN/cm ²
In-plane shear modulus	360 kN/cm ²
Density	6 kN/cm ³
Yield Strength	1.3 kN/cm ²

In order to ensemble the structural model, the support points and loads must be defined. Similar to connection nodes, the support points are also extracted from the relaxed geometry and defined to the ensemble module in Karamba. The support points considered as fixed, therefore, their movement is restricted in every direction.

In terms of loads, the structure's self-weight and relative wind loads are taken into account. Besides the wooden gridshell members, the glass panels that cover the structure are included. The weight of the glass panels with the aluminum profiles for attachment is estimated as 0.5 kN/m².

Considering the wind force (F_w) acting upon the gridshell structure, the following formula is taken into account:

$$F_w = q_b A_s \quad (4.4)$$

where q_b is the dynamic pressure and A_s is the related area of the gridshell. To calculate wind load acting on the system, the dynamic pressure (q_b) and the basic wind velocity (v_b) are found with respect to TS EN 1991-1-4:2007. The lifting and drag forces are not taken into account.

$$v_b = C_{dir} \times C_{season} \times v_{b,0} \quad (4.5)$$

$$q_b = \frac{1}{2} \rho v_b^2 \quad (4.6)$$

where ρ is the density of the air, v_b is the basic wind velocity, C_{dir} is the directional factor, C_{season} is the season factor, $v_{b,0}$ is the fundamental value of the basic wind velocity. According to TS EN 1991-1-4:2007, the recommended value for C_{dir} and C_{season} is 1, ρ is 1.25 kg/m³. As Maden (2015) suggests, the fundamental wind velocity of 26 m/s can be taken into account for Izmir, Turkey. According to the Eq. 4.5, v_b can be calculated as 26 m/s and according to Eq. 4.6, q_b is equal to 0.4225 kN/m².

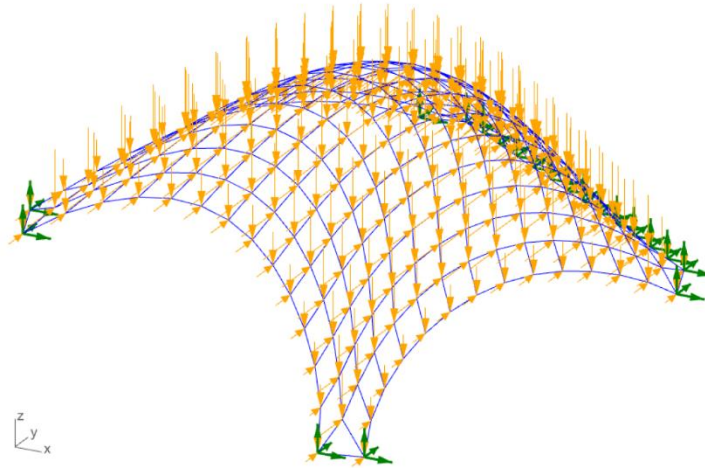


Figure 4.14. Loads and support points of the *Finite Element Model* in *Karamba*.

Following the assembly of the Finite Element Model, the model is analyzed considering deflections. The loads and support point definitions of an example gridshell model are shown in Fig. 4.14.

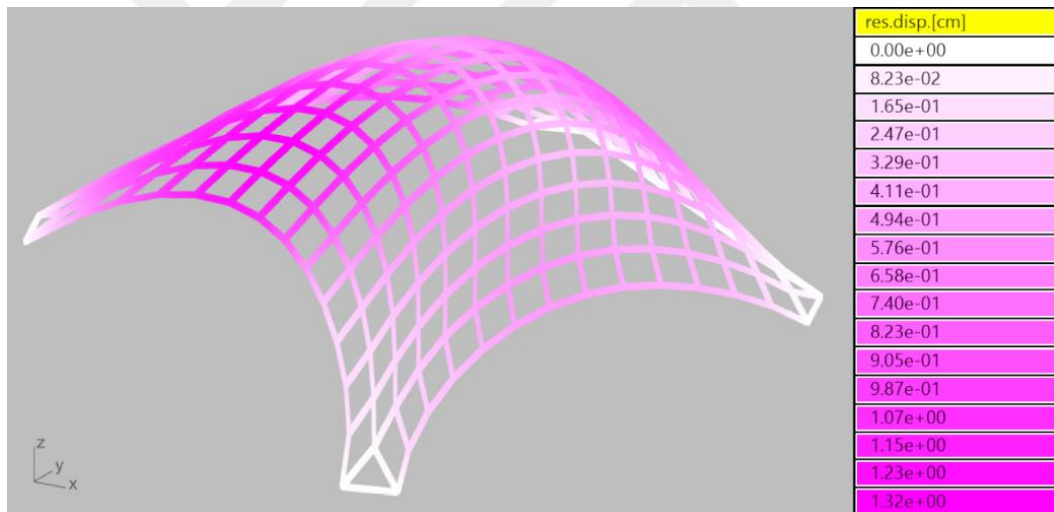


Figure 4.15. Displacement Analysis in *Karamba*.

There is a hard constraint related with the allowable deflections in the Grasshopper workflow, named as Maximum Displacement Constraint (MDC). According to the Turkish Standards, the maximum allowable deflection is limited to the span/200 (TS EN 1990:2002). The deflection of the members of example gridshell model is shown in Fig. 4.15.

4.6. Optimization of Construction Cost and Shape Approximation

In order to create a set of gridshell model alternatives related to the defined design problem, an optimization module is included in the Grasshopper workflow. As mentioned in Chapter 4.1, Octopus, an integrated multi-objective evolutionary solver for Grasshopper is used in the workflow in order to create suitable solutions while satisfying the constraints. By including this module to the Grasshopper workflow, it is possible to select a suitable solution that satisfies the minimum requirements from a list of alternatives.

The parametric Grasshopper workflow includes an optimization algorithm that can generate different solutions by altering the decision variables subject to the user-defined inputs and constraints. In addition to the user-defined inputs, some decision variables are included in the workflow in order to enrich the output variety. The decision variables in the workflow are shown in Tab. 4.3.

Table 4.3. Decision Variables.

Decision Variables	Notation	Type	Range
Surface Division Count in U - Direction	N_{u-div}	Odd Numbers	[11,45]
Surface Division Count in V - Direction	N_{v-div}	Odd Numbers	[11,45]
Edge Condition	E_{con}	Boolean	[0,1]
Surface Triangulation	S_t	Boolean	[0,1]
Force Factor	F_x	Floating Point Numbers	[1.00,5.00]
Cross-Section Selection	A_{x-sel}	Integer Numbers	[0,2]

Surface Division decision variables, as mentioned in Chapter 4.2, define the tessellation of the initial surface. The Surface Division Count in U-Direction (N_{u-div}) defines the division count in U-direction whereas, The Surface Division Count in V-Direction (N_{v-div}) defines the division count in V-direction of the surface. These variables control the amount of generated sub-surfaces, therefore, the discretization of the initial surface. Considering the resulted point count on the surface edges following the division, the type of these variables is defined as odd numbers in relation with the Edge Condition (E_{con}) which is mentioned in Chapter 4.3.2, to allow the selection of 2 points in the middle of each edge in order to define them as support points. Following the Edge Condition input which is defined by the user, E_{con} the decision variable is utilized by the optimization algorithm to create additional supports on the edges if it is necessary, therefore, it is defined as a Boolean toggle.

The Surface Triangulation (S_t) decision variable defines the tessellation type of the surface as mentioned in Chapter 4.2. It is defined as a Boolean toggle which makes a selection between triangular tessellation and quadrilateral tessellation. As mentioned in Chapter 4.2, the quadrilateral tessellation is a material efficient tessellation type whereas, the triangular tessellation can be considered as the most structurally effective tessellation type due to its geometry. Considering the maximum allowable deflection limit, a selection possibility is created for the optimization algorithm, since the quadrilateral tessellation is the most cost-effective tessellation type, however, in some cases, it may not be able to achieve structural stability.

Force Factor (F_x) is a decision variable in the workflow, mentioned in Chapter 4.3.3, which is included to increase the output variation. It defines the amount of force included in the Dynamic Relaxation Procedure, therefore it affects the height of the relaxed surface. It is defined as a floating-point value which has a range of [1.00,5.00] related with the multiplication of the force. As mentioned on Chapter 4.3.3, F_x value of 1.00 is equal to the initial calculated force whereas, the value of 5.00 can be translated as 5 times of the initial calculated force on the nodes.

Cross-Section Selection (A_{x-sel}) is a decision variable mentioned in Chapter 4.3.3 which is defined in order to make a selection between three possible cross-sections for wooden elements. This decision variable can be utilized by the optimization module in order to achieve structural stabilization since the different cross-section areas affect the maximum deflection of the system in the expense of increasing construction cost.

4.6.1. Problem Formulation

In the Grasshopper workflow, two objective functions are considered: minimization of the construction cost of gridshell (C_g) and the better approximation of the smooth surface (A_{shape}). Moreover, user defined constraints Maximum Height Constraint (H_{max}) and Useful Area Constraint (R_{UA}) constraint as well as the Maximum Allowable Deflection Constraint (Δ_{max}) are taken into account. As an example, the multi-objective optimization problem is formulated as follows:

$$\text{Min} \left(C_g, \frac{1}{A_{shape}} \right)$$

where

$$C_g = C_b + C_j \quad (4.7)$$

$$A_{shape} = N_{u-div} \times N_{v-div} \quad (4.8)$$

Subject to:

$$6 \text{ m} < H_{max} \quad (4.9)$$

$$60\% < R_{UA} \quad (4.10)$$

$$\frac{l_{max}}{200} < \Delta_{max} \quad (4.11)$$

As shown in Eq. 4.7, the total cost of the gridshell (C_g) can be found as the addition of the total cost of the wooden beams (C_b) and the total cost of the joints (C_j). These costs depend on the cross-section area of the wooden beams (A_x), the total length of the wooden beam elements in the system (L_T) and the surface division count.

$$C_b = L_T \times A_x \times C_w \quad (4.12)$$

$$C_j = [(N_{u-div} + 1) \times (N_{v-div} + 1)] \times C_n \quad (4.13)$$

The unit price of wood (C_w) is considered as 1500 TL per m^3 and cost of the one joint element (C_n) is considered 50 TL per element. Related to the calculation of discrete wooden structural elements (C_b) as shown in Eq. 4.12, the total volume of the elements is calculated and multiplied by the unit m^3 cost of the selected wood type. Considering

the total cost of the joints (C_j) as shown in Eq. 4.13, the total joint count in the system is calculated in relation with the surface division and multiplied by the unit cost of the joint elements (C_n).

Even though the glass and aluminum cover for the structure is taken into account in the structural calculations, it is not included in the price calculations. The cost calculations are defined related with the skeleton of the gridshell since the cover material may have a significant effect which results with domination in the construction cost calculation since it is an expensive material. The glass cover with aluminum structure is one of the heaviest cover materials. Therefore, it is chosen to include the structural calculations since the structure which is capable of carrying this material can carry most of the different cover materials.

Following the generation of a relaxed surface by using Dynamic Relaxation procedure as mentioned in Chapter 4.3.4, a smooth surface is obtained. In order to create a gridshell geometry, the smooth surface should be transformed to a discrete surface with planar faces. This includes the transformation of surface iso-curves to polylines since the elements of the gridshell are discrete. As Pottmann et al. (2007) mentioned, the more the polylines are refined and the discrete transformation the better the discrete version will approximate a smooth surface. An example of the discretization of a sphere is shown in Fig. 4.16.

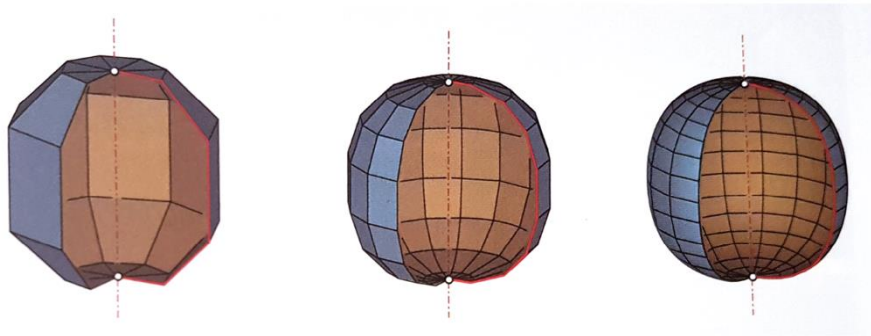


Figure 4.16. Discretization of a smooth sphere (Pottmann et al., 2007).

As shown in Eq. 4.8, the objective function related to the approximation of the smooth surface (A_{shape}) is formulated in relation with the number of sub-surfaces that are generated. Since, the discretization of a surface is directly related with the surface division count as shown in Fig. 4.16, an increase in the generation of sub-surfaces results with a better approximation of the initial smooth surface. This can be

considered as an architectural result, since the construction of a smooth gridshell surface with linear structural materials is practically not possible, the transformation of the smooth surface to a discrete equivalent should be as close as possible.

As shown in Eq. 4.9, 4.10 and 4.11 the constraints are defined to the optimization module. As mentioned in Chapter 4.4, the Maximum Height constraint (H_{max}) and the Useful Area constraint (R_{UA}) are user defined constraints in relation to the architectural decisions. As mentioned in Chapter 4.5, the Maximum Displacement constraint (Δ_{max}) is a constraint related with the maximum allowable deflection. As shown in Eq. 4.11, in respect to the Turkish Standard (TS EN 1990:2002), it must not exceed the limit of the largest span of the structure (l_{max}) divided by 200.

4.6.2. Optimization Procedure

Considering the multi-objective optimization problem described in the previous chapter, Octopus, multi-objective evolutionary solver for Grasshopper, is included in the workflow. As mentioned in Chapter 4.1, Octopus includes Evolutionary Algorithms to coop with various optimization problems. The main difference between traditional algorithms and Evolutionary Algorithms is Evolutionary Algorithms can be considered as dynamic since they can evolve whereas, the traditional algorithms can be considered as static due to the lack of the evolutionary operators. The Hypervolume Estimation Algorithm for Multi-objective Optimization (HypE) is the default evolutionary algorithm defined within the Octopus plug-in which is used in the optimization process. The workflow diagram of the optimization procedure is shown in Fig. 4.17.

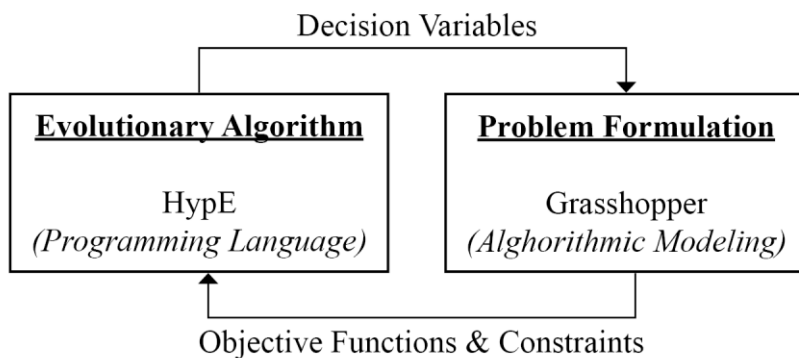


Figure 4.17. Workflow diagram of the optimization procedure.

Following the translation of the gridshell design problem into a multi-objective evolutionary optimization problem, a set of design alternatives can be achieved. In order to create a ranking between the alternatives, the range-independent Pareto non-dominated sorting method is included in the Octopus. As mentioned by Goldberg (1989), this range-independent method and variants of it are commonly used. As depicted by Bentley and Wakefield (1998), “*The fitnesses of the different objectives are treated separately and never combined, with only the value for the same objective in different solutions being directly compared. Solutions are ranked into 'non-dominated' order, with the fittest being the solutions dominated the least by others.*”

4.7. Evaluation of the Set of Alternatives

In order to evaluate the outputs of the Grasshopper workflow, the workflow is applied to two different gridshell design problems. In the first case, a rectangle is defined as the input geometry whereas in the second case, an asymmetrical quadrilateral geometry is defined as the input geometry. The dimensions and areas of the input geometries for each case are shown in Fig. 4.18.

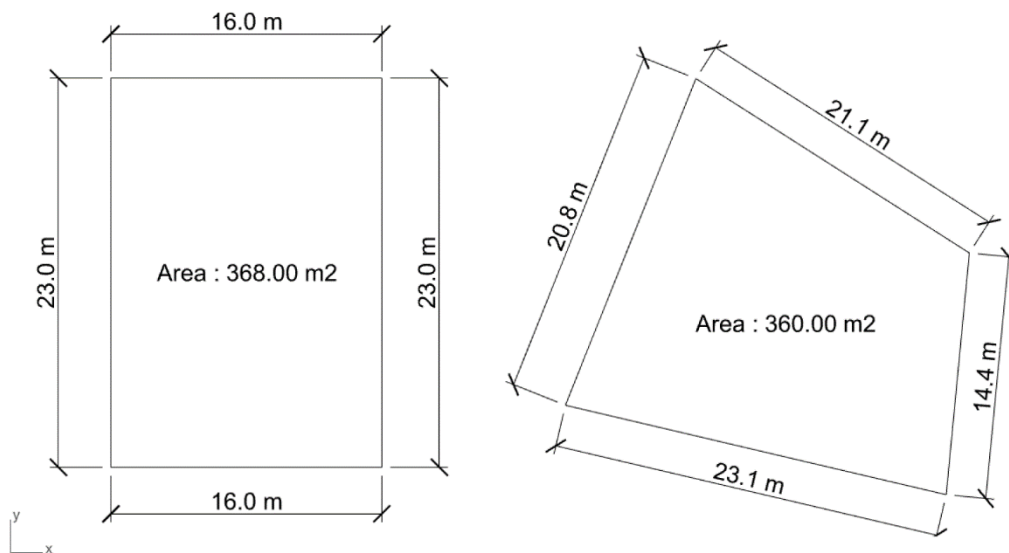


Figure 4.18. The input geometries for case 1 and case 2, respectively.

In order to define the support points, Edge Condition input mentioned in Chapter 4.3.2, is defined for each case. Following the numbering of the edges as shown in Fig. 19, the edge condition values are defined as “0” for edges numbered 0 and 2 whereas “1” for edges numbered 1 and 3. This means on edges that numbered as 1 and 3 additional

supports will be created if it is necessary whereas, on edges 0 and 2 the supports are placed on the corners only. User-defined inputs and constraints which are Useful Height (UH), Maximum Height Constraint (H_{max}) and Useful Area Constraint (R_{UA}) are defined as follows:

$$H_{useful} = 2.00 \text{ m} \tag{4.14}$$

$$H_{max} < 6.00 \text{ m} \tag{4.15}$$

$$R_{UA} > 60\% \tag{4.16}$$

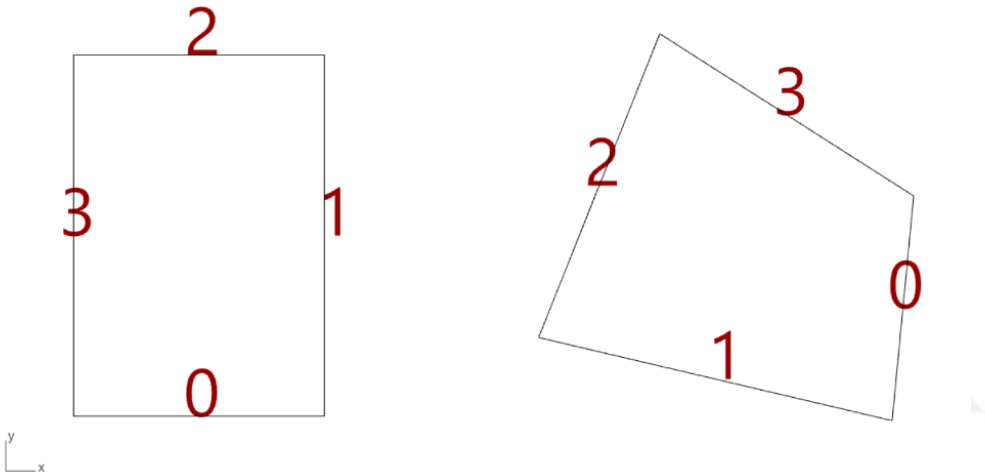


Figure 4.19. Edge numbers of input geometries in case 1 and case 2, respectively.

Following the definition of user-defined inputs and constraints, the iterative design process is started. The default settings in Octopus related to the evolutionary operations and population size is employed in the optimization process. These settings are shown in Tab. 4.4. The algorithm is stopped at the 200th generation in each case.

Table 4.4. Octopus settings.

Elitism	Mutation Probability	Mutation Rate	Crossover Rate	Population Size
0.5	0.1	0.5	0.8	100

4.7.1. Evaluation of Case 1

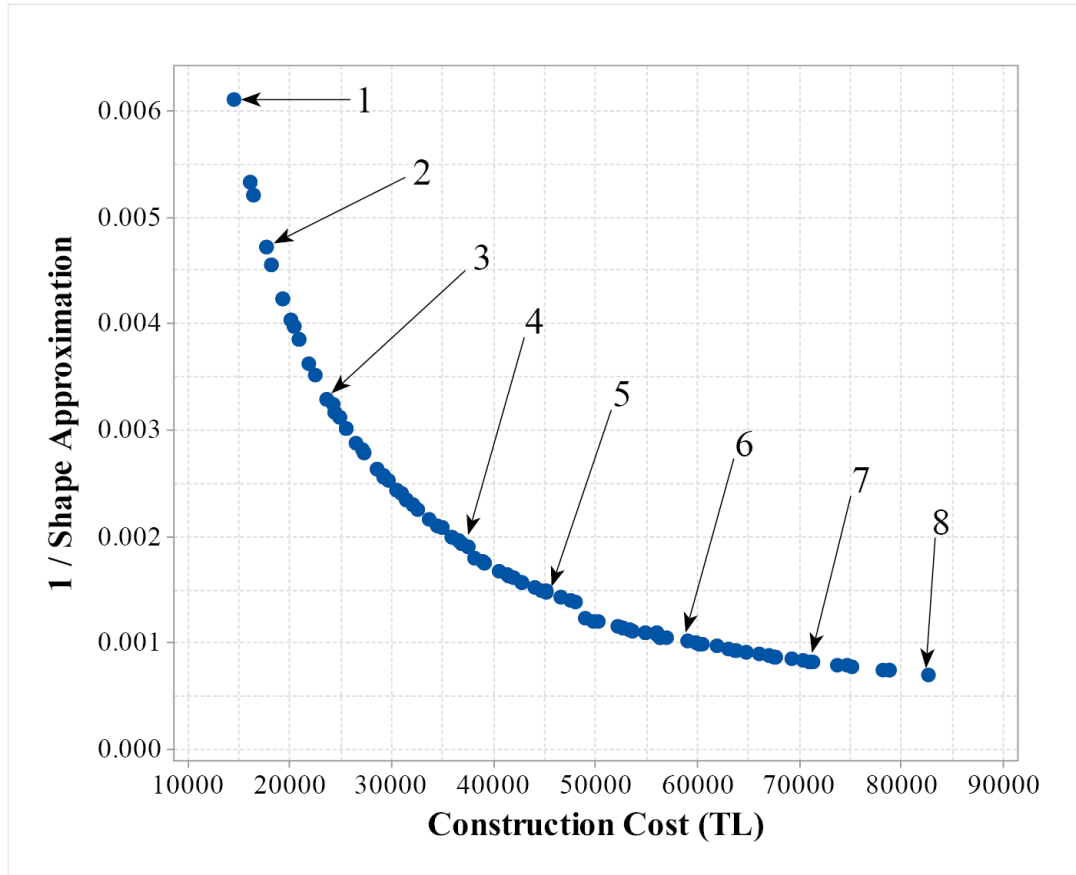


Figure 4.20. Pareto Chart of Non-Dominated Solutions at the 200th generation for case 1 with selected alternatives.

The Pareto chart of non-dominated solutions at the 200th generation for 100 population size for case 1 is given in Fig. 4.20. Following the end of the optimization process, the optimization module was able to discover 84 different design alternatives for the given problem. After the optimization process concluded for case 1, it is observed that the optimization algorithm discovered cost values within a range between 14,309.73 TL and 82,516.35 TL. Concerning Shape Approximation (A_{shape}), the optimization algorithm is suggested results between 164 and 1424. The average values for total construction cost and shape approximation are 44,583.24 TL and 699, respectively. Eight different design alternatives selected from the non-dominated solutions are presented in Fig. 4.21 and Fig. 4.22. The values of the decision variables and objective functions for selected alternatives for case 1 are presented in Tab. 4.5 and Tab. 4.6. The values of the parameters and the objective functions for all generated alternatives in case 1 are presented in Appendix 1.

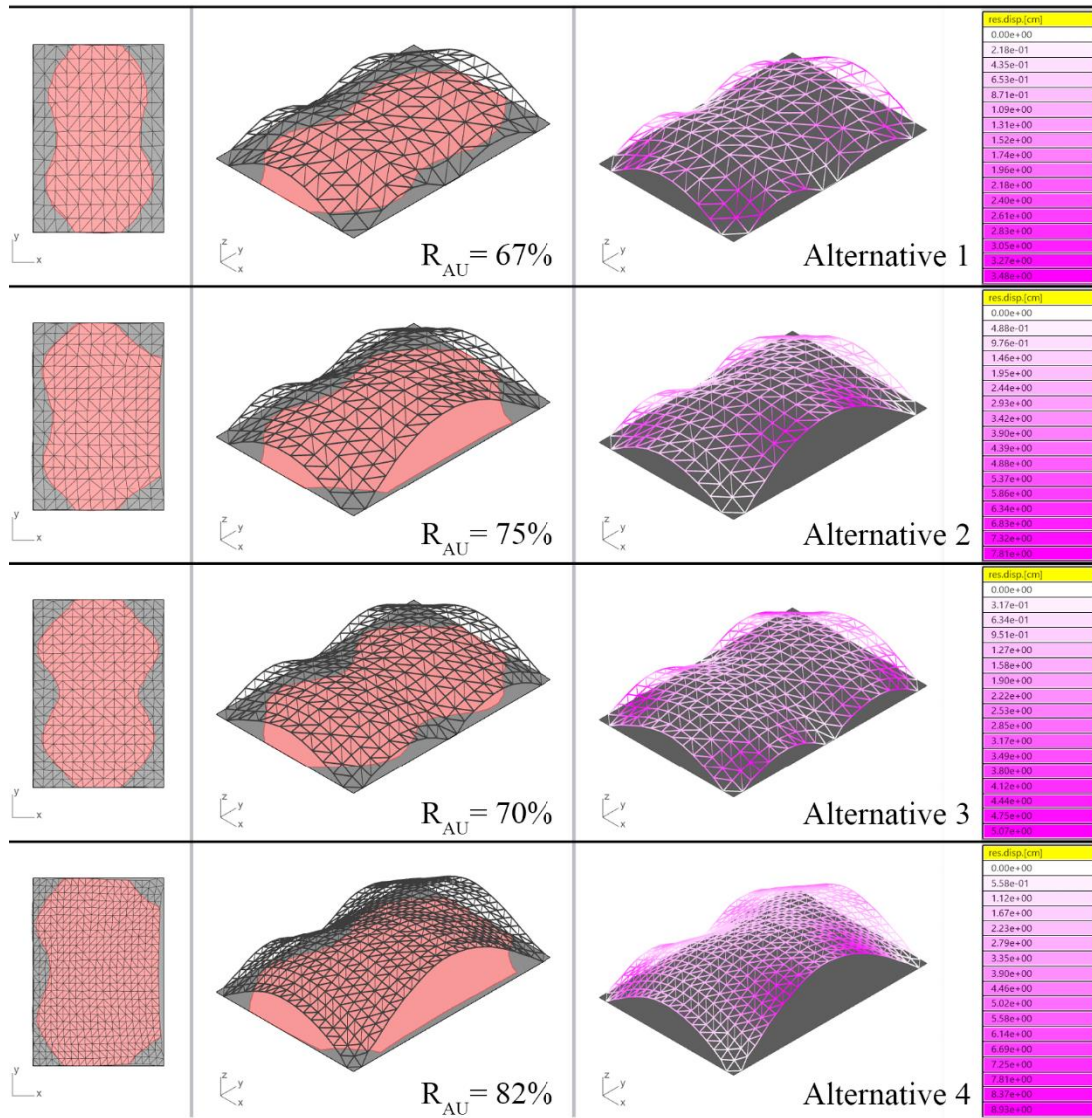


Figure 4.21. Alternatives selected from the non-dominated solutions for case 1.

Table 4.5. Values of decision variables and objective functions for selected alternatives for case 1.

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
1	11	13	1-1	1	4.6	0	14,309 TL	164
4	11	17	0-1	1	3.6	0	17,531 TL	212
12	13	21	1-1	1	3.5	0	23,558 TL	304
34	19	25	0-1	1	4.1	0	36,760 TL	516

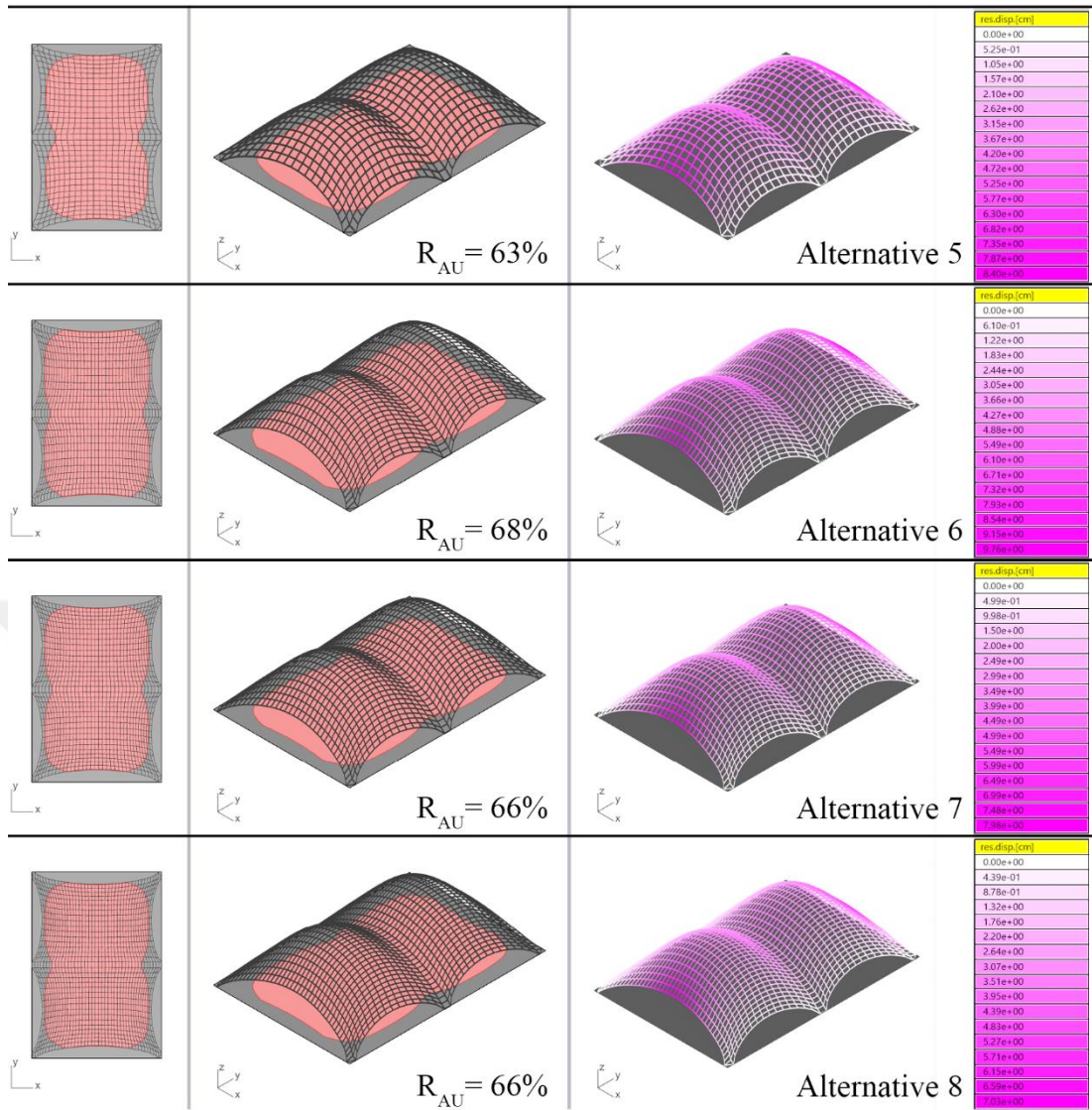


Figure 4.22. Alternatives selected from the non-dominated solutions for case 1.

Table 4.6. Values of decision variables and objective functions for selected alternatives for case 1.

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
47	21	31	1-1	0	2.8	1	46,432 TL	700
62	37	25	1-1	0	3.9	0	58,864 TL	984
78	35	33	1-1	0	3.3	0	71,280 TL	1220
84	41	33	1-1	0	3.2	0	82,561 TL	1424

When the selected alternatives which are shown in Fig. 4.21 and Tab. 4.5 are examined for case 1; it can be seen that until some point the algorithm selected Surface Triangulation decision variable (S_t) as 1 in order to adopt triangular tessellation type in lower surface division values. As mentioned in Chapter 4.2, triangular tessellation is one of the most effective tessellation types in terms of structural performance and stability, therefore, it is possible to create a gridshell model which satisfies the structural constraints with the usage of fewer materials by adopting triangular tessellation. Furthermore, by connecting the elements diagonally, diagonal stiffness is achieved and the shear forces can be transmitted from one edge of the gridshell element to the opposite one. However, in terms of Shape Approximation (A_{shape}) employing lower division values in discrete surface transformation process results with the rough equivalent of the initial surface rather than a smooth approximation.

Since the transformation of the discrete surface by employing triangular tessellation results with a satisfying structural performance, the heights of the gridshell outputs are relatively lower than the quadrilateral equivalents. This results in employing larger Force Factor values (F_x) by the algorithm in order to satisfy the Useful Area Constraint (UAC). In Tab. 4.5, it can be seen that the alternatives which created by triangular tessellation have larger F_x values than the quadrilateral tessellated surfaces which are shown in Tab. 4.6.

Regarding cross-section selection for discrete linear structural elements (A_{x-sel}), the algorithm tends to choose smaller cross-sections since it has a great effect on the construction cost calculation. In most of the alternatives, the cross-section for the beams are selected as 10cm x 5cm since the largest span can be crossed by using smaller sections in this case.

In terms of Edge Condition (E_{con}) for edges numbered a 1 and 3, the algorithm chose to create addition supports on middle points in order to satisfy the Maximum Displacement constraint in most of the alternatives. In alternatives 2 and 4, the algorithm did not employ additional supports due to they already satisfy the constraint, this results with higher Useful Floor Area (R_{UA}) ratios.

4.7.2. Evaluation of Case 2

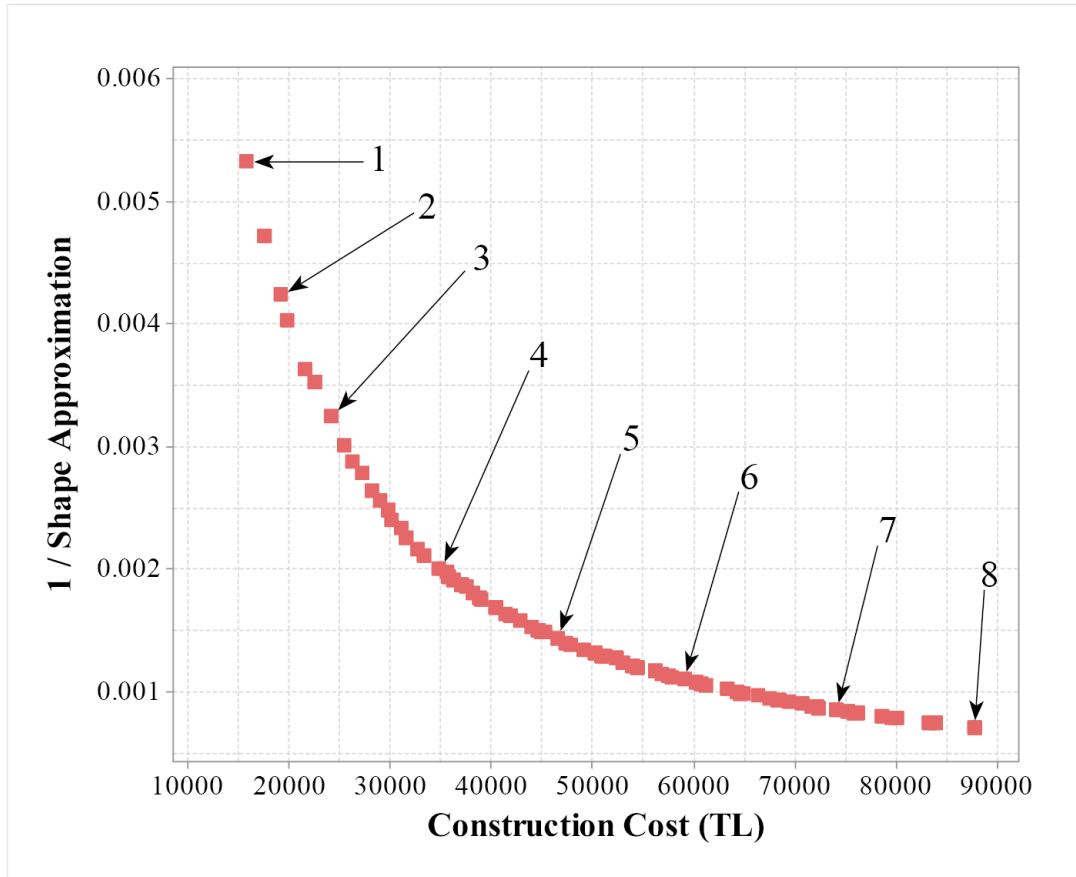


Figure 4.23. Pareto Chart of Non-Dominated Solutions at the 200th generation for case 2 with selected alternatives.

The Pareto chart of non-dominated solutions at the 200th generation for 100 population size for case 2 is given in Fig. 4.23. Following the end of the optimization process, the optimization module was able to discover 77 different design alternatives for the given problem. After the optimization process concluded for case 2, it is observed that the optimization algorithm discovered cost values within a range between 15,719.59 TL and 87,759.55 TL. In terms of Shape Approximation (A_{shape}), the optimization algorithm is suggested results between 188 and 1424. The average values for total construction cost and shape approximation are 49,891.35 TL and 760, respectively. Eight different design alternatives selected from the non-dominated solutions are presented in Fig. 4.24 and Fig. 4.25. The values of the decision variables and objective functions for selected alternatives for case 2 are presented in Tab. 4.7 and Tab. 4.8. The values of the parameters and the objective functions for all generated alternatives in case 2 are presented in Appendix 2.

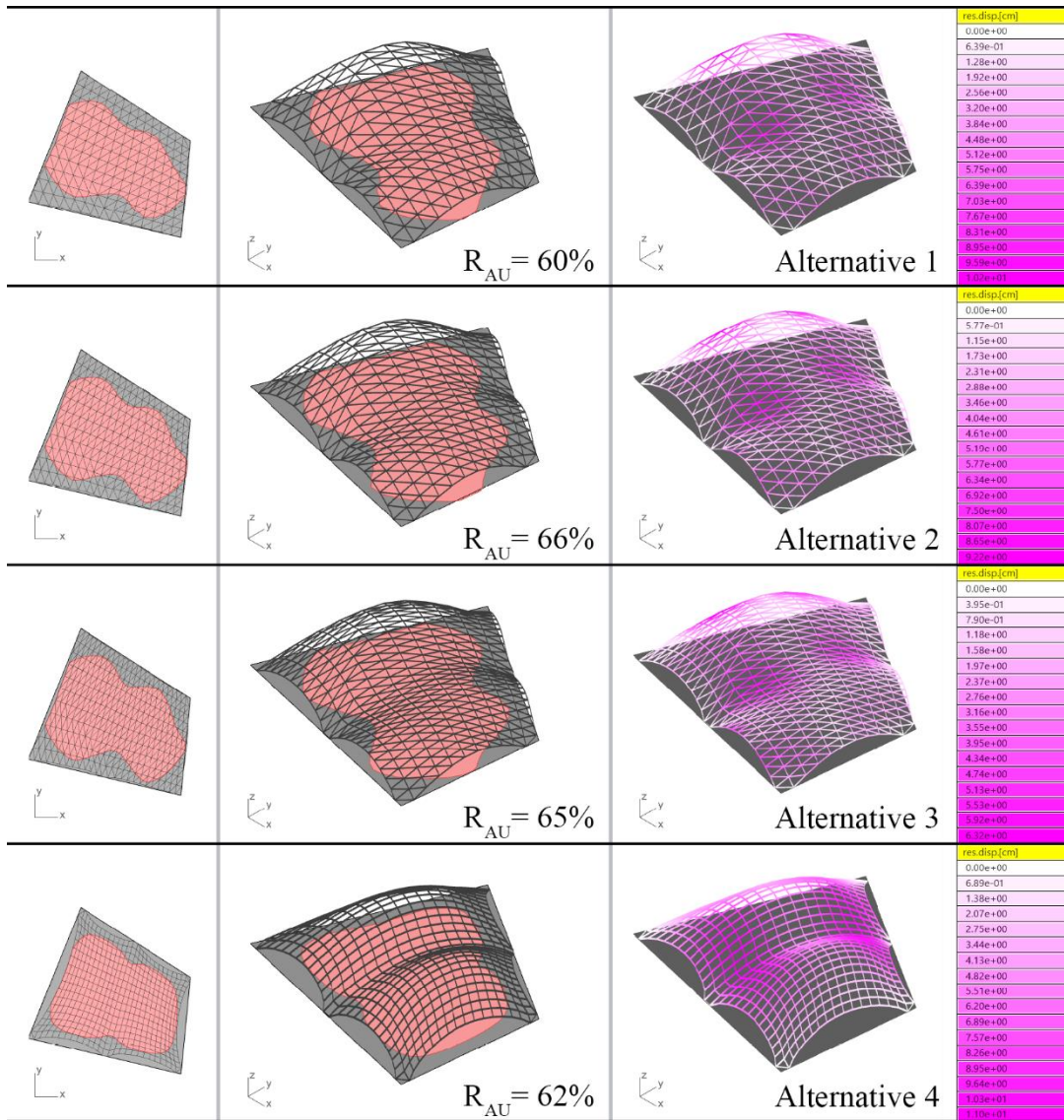


Figure 4.24. Alternatives selected from the non-dominated solutions for case 2.

Table 4.7. Values of decision variables and objective functions for selected alternatives for case 2.

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
1	11	15	1-1	1	3.2	0	15,719 TL	188
3	11	19	1-1	1	3.8	0	19,046 TL	236
7	11	25	1-1	1	3.1	0	24,095 TL	308
19	17	27	1-1	0	2.7	1	34,718 TL	500

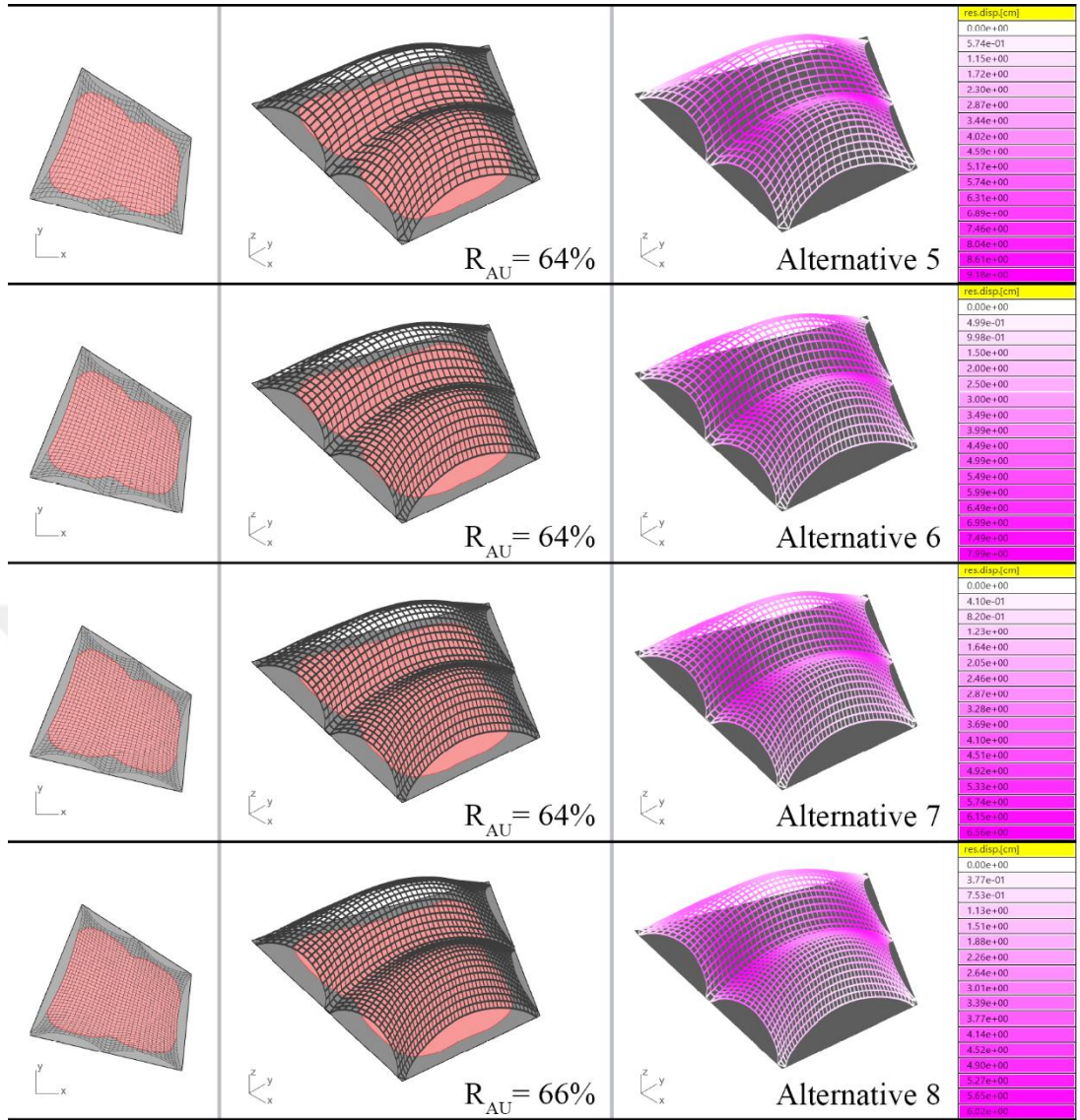


Figure 4.25. Alternatives selected from the non-dominated solutions for case 2.

Table 4.8. Values of decision variables and objective functions for selected alternatives for case 2.

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
37	21	31	1-1	0	3.1	1	46,450 TL	700
52	35	25	1-1	0	2.5	1	60,172 TL	932
69	39	29	1-1	0	2.5	1	75,101 TL	1196
77	41	33	1-1	0	2.9	1	87,759 TL	1424

For case 2, an asymmetrical quadrilateral geometry is utilized as the input shape. Similar to the case 1, the algorithm achieved smaller cost values by employing smaller values for the surface division with triangular tessellation in the expense of the Shape Approximation (A_{shape}) values. In addition, discretization of the surface by triangular tessellation allows the utilization of small cross-sections on structural elements.

In Tab. 4.7 and Tab. 4.8 as well as Appendix 3, it can be seen that the selection of the tessellation type has a direct relationship with the cross-section selection (A_{x-sel}). In every gridshell alternative which are tessellated triangularly, the cross-sections are chosen as 10cm x 5cm whereas, in the quadrilateral tessellation examples the cross-sections are selected as 15cm x 5cm. The main reason for this selection is the quadrilateral tessellation is not structurally efficient as triangular tessellation because of its geometrical nature, therefore, the utilization of smaller cross-sections in these examples cannot satisfy the Maximum Displacement Constraint. This results with larger cross-sections in most of the alternatives even though it has a strong impact on the construction cost.

Similar to the case 1, the Force Factor (F_x) values for discrete surfaces that have triangular tessellation are relatively higher than their quadrilateral equivalents. In terms of Edge Condition (E_{con}), the algorithm created additional supports in every alternative due to the irregular initial geometry in order to achieve structural stability. This results with relatively low Useful Floor Area (R_{UA}) ratios in each alternative. Moreover, due to the irregular initial boundary of the geometry and larger span, in case 2 the price range and the average price is higher than the case 1.

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

5.1. Summary

Shell structures have been used for ages to cope with a variety of different architectural design problems due to their structural stability and efficient material usage as a result of their geometrical features. They can be considered as an important part of the formal exploration process in Contemporary Architecture since they can be built in any shape from simple symmetrical geometry to more complex free-form asymmetrical shape. Gridshells can be considered as a kind of traditional shell systems. They are discrete versions of continuous shells that consist of discrete structural elements. The primary goal of this study was making research and examinations in order to develop a computational design method for gridshell design.

Gridshell systems can be utilized to cross large spans without needing any additional supports since the elements of the gridshell are the load-bearing element of the structure besides creating an overall visual appearance. Therefore, throughout the study, it has been realized that designing a gridshell is often described as a hard task since it requires the ability to manage complex structural engineering calculations while making architectural decisions in the design process. Furthermore, in the light of the reviewed literature, one may state that the recent developments in computational design techniques create a new approach to the gridshell design by utilizing novel methods related with generating and analyzing even the most complex geometries.

The questions achieve the conclusion of this study based on the information that has been gained throughout the literature review, and the obtained knowledge is expanded through the development of a computational gridshell design method. Consequently, research questions are concluded as follows:

5.2. Concluding the Research Questions

5.2.1. Research Question 1

What are the geometrical features and structural principles of shell systems?

To understand the nature of the shell systems the categorization of the shells in the current literature regarding geometry, material usage, and structural principles have been surveyed. The shell systems have been studied in two titles;

1. *'Continuous Shell Structures' and*
2. *'Gridshell Structures.'*

The similarities and differences of the two shell systems in terms of geometry and structure are depicted by examining the existing continuous shell and gridshell examples as well as the current literature. It is noticed that, even though the two structural systems are developed by following the same principle; carrying loads through their geometries, their structural behaviors are different from each other due to the utilization of their structural elements. The structural principles of shell systems are examined under two sub-titles;

1. *'Flat Plates and Plane Stress' and*
2. *'Bending Theory and Buckling.'*

Utilization of a continuous material all along the shell structure empowers the shell's ability to carry the loads through the distributed in-plane stress fields whereas, the discrete nature of a gridshell results with the resistance to the loads and out-of-plane bending through axial forces. Besides, the bending stiffness of the gridshell elements prevents any inconvenient membrane behavior.

The shell structures are gained their strength by their geometry, and they are able to possess double-curved shape. Under the title *'Surface Curvature'* the definition of double-curved geometry in the current literature, as well as the categorization of the geometrical shaped by considering the Gaussian curvatures that they possessed, are reviewed.

5.2.2. Research Question 2

What are the methods related to form-finding and structural analysis of a shell and can we determine a design method in the utilization of a gridshell?

In order to explore the form-finding and structural analysis methods related to designing a gridshell, traditional and computational methods are surveyed. The methods have been studied in four titles;

1. *'Hooke's Hanging Chain Law,'*
2. *'Physical Modeling,'*
3. *'Computational Form Finding' and*
4. *'Finite Element Method and Computational Structural Analysis.'*

Hooke's Hanging Chain Law is one of the most essential fundamental approaches in physics-based form-finding process of a shell structure. The simplicity of the idea makes its application possible to numerous different methods. From traditional form-finding methods to more developed computational form-finding methods, the principles of this law are utilized. One may state that one of the most used traditional form-finding methods can be considered as the Physical Modeling Method that is used by various designers throughout history. The Physical Modeling method has been studied in two titles;

1. *'Scale-Independent Models' and*
2. *'Scale-Dependent Models.'*

Prior to the development of computers, the calculations related to the structural stability of a shell were done manually. Alongside the mathematical equations in the context of a specific shell design problem, the small-scale physical models of the relative structure were also created. Building small-scale physical models of a structure give an insight into the construction of the full-size structure. Therefore, the method is often included in the architectural decision-making process. The structural complexity of the design problem affects the physical modeling approach. Because of this situation, some design problems require the scaling of the applied forces to create a precise test

model. A physical modeling method is a viable approach that is still used by architects and engineers.

The translation of architectural design problems to the digital environment alongside the development of computational design programs has created an opportunity to utilize interactive form-finding methods in the design process. In the literature review, it has been seen that the computational methods are examined under two categories; '*Geometry Oriented Methods*' and '*Material Oriented Methods*.' One of the most used methods for each category are examined, and the approaches are compared to each other. The subjected methods are;

1. '*Force Density Method*' and
2. '*Dynamic Relaxation Method*,' respectively.

In the literature, it is stated the outcome of the material-oriented form-finding methods can be considered more accurate. Since the material properties are known for the study, the *Dynamic Relaxation Method* is chosen as the form-finding method in the context of this thesis.

Following the review of the existing structural analysis methods in the current literature, the '*Finite Element Method*' is a viable method for calculating nodal displacements, stress, and stability of the structural elements. The discrete and complex nature of a gridshell makes a good case for '*Finite Element Analysis*.' Considering these criteria alongside the availability of the FEA programs, the *Finite Element Method* is chosen for structural analysis part of this study.

5.2.3. Research Question 3

Can we develop a computational workflow for a generic gridshell design process that includes the form-finding process as well as an optimization between geometrical features and construction cost?

Following the literature review to understand the principles of gridshell structures, form-finding methods and structural analysis a computational method for gridshell design has been developed. The method consists of five consecutive steps as follows;

1. *'Initial Geometry Creation,'*
2. *'Physics-based Form Finding Procedure,'*
3. *'Evaluation of the User Defined Constraints,'*
4. *'Assembly of the Structural Model and Structural Performance Evaluation' and*
5. *'Optimization of Construction Cost and Shape Approximation.'*

The *'Initial Geometry Creation'* step creates the adaptive characteristic of the computational workflow. The step includes user decisions related to a gridshell design problem. Thanks to the generative approach of the workflow, the method can be applied to various gridshell design problems following the definition of the area boundaries and desired architectural criteria. In this step, an intermediate flat shape of the respective gridshell that is divided into sub-surfaces is created in relation to the user-defined inputs.

The second step of the workflow which is named as *'Physics-based Form Finding Procedure'* includes a real-world physics simulation to create a discrete gridshell geometry. In this step, the *Dynamic Relaxation* procedure is applied to the flat sub-surfaces which are generated on the first step. In small time increments, the sub-surfaces collapse under their self-weight in accordance with the initial surface until they form a shape that is statically in equilibrium. When the principles of *Hooke's Hanging Chain Law* are considered, the reverse shape of the collapsed surface defines a shell shape that is statically in equilibrium. By including a physics simulation to the workflow, an equilibrium shape for a gridshell is acquired since the probability of

satisfying structural requirements with employing an arbitrary surface shape is relatively low.

The third step of the workflow is the evaluation of the gridshell's shape considering the user-defined criteria. Since the shape of the gridshell is the result of the Dynamic Relaxation procedure, constraints are required in order to evaluate the output if it satisfies the desired criteria. The iterative nature of the computational workflow requires this step to create gridshell alternatives that can be considered as suitable in the context of the given design problem. This step is directly related to the optimization part of the workflow. By including this step to the workflow, the iterative process does not proceed to the next step and restarts until the user-defined criteria are satisfied.

In the fourth step of the workflow, a *Finite Element Model* is assembled in accordance with the generated gridshell shape. By using the *Finite Element Analysis*, the displacement analysis is performed. In relation to the Turkish Standards, a structural constraint is defined in order to limit the maximum allowable displacement. The constraint is included in the iterative workflow in order to consider the outputs which do not meet the serviceability criteria as unfeasible.

In the fifth step of the workflow, the iterative character of the method is created by including an optimization process. Two objective functions are considered for the optimization which are; minimizing the construction cost of the gridshell while creating a discrete transformation of the smooth parent surface as close as possible. Since the generated gridshell alternatives consist of linear structural elements and planar faces, they cannot possess a smooth surface. The discretization of a smooth surface is necessary to define the form of a gridshell. In the discrete transformation process, the amount of sub-division of the parent surface plays a significant role concerning shape approximation. However, as the number of sub-surfaces increases, the construction cost also rises due to the increase in the numbers of the structural elements and connection nodes. Therefore, two conflicting objectives make a good case for a multi-objective evolutionary optimization process.

The objective of the study was determined as creating a computational workflow with an iterative characteristic that can be adapted to different gridshell design problems. As a result of the workflow, a set of gridshell alternative can be achieved that requires

the user-defined criteria. In the end, it is possible to select suitable solutions for the given problem by considering the decisions related to construction cost and geometry.

The computational workflow has been applied to two different cases. In these cases, one symmetrical rectangle geometry and one asymmetrical quadrilateral geometry have been defined. Besides the different boundary conditions, the user-defined constraints have been defined as the same for each case. The workflow has managed to achieve a set of gridshell design alternatives in each case. It has been observed that even though the algorithm is stopped at the 200th generation in each case, the resulting number of design alternatives were different. In light of this information, one may state that the number of generated design alternatives depends on the features of the given design problem.

In each case, the workflow has defined triangular tessellation in terms of surface division type in order to cross the relative span by creating the minimum number of structural elements. Until a threshold, the alternatives have been created by using triangular tessellation to meet the minimum cost objective. This is mainly because, although the quadrilateral tessellation type results with a lesser number of elements, the lack of the diagonal elements result with larger deformation than the allowable limit.

It has been observed that with the increase in the number of sub-surfaces the optimization algorithm has preferred the quadrilateral tessellation type over the triangulation since the quadrilateral tessellated gridshell alternatives result with lower construction cost than their equivalent triangulated alternatives. Also, the increase in the number of sub-surfaces has resulted in a closer approximation of the parent shell surface in the expense of the construction cost.

5.3. Research Contribution

Architectural design can be described as the creative progression towards the achievement of specific design objectives in the context of a particular design problem. Every architectural design problem requires a conceptual idea development process in order to coop with the specific characteristics of the given problem. In the conceptual design process, the design parameters and constraints are defined in order to satisfy the design objectives. By controlling these parameters within the limits of the constraints, different responses to an architectural design problem can be made. These

sketch responses are an essential part of the idea development process since they allowed architects to visualize and evaluate the results of their decisions.

The conceptual design phase of an architectural problem can be richened with the inclusion of digital sketches. By considering this approach, a computational workflow related to the conceptual design phase of gridshells is developed in this thesis. For this study, by extending the sketches to the digital environment, the gridshell design is not only considered regarding form and geometry, but also instant feedbacks related with the structural performance is included to the workflow.

The proposed method of this study can be considered as an innovative decision-making process for designers and architects. The form-finding process in this method is based on real-world physics simulations which result in shapes that are structurally in equilibrium. Besides the Dynamic Relaxation process which is included in the form-finding phase of the study, the structural performance evaluations of the generated forms are done by using Finite Element Methods. Besides, by creating the consecutive steps in the workflow that can be controlled parametrically, it is possible to create alternative gridshell designs by changing relevant parameters. This aspect of the workflow is translated to an optimization process in order to create a set of alternative gridshell designs for a given design problem.

The computational and iterative workflow is developed by using digital tools for physics simulations, structural analysis, and optimization, Kangaroo, Karamba, and Octopus, respectively. By introducing these tools to the design method, the users can adapt an intuitive approach in order to see the relationship between the formal exploration process and the structural performance. Furthermore, a response to the defined design problem can be chosen from the set of alternative gridshell designs as the result of the optimization process that satisfies the related design constraints. Therefore, one may state that, by including this method to the conceptual design phase, the users can generate multiple gridshell design alternatives without needing any advanced knowledge of structural analysis and optimization.

The adaptive nature of the proposed method draws its power from the consecutive computational steps which are included in the workflow. By defining design-specific input parameters and constraints, the method can be utilized in various gridshell design problems with different characteristics. In addition, by including constraints related to

the architectural decision, it is possible to develop a design by considering the architectural aspects of the structure such as the architectural program and function.

As a result of the optimization procedure, a set of gridshell design alternatives can be acquired for a given design problem. The user can evaluate these alternatives in terms of construction cost and gridshell's overall geometry. Theoretically, any solution that is acquired by using this method can be utilized as the response of the design problems. This is because the alternatives which are included in the resulted set satisfy the constraints of the problem. However, a decision should be made considering the overall cost and form. Since the approximation of the discrete gridshell geometry to its parent smooth surface affect the construction cost, the decision should be made in accordance with the project's budget.

5.4. Recommendations and Further Research

The thesis focuses on developing a computational design method for generic gridshell design problems. The method includes a physics-based form-finding procedure, a finite element structural analysis workflow, and an optimization module. By considering the design related objectives and constraints, the method can be customized and applied to different gridshell design problems.

In the context of this thesis, an application and design method has been developed in order to create a gridshell that consists of unstrained, linear wooden structural elements. The utilized generic wood material is chosen due to its availability within the digital tools which are used in this study. Different types of wood materials can be studied, and because of the difference in their material properties, the structural analysis may result different accordingly. In addition, the utilization of unstrained elements creates a possibility to use different types of materials such as steel and structural aluminum. Furthermore, circular cross-sections can be utilized with the introduction of these materials. This situation makes the prediction of the bending behavior of the structure more accurate and diminishes the effect of human error in the construction process.

A gridshell's structural performance highly depends on the characteristics of its structural elements as well as the overall tessellation. Any change in the elements' geometries such as their dimensions and shapes of their cross-sections effects the overall structural performance. Each part of the gridshell can be analyzed and

redesigned through an optimization process in order to achieve a satisfying structural performance.

Due to their complex erection schemes and complicated force calculations, the strained gridshell systems are excluded from the scope of this study. The elastic nature of the wood material allows the utilization of pre-tensioned linear wooden elements that may result in more resistant gridshell structures. Thus, they can cross larger spans with lower material consumption. However, this type of study should include an innovative method related to the erection process of the flat-gridshell surface and calculations should be done accordingly.

In this study, a generic cover material is defined as glass panels with structural aluminum frames for the gridshell alternatives. The calculations related to structural stability are done accordingly. In these calculations, the self-weight of the gridshell structure and downward wind forces are defined as the applied loads. For further researches, the uplift wind forces and earthquake loads can also be integrated into the context of the study. Besides, different cover materials can be studied in order to enrich the architectural decision-making process.

The gridshell design problem is formulated as a multi-objective evolutionary optimization problem, and HypE algorithm is employed in the optimization process. The performance of the algorithm is not evaluated since only one optimization algorithm is used throughout the study. The algorithm is chosen considering its availability within the Octopus plug-in. A further study can be done by using multiple evolutionary algorithms in order to compare their performance.

REFERENCES

- Addis, B. (2014). Physical modelling and form finding. In *Shell Structures for Architecture: Form Finding and Optimization* (pp. 47-58): Routledge.
- Adriaenssens, S., Block, P., Veenendaal, D., & Williams, C. (2014). *Shell structures for architecture: form finding and optimization*: Routledge.
- Argyris, J. H., & Kelsey, S. (1960). *Energy theorems and structural analysis* (Vol. 960): Springer.
- Asmaljee, Z. (2014). *Form-finding of thin shell structures*.
- Bächer, M., & Burkhardt, B. (1978). Multihalle Mannheim-Dokumentation über die Planungs- und Ausführungsarbeiten an der Multihalle Mannheim. *Stuttgart: Krämer*.
- Bentley, P. J., & Wakefield, J. P. (1998). Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In *Soft computing in engineering design and manufacturing* (pp. 231-240): Springer.
- Block, P., & Veenendaal, D. (2014). Comparison of Form Finding Methods. In *Shell Structures for Architecture: Form Finding and Optimization* (pp. 115-130): Routledge.
- Cevizci, E., & Kutucu, S. (2017). Particle-based Structural System Modelling on Asymmetric Shaped Generic Shell Design. *MSTAS 2017*, 137.
- Chilton, J., & Isler, H. (2000). *Heinz isler*: Thomas Telford.
- Cotterell, B., & Kamminga, J. (1992). *Mechanics of pre-industrial technology: An introduction to the mechanics of ancient and traditional material culture*: Cambridge University Press.
- Dimcic, M. (2011). *Structural optimization of grid shells based on genetic algorithms*.
- Engel, H., & Bandel, H. (1967). *Tragsysteme: structure systems*: Deutsche Verlags-Anstalt.
- Fleischmann, M., Knippers, J., Lienhard, J., Menges, A., & Schleicher, S. J. A. D. (2012). Material behaviour: embedding physical properties in computational design processes. *82(2)*, 44-51.
- Fresl, K., Gidak, P., & Vrančić, R. J. G. (2013). Generalized minimal nets in form finding of prestressed cable nets. *65(08)*, 707-720.
- Fuller, B. (1982). Self-disciplines of Buckminster Fuller. In *Critical Path*: St. Martin's Press New York.
- Garcia, J. R. (2011). *Numerical study of dynamic relaxation methods and contribution to the modelling of inflatable lifejackets*. Université de Bretagne Sud,

- Gidak, P., & Fresl, K. (2012). *Programming the force density method*. Paper presented at the IASS-APCS 2012, From spatial structures to space structures.
- Happold, E., & Liddell, W. (1975). Timber lattice roof for the Mannheim Bundesgartenschau. *The structural engineer*, 53(3), 99-135.
- Hecker, C. (1996). Physics, the next frontier. *Game Developer*, 12-20.
- Hoefakker, J. (1900). *Theory review for cylindrical shells and parametric study of chimneys and tanks*: Eburon Uitgeverij BV.
- Huerta, S. (2006). Structural design in the work of Gaudí. *Architectural science review*, 49(4), 324-339.
- Hüttner, M., Máca, J., & Fajman, P. (2014). *Analysis of cable-membrane structures using the dynamic relaxation method*. Paper presented at the Proceedings of the 9th International Conference on Structural Dynamics.
- Kaveh, A. (2014). *Computational structural analysis and finite element methods*: Springer.
- Kleiner, F. S., & Mamiya, C. (2005). *Gardner's Art through the Ages: The Western Perspective, Volume II*: Belmont, California: Wadsworth Publishing.
- Knippers, J. (2000). Johann Wilhelm Schwedler: Vom Experiment zur Berechnung. *Deutsche Bauzeitung*, 134, 105-110.
- Lewis, W. J. (2003). *Tension structures: form and behaviour*: Thomas Telford.
- Linkwitz, K. (2014). Force density method. *Shell Structures for Architecture: Form Finding Optimization*, 59-71.
- Lozano-Galant, J. A., & Payá-Zaforteza, I. J. E. S. (2011). Structural analysis of Eduardo Torroja's Frontón de Recoletos' roof. 33(3), 843-854.
- Maden, F. (2015). Novel design methodologies for transformable doubly-ruled surface structures.
- McNeel, R. (1993). Rhinoceros: NURBS modeling for Windows. *Computer software*.
- Owen, D., & Feng, Y. (2012). Fifty years of finite elements—a solid mechanics perspective. *Theoretical Applied Mechanics Letters*, 2(5).
- Pedron, C. (2006). *An innovative tool for teaching structural analysis and design*: vdf Hochschulverlag AG.
- Piegl, L., & Tiller, W. (1987). Curve and surface constructions using rational B-splines. *Computer-Aided Design*, 19(9), 485-498.
- Piker, D. (2013). Kangaroo: form finding with computational physics. *Architectural Design*, 83(2), 136-137.
- Poleni, G., & Poleni, J. (1748). *Memorie storiche della gran cupola del tempio*

Vaticano e de'danni di essa e detristoramenti loro, divisi in libri 5: Nella Stamperia del seminario.

- Pottmann, H., Asperl, A., Hofer, M., & Kilian, A. (2007). *Architectural geometry* (Vol. 724): Bentley Institute Press Exton.
- Preisinger, C., & Heimrath, M. (2014). Karamba—A Toolkit for Parametric Structural Design. *Structural Engineering International*, 24(2), 217-221.
- Przemieniecki, J. S. (1985). *Theory of matrix structural analysis*: Courier Corporation.
- Rutten, D. (2007). *Grasshopper & galapagos*. Seattle: Robert McNeel Associates.
- Rutten, D. (2013). Galapagos: On the logic and limitations of generic solvers. *Architectural Design*, 83(2), 132-135.
- Scarre, C. (2002). *The seventy wonders of the ancient world: the great monuments and how they were built*: Thames & Hudson.
- Schek, H. (1974). The force density method for form finding and computation of general networks. *Computer methods in applied mechanics engineering*, 3(1), 115-134.
- Southern, R. (2011). The force density method: a brief introduction.
- Stevens, K. (1981). The Visual Interpretation of Surface. *Artificial Intelligence*, 17(19m), 47-73.
- Stevin, S. (1586). 1586. De beghinselen der weeghconst. *The Principal Works of Simon Stevin, 1*.
- Timoshenko, S., & Goodier, J. (1970). *Theory of Elasticity*, (216). In: McGraw-Hill.
- Tomlow, J. (2011). Gaudí's reluctant attitude towards the inverted catenary. *Proceedings of the Institution of Civil Engineers-Engineering History and Heritage*, 164(4), 219-233.
- Toussaint, M. (2007). A design tool for timber gridshells: the development of a grid generation tool.
- TSE. (2002). TS EN 1990. In *Eurocode - Yapı Tasarımının Esasları*. Ankara: Türk Standardları Enstitüsü.
- TSE. (2007). TS EN 1991-1-4. In *Eurocode 1 – Actions on Structures– Part 1-4: General Actions – Wind Actions*. Ankara.
- Turner, M. (1956). Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, 23(9), 805-823.
- Veenendaal, D., & Block, P. (2012). 35 Computational form-finding of fabric formworks: an overview and discussion.
- Veenendaal, D., & Block, P. (2012). An overview and comparison of structural form

finding methods for general networks. *International Journal of Solids Structures*, 49(26), 3741-3753.

Vejrum, P. (2013). *A Generative Gridshell Form Finding Tool: Development of a generative parametric gridshell form finding tool*. (Master Thesis), Aarhus University,

Vierlinger, R., & Schneider, C. G. (2014). Octopus. Vienna. www.food4rhino.com/app/octopus. Bollinger+ Grohmann Engineers.

Williams, C. (2014). What is a shell? In *Shell Structures for Architecture* (pp. 35-46): Routledge.

Zienkiewicz, O. C. (2000). Achievements and some unsolved problems of the finite element method. *International Journal for Numerical Methods in Engineering*, 47(1-3), 9-28.

Zienkiewicz, O. C., & Cheung, Y. (1967). *The Finite Element Method in Engineering Scheme*.

APPENDIX 1 – Values of Objective Functions and Parameters for Case 1

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
1	11	13	1-1	1	4.6	0	14309.73	164
2	11	15	0-1	1	4.9	0	15915.80	188
3	13	13	1-1	1	4.6	0	16336.74	192
4	11	17	0-1	1	3.7	0	17531.83	212
5	13	15	1-1	1	4.5	0	18127.63	220
6	11	19	0-1	1	3.6	0	19156.34	236
7	13	17	1-1	1	3.7	0	19929.06	248
8	15	15	1-1	1	4.9	0	20354.01	252
9	11	21	1-1	1	3.3	0	20788.01	260
10	13	19	1-1	1	3.3	0	21739.76	276
11	15	17	1-1	1	3.5	0	22342.37	284
12	13	21	1-1	1	3.5	0	23558.53	304
13	11	25	0-1	1	3.6	0	24068.48	308
14	15	19	1-1	1	4.6	0	24340.39	316
15	17	17	1-1	1	3.5	0	24768.45	320
16	13	23	0-1	1	3.5	0	25384.26	332
17	15	21	0-1	1	4.8	0	26347.01	348
18	17	19	1-1	1	3.6	0	26955.01	356
19	13	25	1-1	1	3.9	0	27215.98	360
20	15	23	1-1	1	4.4	0	28361.22	380
21	13	27	1-1	1	4.8	0	29052.88	388
22	17	21	0-1	1	4.8	0	29150.42	392
23	19	19	1-1	1	3.2	0	29580.96	396
24	15	25	1-1	1	4.8	0	30382.11	412
25	13	29	1-1	1	4.5	0	30894.24	416
26	17	23	0-1	1	4.7	0	31353.82	428
27	19	21	1-1	1	4.9	0	31966.18	436
28	15	27	1-1	1	4.8	0	32408.86	444
29	17	25	0-1	1	4.7	0	33564.35	464
30	19	23	1-1	1	4.6	0	34359.55	476

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
31	21	21	1-1	1	4.6	0	34792.10	480
32	17	27	1-1	1	4.7	0	35781.24	500
33	15	31	1-1	1	4	0	36477.14	508
34	19	25	0-1	1	4.1	0	36760.33	516
35	21	23	1-1	1	4.2	0	37376.27	524
36	21	25	1-1	0	2.9	1	38750.57	568
37	17	31	1-1	0	3.2	1	38995.69	572
38	19	29	1-1	0	2.9	1	40353.47	596
39	17	33	1-1	0	4.6	1	41156.48	608
40	21	27	1-1	0	4.6	1	41311.31	612
41	23	25	1-1	0	4.2	1	41869.20	620
42	19	31	1-1	0	3.9	1	42714.22	636
43	21	29	1-1	0	4.7	1	43872.01	656
44	23	27	1-1	0	3.9	1	44629.85	668
45	25	25	1-1	0	2.9	1	44987.73	672
46	19	33	1-1	0	3.9	1	45074.93	676
47	21	31	1-1	0	2.8	1	46432.68	700
48	23	29	1-1	0	4.6	1	47390.47	716
49	25	27	1-1	0	3.9	1	47948.31	724
50	23	33	1-1	0	4.8	0	48807.76	812
51	25	31	1-1	0	4.8	0	49712.94	828
52	27	29	1-1	0	4.6	0	50218.13	836
53	35	23	1-1	0	3.9	0	52079.27	860
54	25	33	1-1	0	3.1	0	52553.28	880
55	27	31	1-1	0	3	0	53258.45	892
56	29	29	1-1	0	4.6	0	53563.64	896
57	37	23	1-1	0	3.3	0	54824.71	908
58	41	21	1-1	0	3.3	0	55875.32	920
59	35	25	1-1	0	3.9	0	55919.51	932
60	27	33	1-1	0	3.3	0	56298.76	948
61	29	31	1-1	0	4.2	0	56803.93	956
62	37	25	1-1	0	3.9	0	58864.92	984

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
63	35	27	1-1	0	3.9	0	59759.73	1004
64	29	33	1-1	0	4.8	0	60044.21	1016
65	31	31	1-1	0	3.6	0	60349.37	1020
66	39	25	1-1	0	4.2	0	61810.31	1036
67	37	27	1-1	0	3.9	0	62905.13	1060
68	35	29	1-1	0	3	0	63599.96	1076
69	31	33	1-1	0	3.1	0	63789.63	1084
70	41	25	1-1	0	4.6	0	64755.67	1088
71	39	27	1-1	0	4.3	0	66050.50	1116
72	37	29	1-1	0	3.6	0	66945.33	1136
73	35	31	1-1	0	4.2	0	67440.17	1148
74	33	33	1-1	0	3.1	0	67535.02	1152
75	41	27	1-1	0	4.3	0	69195.85	1172
76	39	29	1-1	0	3	0	70290.69	1196
77	37	31	1-1	0	3.6	0	70985.53	1212
78	35	33	1-1	0	3.3	0	71280.38	1220
79	41	29	1-1	0	4.2	0	73636.02	1256
80	39	31	1-1	0	4.2	0	74530.87	1276
81	37	33	1-1	0	3.3	0	75025.72	1288
82	41	31	1-1	0	4.2	0	78076.18	1340
83	39	33	0-1	0	2.9	0	78771.05	1356
84	41	33	1-1	0	3.2	0	82516.35	1424

APPENDIX 2 – Values of Objective Functions and Parameters for Case 2

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
1	11	15	1-1	1	3.2	0	15719.59	188
2	11	17	1-1	1	3.2	0	17377.86	212
3	11	19	1-1	1	3.8	0	19046.36	236
4	13	17	1-1	1	3.1	0	19680.79	248
5	13	19	1-1	1	3.9	0	21533.04	276
6	11	23	1-1	1	3.1	0	22406.77	284
7	11	25	1-1	1	3.1	0	24095.87	308
8	13	23	1-1	1	3.1	0	25265.57	332
9	15	21	1-1	1	3.8	0	26087.97	348
10	13	25	1-1	1	3.9	0	27142.77	360
11	15	23	1-1	1	3.1	0	28144.98	380
12	17	21	1-1	1	2.5	0	28798.93	392
13	11	33	1-1	0	2.6	1	29684.36	404
14	13	29	1-1	0	2.5	1	29968.34	416
15	17	23	1-1	1	3.1	0	31042.80	428
16	15	27	1-1	0	2.5	1	31446.44	444
17	17	25	1-1	0	2.5	1	32524.65	464
18	19	23	1-1	0	2.2	1	33202.95	476
19	17	27	1-1	0	2.7	1	34718.61	500
20	15	31	1-1	0	3.1	1	35434.34	508
21	19	25	1-1	0	2.7	1	35596.86	516
22	21	23	1-1	0	2.2	1	36075.21	524
23	17	29	1-1	0	2.5	1	36912.48	536
24	15	33	1-1	0	2.4	1	37428.16	540
25	19	27	1-1	0	2.5	1	37990.70	556
26	21	25	1-1	0	2.4	1	38669.00	568
27	23	23	1-1	0	2	1	38947.37	572
28	19	29	1-1	0	3	1	40384.47	596
29	21	27	1-1	0	2.9	1	41262.73	612
30	23	25	1-1	0	2.5	1	41741.05	620

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
31	19	31	1-1	0	2.5	1	42778.18	636
32	21	29	1-1	0	3.9	1	43856.41	656
33	23	27	1-1	0	2.8	1	44534.69	668
34	25	25	1-1	0	2.5	1	44813.04	672
35	19	33	1-1	0	2.7	1	45171.83	676
36	21	31	1-1	0	3.9	1	46450.03	700
37	23	29	1-1	0	2.9	1	47328.29	716
38	25	27	1-1	0	2.5	1	47806.60	724
39	21	33	1-1	0	3.9	1	49043.61	744
40	23	31	1-1	0	2.1	1	50121.84	764
41	25	29	1-1	0	2.6	1	50800.12	776
42	27	27	1-1	0	2.6	1	51078.44	780
43	35	21	1-1	0	2.4	1	52185.46	788
44	23	33	1-1	0	2.4	1	52915.35	812
45	25	31	1-1	0	2.1	1	53793.60	828
46	27	29	1-1	0	3.9	1	54271.90	836
47	35	23	1-1	0	2.1	1	56178.76	860
48	25	33	0-1	0	2.4	1	56787.06	880
49	27	31	1-1	0	2.1	1	57465.33	892
50	29	29	1-1	0	2.7	1	57743.64	896
51	37	23	1-1	0	2.2	1	59050.46	908
52	35	23	1-1	0	2.5	1	60172.05	932
53	27	33	1-1	0	2.9	1	60658.73	948
54	29	31	1-1	0	2.9	1	61137.01	956
55	37	25	1-1	0	2.5	1	63243.71	984
56	35	27	1-1	0	2.5	1	64165.33	1004
57	29	33	1-1	0	2.9	1	64530.36	1016
58	31	31	1-1	0	2.5	1	64808.66	1020
59	39	25	1-1	0	3.9	1	66315.32	1036
60	37	27	1-1	0	2.6	1	67436.95	1060
61	35	29	1-1	0	2.7	1	68158.60	1076
62	31	33	0-1	0	2.9	1	68401.97	1084

No	N_{u-div}	N_{v-div}	E_{con} 1-3	S_t	F_x	A_{x-sel}	C_g	A_{shape}
63	41	25	1-1	0	2.9	1	69386.89	1088
64	39	27	1-1	0	2.6	1	70708.52	1116
65	37	29	1-1	0	3.9	1	71630.18	1136
66	35	31	1-1	0	2.5	1	72151.84	1148
67	33	33	1-1	0	2.4	1	72273.54	1152
68	41	27	1-1	0	2.7	1	73980.07	1172
69	39	29	1-1	0	2.5	1	75101.72	1196
70	37	31	0-1	0	2.6	1	75823.39	1212
71	35	33	1-1	0	2.5	1	76145.08	1220
72	41	29	1-1	0	2.9	1	78573.24	1256
73	39	31	1-1	0	2	1	79494.91	1276
74	37	33	1-1	0	2.7	1	80016.60	1288
75	41	31	1-1	0	2.5	1	83166.40	1340
76	39	33	1-1	0	2.5	1	83888.09	1356
77	41	33	1-1	0	2.9	1	87759.55	1424