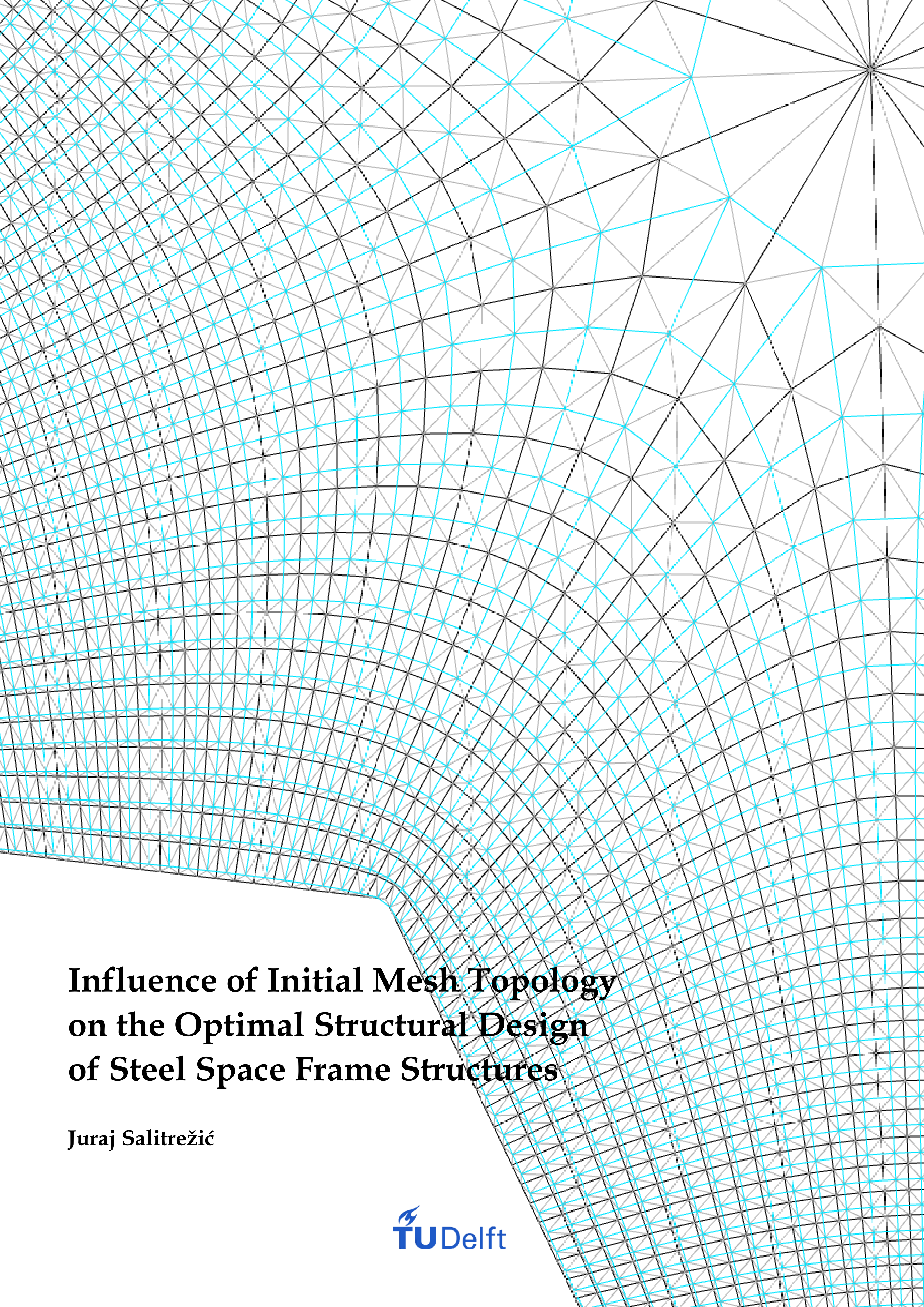


Influence of Initial Mesh Topology on the Optimal Structural Design of Steel Space Frame Structures

Juraj Salitrežić

Influence of Initial Mesh Topology on the Optimal Structural Design of Steel Space Frame Structures

by

Juraj Salitrežić

(4503236)

in partial fulfilment of the requirements for the degree of
Master of Science
in Building Engineering
Delft University of Technology

Thesis Committee:	Dr.ir. H.R. Schipper	TU Delft, Chair
	Ass. Prof. Dr. M.A. Popescu	TU Delft, Supervisor
	Dr. F. Kavoura	TU Delft, Supervisor
	Dr.ir. P.C.J. Hoogenboom	TU Delft, Supervisor

Project duration: Dec. 15, 2021 - Dec. 15, 2022
To be defended publicly: Dec. 15, 2022
An electronic version of this thesis: http://repository.tudelft.nl/

Nov. 29, 2022

Abstract

Space frame structures today are a common choice of load bearing system for achieving long spans with minimal interruptions of the floor plan beneath. On top of that, the versatility of space frame structures to conform to any shape makes them particularly interesting today, especially in the context of free-form geometry which is becoming ever more common. In structural design, and especially for space frame design, the creation of a structural model is quite a labour-intensive process, and it is highly beneficial to automate the structural model generation. Furthermore, the design of such structures can benefit from an exhaustive preliminary design space investigation. Thus, this MSc thesis deals with the parametric design, engineering and optimization of space frame structure typologies based on different initial surface discretization of the input free-form surface geometry, and different topological relations of space frame top and bottom layers. These topological relations are based on Conway operators most relevant for the structural patterns occurring in space frame structures specifically, dual, kis and ambo. These operators are always applied on an initial or seed meshing of the desired free-form surface, to create different space frame layouts for them to be compared by their structural performance, primarily in terms of mass. The scope of initial meshing options is kept to tri, quad, and skeleton-based quad meshing. These three different types of mesh options, combined with the three possible Conway operator options, constitute the main nine combinations for each case study example.

In essence, every space frame design, due to the linear and geometric nature of the structural elements (nodes and bars seen as points and lines), can be considered as a literal structural translation of the final desired free-form shape. This free-form shape is always tessellated or discretized in certain configurations. The aim was to gain more insight into how the initial tessellation affects the behaviour of space frame structures as well as how the process of optimization of such structures is influenced regarding the initial tessellation. To gain insight into this influence qualitatively and quantitatively a parametric tool was developed to conduct case studies. The tool allows for the generation and cross-section optimization of various space frame structures based on an input surface. This parametric tool was developed using Rhino, Grasshopper, and karamba3D structural analysis plugins for grasshopper.

Keywords

parametric design, structural design, free-form, space frame, optimisation, discretization, tessellation, meshing algorithm-aided design, computational design, structural patterns, Conway operators, topological skeleton, design space investigation, preliminary design

Acknowledgements

I wish to thank to everyone who made it possible and helped me bring to an end this great, for me at time daunting, but wonderful adventure, studies at TU Delft as well as writing my master's thesis.

I could not have undertaken this journey without my graduation committee which provided guidance, Roel Schipper, Mariana Popescu, Florentia Kavoura and Pierre Hoogenboom. I am highly grateful for your guiding during the preparation of this paper especially for all your constructive remarks, advice, and instructions with full encouragement and enthusiasm.

Special thanks to Dr. Robin Oval for his efforts on compass_singular and hearing my thesis ideas and providing more insight into the more theoretically complex parts of his research. Your research, shown to me by prof. Mariana Popescu, has greatly inspired the direction my thesis.

In the end, I would like to point out, that without the great help, encouragement and understanding of my girlfriend and family all this would have been "mission impossible".

J. Salitrežić
Delft, Dec. 2022

Contents

List of Figures

06 Optimization

07 Space Frame Parametric Logic

08 Space Frame Parametric Tool Procedure

09 Pentagram

PART I:
RESEARCH DEFINITION

01

Introduction

01.1 Motivation

In recent years, the development of algorithm-aided design has facilitated the overall architectural and structural design of complex structures. This advancement in computational design allows engineers and architects to design structures not deemed feasible before. Additionally, parametric structural design with tools such as karamba3D and grasshopper has opened a potential for creating evermore complex geometries and optimizing structural designs, especially in the preliminary design phase.

Figure 01.1
Gallery of the new Milan Trade Fair
[Source: archello.com]

Space frame building structures have become more interesting than before due to the development of these computational design tools, as they allow the designer to create adequate parametric models to investigate the relationship between geometry and load bearing behaviour in the early design phases. Furthermore, Space frames are excellent load bearing structures for creating large spans and open spaces, while at the same time having a certain aesthetic value. This aesthetic value although hard to measure or comment, is primarily in the structural pattern created by the configuration of the bars and nodes. The modularity and assembly of Space Frames makes it possible to cover large spans of irregular, free-form, curved geometries.

Being a complex system of bars and nodes, with 3D load bearing mechanism hand calculations come with many assumptions, providing rough estimates for simple geometries. Moreover, Space frame structural behaviour and their respective potential structural patterns or layouts/configurations can only be properly assessed by computational methods. Parametric Design tools and 3D modelling environments such as Rhino, grasshopper and Karamba3d have the potential to bring more insight into the structural behaviour and size optimisation of Space frame structures. Considering the aforementioned, rationale for research about space frames structural patterns is presented.

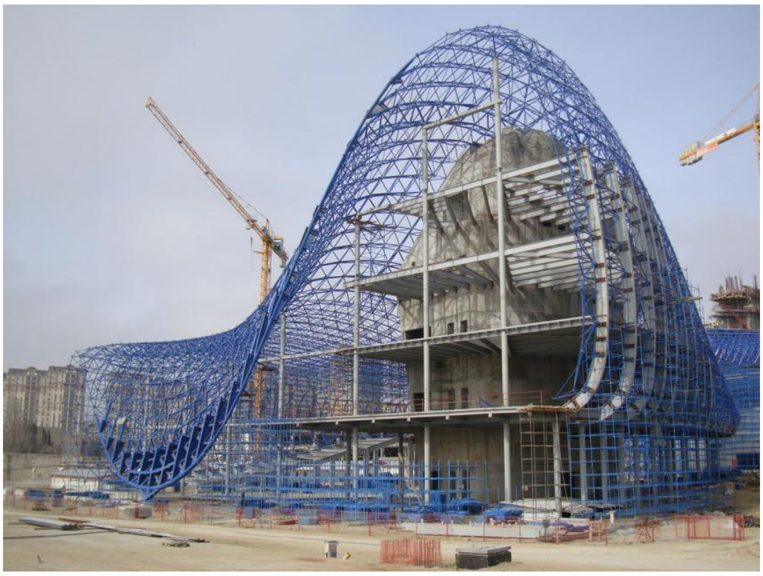


Figure 01.2
Heydar Aliyev Cultural Cent
[Source: buildpedia.com]

However, another characteristic aspect, which further motivates this thesis research, are the current tendencies in engineering practice, a sort of unification of the roles of architects and structural engineers, reminiscent of renaissance builders such as the famous “homo unviersalis”, Leonardo da Vinci. This tendency is facilitated by the ever-evolving computational design capabilities and software tools, an ever-expanding toolbox from which a new type of engineer emerges, one which can “equip” himself with as many tools as their profession or curiosity dictates. Thus, the process of design, from the initial idea and preliminary design to detail design, is becoming one integrated approach in which the collaboration between architects and structural engineers is paramount for successful design from the start of the design process. In recent years, these tendencies have resulted in numerous inspiring, novel, and complex structures, state of the art from the point of architectural, computational, and structural design.

01.2 Why parametric?

Nowadays, due to climate change and the absolute necessity of humanity to mitigate it, more and more emphasis is being put on sustainability in engineering design. Civil engineering sector, especially the design professionals, are being challenged to continuous learning and updating of their knowledge to keep up and implement the latest developments, to create more economical and sustainable designs than before. The structural design of today, not only has to assure functionality, durability and reliability of structures designed but sustainability and low environmental impact as well. Structural efficiency and economy in terms of lower material usage/costs and thus carbon footprint is becoming vital. These tendencies have created a new environment in engineering practice where the design process is becoming ever more a computational performance driven optimization-based design process, intersecting with the fields of evolutionary optimization, computational design and building performance simulation *(Figure 01.3.)* However, structural optimisation, has always been inherent in structural design in terms of preventing unnecessary wastage and material costs, although methods and tools for structural optimisation were not developed and sophisticated as today. Developments in computational tools for Parametric engineering and Structural optimisation now make it possible to optimize and design more complex structures than ever before. The main benefit of using parametric design or algorithm-aided design is in the possibility to automate the creation of complex structural system geometry which will serve as a basis for a parametric structural model. Another benefit is the potential for creating several different designs depending on input variables to gain more insight into which structural configuration is better in respect to some target parameters such as weight, displacement etc. All of the mentioned methods help to achieve a performance-based design optimization.

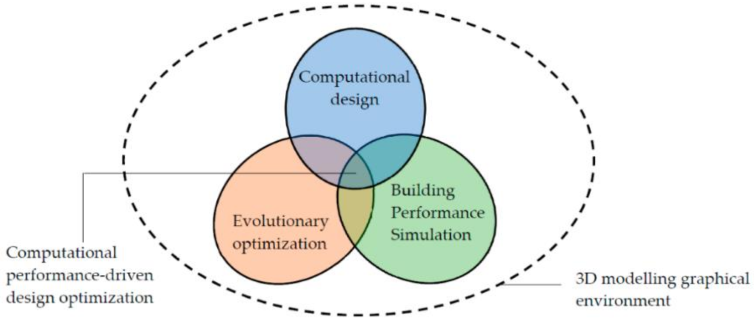

Computational design
Building Performance Simulation
Evolutionary optimization
Computational performance-driven design optimization
3D modelling graphical environment

Figure 01.3
Computational performance-driven design optimization is the combination of computational design, evolutionary optimization, and BPS in a 3D modelling graphical context. [1]

To conclude, it is important to note that the development of parametric design has opened a path towards more synergy and collaboration between architects and structural engineers, while at the same time considering the MacLeamy curve *(Figure 01.4.)* facilitating more insight available in the preliminary design phase, when the design decisions have the most impact for later design stages, potentially saving material and time costs down the line.

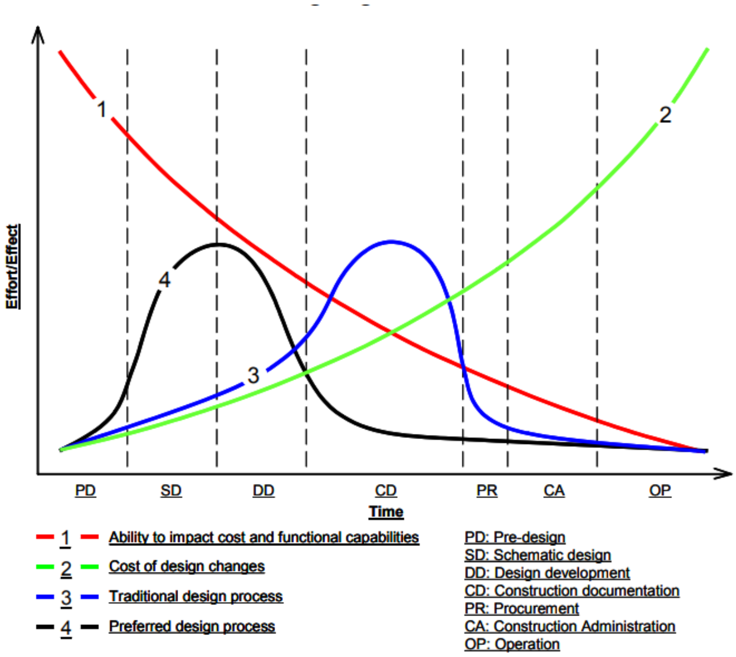


Figure 01.4
MacLeamy curve [2]

01.3 State of the art

A first literature review has been done to find out what has been researched on the topic of structural patterns/configurations/layouts of space frame structures. The search was done through google scholar by using key words relating to the topic stated. The main conclusion is that this topic as specified has not been researched to my knowledge. However, the key word search has resulted in finding useful articles and literature related to some extent with the posed problem and identified a research gap regarding structural pattern space frame research.

The research done by *Koronaki et al. [3]* and *Shepherd and Pearson [4]* relates to the layout optimization of space frame structures. The main takeaway of these articles is that space frame tessellation can be achieved by using certain Conway operators to mathematically define them which can facilitate the modularity of the structural configuration. Furthermore, another interesting approach outlined by *Oval et al. [5]* in short shows a procedure for creating structural patterns by topology finding of structural patterns for shell-like structures (grid shells, shells, voussoir tessellation for masonry vaults) based on the design of singularities in the pattern. More importantly the author shows how quad mesh patterns can be created based on a topological skeleton of the underlying surface. This gives a direction for further research into how to formulate different tessellation strategies to form various modules or structural patterns of the space frame structure, this type of discretization has not been compared to other known types for space frame design, showing a research gap which is to be addressed by this MSc Thesis. Another article by *Koronaki et al. [6]* shows a possible approach to rationalizing space frame structures by reducing variability in joints, which could also be an interesting optimization goal. More specific details, and relevant knowledge and information from the mentioned sources, are presented in the Literature Review, part II of this MSc Thesis.

To conclude, most of the literature related to the topic used a parametric design approach, showing that it is a valid research approach. In short, it can be said that the found literature mostly relates to specific sub problems which could help find answers to the originally stated topic. A research gap was identified regarding lack of comparison between various possible space frame discretizations based on different initial meshings and Conway operator relations of top and bottom space frame layers.

02

Research Definition

Based on the introductory and motivational chapters, the main idea of researching various discretizations of input freeform surfaces to create space frames and gain insight into how they influence the load bearing behaviour, structural design and optimization has been laid out. To formalize and give context to this idea, a problem statement is further presented.

02.1 Problem Statement

The problem is structured as follows: To create a space frame based on a free-form surface, this free-form surface geometry must be first discretized/tessellated/meshed in a certain configuration. There is a certain design freedom in choice of this initial tessellation or meshing, meaning there is more than one configuration. This problem raises two main questions:

1. ***Given an architectural free-form surface model to discretize into a steel space frame configuration, what is the influence of this structural pattern configuration regarding the optimal structural design and behaviour of a space frame?***
2. ***For a given architectural free-form surface model, how does one generate an appropriate parametric structural tool to test various space frame structure typologies?***

The first stated problem question forms the basis of the main research question stated in the Research questions part of this MSc Thesis. It, of course, implies several sub-questions which are explicitly stated as well. The second problem question can be perceived as one of the many necessary steps to answer the first main question. Namely, after researching initial approaches, and many iterations, it can be concluded that this problem can be approached by using parametric design tools, applying different meshing strategies to form the basis of a space frame structure. The following images, *(Figures 02.1, 02.2 and 02.3)* show an example of possible mesh topologies for discretizing an input double curved pentagonal surface to be translated into a space frame structure.

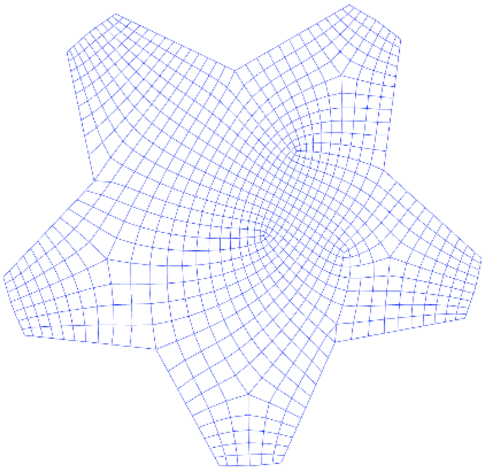


Figure 02.1
Quadrilateral tessellation example for bottom layer of a space frame based on a concave polygon surface

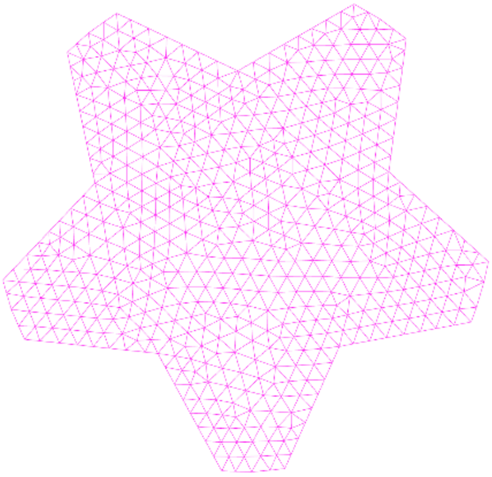


Figure 02.2
Triangular tessellation example for bottom layer of a space frame

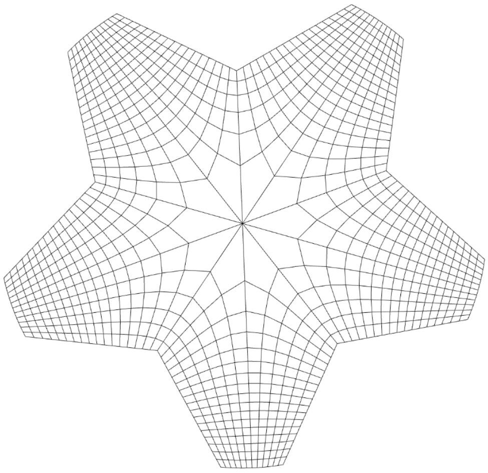


Figure 02.3
Topological skeleton based Quadrilateral tessellation example for bottom layer of a space frame

02.2 Research Objective

Space frame structures have been used extensively in the past to create large spans and column free spaces. With new developments in computational and algorithm aided design specifically space frame structures can achieve forms before deemed as unfeasible. A parametric structural design or generally computational design approach gives the opportunity for algorithmically generating space frame configurations based on an input surface as well as potentially giving significant insight into structural behaviour in the preliminary design phase.

The main research objective is to investigate applications of different structural patterns using a computational design approach for space frame design and to evaluate each solution in terms of multiple optimality criteria. The computational design approach entails the development of a grasshopper definition to serves as the main research tool, which could potentially afterwards also be used in design practice during the preliminary phase design of a space frame structure. Possible optimality criteria are mass, stiffness, aesthetics, and fabrication aspects. Least mass is chosen as the primary optimality criteria. Each of the design variants should then be evaluated regarding chosen optimality criteria. In essence the relationship between geometry, structural pattern and load bearing behaviour is investigated.

This approach will of course entail a certain design space of optimized solutions not only one "optimal" solution. The term optimal throughout the MSc thesis is always meant in the context of formulated optimality criteria, not in the sense of only one ultimate optimal solution.

02.3 Research Questions

The main research question is as follows:

Given an architectural free-form irregular surface model to discretize into a steel space frame con- figuration, what is the optimal structural pattern configuration regarding multiple optimality criteria (mass, fabrication, aesthetics, stiffness in regards of chosen pattern) and their realistic constraints (load bearing behaviour, deflection, available types of steel c/s)?

This main research question leads to several sub-questions which will lead to possible answers to the main question. The sub-question are as follows:

1. *What are relevant surface tessellations/discretization's/meshing options for generating space frame structures?*
2. *Is there any noticeable influence on structural performance of space frames in regard to chosen tessellations?*
3. *Which space frame configuration is most appropriate considering optimality criteria?*
4. *Which parameters governing the space frame structure configuration are most relevant for optimizing space frames?*
5. *What is the influence of the relevant parameters?*
6. *Which structural pattern discretization strategy is the most appropriate?*

02.4 Approach

Part II, Literature Review is performed to gather relevant knowledge for the prescribed research topic, further expanding and substantiating the initial state of the art overview. The contents of the literature review are roughly divided into four main chapters dealing with the four main thematic aspects of the research, namely, space frame structures, structural patterning, parametric design, and optimization methods. In part III the knowledge gathered during the Literature review is applied in creating the full parametric space frame tool. Furthermore, the Parametric Tool Development process entailed investigating more than one parametric approach depending on desired functionality and computational limitations. Of course, only the relevant and latest approach is shown. It can be considered as the most practical part of the MSc thesis. In part IV a variant study shall be done to benchmark the developed parametric tool, show its functionality and possible limitations. Finally in Part V the conclusion and further recommendations are laid out based on research done in previous chapters especially the ones in part III and part IV of the MSc Thesis.

PART II
LITERATURE REVIEW

03

Space Frame Structures

In context of this MSc thesis, a Space frame structure can be considered as a load-bearing structural assembly of linear, axially loaded only, rod or bar elements, which are connected at structural nodes, often considered as hinges in terms of structural behaviour. These bar and node elements connect to form a three- dimensional load bearing truss module, which is then repeated throughout the structure. They are often used for achieving long-spans with minimal intrusions to the spanned space below, consequently freeing the space below to be organised and used in a more efficient manner.

03.1 Elements and terminology of space frame structures

When talking about space frames it is important to distinguish several terms which refer to the different elements of space frames. From a material point of view the space frame elements considered are bars and nodes. However, from a more abstract point of view we can talk about space frame elements in geometrical terms of vertices, edges, and faces which are populated with cells, three-dimensional truss modules. Furthermore, considering a space frame is essentially a 3d truss system we can talk about three distinct layers of the structure, the top chord, bottom chord, and web layer. This is important in understanding the structure of a space frame as there can exist certain topological relations between the layers and thus give us a terminology to name certain configurations of space frame structures. This idea is explained in detail in Chapter 04, Structural patterning, while specific implementation of these relations is explained in Chapters 7 to 11. Nevertheless, although we talk about 3 layers in space frames relating to truss terminology, in space frame structure terms these are essentially double layer systems. Basically, any space frame structure with a structural height, or web layer, is considered a double or more layered structure. Single layered space frames are what is often called today as grid-shells.

One way of categorising space frames is according to their main shape. They can be flat, singly curved, doubly curved or a combination of single and doubly curved parts. However today free-form space frames are commonplace, thus making shape classifications redundant, the reader is referred to many online sources showing such classifications if interested.

03.2 Advantages and disadvantages of space frames

The case for considering a space frame structure in a certain design situation is best reflected through the following listed advantages and disadvantages. It should be noted that this is not an exhaustive list, but a general indication of most important pros and cons, compiled from *Chilton [7] and Lan [8]*.

Advantages:

Load sharing - space frame structures, due to their three-dimensional structural behavior, contribute with all their members in carrying loads, effetely distributing both concentrated and uniform loads throughout the structure towards the supports.

Installation of services - the self-evident open nature of space frame structures allows the possibility to integrate installations such as mechanical, electrical and ventilation services within its structural height. The support of such systems can be achieved by supporting them at the structural nodes. However, one must consider these loads into the structural model, especially if they are of considerable weight.

Robustness - Although considered lightweight structures, space frame structures have considerable rigidity and redundancy. Meaning compression or buckling failure of one of the elements will not induce total progressive collapse of the structure. This robustness comes from the 3D load bearing mechanism which allows for load sharing and utilization of all members in the structure.

Modular components - an overwhelming majority of space frame structures are prefabricated in factory facilities. The factory fabrication conditions allow for high quality production of elements, with highly precise tolerances, surface finishing and accurate dimensions. The space frame elements are also easily transported, due to both the compact nature of the product and its relatively small size in relation to other structural steel elements, such as girders. Furthermore, the modularity of components gives an opportunity in terms of a finite number of chosen standardized cross-sections and nodes which make up to final structure. The main strength of modularity is the possibility to reduce the amount of unique or one-off bar elements and nodes in the structure.

Freedom of choice in support locations - space frame structures are ideal for creating long uninterrupted spans allowing for a certain freedom in choice of support locations for the space frame. This aspect is particularly useful for the architectural layout below the space frame, giving the possibility to utilize the space more efficiently, with less interruptions in the floor plan. The space frame can be supported in basically any node in the structure. However, a designer must consider the possibility of high local forces at support locations, especially if the space frame is considered as point supported. Utilization of linear supports will allow more uniform force distributions.

Ease of erection - No matter the final size of the space frame structure, it is always assembled from smaller elements, in-situ, giving the possibility to safely assemble parts of the structure and lift them up to their final configuration.

Lightweight - the main load transfer mechanism in space frames is axial in terms of tension or compression forces in the bar elements. Considering that this structural action is much more effect than bending, the elements applied in the structure can be almost fully utilized and of relatively smaller dimensions, structural height of cross-sections are not as pronounced as in structural elements under bending. The axial load bearing mechanism allows for a smaller self-weight while being able to achieve large spans at the same time.

Form and shape versatility - considering the linear nature of space frame structures, they can adhere almost to any shape or form be it flat or free-form. This makes space frame structures or space grid structures a highly suitable structural system for materializing modern, highly free-form organic, large span, interruption free, architecture.

Disadvantages

Cost - when utilized for relatively smaller spans, 20-30m, the cost of space frames can be rather high in comparison to alternatives for achieving such a span. The main cost driver is the number of nodes, considered as the most expensive part of the structure, while also being 20-30% of the total weight. Hence to achieve a cost-efficient design the number of nodes might serve as a good metric to keep under consideration. Another cost driver to consider is the complexity of the structural nodes. The number of nodes differing from each other in terms of numbers of bars connected and the angles at which they connect, should be minimized to have more uniform elements and thus less fabrication and erection complications.

Erection time - The complexity of structural nodes and their number can lead to longer erection times. Care should be taken when designing to avoid too much heterogeneity in structural elements. The final solution is neither a fully uniform space frame structure, with all bar and node elements being the same, nor one where each bar and node is unique.

Fire protection- considering the high number of elements and their relatively large total surface area, it is difficult to achieve economical fire protection of space frame structures. Case specific situations can make fire protection a highly important aspect of space frame design. As for any steel structure, same principles of fire protection stand, with the choice of active and passive fire protection measures.

Load sharing at supports - the same load sharing capacity allowing space frames to distribute concentrated and uniform loads efficiently through the structure can cause problems at support locations. Considering the case of supporting a space frame in its bottom node, usually four diagonals of the web layer join in a node. These diagonals tend to have primarily compression forces at the supports, where buckling or compression failure of only one of the members, could cause a redistribution of forces to other remaining elements, possibly triggering a partial or even total progressive collapse of the space frame structure. This aspect becomes more important the less uniform the loading conditions are, high localized loads at support locations can be problematic.

03.3 Short overview of early development and examples of Space Frame structures

According to *Chilton [7]* the earliest example of a Space Frame structure can be traced back to an experimental design for kite construction by none other than the inventor of the telephone Alexander Graham Bell. First shown in National Geographic Magazine in 1903, Bell demonstrated that with the use of tetrahedral cells a lightweight, yet robust structure can be formed. Although it was an experimental kite construction design he commented: *"Just as we can build houses of all kinds out of bricks, so we can build structures of all sorts out of tetrahedral frames, and the structures can be formed as to possess the same qualities of strength and lightness which are characteristic of the individual cells. ".*

In 1907, Bell corroborated his comment by constructing the first steel space frame structure, the observation tower at Beinn Breagh, USA. It had tubular members and cast nodes. He successfully demonstrated that lightweight and robust steel structures are possible if using a space frame structural system. However, his structural innovation did not result in widespread commercial applications in architecture and construction.

The first successful and widely available commercial space frame system was the brainchild of German engineer Dr. Ing. Max Meringhousen in 1943, named MERO system. The system consists of tubular member connected at ball shaped nodes. The MERO System at the time was innovative due to the industrial fabrication of its components, and relative simplicity of connections. While manufacturing methods today are hard to compare with those in 1943 and have much improved, the same principle of tubular members and node ball joints is present even today. The system inspired numerous other spatial structural systems based on similar principles.

However, here I want to present two of the most famous space frame structures from my country, Croatia, namely the Poljud stadium *(Figure 03.1)* and Zagreb Airport *(Figure 03.2)* The Poljud stadium (built from 1977-1979) in Split, Croatia, designed by architect Boris Magaš, is considered in public opinion to be one the most beautiful stadiums ever built, both because of the location and view towards the Adriatic Sea and the shell-like curved flowing shape of the roof achieved by applying MERO system for construction of the space frame structure. The Poljud stadium roof was the largest spanning Mero structure built to date at the time, spanning an astonishing 206 meters by 47 meters. The main layout of the space frame can be considered as Quadrilateral mesh, with a topological dual

Conway operator relation between top and bottom space frame layers, it is the most common space frame configuration.

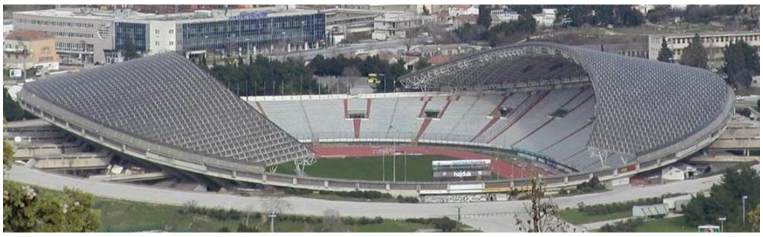


Figure 03.1
Poljud Stadium - Split RH
[Source: wikipedia.com]

The Zagreb Airport designed by Neidhardt and Kincl, finished in 2017, received the BigSEE Architecture Award 2022 in the Public and commercial architecture category. The most notable part of the structure is the roof envelope. The primary choice for a space frame structure was due to facilitating an open plan as possible to efficiently utilize the space below. All of the bars in the structure are of different length due to the shape of the roof, which shows the possibility of modern space frame construction, which allows for highly custom structures, tailored to the needs of the client and project specifically. This is primarily because of the custom industrial approach, which has advanced through the years, as to not subordinate the client to its technology but adjusts to the client or market requirements, opening the possibilities for more complex structures with less limitations than before.

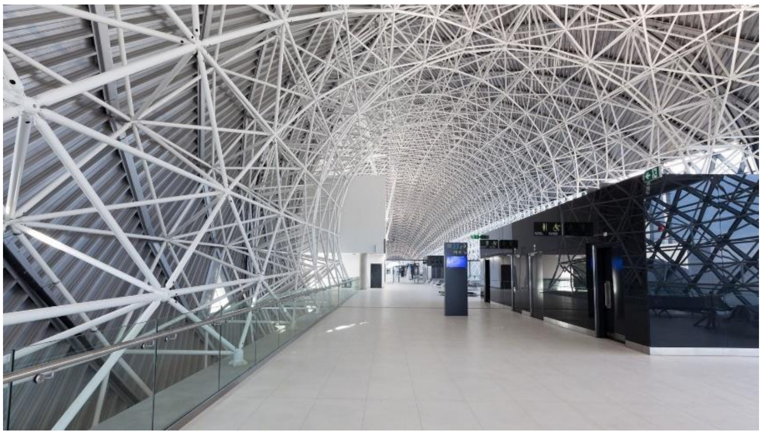


Figure 03.2
Franjo Tuđman Airport interior - Zagreb RH
[Source: archdaily.com]

As mentioned, space frame structures transfer loads primarily in an axial manner, allowing only compression and tension members through the structure, and none should be subject to bending. Overall, the structural behaviour of a space frame structure is dependent on its geometry. Depending on the shape of the structure the global load-bearing behaviour always consist of three main load bearing mechanisms in varying degrees of arch, shell and plate bending. A flat space frame will have dominant plate bending out of plane action, where the truss height is governing. A singly curved space frame will have primarily arching action, and a doubly curved will have a shell-like load transfer. A free-form space frame will have all three depending on the geometry of the space frame. Although every time, locally, the structure is a three-dimensional truss of compressive and tensile members, the global shape of the space frame surface (assuming only self-weight as load, not any asymmetric snow loads or point loads (for example football score screens on a stadium space frame roof) will influence how the forces are distributed between the members. Thus, one can see why it would be important to choose a sensible shape for a space frame, to achieve the most economical load bearing behaviour. This can be achieved through optimization but also through form finding.

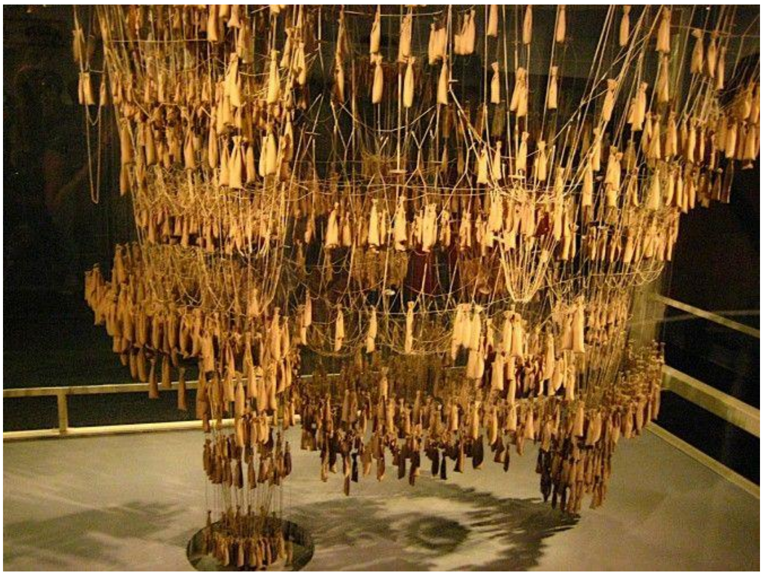


Figure 03.3
Hanging chain model of Sagrada Familia by Antoni Gaudi
[Source: pinterest.com]

03.4 Form Finding

"*Form finding - originally coined in the latter half of the 20th century as 'Formfindung' in German - is the methodology for finding the equilibrium geometry for a given set of external loads, internal forces, and boundary conditions. Typically, it is assumed that the topology (connectivity) of the structural elements is defined, and the form-finding algorithms solve for the unknown nodal (XYZ) coordinates.*" Popescu and Oval [9]. However, early form finding can be traced back to physical hanging chain models made by Antoni Gaudi for the Sagrada Familia *(Figure 03.3)* or Heinz Islers hanging membranes for his shell structures *(Figure 03.4).* Originally a physical modelling process, today it is often applied computationally utilizing various algorithmic form finding procedures such as Force Density method, Dynamic relaxation, and Stiffness matrix methods. A full exhaustive explanation of those methods is out of scope for this thesis. However, in short according to *Veeenendaal and Block [10]:* "*Force density methods refer to all methods that use the concept of the ratio of force to length (or stress to surface area) as a central unit in the calculations. Dynamic relaxation methods use the analogy with motion, where residual forces are converted to velocities and the mass of the nodes determines acceleration. Stiffness matrix methods use real material stiffness matrices in the calculations* ".

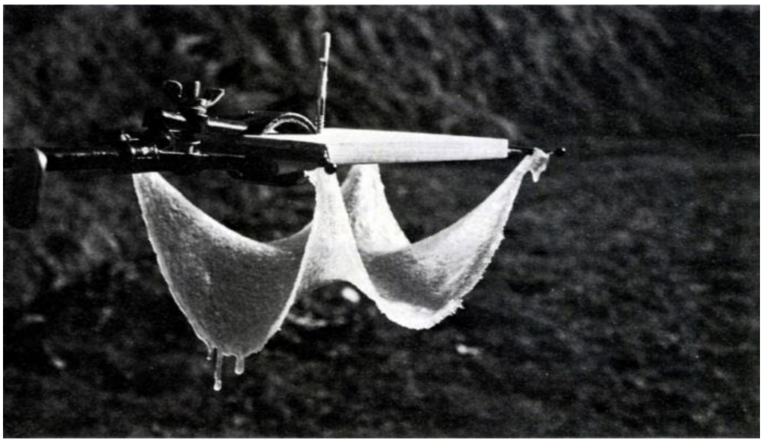


Figure 03.4
Hanging membrane model by Heinz Isler
[Source: baunetz.de]

Although form finding is primarily used for modelling of tension structures or shell structures, space frame structures also benefit through the form finding of the shape of the final surface to be discretized into a space frame structure, as will be shown in this thesis.

04 Structural Patterning

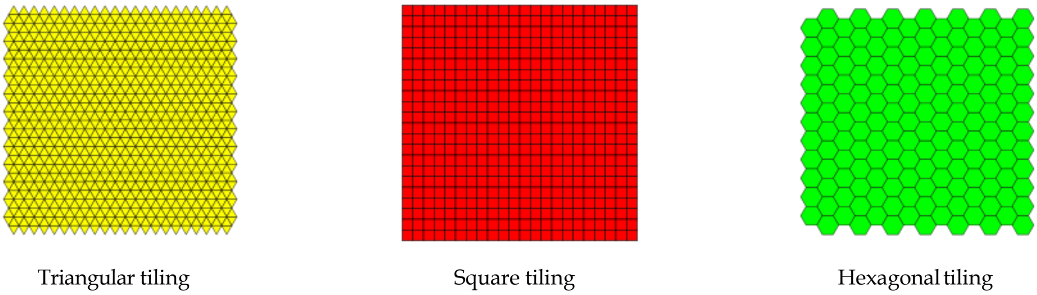
A core focus of this MSc research is on the structural pattern of space frame structures. The structural pattern in space frames is made up of the topology of its bottom grid, top grid, and inter-connectivity between them. A core question is how these structural patterns influence the load bearing behaviour and the structural design and optimization of such structures. A Literature review is done to identify current research available on the various topics connected to the structural pattern of space frames. Namely, this concerns topics such as application of Conway Operators to space frames, and Topological Skeleton based quad meshes. Furthermore, in the following sections appropriate terminology will be introduced to help understand the various configurations of space frames investigated further in the thesis along with the literature review.

The design of space frame structures can be viewed as a process of creating structural patterns by utilizing various geometrical and topological relations to explain the shape of the assembled structural elements (bars and nodes). Furthermore, to be able to adequately describe a particular space frame design, a certain vocabulary is needed. Part of this vocabulary was introduced in chapter 3 regarding Space Frame structures in general, another more specific and relevant for the parametric research done is introduced in the following sections. This vocabulary is not only related to the fundamental problem of discretizing, tessellating, or meshing a surface to create a space frame structure but also to the problem of how certain space frame configuration can be called according to how they relate in terms of topology. Essentially much of the following terminology can be interchangeable as they talk about the same thing but from differing points of view.

04.1 Tessellations

A tessellation or tiling is the covering of a surface, often a plane, using one or more geometric shapes, called tiles, with no overlaps and no gaps. As space grid/space frame/space truss structures can be considered as tessellations of a surface by bars and nodes, it is useful to borrow some concepts and terminology from this area of mathematics to formalize the structural patterns expressed by space frame structures.

Essentially all elementary space frame configurations are based on so called Euclidean tilings by convex regular polygons. There are three basic Euclidean regular tilings, Triangular tiling, Square tiling and Hexagonal tiling *(Figure 4.1).* In this MSc thesis research, triangular and square tilings are chosen as initial possible tessellation shapes. This choice is based on the prevalence of these two types of tessellations in space frame structures. Furthermore, it is important to note, that tessellations in this MSc research always refer to the layout and shape of the bottom layer a space frame structure. The initial free-form surface is always tessellated in either a square or triangular tiling, while the top layer is created based on topological relations with the bottom layer, which serves as a sort of seed tessellation. Hexagonal tiling is not investigated specifically, although this tiling will appear because of a dual Conway operator on a triangular seed tessellation. This is explained further in detail in the section regarding Conway operators.



Triangular tiling

Square tiling

Hexagonal tiling

Figure 04.1
Regular Euclidean tilings by convex polygons [Source: wikipedia.com]

04.2 Meshes

In this research the notion of a space frame structure, more specifically its layout, is heavily dependent on the initial "tessellation", or in computer graphics terms, "meshing" of the initial input NURBS surface, which is supposed to be translated into a space frame structure. Patterns for structural applications can effectively be modelled computationally by meshes.

Meshes are computational representations of 2D or 3D objects. They consist of vertices(points), edges(lines) and faces (areas traced by edges, which are defined by vertex connectivity). This elementary mesh structure consisting of connected points can form various polygons. The two basic two-dimensional polygon meshes are the triangular (tri) and quadrilateral (quad) meshes.

Mesh data structure

To create meshes computationally there needs to be a certain data structure to program the meshing procedure. Mesh data structures contain information about mesh geometry and topology. For example, in grasshopper, the data structure of meshes is simply defined by face vertex connectivity, meaning that with a list of vertices and an ordered groups of those vertices to constitute faces, a mesh is constructed. There exist other mesh data structures such as half edge and winged edge structures. The compass_singular python library, implemented in this MSc Thesis research with its accompanying compass_singular grasshopper plugin components, uses half edge data structures. A full explanation of half edge data structures is out of scope of this thesis. However, it should be noted that at the same time they: encode more information and are much more elaborate than face-vertex data structures, allowing efficient and complex meshing procedures along with editing and exploration of mesh patterns.

Structured and Unstructured grids

Meshes with their tessellations form grids. These grids can be either structured, or unstructured. Structured grids have regular connectivity meaning a uniform polygon structure some examples are shown in figures below. These structured grids co-relate to the basic Euclidean regular tilings by convex polygons (Triangular and Square tiling). *(Figure 4.2)* below illustrates the qualitative difference between structured and unstructured meshes.

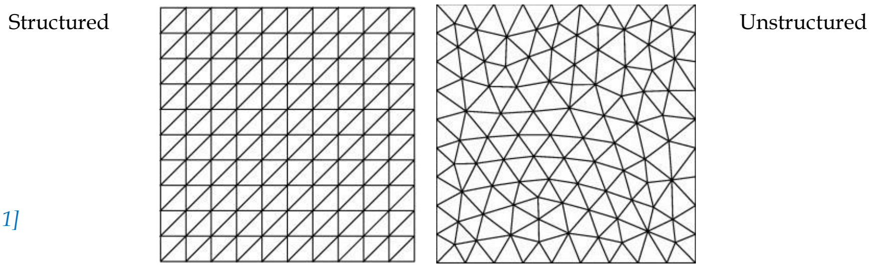


Figure 04.2
Structured vs. Unstructured grids [11]

While this might be self-evident, it is important to mention to distinguish between the terms of tessellation and meshing. Further in this text grids are equivalent terms to meshes and are used interchangeably according to the focus to be conveyed.

The MSc thesis problem of the influence of initial mesh topology on the load bearing behaviour on space frame structures, is essentially looking at the problem of how the load bearing behaviour is influenced if the space frame structure is based on structured or unstructured meshes. Moreover, there is a question of how the load-bearing behaviour differs in between the various structured grids possible. The main notion behind this is to see if there is an obvious difference between the load bearing behaviours and is there an obvious preference for structured grids (and which ones) as opposed to unstructured.

Considering the load sharing feature of space frame structures, at first one could say that there would not be an obvious difference and preference between the two. However, to properly assess the question from a qualitative and quantitative aspect, and to see if load-sharing is so substantial that there is no difference between structured and unstructured grids, research is done in terms of appropriate case studies through the developed parametric tool, which details are contained within part IV of this MSc thesis.

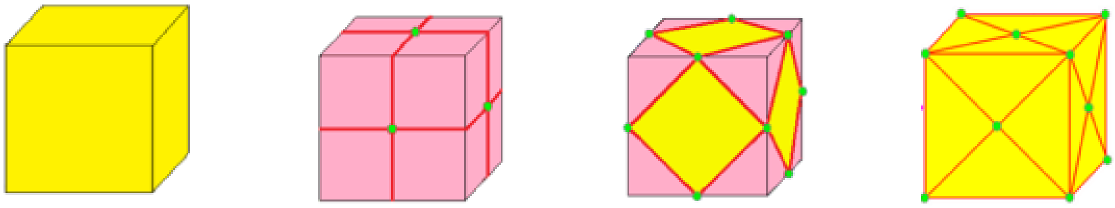
Types of meshes

In this MSc thesis research two basic two-dimensional polygon meshes are of interest. Namely the triangular (Tri), quadrilateral (Quad) meshes, and a specific subset of Quad meshes based on topological skeletons of shapes, named Skeleton-based quad meshes (more detail in section 04.4). In the figures below, the layout of these meshes is shown applied on a doubly curved 3D pentagon shaped surface. These two types of meshes are well implemented within the Rhino 7 and Grasshopper environments. QuadRemesh and TriRemesh grasshopper components are stock components, meaning no plugin is required. Furthermore, they have proven to be quite stable and fast, able to appropriately mesh almost any surface geometry, if it is based on convex polygons (floor plan projection of surface). Without these new components, the initial meshing of surface geometry would have to be done in a more manual manner or would require programming the meshing logic, which is not in the scope of this research. It is important to note that when developing the grasshopper research tool for case studies further in the thesis, the dependency on various plugins was kept to a practical minimum. However, the Skeleton-based quad meshes are generated by compass_singular grasshopper plugin, developed by Oval. This is explained in detail in chapter contained within part III of this thesis, which concerns the details of developing the parametric tool.

04.3 Conway Operators

The above depicted, triangular, quadrilateral meshes and subsequent operations of editing those meshes into new patterns can be mathematically described and formalized. More specifically, they can be aptly described by so called Conway polyhedron notation. This mathematical notation, invented by *Conway et al. [12]*, popularized by *Hart [13],* is based on operators which are simply applied on “seed” polyhedra in order to create other polyhedra. Conway polyhedron notation is thus based on a predefined set of operators, named Conway operators, which modify an initial seed geometry and thus output a new one. This modification procedure through operators is applied sequentially from left to right like mathematical functions.

In *(Figure 4.3)* the results of applying the first three basic Conway operators are shown, dual, ambo and kis on a seed polyhedra, a cube in this case. Although Conway operators can be applied on any polyhedra, the easiest way to understand how they function is to look at the example of a cube. The dual operator replaces each face with a vertex and each vertex with a face. The ambo operator converts edge midpoints into vertices. Kis operator converts each face into a pyramid, however its height can be positive, negative or zero, in *(Figure 4.3)* the height is zero. The same operators applied on polyhedra can of course be applied on polygons as well. Looking at the faces of the cubes one can imagine how the operators would influence Quad mesh geometry establishing a topological relation between two meshes the seed and the operated one.



Seed Ambo Dual Kis

Figure 04.3
Relevant Conway Operators [Source: wikipedia.com]

In *(Figure 4.4)* the results of applying the three basic Conway operators dual, kis and ambo, denoted by pink arrows, shows the transformations achieved by operators on subsequent seed geometries. With the three basic operators 12 other compound operators can be achieved. Each of them is denoted as a string representing the sequentially applied operators on the seed geometry. For example, applying a dual operator on a beforehand kis operated geometry creates the so-called Zip operator which can mathematically be denoted as *dkC*, meaning, dual operator applied first, kiss secondly on a Cube seed geometry. This the earlier mentioned function like property of the operators. Furthermore, the dual of an already dual operated geometry will revert it back to its seed topology, meaning *ddC* = *C*, and if we take into consideration the definition of the dual operator, it can easily be seen why it is so.

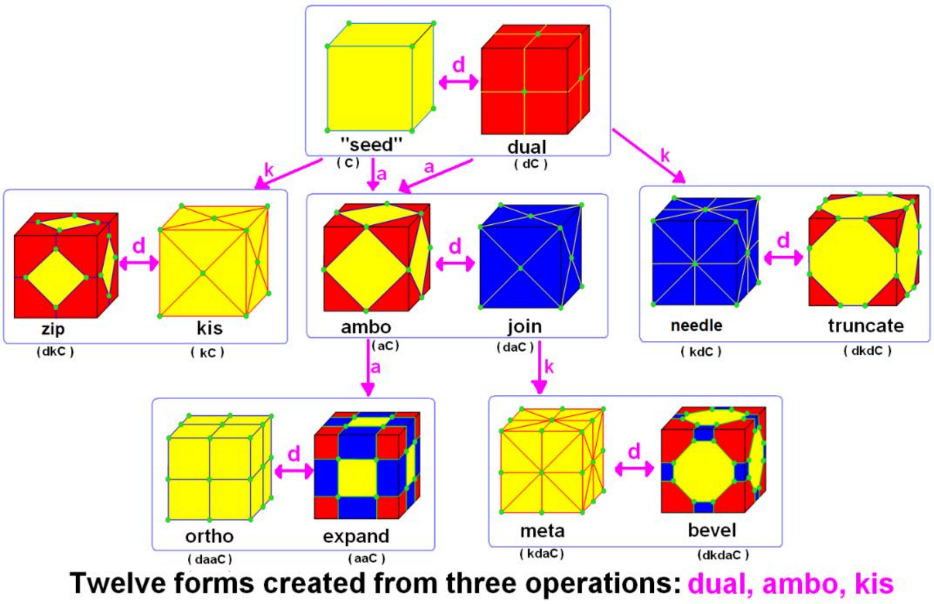


Figure 04.4
Relevant Conway Operators [Source: wikipedia.com]

Application to Space frames

The application of Conway Operators to Space Frame design was first implemented by *Shepherd and Pearson [4]*. In their paper they have shown that space frame geometries, top and bottom layer specifically, can be effectively described and generated by Conway operators, specifically dual, ambo and kis. In their research, these top and bottom layers, generated by applying Conway operators on a hexagonal base mesh, are then linked, or connected, on a limited proximity basis between vertices of the top and bottom layer. The following figures show in blue lines always the original hex mesh and orange parts show the pattern created by respective Conway operators and the layout of the generated space frame.

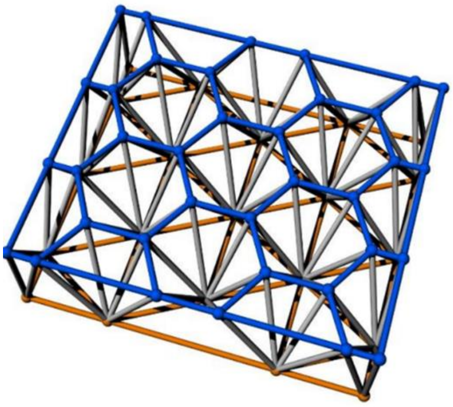


Figure 04.5
Generated space frame [4]

Furthermore, they have addressed topology optimization by applying a new "member adding" scheme on the basis of a so called "feasible ground-structure" method, showing that the pervious classic ground-structure method where all vertices are connected to each other, a highly redundant and computationally heavy structure, can be replaced by one where the ground-structure is sparser, although structurally sound, noting *"Rather than simply removing unused members for a large list of potential members, this approach can start from a sparsely connected structure and can add in missing members which are required for optimality"*. Moreover, they critically assess their approach stating: "However the highly mathematical implementation of linear programming means that it is not easy to incorporate directly into modelling software". The authors also mention how in the end the inner workings of this implementation are hidden, meaning that to a practicing roof designer a detailed grasp of the solver is unlikely, also mentioning "*and if an optimal problem to the solution is not found it is often difficult to know exactly what needs to be done to fix it*.".

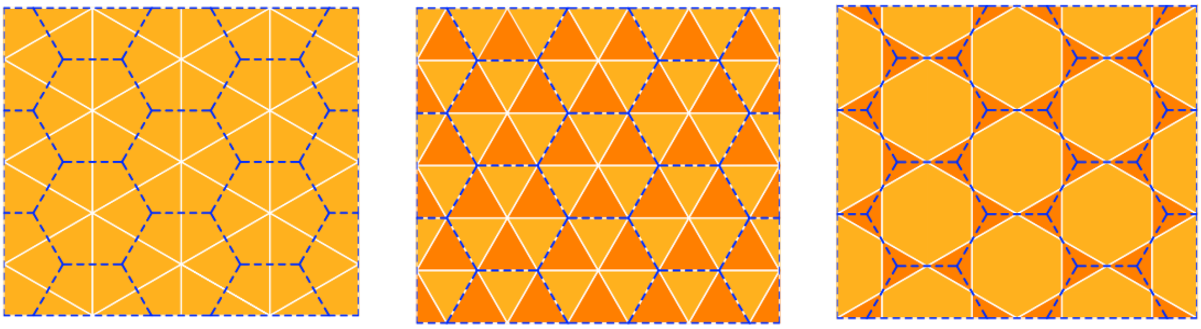


Figure 04.6
Blue dashed lines showing base hexagonal mesh, orange showing mesh created by applying Conway operator [4]

The presented approach in their paper has resulted in optimized layouts of their case study example which was the famous Chris Williams design of the British Museum Great Court Roof. They report a 16% saving in material within 13 optimization iterations. The depicted ground-structure and optimized layout show the complexity of both the initial grid and the final optimized design.

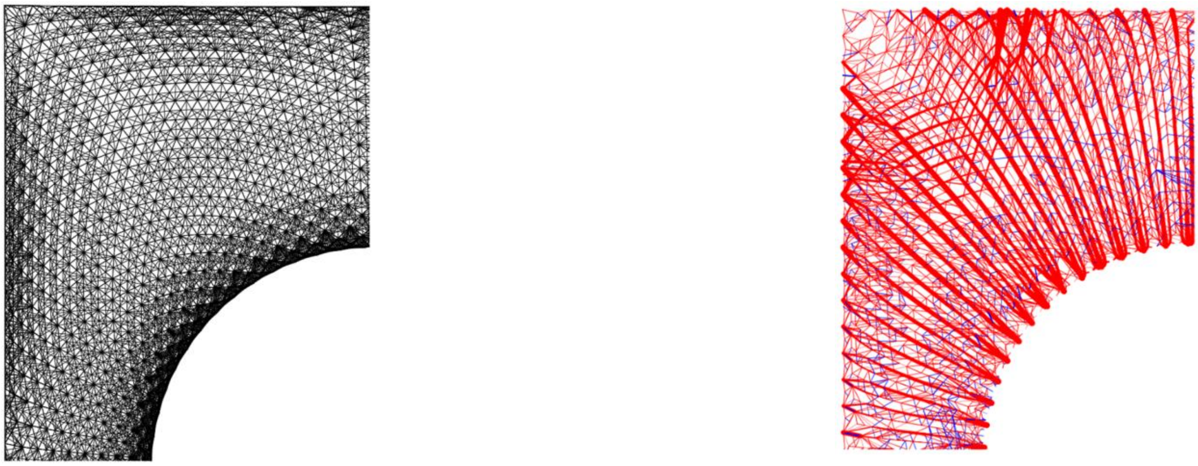


Figure 04.7
Ground structure (left) and Optimized layout (right) [4]

While their novel approach of both applying Conway operators and using a new-member adding scheme based on feasible ground-structures has its merits, there is a question of how practical the achieved final optimized layout in terms of constructability is. Furthermore, even though their proposed optimized structure might bring material savings and be an efficient load bearing system, the original solution by Chris Williams is already considered as highly aesthetically pleasing and efficient structure, posing the question of would this alternative optimized design, if present at the time, be accepted in relation to fabricational, architectural and structural requirements and considerations.

Moreover, the optimized solution having 16% material savings might be diminished in terms of impacting the cost of construction, considering the highly complex connectivity. General design awareness postulates regularity and predictability in patterns as an imperative. Thus, in this MSc thesis, ground structure methods are not included in the scope of research. Nevertheless, the application of Conway Operators on Space Frames is noted and serves to further inspire and formalize my MSc research.

Continuing the application of Conway Operators to space frame design, *Koronaki et al. [3]* present in their paper a computational workflow for generating space frames using Conway Operators (dual, kis and ambo) on quadrilateral grids, and further optimize the structure, obtaining insight into structural behaviour of each operator applied. Depicted in the images below we can see how the three Conway operators create the three different grid topologies, as well as how they are connected. The mentioned paper also gives a direction of how to research in terms of creating a case study (Oguni dome) and evaluating it in terms of structural performance. The authors present results in terms of graphs showing the total length of members within each of the three layers (top, bottom, and web layers) of a space frame created by the three mentioned operators. In short, they found that the dual topology or layout of the original space frame in question (Oguni dome) is considered as the optimal topology, meaning the least structural mass in relation to other applied operators. Further they found, that the kis topology performed better than ambo in terms of structural mass. Thus, we can see a clear ranking from dual, kis to ambo, by leased to most structural mass. Another important insight gained by the authors was regarding the utilization of members in each of the space frame layers. They conclude that “*here seems to be a direct relationship between the distribution of the tension and compression areas throughout the structure and its performance*” further stating that „*areas of pure tension or compression are minimized* “

The approach by *Koronaki et al. [3]* has its limitations, mainly in the application of Conway operators only on Quadrilateral regular meshes. Thus, certain gaps are identified, for example, Triangular based meshes are not included in the overall research. Furthermore, their approach limits the research on only convex based surfaces, leaving concave surfaces out of the scope. This poses the question of grid generation on concave surfaces as opposed to convex surfaces. Convex shapes can adequately be meshed by Quadrilateral and Triangular faces creating often structured grids, especially in the standard case of a rectangular shaped surface. However, on concave shapes such as a five-pointed star, the Quadrilateral and Triangular meshes often produce unstructured grids, posing the question of how to generate structured grids on concave shapes. A potential answer is identified in the following section, Topological skeleton-based Quad meshing.

It is important to note that the notion of grid generation for basing space frame structure can be understood as purely a process of creating structural patterns. The central question of this MSc thesis is how this choice of structural pattern (initial mesh topology), influences the load bearing behaviour and structural design and optimization of the space frame structure. The earlier question posed of how one finds and creates different structural patterns on convex shapes, which would be structured grids, is presented through the findings of Dr. Robin Oval's PhD thesis, "Topology Finding of Patterns for Structural Design" [10].

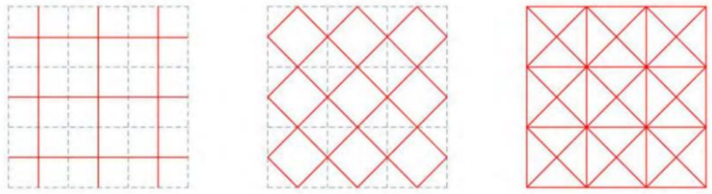


Figure 04.8
Grids generated by applying Conway operators on a Quad mesh [3]

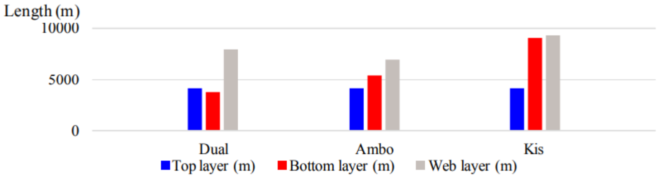


Figure 04.9
Lengths of elements per space frame layer [3]

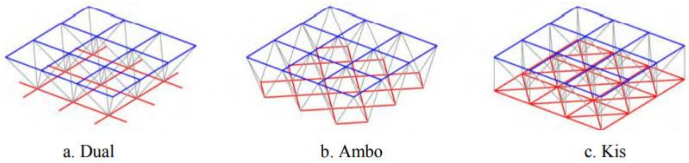


Figure 04.10
Created space frame configurations by applying Conway operator [3]

04.4 Topology Finding of Patterns for Structural Design

In his PhD thesis, Oval [14] shows a novel and complex approach for Topology finding of Structural Patterns. He motivates his research stating that "*topology of these patterns' constraints their qualitative and quantitative modelling freedom for geometrical exploration. Unless topological exploration is enabled*". These patterns are researched on shell-like surfaces or structures, based on a Quad mesh approach. In his PhD thesis, his topology finding of patterns approach is limited to grid shells, nets and masonry vaults.

However, the same approach can be applied for exploring different structural patterns (meshings) of a surface to create for example the bottom layer of a space frame structure, thus giving a means of researching structural patterns of space frames.

This mentioned approach is based on "*geometry-coded exploration*" which relies on a so called "skeleton-based" quad decomposition of a surface, potentially including point and curve features to appear in the mesh additionally. These point and curve features can stem from many reasons, from the statics of the structural system (points, for point supports) to the curvature of the shell (main curvature, curve feature). This skeleton-based decomposition relies on the concept of a medial axis or topological skeleton of a shape. For example, in *(Figure 4.11)*, one can see how the initial surface boundaries relate to the topological skeleton. One can identify as well, so called singularity points which occur as the result of the mesh topology the mesh topology.

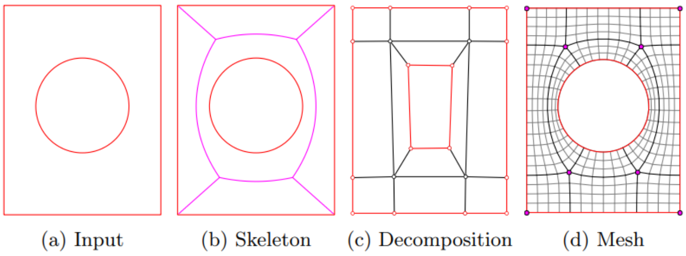


Figure 04.11
Skeleton-based surface decomposition to yield a pattern aligned with the boundary and point features on the boundary [14]

These singularity points can be understood as simply vertices which have an odd number of surrounding mesh faces, so called "valency" of the singularity points. Singularities always appear on the crossings of the boundaries of the surfaces, as well as in the mesh generated in between the surface boundaries. The author shows different generated mesh, including point and curve features, *(Figure 4.12)*. While including point and curve features creates interesting patterns for single-layered structural systems such as gridshells, for space frames they create a too dense mesh for this feature to be of interest, as the fabrication of this space frame if the corner points were included to guide the mesh generation would be highly impractical.

However, Oval's concept of skeleton-based quad decomposition of surfaces, gives an excellent start for creating structured grids on concave shaped surfaces. A topological skeleton of a shape or surface can be considered as a dimensional reduction of the surface into a set of curves, which keeps all the relevant geometric and topological relation of the underlying shape.

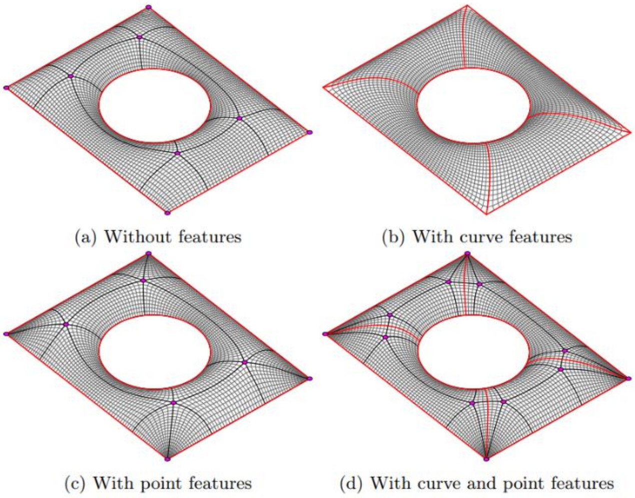
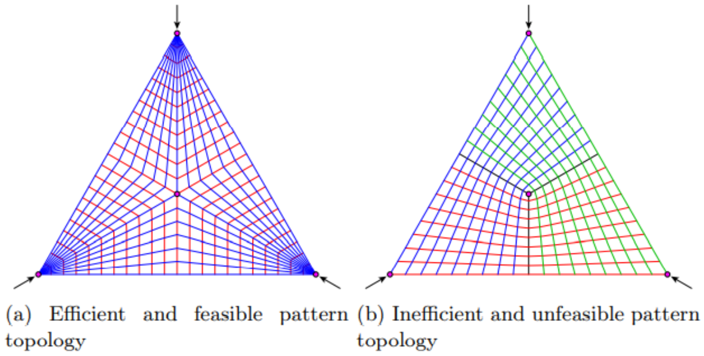


Figure 04.12
Quad-mesh patterns for British Museum courtyard roof including point and curve features to influence the topology [14]

This concept of surface decomposition is highly applicable for space frame design, even though the author does not investigate this concept on space frames. Nonetheless, the concept of surface decomposition based on the topological skeleton of the surface for space frame design is of interest, especially in situations where the initial surface is based on concave shapes. The interest lies primarily in the fact that the topological skeleton of shape captures its most important features, thus giving a valid discretization logic to create structured grids on concave shapes. Furthermore, the author points out a famous example of the CNIT in Puteaux (France) a corrugated reinforced concrete shell structure with a span of 218 m. With this example the author shows how important is the initial discretization of a surface, as this will dictate whether the load bearing mechanism and patterns is structurally efficient or feasible. Specifically in the case of the CNIT structure (figure), the blue lines which go towards the corner point supports give the structure efficiency, while the feasibility stems from the non-overlapping red and blue elements which introduce the stiffening corrugations over the structures surface. The efficient and feasible pattern can be found using the skeleton-decomposition based quad surface meshing algorithm developed by Oval.


(a) Efficient and feasible pattern topology

(b) Inefficient and unfeasible pattern topology

Figure 04.13
Feasible and unfeasible patterns for CNIT structure [14]

This poses a question of do space frame structures benefit in terms of structural behaviour, material efficiency if based on skeleton decomposition logic? The answer to the postulated question is highly dependent on the initial shape of the surface to be discretized. In any case, whatever the shape is, be it concave or convex, the skeleton-based surface decomposition provides often structurally sound, geometrically, and topologically relevant structured grids to base space frames on. Oval's PhD concludes that efficient patterns can be found for single-layered structures, following the skeleton-based decomposition logic. The goal of this MSc research is to gain insight if in relation to other most common discretizations of surfaces for space frames (quad, tri, skeleton based, structure vs unstructured grids), the skeleton based one improves structural performance. Other structured grids would be possible to investigate, primarily generating quad meshes based on the principal stress directions, however their relevance is mostly for concrete structures, where corrugations or reinforcement can follow those directions. For space frame structures, which are 3-dimensional truss structures, this principal stress direction logic is not applicable, more so since in engineering practice the amount of load cases, symmetrical and unsymmetrical is large, thus giving unique principal stress directions which differ for each load case. This would then postulate the question of how to consider all the different principal stress states to create a sensible space frame structure discretization which satisfies each case as much as possible while still being feasible. Thus, this approach although interesting, is out of scope for this MSc thesis research, and would be an interesting research subject in its own right.

05

Parametric Design

A central part of this thesis is the development of a parametric tool in order to research the relation between space frame load-bearing behaviour regarding three possible initial meshings for the bottom layer of the space frame and three possible Conway operator relations with the bottom layer configurations to create the top layer. To understand the developed tool, it is important first to understand what parametric design is.

There exist numerous definitions of Parametric design by various authors. All of the definitions vary in scope and terminology, however according to *Caetano et al. [15]* analysis of literature and subsequent definition synthesis, parametric design can be defined simply as" a design process based on algorithmic thinking that uses parameters and rules to constrain them". In contrast to traditional design process the Parametric Design process entails a new mindset for the practicing designer. Parametric thinking, described by *Swartout [16]* as "a thinking process that relates and outputs calculated actions to generate solutions to problems rather than simply seeking them", can be considered as a necessary minimum perquisite skill in order to even start with Parametric Design.

Without parametric thinking skills, parametric design cannot be utilized to its full potential. Before even starting to create a parametric model, one should have a clear vision of what and how ones to achieve, otherwise it's purely a blind trial and error, highly ineffective process. In parametric design the focus is thus shifted from the end product as opposed to traditional design, towards the procedure of exploring the possible design space of the end product through a rigours defined set of parameters and their relations.

In the last ten years, Parametric Design has resulted in the advent of a new architectural style aptly named Parametricism, whose famous proponents among many others are Patrick Schumaher and Zaha Hadid. While not without its critics it has resulted in a surge of interest in parametric design and modelling, from companies to university courses, with parametric design tools becoming an expected skillset in future engineers and architects.

05.1 Parametric thinking

Parametric thinking, also sometimes called Parametric design thinking, has be defined by *Woodbury [17]* as to having three distinct characteristics: thinking with abstraction, thinking mathematically, and thinking algorithmically. The author would like to point out that the order of the parametric design characteristics by *Woodbury [17]* is intentional, it is a sequence of intellectual activities preceding parametric modelling itself. The highly skilled parametric designer might find these steps trivial and obvious, as with experience one will develop these modes of thinking as a habit. However, for the newly initiated into the world of parametric design and parametric modelling this might present a total paradigm shift, with a daunting learning curve.

Nonetheless, in contrast to traditional CAD process, parametric design offers an opportunity for both architecture and structures of greater complexity but also for a more mindful, strategical design process. However sometimes parametric design is equated with the Parametricist style, which is often critiqued for the lack of societal relevance of the achieved complex geometries. However, the author would like to point out that the use of parametric tools does not necessitate creation of Parametricisit architecture. According to Karle and Kelly [18], "*Parametric design can be defined as a series of questions to establish the variables of a design and a computational definition that can be utilized to facilitate a variety of outcomes. Parametric design sets up measurable factors of rule-sets to determine behaviour* ", this definition encompasses the generality of the parametric design approach, negating the notions of equating parametric design with parametricist style.

Parametric modelling can be effectively used for impressive curvilinear, "science-fiction" like architecture but also for creating parametric models of typical dairy cow farms or for example on-shore wind turbines. It should be primarily understood simply as a design tool allowing for meaningful exploration of the many abstracted relations between parameters of a project being designed. For this to be successful one must utilize his capabilities of abstract, mathematical, and algorithmic thinking. Starting a parametric model without first having a developed flow chart, either mentally or manually, accompanied with an understanding of how data is processed by the parametric software is not going to result in a success.

Thinking abstractly

Before setting out to create parametric models for a particular design, one must first abstract the design project into a certain set of parameters accompanied with the relations and limitations between them. Woodbury [17] aptly states:" To abstract a parametric model is to make it applicable in new situations, to make it depend only on essential inputs and to remove reference to and use overly specific terms" further stating "*If part of one model can be used in another, it displays some sort of abstraction by the very fact of reuse. Well crafted abstractions are a key part of efficient modelling*". This aspect of abstract thinking is required due to the programming nature of parametric tools. Similarly, a computer programmer who writes lines of code and abstracts classes and functions to have as much utility as possible without repeating unnecessary code parts in order to achieve as much generality as possible, the parametric designer often uses visual programming components to daisy chain a number of components (possibly augmented with the designers own custom scripted nodes/components), creating an algorithm, to abstract a design and create relations between its parameters (while understanding the underlying data structures being manipulated) in order to create the parametric model.

To conclude, a short example of abstract thinking applied during initial steps of this MSc thesis research is as follows:

A space frame consists of a top and bottom layer with a certain connectivity rule between them. The space frame geometry should be generated on an input surface, and any input surface should be processable, convex, non-convex, trimmed or untrimmed. There were two initial approaches for creating space frame geometry based on an input surface. Either dividing the initial NURBS surface into a subset of smaller surfaces, with a prescribed number of U and V divisions of the surface, or by using meshing procedures.

Both approaches were tested and the more general one was applying a meshing procedure due to it always being bound by the surface boundaries be the shape convex or non-convex.

The subsurface approach is lacking due to it not respecting the shown trimmed geometry but the underlying untrimmed one, thus requiring deletion of certain subsurface (which also impacts the speed of the developed script due to additional process), which in the end do not result in a surface with smooth edges as in the meshing procedures, but a surface which appears more approximated by rectangular subsurfaces, giving jagged edges, rather than being populated by rectangular cells as in meshes.

Thinking Mathematically

To further develop the beforehand abstracted idea of the parametric model, one also must have some understanding of certain mathematical and geometrical principles. Furthermore, the more advanced model and processes a designer wants to implement the more advanced the accompanying mathematical concepts are. Even so, one does not need to be a mathematician to be a parametric designer, however, the more extensive the mathematical background and understanding, the more successful the designer can be. The ability to use mathematics is much more important for a designer than to do mathematics.

Concepts such as vectors, tangents, vertices, edges, meshes, connectivity, surface curvature, normal vectors, plane projections, plane equations should be familiar for most practicing designers with an engineering background. The understanding of how to use them helps to formalize parameter relations and achieve the desired result. For example, the famous British Museum courtyard roof shape designed by Chris Williams, is a perfect example of applying complex mathematical plane functions to generate the roof surface geometry. Furthermore, the structural grid is also laid out according to certain mathematical rules. Thus a "free-form" design which can start from a hand drawn blob on a piece of paper is formalized in to an exact three dimensional mathematically designed surface.

One can say that sculpturally and flow of a shape is transformed into something more than pure creative whims and sketching (no underestimation intended) once it is mathematically formalized. If the initial shape was considered aesthetically pleasing than the mathematical background of it elevates the whole aspect of design on a higher level which should be appreciated even more.

On the other hand, sometimes, special mathematical concepts are needed for creating parametric models, which are rarely part of standard engineering mathematics curricula.

For example, in this thesis a novel mathematical concept for the author was the notion of Conway operators. The literature while exhaustive and detailed is highly abstract and theoretical and would require a dedication not in the scope of reality for the time available to achieve the same or even approximate the understanding as the author of the literature. However, it is possible to grasp the concept well enough to be able to utilize while designing the parametric model. Thus, the ability to use mathematics is much more important than the ability to do the proofs and understand all the minute details of such a niche mathematical field. Another fine example would be the understanding of topology, to understand what a Skeleton-based quad decomposition is. Topology is another highly abstract field of mathematics, which is rarely taught in engineering curriculum (except for structural topology optimization courses) which to understand one can spend a lifetime researching it. Its applications range from knot theory, dynamical systems, to string theory in physics.

However, basic understanding of the underlying principles to convey what a topological skeleton of a shape is, how this can be utilized for space frame configurations is possible.

In conclusion, while sometimes mathematical knowledge of a particular designer might be lacking it should not serve as discouragement, rather it should serve as a challenge for the designer to improve his knowledge every step of the process, once one is able to use mathematical concepts to achieve the desired parametric model behaviour and results then the concepts become much less daunting and become part of the designer's skill set.

Thinking algorithmically

After the initial steps of abstracting and mathematically grasping the design problem the next step is to apply the idea algorithmically within a parametric modelling environment. Parametric modelling using visual programming tools is essentially creation of an algorithm to process the inputs and return the desired outputs for a design. Like programming, but more open to designers without practical programming knowledge. The ability to understand algorithms and what they are is still essential. Below I include the definition of an algorithm *Berlinski 1999, as cited by Woodbury [17]* which I find rather poetic and highly illustrative:

An algorithm is

> a finite procedure,
> written in fixed symbolic vocabulary,
> governed by precise instructions,
> moving in discrete steps, 1,2,3,…,
> whose execution requires no insight, cleverness,
> intuition, intelligence, or perspicuity,
> and that, sooner or later, comes to an end.

As *Woodbury [17]* comments, although the definition is less formal than normally found in literature, it still encompasses the full meaning of what an algorithm is.

The algorithmic thinking aspect of parametric design is best reflected through the development of flowcharts which represent what are the main inputs, processes, sub-processes, and outputs of the parametric model to be created. One might say that the traditional design processes can also be represented algorithmically, however the word algorithmically in the context of parametric design relates to the procedural aspect and precision of algorithms.

The result will only be as good as the underlying developed algorithm, which must be carefuly developed. This means that the proposed relations of parameters for the algorithm to encompass must be developed with careful intent. Algorithms do not care for context or what you thought you programmed; they care for the execution of code line by line, what is written. If there is a mistake or misconception, the algorithm will not result in what the designer expected.

05.2 Parametric software

There are various software programs with capabilities for parametric modelling, for example CATIA, Generative Components, Autodesk Revit and Dynamo to name a few. However, in this thesis the primary choice is the 3d-modelling software Rhino developed by McNeel, with its visual programming plugin Grasshopper 3D developed by David Rutten, a former alumnus of TU Delft.

The choice of software was made upon the fact that in many of articles and literature researched during this thesis, grasshopper was the common choice of design tool for parametric modelling. Furthermore, the community, amount of learning resources available online, and amount of various useful plugins available have greatly facilitated this thesis research. Thus, in the following paragraphs I present an overview of basic principles of parametric modelling with grasshopper and mention other important plugins for grasshopper which facilitated this thesis research.

Grasshopper 3D

Grasshopper is a visual programming plug-in for Rhinoceros, it is not a standalone application. It always runs parallel to Rhinoceros. Essentially the same modelling commands which can be used for 3D-modelling in Rhinoceros for 3d modelling are available in Grasshopper in the form of nodes or visual programming components. The whole process of creating a parametric model is thus done within the grasshopper editor environment, by setting desired visual scripting components and connecting them with other to achieve certain functionality. It can be understood as the creation of a visual algorithm. Grasshopper by itself has one directional data flow, from left to right, meaning loops are not possible.

Figure 05.01. shows a typical grasshopper component. From left to right it has an input, a name (or icon depending on display settings) and an output. The component draws a parametric rectangle whose dimension depend on the input number sliders. With setting the two sliders to the same number the component will output a square, the same thing will happen if one slider is connected to both x and y, however then you can only create squares, as you have no Y input available, it is constrained to one slider.

Figure 05.1
Grasshopper Rectangle Component with its inputs and outputs

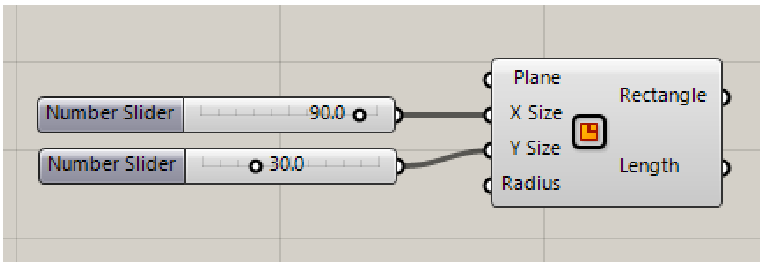


Figure 05.2
Grasshopper Rectangle Component Output in Rhino

To further illustrate how one creates parametric models with Grasshopper I will show one simple parametric model which can stem from this drawn rectangle.

Example – Flat slab 5x5 m, corner supported by columns 3 m height.

The illustrative parametric model is a geometrical model of a hypothetical flat slab 5x5 m with 3 m high columns, supported on four corners. For example, this simple model can then be used to generate a parametric structural model using the grasshopper Karamba3d plugin.

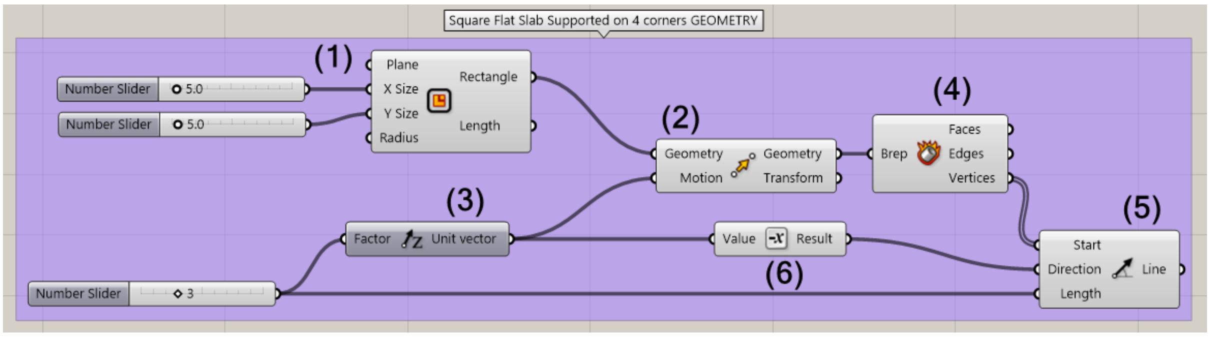


Figure 05.3
Grasshopper Rectangle Component chained with other components to create a simple flat slab geometry with corner column supports

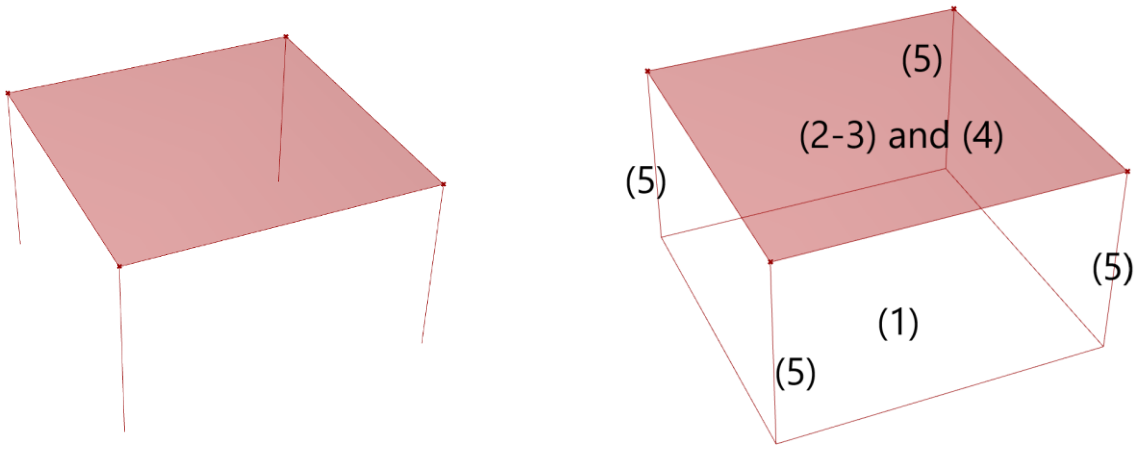


Figure 05.4
Output of Grasshopper definition, left – all modelled geometry previewed, right - only desired geometry previewed

To explain how the Grasshopper definition in 05.3. is set up to create the result in 05.4 right, the components used from (1) to (6) are named along with the results that occur due to putting down the component in the grasshopper editor:

(1) Rectangle component – Creates a Rectangle based on the X size and Y size, in the default base plane z=0 in Rhino if Plane input is unconnected.
(2) Move component – Moves geometry by a prescribed motion (vector direction and amplitude)
(3) Unit Z Vector – Vector of unit length in the Z direction, the factor controls the vector amplitude
(4) Deconstruct Brep – Deconstructs boundary representations (breps) into Faces, Edges and Vertices.
(5) Line SDL – Creates lines based on start direction and length.
(6) Negative – changes the sign of a value.

Modelling procedure:

1.) Generate a Rectangle , 5 x 5 m
Put “Rectangle” component (1) in the Grasshopper editor, two sliders set to 5, connect one to “X size” other to “Y size”. This generates a 5 x 5 rectangle in the base plane z=0

2.) Move the generated rectangle to 3 m height
Put “Move” component (2), connect the “Rectangle” output from (1) into “Geometry “ input of (2). Put “Unit Vector Z “(3) and a slider set to 3. Then connect the “Unit vector” output from (3) into the “Motion” input of (2). This moves the rectangle to a 3 m height.

3.) Find the vertices of the rectangle to serve as base points for lines which will represent columns
Put “Deconstruct Brep“(4). Connect the “Geometry” output from (2), the previously moved rectangle to 3 m height, to the “Brep” input of (4). This deconstructs our rectangle into “Faces”, “Edges” and “Vertices”. Now thanks to the “Vertices” output of (4) the base points to create column lines are created.

4.) Create Lines from each corner of the “floating” rectangle in the negative Z-direction and set the length to 3 m

Put ”Line SDL” component (5). Set the “Direction” input by using the already existing “Unit vector” output of (3) which should have the value of 3 m translation but in the negative Z direction (downwards). By using “Negative“(6) we make sure that the line direction is in the negative z direction. To make sure the length of the column is coupled with the slab height, use the factor slider connected to (3) to input “Length” parameter of (5). Next connect the “Vertices” output from (4) into the “Start” input of (5).

The simple geometry of the flat slab supported by corner columns is thus generated. Now using the three sliders we can change the X and Y dimensions of the slab, and the Height of our slab and columns. It is important to note that this is just one possible approach for creating the desired flat slab geometry. There are other possible approaches to create the same geometry.

For example, one approach could be instead of starting from a rectangle, one can start from modelling the columns first, this alternative grasshopper definition is shown in *Figure 05.5,* the reader is invited to study and compare with the original one as an exercise to check basic understanding. In the alternative definition, First create four base points, use the base points to set up lines in the vertical direction, find their top end points, use a polyline to connect the top end points, thus creating a rectangle shape on top of the four lines. The result is the same, however, the definition and control of the model is entirely different.

Instead of controlling X and Y directions by setting rectangle dimension now the user must set the X,Y,Z coordinates of the base points in order to control the X and Y sizes of the rectangle. In the end there is no right or wrong approach for creating parametric geometry in general.

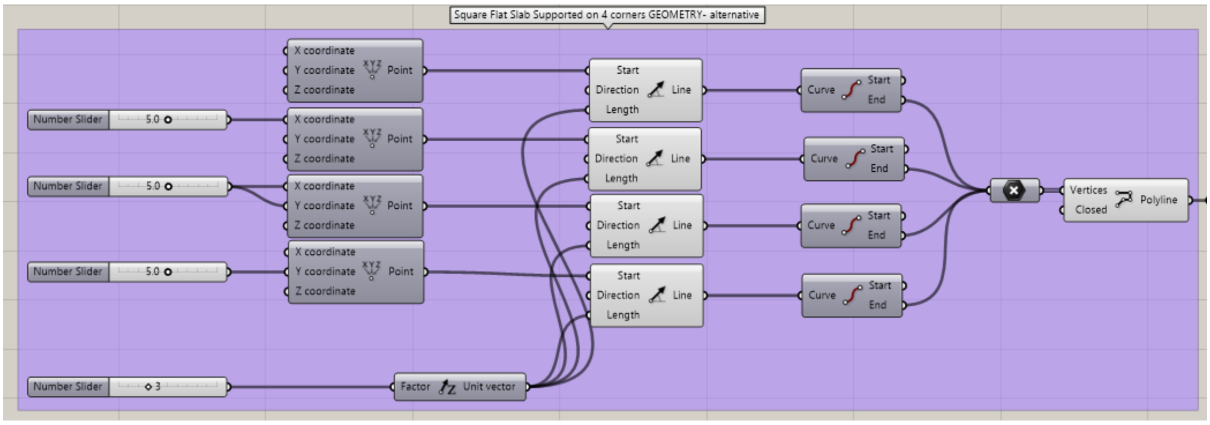


Figure 05.5
Alternative Flat slab geometry definition; Unconnected inputs of components have default values, 0 for the coordinates inputs of the point components, unit z vector direction for the Line SDL components

However, depending on specific cases and how the model is supposed to be controlled, each component is put with specific intent, to achieve desired control of parametric models. In the comparison between the two flat slab geometries one can see why the first definition is more straightforward and easier to manipulate. Less number of components, less repetition, and straightforward control of X and Y dimensions. It is simply easier to control the rectangle geometry with X and Y sizes then with setting base points to specific coordinates.

Karamba3d

Karamba3d is a grasshopper plugin for parametric structural engineering modelling tool developed by *Preisinger [18]*. This plugin is used to translate the parametric geometrical models generated in grasshopper into parametric finite element models which can be used for calculation, analysis and optimization of the generated parametric geometry. This tool is used as the main structural analysis software in this MSc thesis research. To illustrate the basic setup of a karamba3d model within a grasshopper definition, I will build upon the already shown example of the flat slab geometry, however a very detailed explanation of the karamba3d part of the grasshopper definition is omitted here, only a short overview of the main logic is presented, the detailed explanation is contained within Annex of this thesis.

Figure 05.6
Grasshopper definition – Parametric Structural model utilization of elements results, deformation is exaggerated by a factor 4000.

In Figures 05.6. the results of parametric structural model are displayed, left undeformed, right deformed structure. One can also extract the calculated utilizations per finite element as well as internal forces. By setting a deformation display factor one can exaggerate the deformation to see if it is as expected. Thus, while modelling in karamba3d one can easily see if something was modelled wrong due to instant visualization of results. In this case the deformation of the model is as expected, thus correct for the set inputs. Figure 05.7. displays the total grasshopper definition for a parametric structural model of a flat slab floor supported on corners by columns is shown. To better understand its structure, it has been divided into general code block categories from A to J which explain what is happening in the code. The categories are as follows:

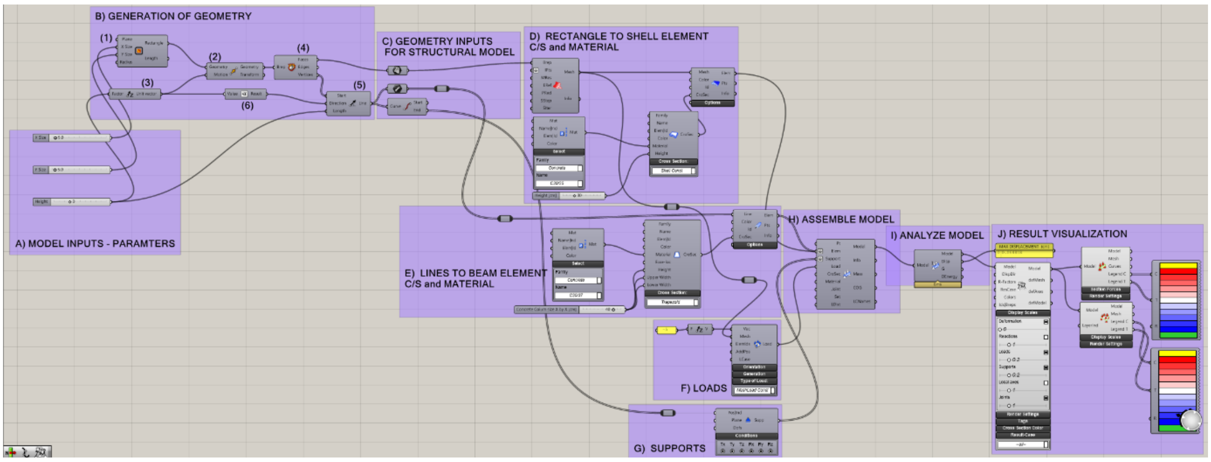
A) Model inputs – parameters
B) Generation of geometry
C) Geometry inputs for structural model
D) Rectangle to Shell – c/s and material
E) Lines to beams – c/s and material
F) Loads
G) Supports
H) Assemble model
I) Analyse model
J) Result visualization

Code blocks A and B are the geometry model shown in *Figure 05.3.,* code block C just collects all the important generated geometry to be used for the structural model, the rectangular face for the slab and the four lines for the columns, end points of lines to specify support conditions.

For any FEA program there are three general steps. Pre-processing, Calculation and Post-processing. The same three general steps are also applied during creation of the parametric structural model with Karamba3d. Code blocks A – G can be considered as pre-processing, G – I is calculation (processing) and J as post-processing.

Code blocks D – J contain Karamba3d grasshopper plugin components. The central component is “Assemble model” component in code block J (*Figure 05.10*) and in code block I “Analyse model I”, which calculates the assembled model according to first order theory (small deflections). The necessary inputs to create and calculate the structural model are “Elements”, “Supports” and “Loads”. Code block D (*Figure 05.8*) contains components which translate the meshed rectangular face to a shell element, assign a concrete material C20/25 and constant cross section thickness of 30 cm. Code block E (*Figure*

05.9) contains components which translate the four lines in to beam elements, assign a concrete material C30/37 and a rectangular solid cross section of 40 x 40 cm. The other inputs like CroSec and Material have default values set so one can run the model, however there is a possibility of specifying exact. This was done in the shown definition. The supports points are chosen by selecting lower end points of the 4 lines, with fixed boundary condition applied. The loads is a mesh load of -3 kN/m2 (the negative sign indicate the load direction is downwards)



B) GENERATION OF GEOMETRY
(1)
(2)
(3)
(4)
(5)
(6)
C) GEOMETRY INPUTS FOR STRUCTURAL MODEL
D) RECTANGLE TO SHELL ELEMENT C/S and MATERIAL
A) MODEL INPUTS - PARAMTERS
E) LINES TO BEAM ELEMENT C/S and MATERIAL
H) ASSEMBLE MODEL
I) ANALYZE MODEL
J) RESULT VISUALIZATION
F) LOADS
G) SUPPORTS

Figure 05.7
Grasshopper definition – Flat slab supported by columns in corners – Parametric Structural model

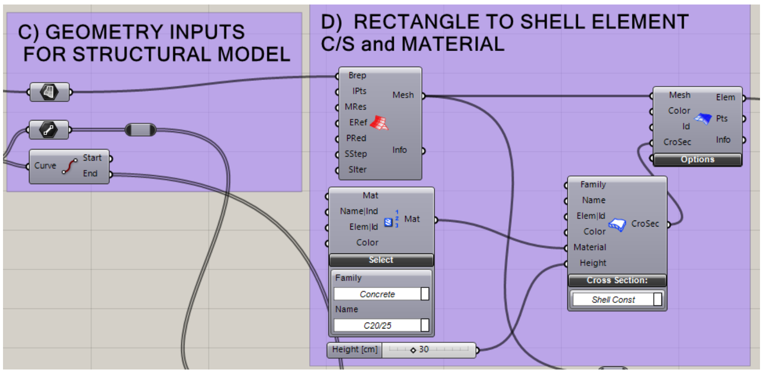


Figure 05.8
Grasshopper definition – C and D code blocks

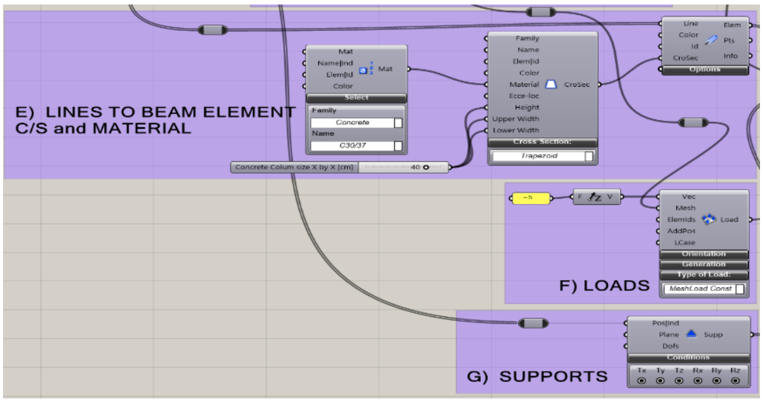


Figure 05.9
Grasshopper definition – E – G code blocks

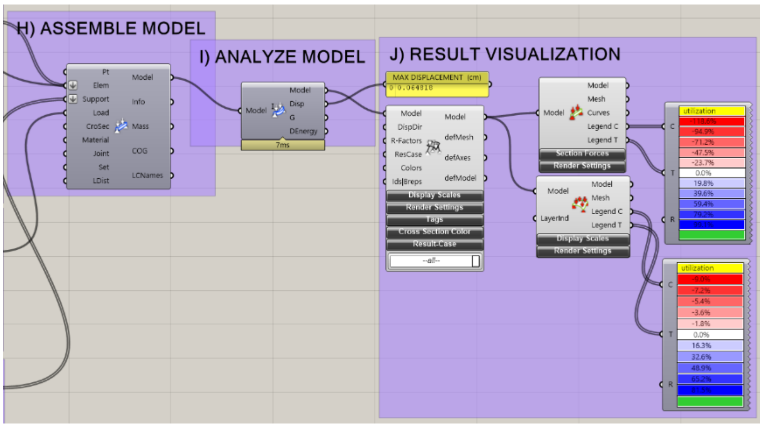


Figure 05.10
Grasshopper definition – H – J code blocks

06

Optimization

Rajput and Datta [20] define optimization as "the demonstration of ideal the best outcome under given conditions ". According to *Kelly [21]* "The optimization stage of the engineering design process is a systematic process using design constraints and criteria to allow the designer to locate the optimal solution". Thus, to surmise, optimization can simply be defined as the search for the best possible solution regarding certain posed criteria Considering how a problem is defined in terms of abstracted parameters which model all the relevant behaviour, there exists an imaginary design space where the combination of all possible parameter states, which represent potential solutions to the problem, exists. Thus, one can think of Optimization as a process of navigating the created design space of a specified problem, a search for the right combination of design parameters to achieve the desired result regarding set limitations. The size and shape (*Figure 06.1*) of this design space containing all possible solutions will depend on the number of parameters describing the problem and the size of their respective domains. In this chapter the most important concepts relating to methods of optimization, specifically structural optimization are laid out.

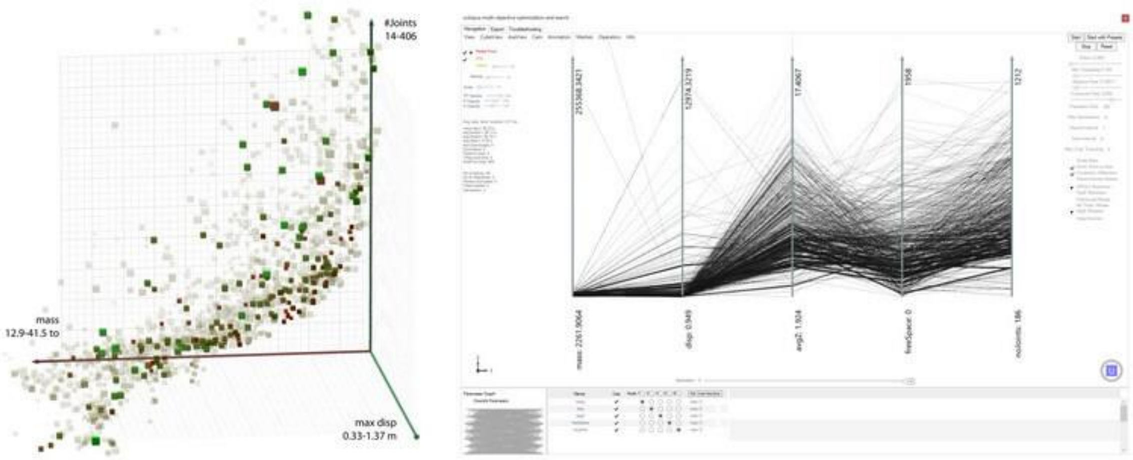


Figure 06.1
Possible visualization of design space

06.1 General concepts

Optimization in civil engineering, specifically structural engineering, has always been inherent, economy of design while respecting desired performance and functionality to be achieved has always been an essential principle. Typical for the traditional design process is the sequential development of a design throughout its design stages, following a deductive approach (from broad to specific, from preliminary design to detail design), accompanied with several iterations and reworks of the stages as needed. However, today with the tools available and the mostly computational approach in civil engineering design, optimization has moved from being an iterative procedure throughout the design phases to being a dedicated process done best in the preliminary phase, where the impact of decisions is the highest, and costs of changes are the lowest.

Structural optimization

The term structural optimization can be defined in a few different ways. One definition explains structural optimization as the process of finding the optimal design for a load bearing structure *[22]*. *Eschenauer et al. [23]* defined structural optimization as "the rational establishment of a structural design that is best of all possible designs within a prescribed objective and a given set of geometrical and/or behavioural limitations". Furthermore, it can be explained as the use of various numerical optimization methods for designing material efficient and/or cost-effective structures. In essence every optimization problem is described with an objective function which represents the desired goal to be minimized or maximized, and the set of variables with their respective ranges which define the behaviour of the problem and influence the objective function.

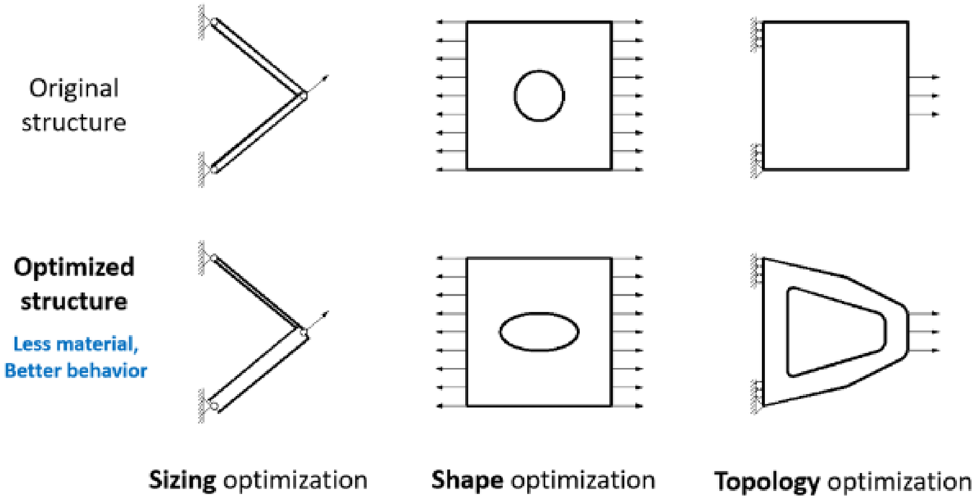


Figure 06.2
Structural optimization categories [22]

Categories of Structural optimization

According to *Mei and Wang [24]*. Structural optimization (*Figure 06.1*) can be categorized in the following:

- Size optimization, refers to cross-sectional areas of structural members as design variables
- Shape optimization, refers to disposition or configuration of nodal coordinates as variables
- Topology optimization, refers to nodal connections and supports, deleting unnecessary members to achieve optimal design
- Multi-objective optimization refers to combination of any of the above stated optimization to achieve even more specific optimization results.

In this thesis, the posed research problem of finding an "optimal" configuration of space frames in regard to three possible base mesh layers which form the bottom layer of the space frame (Tri, Quad, Skeleton-based Quad) and three Conway operator relations (Dual, Kis and Ambo) with the base meshes which form the top space frame layer, falls into the following categories of structural optimization:

1.) Size optimization (each space frame structure variant has optimized c/s sizes),
2.) Shape optimization (Tri, Quad, Skeleton-based quad)
3.) Topology optimization (Operators, Dual, Kis and Ambo)

Objectives of Structural optimization

Mei and Wang [24] summarize the following objectives of structural optimization occur most in literature.

- Cost minimization
- Structural performance improvement
- Environmental impact minimization
- Multi-objective, combining more than one of the above objectives

In this thesis the main objectives for evaluating optimality of designs are cost minimization (in this thesis equated with minimization of total mass of steel used in a particular design variant) and structural performance improvement (deflection, stiffness).

06.2 Methods of Optimization

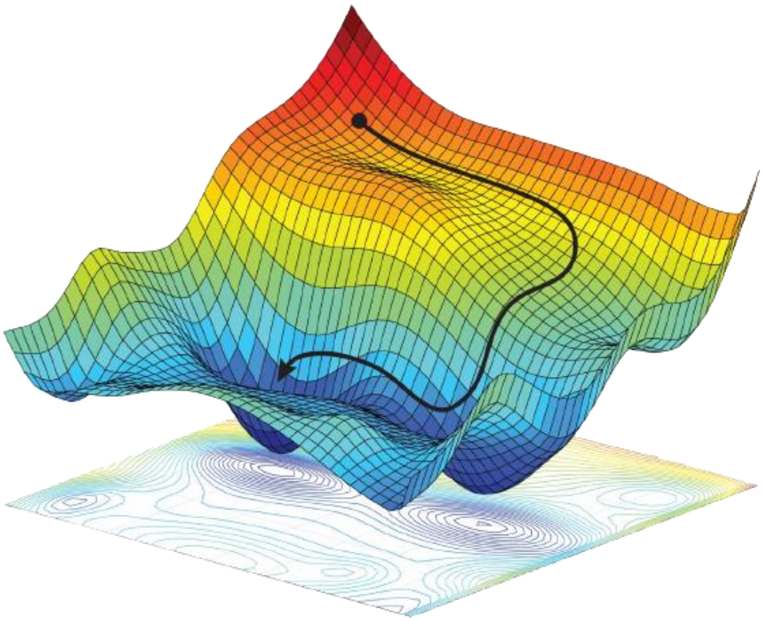


Figure 06.3
Gradient-based optimization visualization
[Source: optimal.uva.nl]

The methods of optimization can generally be classified as gradient-based approaches or heuristic approaches *[24]*. Gradient-based approaches (*Figure 06.3*) are explicitly formulated in terms of mathematical approaches such as linear and non-linear programming methods, optimality criteria methods and feasible direction methods. Heuristic approaches utilize a rule-based trial and error process. A sub-type of heuristic approaches named Meta-heuristic approaches are currently often applied to find optimized solutions without getting stuck in local optima, which often happens with gradient-based approaches. These Meta-heuristic optimization methods are often based on principles found in nature such as evolution (Genetic Algorithms), ant colony behaviour (Ant Colony Optimization), swarm behaviour (Particle Swarm Optimization), heating and controlled cooling of material (Simulated Annealing) etc. The scope of this thesis does not allow for full explanation of each of meta-heuristic algorithm available, only the two most common and understood ones will be shortly explained, namely Genetic Algorithms and Simulated Annealing.

All the above stated methods have been invented to avoid the most straightforward but not entirely feasible idea of opting for a brute force approach to calculate all the possible solutions for a design problem and then simply choose the best one. However, in using the meta-heuristic or heuristic methods of optimization, while great for traversing the design space in small enough time to achieve good enough results, one is always presented with a black box process. One sets the parameters and formulates desired goals to achieve and leaves all the heavy work to algorithms based on various underlying metaphors of natural phenomena.

Because of this black box process, and the complexity behind understanding the parameters and methodology of each one, one cannot always be sure if the achieved results are actually optimal, nor can the user grasp the design space of the problem. Thus, in this thesis both a Meta-heuristic approach was applied using Galapagos component in grasshopper for optimization (Genetic Algorithm and Simulated Annealing) to understand at what range of parameters a supposedly optimal solution is located at, and a constrained brute force approach (each parameter with a limited) utilizing Colibri plugin and Design Explorer visualization to understand the design space of the problem and compare with the meta-heuristic solution.

The Colibri and Design Explorer limited brute force approach was central in researching how particular space frame designs are influenced in terms of mass, stiffness, and deflection regarding the three possible initial meshings, three Conway operator relations and form finding height. Specific details of the process involved are described in Chapter 7 and 8.

Meta-heuristic methods of optimization

To grasp the concept of metaheuristic methods of optimization, a good start is to first understand the terminology. The term Metaheuristic, first coined by *Glover [25]* is constructed from Greek prefix meta, signifying something above, on a higher encompassing level, and heuristic, meaning to search or to find. In computational terms a heuristic is understood as a rule-based process. There exists a vast number of meta-heuristic methods, a good overview of such methods is presented by *Bandaru and Deb [26]*. In this article, he mentions 14 of the most applied meta-heuristic approaches and further lists 76 other possible meta-heuristic methods. *Yang et al. [27]*. give an excellent overview of meta-heuristics applicable in civil engineering design optimization problems, illustrating how many of the meta-heuristic methods can be applied to specific problems. One can see that the field of meta-heuristic algorithms is complex and ever evolving and each possible natural phenomena which can lend itself to an optimization allegory can almost certainly be turned into a meta-heuristic algorithm. The question of which meta-heuristic algorithm is best for a certain problem is still unanswered. While there are studies comparing a certain group of meta-heuristic approaches in between each other, for example *Zavala et al. [28],* there exists no study which encompasses a fully exhaustive comparison and listing of all possible meta-heuristic approaches. Thus, in this research the application of meta-heuristics is limited to the two most common and well-known metaheuristic approaches, Simulated Annealing and Genetic Algorithms.

Genetic Algorithm

Genetic algorithms are one of the oldest metaheuristic algorithms, invented by *Holland [29]*, they are based on the theory of natural selection which drive evolution as postulated by Charles Darwin *[27]*. Properties of natural selection include crossover, recombination, mutation, and selection. The procedure of a genetic algorithm can be summarized in the following steps:

1. Optimization of objective is encoded
2. Fitness function for selection of a particular solution is defined
3. Initialization of population of individual solutions
4. Evaluation of fitness function per individual solution
5. Generation of a new population according to the rules of natural selection.
6. Population is evolved until the prescribed stopping criteria is reached
7. Results are decoded as to obtain solutions to design problem

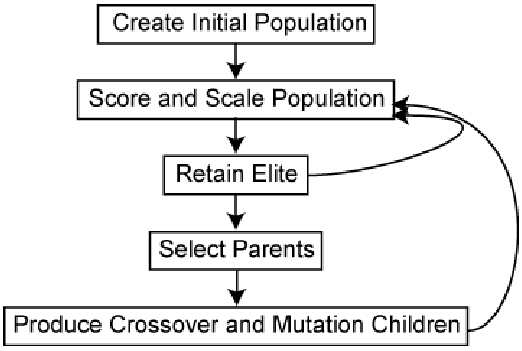

Create Initial Population
Score and Scale Population
Retain Elite
Select Parents
Produce Crossover and Mutation Children

Figure 06.4
Flowchart of main algorithm steps for a Genetic algorithm [Source: mathworks.com]

Put simply, a genetic algorithm modifies a population of individual solutions. In each step of the procedure the algorithm selects a certain set of individual solutions which perform best in regard to posed fitness criteria and selects them to be "parents" which will reproduce further to create bit more optimized solutions or "children" for the next generations (*Figure 06.4*). Thus, with each generation the population of solutions comes a bit closer towards the desired fitness function values, evolving from generation to generation.

Simulated Annealing

The Simulated Annealing, developed independently by *Kirkpatrick et al. [30]* and *Černý [31]*, meta-heuristic algorithm is based on the annealing process in metallurgy and materials science. Annealing is the process of heat treatment and subsequent slow cooling of solids which in turn changes their physical structure on the atomic level, and thus changes the material properties such as ductility and hardness. The process of heating a solid randomly rearranges its particles into the liquid phase. If then followed by a slow cooling process, all the particles begin to rearrange themselves into the lowest energy state, a regular as possible crystalline lattice. Simulated Annealing algorithm generates a random new solution in each step. Then the distance between each current point and subsequent new point is calculated based on a probability distibution propotional to the temprature scale (*Figure 06.6*). The algorithm takes into account the points that minimize the objective function values in regards to constraints and probability distribution. It systematically decreases the temperature and narrows the extent of search.

Steps of simulated annealing can be summarised in the following:

1.) Initialize – at each iteration a random initial placement of a trial point
2.) Distance – the distance between the current point and newly generated one is calculated based on a probability distribution in scale to the temperature
3.) Evaluation – the algorithm evaluates which points lower the objective function values and accepts them, along with some that raise with a certain probability in order to not be stuck in local minima/maxima
4.) Stopping criteria – once the objective function values become lesser than a prescribed tolerance function the algorithm will finish

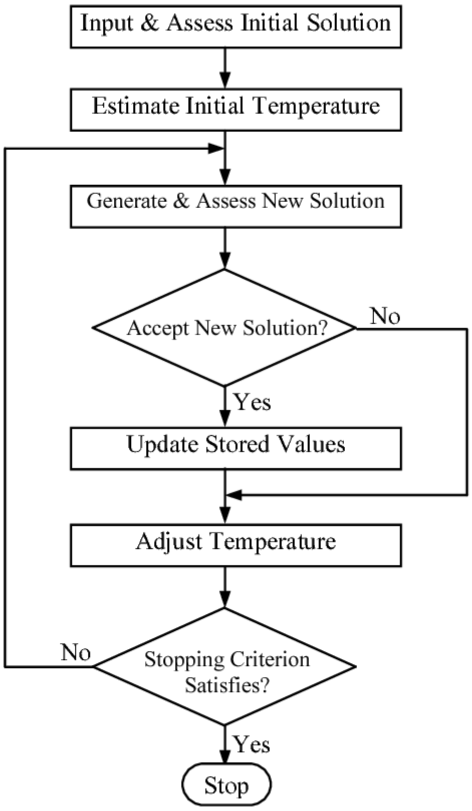

Input & Assess Initial Solution
Estimate Initial Temperature
Generate & Assess New Solution
Accept New Solution?
No
Yes
Update Stored Values
Adjust Temperature
No
Stopping Criterion Satisfies?
Yes
Stop

Fig. 1 The flow chart of a SA

Figure 06.5
Flowchart for a Simulated Annealing algorithm [32]

PART III
PARAMETRIC TOOL DEVELOPMENT

07 Space Frame Parametric Logic

This chapter describes the core ideas and abstracted structure behind the developed main research tool for this MSc thesis. The detailed procedure of how the tool works is presented in *chapter 08.*

The parametric tool was developed to study the influence of initial surface mesh topology on the structural design and optimization of space frame structures. The tool is developed in terms of a grasshopper definition utilizing kramba3d and compass singular plugins for grasshopper, which will be explained in detail in the following sections. It should be noted that this space frame parametric definition was not a result of first thought, but by numerous iterations and testing of functionality and approaches. Initially a different parametric logic was applied for research as opposed to the current one, where the main surface was discretized by creating subsurface using the Isosurface component in grasshopper, which then in turn created problems when dealing with trimmed surfaces. The main "aha" moment was the switch to meshing the surface and then utilizing the clear data structure of meshes to further base the space frame geometry. This has greatly improved the functionality of the tool both in terms of quality of the result and speed of the script. It is also important to note, that in the end, the developed tool for research can be freely used within an engineering design context as well. This was the one of the main guiding principles while developing the tool. Concepts presented in part I and part II, Literature Study earlier in this thesis, from established terminology to specifically the Conway operator implementation inspired by *Koronaki et al [3]*. and *Shepherd and Pearson [4]*, and the Topological Skeleton based quad meshing inspired by *Oval [14]* accompanied by relevant parts of the Python library he developed, named compas_singular, were applied during the development of the research tool.

07.1 Main Scheme

A Grasshopper definition has been developed to serve as the primary research tool. The tool or definition is used to generate space frame configurations. The space frame configurations in question consist of three mesh options; tri, quad, skeleton-based quad, to form the bottom layer of the space frame, and three Conway operator relations with the bottom mesh topology; dual, kis and ambo. These configurations can be researched on various input surfaces, to gain insight into, how the structural pattern of space frame structures (initial meshing of the surface accompanied with respective Conway Operators) influence the structural design and optimization. This research boils down to a question of how mesh topology can influence optimal space frame design (least structural mass).

To summarize, the generation of space frames is based on two approaches. Firstly, the initial input NURBS surface geometry is meshed by one of the three available meshing methods (Quad, Tri, Skeleton based Quad mesh) to create the bottom layer of the space frame structure. Secondly, the top layer is created by applying one of the three Conway Operators (dual, kis and ambo) on the initial bottom layer and afterwards offset by a desired distance (truss height).

After the creation of space frame geometry, it is translated into a parametric structural model, which has been developed to automatically optimize the cross sections for a particular set of space frame configuration input parameters. Next, the design can be further optimized by utilizing meta-heuristic algorithms to find good-enough combination of space frame input parameters in a short-enough time with the goal of minimizing the structural mass.

Thus, a full design loop consists of geometry generation, space frame cross section optimization, geometry optimization while always respecting cross section optimization, with the goal of minimizing the structural mass.

Furthermore, there is an option to use Colibri and Design explorer to carry out a limited brute force calculation of many structural configurations and record all the relevant data for each case in an excel sheet and visualize the design space in a parallel coordinate graph, which can in turn be used to understand how the structural behaviour is influenced by certain parameters. The combination of brute force approach and meta-heuristic optimization form the basis of methodology for researching how the space frame geometry consisting of; mesh topology, form finding height, truss height, and mesh density, influence the space frame structural behaviour. In search for the optimal space frame structural configuration, the brute force and metaheuristic optimization results are combined to visualize the optimization problem design space. The brute force approach results are spread throughout the design space, thus helping to constitute a bigger portion of it as possible, from desirable to undesirable space frame configurations, while the metaheuristic optimization results, are specific points in the design space which indicate where areas of optimums can be found and further compliment the design space resolution.

To fully understand the background of the all the input and output parameters involved in the developed tool, they can be divided in five main parts which constitute the main scheme of the tool:

A) Input data

Case specific data

- Surface geometry
- Support conditions

Definition specific data

- Loads
- Utilization
- Max displacement
- Buckling
- Bending
- Cross section set
- Second order theory

Mesh specific data

- Bottom layer mesh – base mesh
- Top layer mesh – offset mesh

Parametric specific data

- Truss height – mesh offset distance
- Mesh density – target number of faces

Optimization specific data

- Cross section optimization
- Space frame fitness parameters

B) Space frame generation

Geometrical model generation

C) Structural analysis

Structural model generation

Structural analysis

D) Optimization
Cross section optimization
Galapagos – metaheuristic optimization

E) Data recording and visualization
Colibri
Design Explorer

07.2 A) Input data

The input data category is divided into five main subcategories. Each of the categories has been defined to explain how the developed tool inputs should be used and have been used. The first step of utilizing the research tool is to set appropriate values for the input data categories.

Case Specific data

This data category is related to each specific case and is used to investigate a certain space frame configuration based on an input surface. A case consists of an input surface geometry accompanied with boundary conditions. In this MSc thesis research, the investigated case was a pentagon shaped double curved surface, supported on the external boundaries of the surface.

· *Surface geometry*

Firstly, this part deals with the initial input surface, which can be freely chosen, the tool was developed to handle almost any geometry, be it convex polygon based, concave polygon based, then relating to curvature, flat, singly curved, doubly curved, or free form and containing openings or not. This flexibility was developed to accommodate as much architectural freedom as possible and to have as much potential research options as possible. In this research these input surfaces are always long span structures (100 meters or more) mainly because space frames become more appropriate structural systems the longer the span is.

· *Support conditions*

Secondly, case specific data relates also to the support conditions, which can be based on external boundaries of the input surface, internal boundaries, combination of the two and finally point supported in characteristic points.

Definition Specific data

Relates to the general parameters which facilitate the comparison between various cases (input surface geometries accompanied with their discretization (choice of meshing) and configuration (Conway operator relation between top and bottom layer). They facilitate comparison by being invariant in the research, however there is an option to freely change them if one wishes to. However, it is important to pay attention to these parameters to have realistic structural behaviour and analysis.

· *Loads*

Always specified in terms of kN/m2 as a uniform vertical surface load. This is a total load, which encompasses the influence of snow, wind, and non-load bearing layers, considering a symmetrical fully loaded surface. Everything except self-weight which is automatically included. The chosen design load is 2.5 kN/m2, while this number might seem arbitrary, this load is adequate considering usual loadings of structures. A higher load can of course be specified if wished, however it is only important that each space frame configuration has the same load per m2 and same boundary conditions to be able to compare the results.

· *Utilization*

The utilization of space frame elements is limited to 0.9, thus a small safety factor of 1.1 is applied. This 10% reduction is a general precaution due to the complexity and large span of the space frames considered to mitigate possible deviations from the real-world behaviour of the structures, and to avoid the possibility of overstressed members.

- *Max displacement*

For large spans, for example 100m and more, a choice of l/300 is considered adequate. Thus, the maximum displacement is always limited to 30 cm.

- *Buckling*

The structural analysis of the space frame structure can be carried out with or without considering buckling. In all cases buckling is considered, otherwise results would be potentially misleading.

- *Bending*

The structural analysis of the space frame structure can be carried out with or without considering bending as well. In all cases bending is not considered, since space frames are 3-dimensional truss structures, they are considered as pin jointed, or hinged, no bending moments allowed, and loads are only applied in the nodes. Furthermore, the influence of bending moments in a well-designed space frame should be minimal or non-existent. Bending is relevant for space grid structures, which have no truss height to take up bending moments, and their connections must be considered as fixed to carry loads and achieve stability.

- *Cross Section Set*

There is a possibility to include many standard steel cross-sections as well as a possibility to limit their amount, however this limitation will influence the cross-section optimization, potentially leading to skewed or invalid results because of a potential lack of large enough sections or the possible omission of light sections through a set cross-section limitation. In this research circular hollow sections are chosen as the primary set of cross-sections according to EN10210-2, which consists of 251 possible cross section options, ranging from Φ21.3 x 2.3 mm up to Φ1219.0 x 25 mm

- *Second order theory*

The possibility to analyse the structure according to second order theory is present as well. However, this type of analysis is not relevant for axially loaded, pin jointed structures, free of bending moments, and the general assumptions of space frame behaviour. This relevance was tested, and results showed no influence. Thus, first order theory is chosen.

Mesh specific data

This data category relates to the mesh geometries being investigated. This data category is supposed to be changed to investigate how different mesh geometries influence load bearing behaviour of space frames.

- *Bottom layer mesh – base mesh*

This is the most important input data, basis of the research done. An initial mesh topology is chosen to form the bottom space frame layer, on which Conway operators will be applied to form the offset mesh. The relevant mesh options are Quad, Tri and Skeleton based quad mesh types.

- *Top layer mesh – offset mesh*

Using the created base mesh topology, the top layer mesh is generated using Conway operator relations, dual, kis and ambo.

Parametric specific data

This data category contains the main space frame parameters which are to be varied to find the optimal solution for a set case and mesh topology.

- *Truss height /Mesh offset*

Parameter slider from 1m to 5m maximum. This parameter controls the truss height of the space frame, that is the specified offset distance between top and bottom layer of mesh

- *Mesh density*

Target number of faces, the higher the number the smaller the space frame mesh cells.

The larger the density the heavier the computation of the script. The lower limits for the number of faces parameter are set such that the smallest possible discretization of the surface is 1m x 1m cells, and the maximum is 5m x 5m

- *Form finding height*

Expressed as the rise between top and bottom layer of space frame measured at the centroid of the surface shape. This way form finding influence can be easily visualized and measured. The difference between a form found space frame and not form found is that the form found will not have a constant truss height throughout the structure but a slightly cambering one depending on the form finding height. Form finding height is unrelated to space frame truss height.

Optimization specific data

This data category contains optimization specific data. It is intended to be constant in each space frame configuration calculation but can also be adjusted if wished.

- *Cross-section optimization*

By utilizing karamba3d Optimize cross section algorithm component, in each space frame configuration calculation (a specific set of parametric data), the most appropriate cross sections are chosen. The main inputs here are the table of cross sections to be applied in the design, along with the possibility of limiting the displacement.

- *Space frame structural mass optimization – metaheuristic algorithms*

Choice of algorithm for optimization by utilizing native grasshopper component Galapagos used for genetic algorithms or simulated annealing metaheuristic algorithm optimization. The main inputs in this step are the parametric specific data and the fitness function which is set to be maximized or minimized. The parametric specific inputs are iterated based on a meta-heuristic algorithm and a thus created space frame configuration is evaluated based on the fitness function (minimization of mass in case of this research)

07.3 B) Space frame generation

Depending on the chosen input parameters in step A, in this step the space frame geometry is generated instantly. The input surface has been meshed in a certain configuration constituting the bottom layer, and afterwards based on chosen Conway operator relation the top layer is formed. Then the two layers are connected in such a way to form a 3-d pyramid module space frame type, thus completing the space frame geometry model generation procedure

07.4 C) Structural analysis

The generated space frame geometry model from step B is translated into a parametric structural model using karamba3d parametric structural analysis plugin and calculated according to first order theory. In each new iteration the corresponding space frame geometry is structurally analysed and verified.

07.5 D) Optimization

In this step the analysed space frame is optimized by varying the input parameters and evaluating per each parameter set the fitness function (mass). This done by utilizing one of the available meta-heuristic algorithms, the genetic algorithm or simulated annealing algorithm. After several iterations of various parameter sets, an optimized solution is found, meaning the algorithm converges and differences between subsequent solutions become negligible.

07.6 E) Data recording and visualization

Instead of using the black-box approach of optimizing by using meta-heuristic algorithms, there is an option to set the range of each parametric specific data input in order to brute force calculate a large number of space frame configurations by using the Colibri plugin, with all possible parameter combinations (according to set range of parameters) and record all the relevant output variables in order to capture the structural behaviour of the space frame in relation to the iterated input variables. Each solution is thus recorded by Colibri in an excel .csv file which can then be inputted into Design Explorer to view a parallel coordinate graph which encodes all the relations between the chosen parameters and output results. This graph, accompanied with the data in the .csv file is then used for analysing the results and concluding how a space frame configuration influences its load bearing behaviour.

08 Space Frame Parametric Tool Procedure

After presenting the main scheme of the tool in the previous chapter, the detailed procedure of how the tool works is presented here. The main scheme served as the main top-level overview of the logic behind the tool, in this section a much closer look at how the tool functions is shown. The detailed procedure of the developed grasshopper definition is presented in the following steps:

1.) Input free-form surface
2.) Bottom layer mesh – base mesh
3.) Form finding
4.) Top layer mesh – offset mesh
5.) Deconstruct meshes – Conway operator top mesh - connectivity
6.) Forming of space frame geometry
7.) Structural analysis of space frame
8.) Optimization
9.) Data recording and visualization

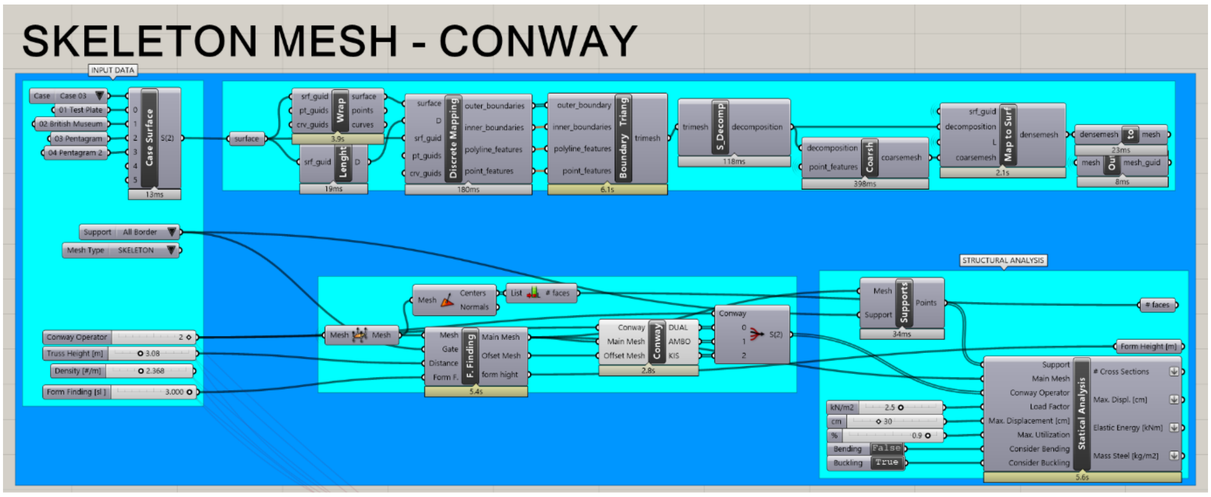


Figure 08.1
Developed parametric tool – grasshopper definition

08.1 Input free form surface

This is the first step in utilizing the tool. The user inputs a desired NURBS surface to be transformed into a space frame structure. The tool allows for complete freedom in choice of the input surface. In this step the case specific data is thus inputted. Some of the examples below illustrate the various surfaces to be translated into a space frame structure. They range from the most basic, flat plane surface with no openings towards the most complex, double curved with openings. The developed tool will mesh each of the inputted surfaces respecting its inner and outer boundaries. Surfaces such as the double curved y shape surface can be created in Rhino by first creating a top view outline of the surface to be created then creating a surface out of this closed boundary line and afterwards turning the control points of the NURBS surface on to manipulate them according to the desired geometry.

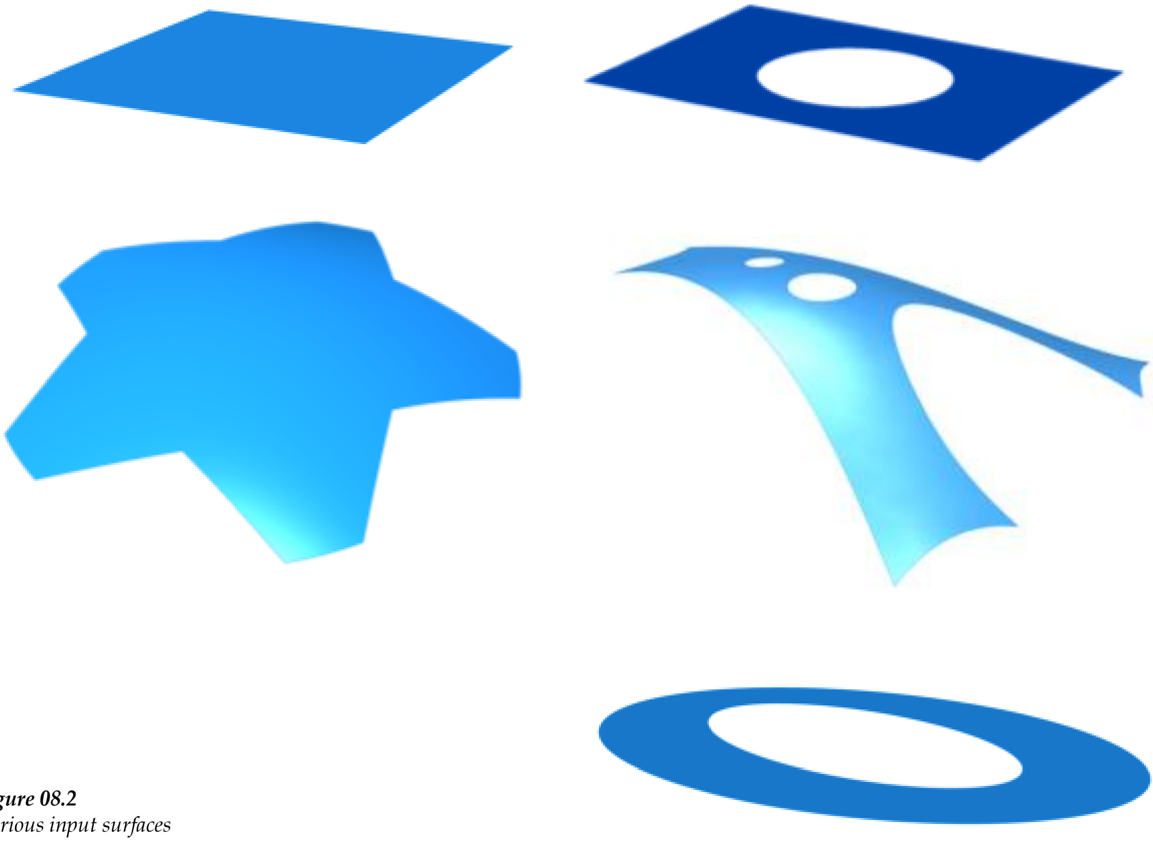


Figure 08.2
Various input surfaces

08.2 Bottom layer mesh – base mesh

After choosing the desired input surface geometry in step 1, the next step is to choose one of the three meshing options to discretize the input surface into the base mesh for forming the bottom of the space frame layer. One can choose triangular, quadrilateral and skeleton-based quadrilateral meshing procedures. The tri and quad meshing procedures are facilitated by native components newly available in Rhino 7 version, QuadRemesh and TriRemesh. The Skeleton-based quadrilateral mesh is generated utilizing compass_singular python library developed by Oval. Following the choice of meshing procedure, one then needs to set the target number of mesh faces, or the mesh density. The higher the density the smaller the average size of the mesh cells, the more intricate the space frame mesh. Mesh density is set either by defining the target number of mesh faces or by setting the average size a x a of the mesh cell.

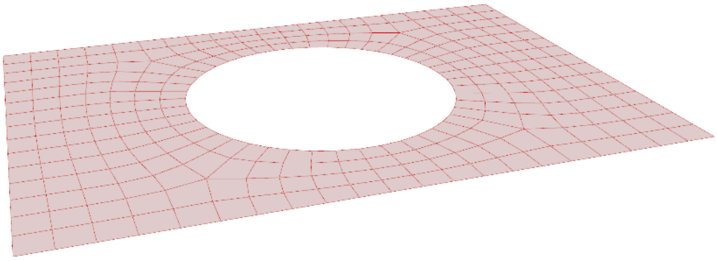


Figure 08.3
Base mesh

Initially the developed definition facilitated all three mesh option types, however later it was divided into three separate definitions, a quad, tri, and skeleton-based definition. This was done to simplify the script and speed up its calculation.

Below an illustration of how each of the three meshes look like on a pentagon shaped double curved surface viewed from a plan view.

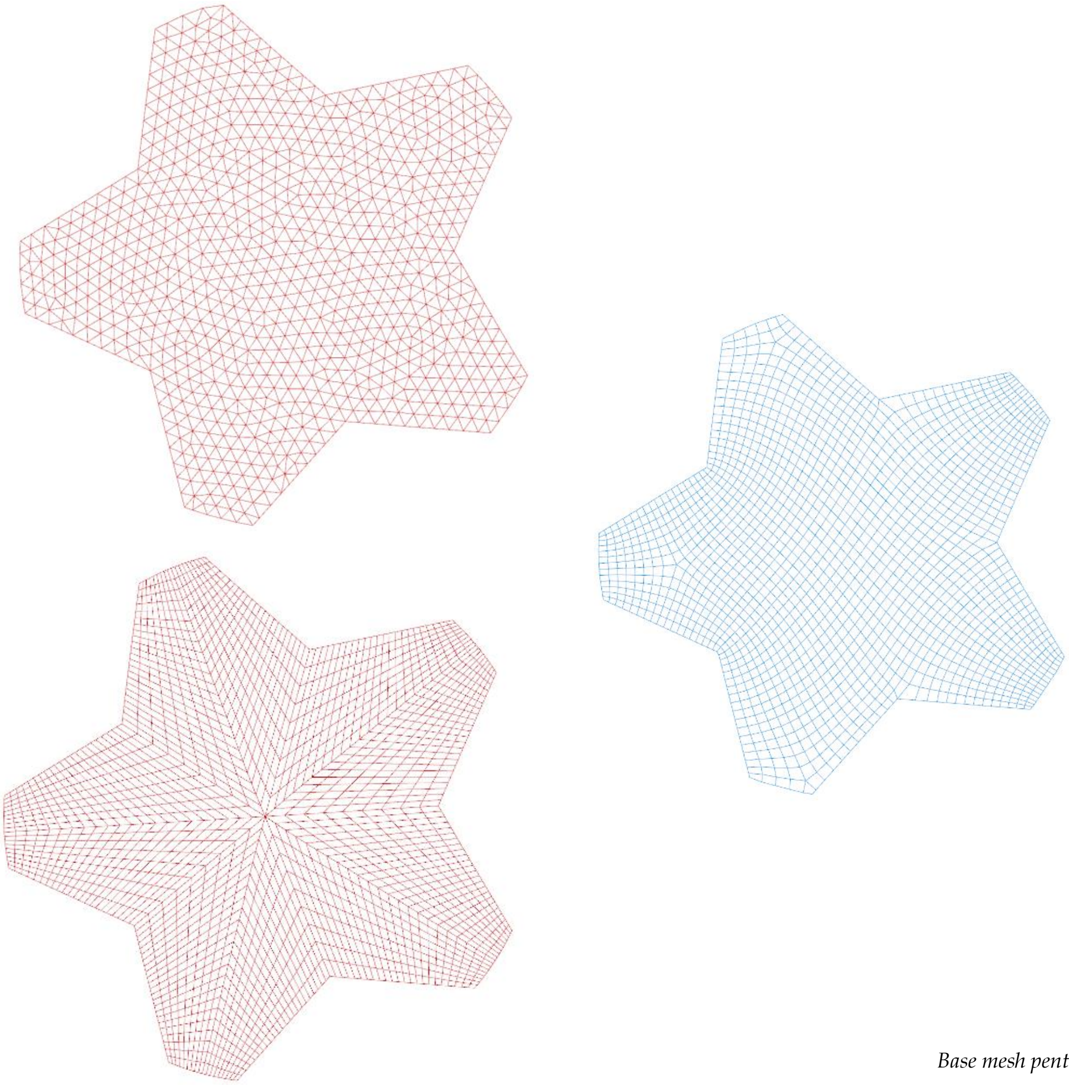


Figure 08.4
Base mesh pentagram example

08.3 Form finding

Once the bottom layer is formed, if specified a flat input surface initially, the option of form finding can be used to achieve a form found shell like surface for the space frame configuration. The bottom mesh is the main input for the form finding which is controlled through the form finding height parameter which is measured as the vertical distance from the centroid of the bottom mesh shape to the centroid of the form found mesh shape. This is also a highly important feature of the tool, as in the case study the influence of this form finding is investigated. It is a question of, starting from a flat space frame, is there a certain benefit in the cambering which can be achieved by form finding. This cambering or the form finding height can be very small for example 1-5 m or if 20 m, then the height turns from being a slight cambering towards the clear height of a shell-like surface with a space frame structure. The cambering or form finding height should be kept within certain limits, specifying, for example 40 m form finding height, one will generate a highly exaggerated surface which is like to an impossibly steep arch in an unfeasible grotesque futuristic gothic cathedral.

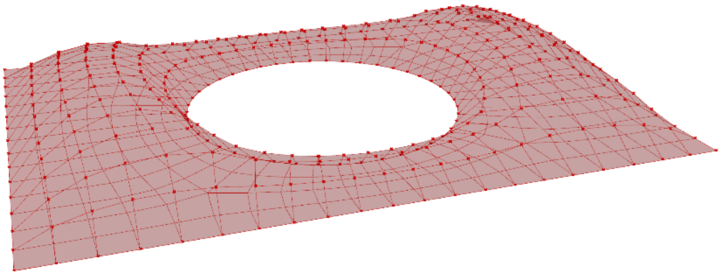


Figure 08.5
Form finding

08.4 Top layer mesh – offset mesh

After forming the base mesh layer, and form finding it if included, this form found mesh (or just the base mesh if the surface was already curved) is simply offset by a specified distance to create the top layer mesh. At this step we have formed the bottom and top layers of the space frame which are currently identical.

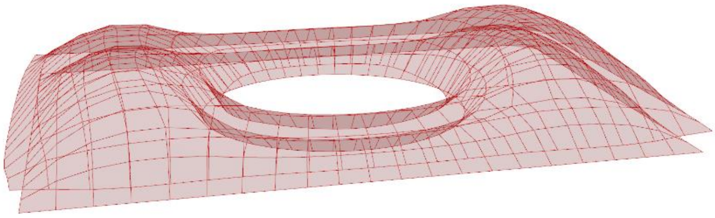


Figure 08.6
Top layer offset

08.5 Deconstruct Meshes – Conway operator top mesh - connectivity

After the meshes have been offset, each of them are deconstructed in to faces, edges, vertices and mid points of the edges. This is due to two main reasons. The first reason is to create the Conway operator relations to form the final geometry of the top layer mesh, by specifying a certain order of connecting the points. The order of connecting vertices is different for each Conway operator. Second reason is to specify the connectivity between the points of the top and bottom layers. The connectivity specified is to achieve a pyramid module. Essentially whatever the discretizations of the surface are, they are always connected to form pyramid modules. The pyramid shapes will depend on the shape of the base they are formed, or more specifically the shape of the mesh faces.

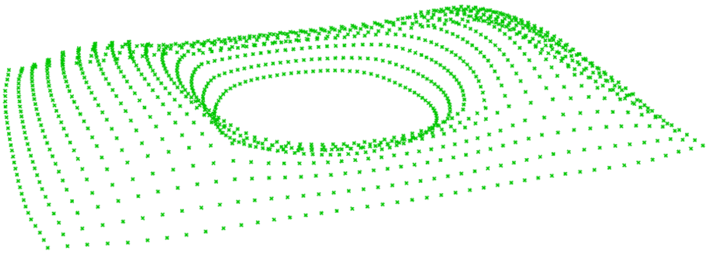


Figure 08.7
Deconstructed mesh points

08.6 Forming of space frame geometry

Based on the deconstructed meshes and set rules to connect the points, the space frame line geometry is created, the top, bottom, and web layer. The development of this part of the definition, the connection of all the generated points into the correct geometries, was one of the more demanding parts. To correctly connect the points with line components, to generate desired geometry, care must be taken in managing the data structure of the points.

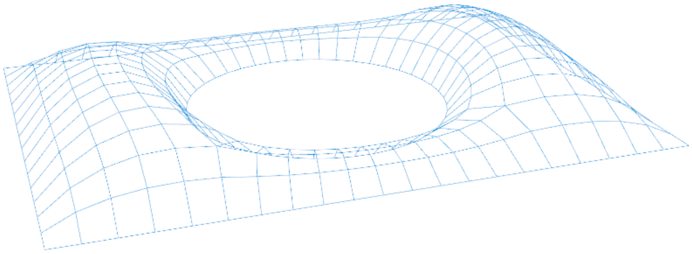


Figure 08.8
Bottom layer

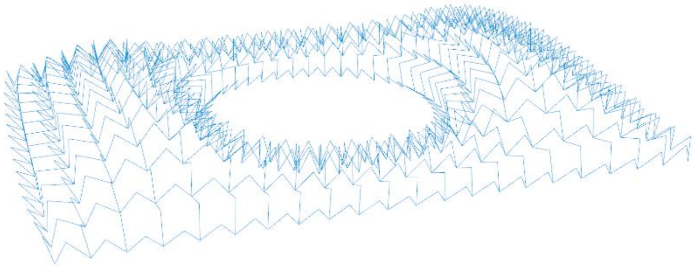


Figure 08.9
Web layer

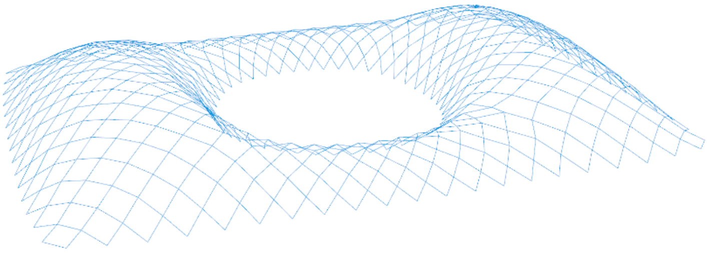


Figure 08.10
Top layer

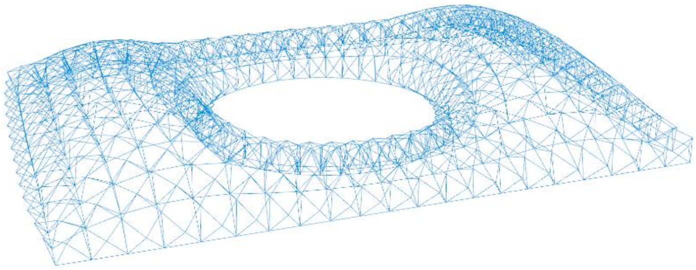


Figure 08.11
Space frame geometry

08.7 Structural analysis of space fame

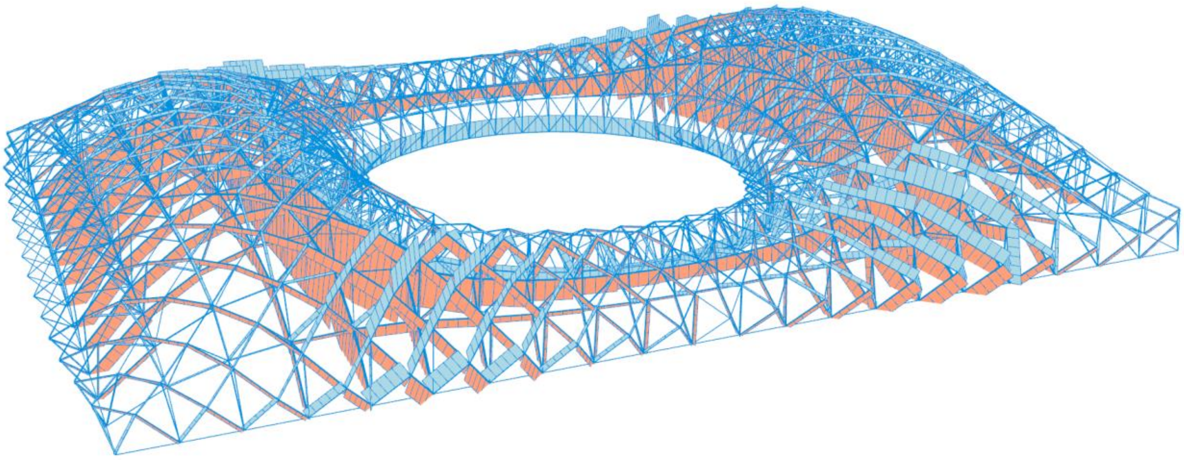
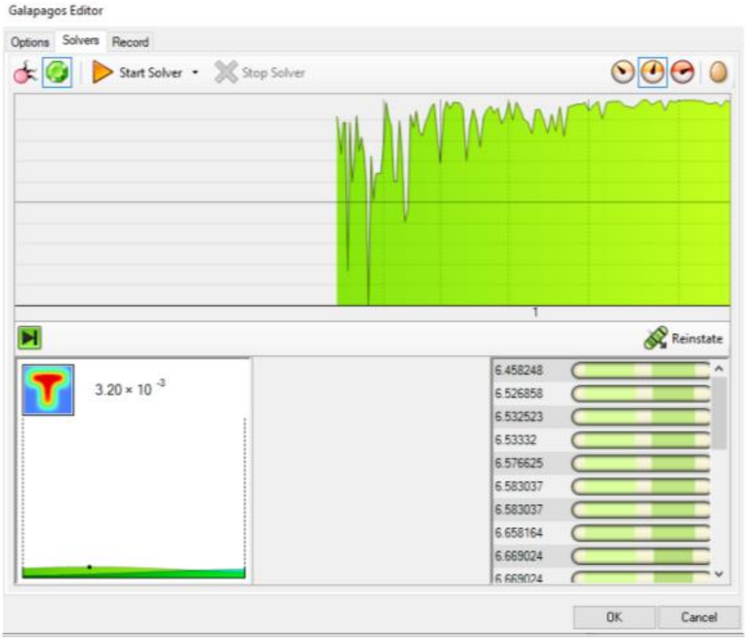


Figure 08.12
Internal forces – structural analysis

The generated parametric space frame geometry model is transformed into a parametric structural model. The lines are translated into truss elements and the points to hinged nodes and the points on the lower outer boundary of the space frame into pinned (translations in x y z directions fixed, rotations free) supports. An initial default set of cross sections are defined, and later automatically dimensioned. Self-weight is considered, a uniform mesh load is specified by using the top mesh as the input, meaning that the specified kN/m2 load is divided into appropriately distributed concentrated loads located in the top mesh vertices in the negative z direction (gravity). Steel S355 is assigned as the material. The dimensioning is done according to Eurocode 3, buckling considered, bending not considered. The structural model results are total mass, number of different cross sections utilized, displacement, internal forces, utilization of members. To minimize the mass of the structure, which is the primary fitness function for metaheuristic optimization, in every iteration the cross sections in the model are optimized along with the geometrical parameters of the space frame truss height, number of faces, and form finding height.

08.8 Optimization

Optimization can is done by using the Galapagos native grasshopper component which has the option of either utilizing Genetic algorithms or Simulated Annealing algorithm for meta-heuristic optimization methods. Optimization is done by varying the geometrical inputs of the space frame model (truss height, face number, form finding height), with a goal of minimizing the fitness function, the structural mass of the space frame. In each run of the algorithm a variant of the space frame geometry is structurally analysed, cross section optimized, its mass recorded. Through iteration the algorithms eventually converge to a particular set of input variable values, which give the lowest structural mass. The final solution can be considered as optimal, but not in the full sense of the word. The nature of the design problem and the size of the design space do not allow for finding just one optimal solution. Instead, the found solution is considered as good-enough achieved in short-enough amount of time. In this thesis research the meta-heuristic optimization mostly serves to pinpoint a plausible location of an optimized solution and arrive at an initial estimate of lowest structural mass. By location, the values of the geometrical parameters of the minimal mass version of the space frame are meant. This approach is complementary to the main limited brute force approach.

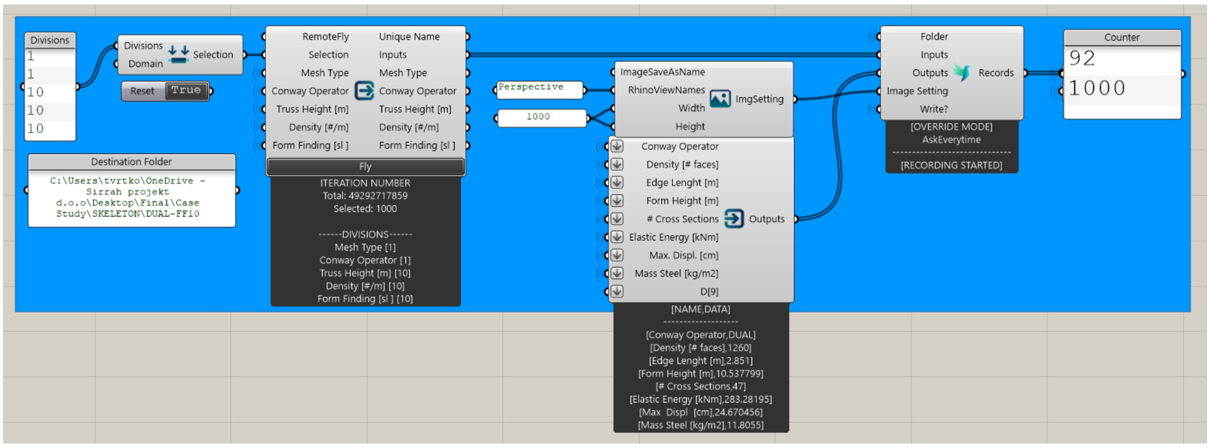


```
6.458248
6.526858
6.532523
6.53332
6.576625
6.583037
6.583037
6.658164
6.669024
6.669024
```

Figure 08.13
Galapagos optimization – Simulated annealing

08.9 Data recording and visualization

To gain insight into how the mesh topology (base mesh, Conway operator relation for top mesh), space frame geometry (truss height, face number, form finding height) influences the load bearing behaviour of space frame (structural mass, number of cross sections), instead of relying on the black box process of meta-heuristic optimization, instead a limited brute force approach is applied. Using the colibri plugin, one can set several parameters of interest, and then set the number of divisions of each parameter value range. For example, if the truss height parameter division is set to 10, and the height ranges from 0 – 5 m, then there will be created 10 space frame configurations of truss heights with an 0.5 m increment (0.5, 1.0, 1.5, …, 4.5, 5.0). This division of each parameter is to avoid a combinatorial explosion. In *Figure 8.11,* the colibri setup is shown for a triangular base mesh case.



```
Divisions
1
1
10
10
10
```

Reset True

Destination Folder: C:\Users\tvrtko\OneDrive - Sirrah projekt d.o.o\Desktop\Final\Case Study\SKELETON\DUAL-FF10

```
ITERATION NUMBER
Total: 49292717859
Selected: 1000

------DIVISIONS------
Mesh Type [1]
Conway Operator [1]
Truss Height [m] [10]
Density [#/m] [10]
Form Finding [sl ] [10]
```

Perspective

1000

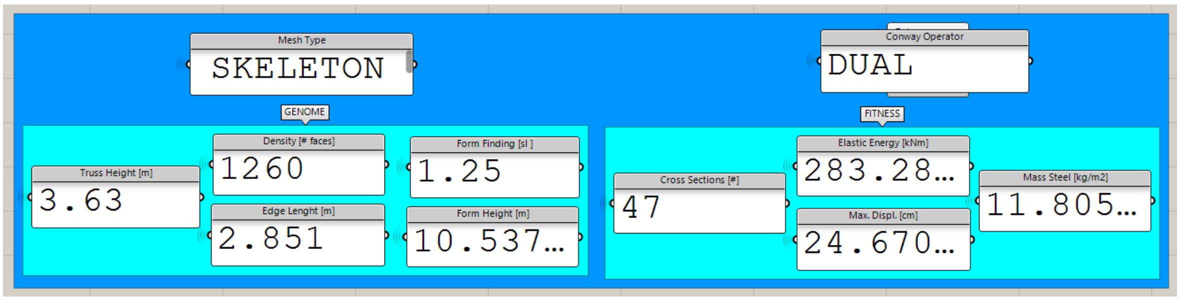
```
[NAME,DATA]
------------------
[Conway Operator,DUAL]
[Density [# faces],1260]
[Edge Lenght [m],2.851]
[Form Height [m],10.537799]
[# Cross Sections,47]
[Elastic Energy [kNm],283.28195]
[Max  Displ  [cm],24.670456]
[Mass Steel [kg/m2],11.8055]
```

```
[OVERRIDE MODE]
AskEverytime
----------------------------
[RECORDING STARTED]
```

```
Counter
92
1000
```

Figure 08.14
Colibri parameter divisions and number of iterations

The relevant parameter and the divisions of their ranges are shown in the black outline below the component in the figure. Due to having three definitions, one for each base mesh type, the division of mesh type is only 1. There are three Conway operator relations to be investigated.



Mesh Type: SKELETON

Conway Operator: DUAL

GENOME

Truss Height [m]: 3.63

Density [# faces]: 1260

Edge Lenght [m]: 2.851

Form Finding [sl]: 1.25

Form Height [m]: 10.537...

FITNESS

Cross Sections [#]: 47

Elastic Energy [kNm]: 283.28...

Max. Displ. [cm]: 24.670...

Mass Steel [kg/m2]: 11.805...

Figure 08.15
Data recorded for each configuration calculated in a .csv file, intended for visualization in terms of a parallel coordinate graph by using Design Explorer

Truss height, density and form finding parameters are all set with divisions of 10. Thus, considering all the parameters and their divisions, there will be 1 x 3 x 10 x 10 x 10 = 3000, space frame configurations to calculate. The total number of iterations possible, based on the size of each selected parameter domain, is 2 894 400 000 (two billion, eight hundred ninety-four million, four hundred thousand). To calculate each solution is not feasible, thus the limited approach with specifying a number of divisions per each parameter. For each of those configurations, results are recorded into a .csv excel file, which can then later be inputted into Design Explorer to visualize, in a parallel coordinate graph, how each of the calculated configurations relates to the input and output parameters. Furthermore, the data collected can be used to create graphs which show how each of the base mesh layers is influenced by each of the parameters (details in Chapter 09).

PART IV
CASE STUDY

09 Star polygon surface

09.1 Input surface geometry

In this chapter the case study procedure and results will be presented. The chosen input for the case study is a non-convex polygon, star polygon shape, flat surface, with a span of 160m. This surface is chosen because it is not clear from the beginning which mesh topology for this surface type would be considered as optimal, or if any of the different space frame configurations based on those mesh topologies have better structural performance than others. In the case of a rectangular flat surface, it is clear a simple structured quad mesh with a dual Conway operator relation to the top layer would be optimal, as this is the standard well known space frame configuration (pyramid modules, square base), executed many times so far.

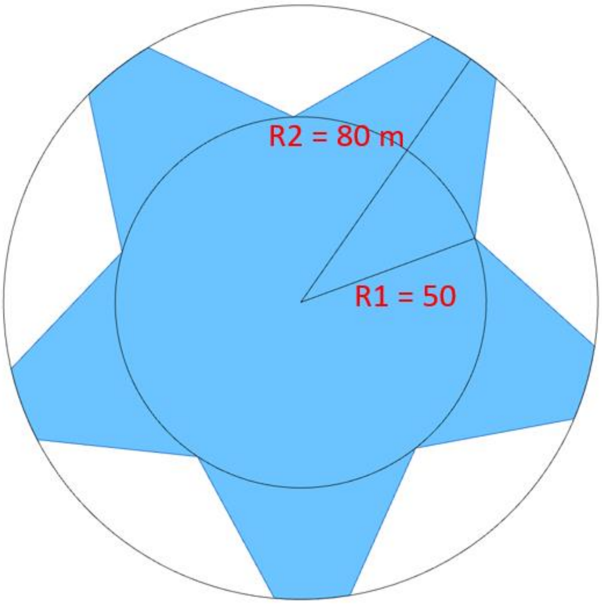


Figure 09.1
***Plan view of input surface with** dimensions*

Figure 09.2
3d shape (with form finding)

09.2 Research Goal

The main research goal of this thesis was to research the influence of mesh topology on the structural performance optimization of steel space frame structures by developing a parametric tool to generate a vast number of space frame configurations.

The mesh topology being investigated in this thesis, consists of the three base mesh types; quad, tri, skeleton-based quad, accompanied with three possible Conway operator relations for each base mesh (dual, kis and ambo) to form the top mesh. Thus, the resulting space frame configurations from investigating those mesh topologies (on a certain input surface), can be put in the following categories based on mesh and operator combination:

1.) Quad base mesh + Dual, Kis, Ambo (Q+D/K/A) – *Figure 09.3 – 09.9*
2.) Tri base mesh + Dual, Kis, Ambo (T+D/K/A) – *Figure 09.10 – 09.16*
3.) Skeleton-based quad base mesh + Dual, Kis, Ambo (S+D/K/A) – *Figure 09.17 – 09.23*

The letters in the parenthesis represent a shorthand way of writing the names of result categories. To generate the various space frame configuration based on different pairs of base mesh and operator, a brute force approach was used to generate and record 3000 thousand solutions per category by utilizing Colibri plugin and then visualize the recorded data in a parallel coordinate graph with Design Explorer.

Afterwards, the same recorded data was used to form graphs which show the relation between mesh operator pairs evaluated against both fitness functions, namely, mass (F1 vs Form Finding - *Figures 09.35-09.44*) and stiffness (F2 – vs Form Finding – *Figures 09.45.-09.54)*. As the initial brute force approach results showed that form finding has significant influence on optimal structural configurations, it was necessary to investigate in more detail, how exactly it influences the optimal solutions.

Furthermore, metaheuristic optimization is also carried out and its results are recorded in the same graphs. The parallel coordinate graphs (*Figures 09.24-09.33.*) are used to investigate trends of how space frame configuration parameter relate to optimality criteria. With this case study three general questions should be answered regarding the space frame configurations on the pentagram surface:

Q1.) Which combination of mesh and operator, out of the nine possible, will perform best and in which category?
Q2.) Is there any benefit to form finding, what is the influence on F1 and F2?
Q3.) How do the input parameter values influence the optimal solution?

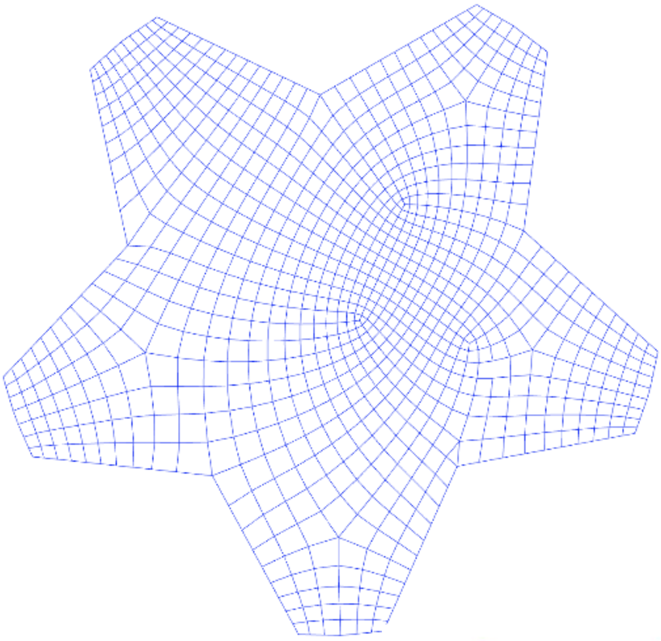
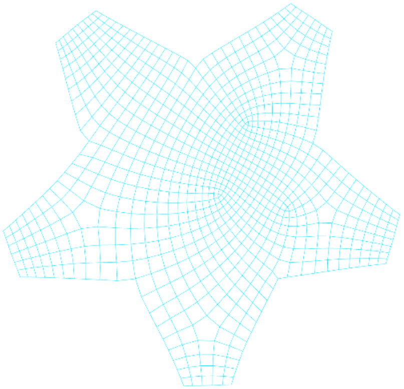
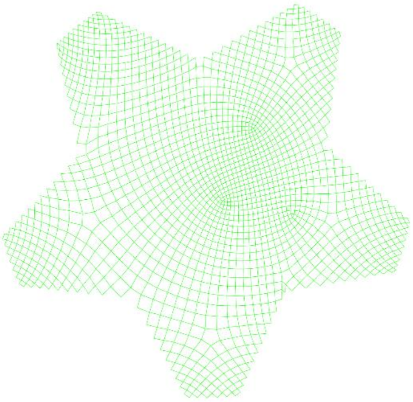


Figure 09.3
Quad mesh





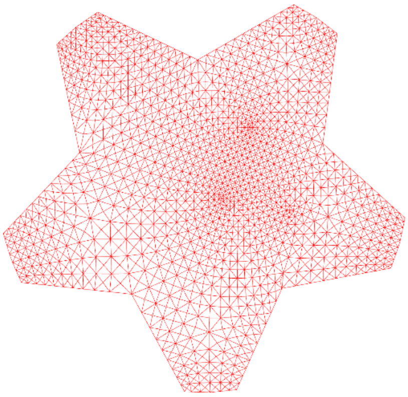


Figure 09.4: *Dual of quad mesh*

Figure 09.5: *Ambo of quad mesh*

Figure 09.6: *Kis of quad mesh*

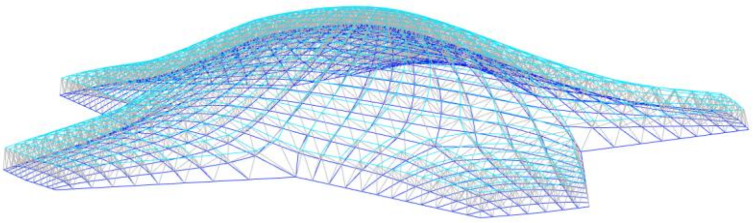


Figure 09.7
Quad – dual, 3d + form finding (FF)

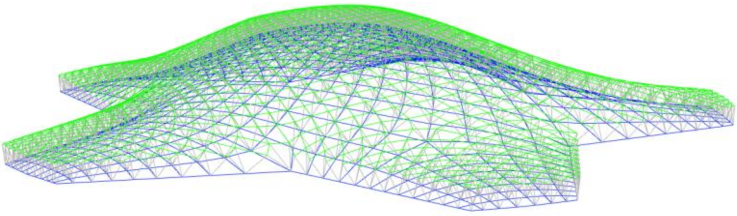


Figure 09.8
Quad – ambo, 3d + FF

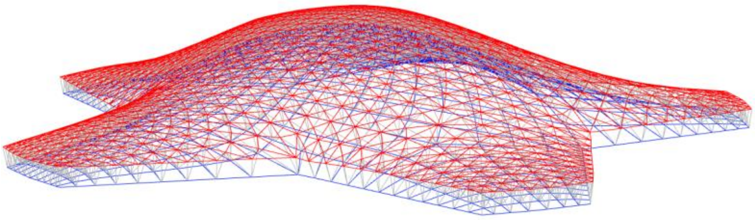
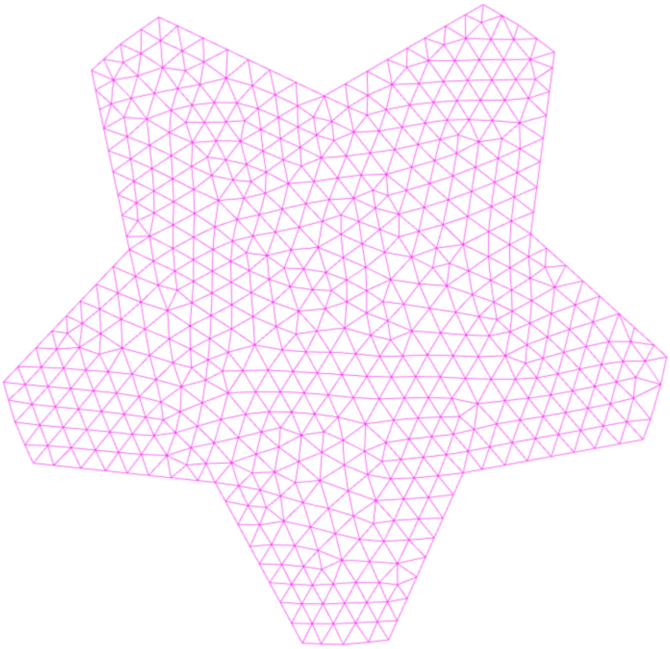
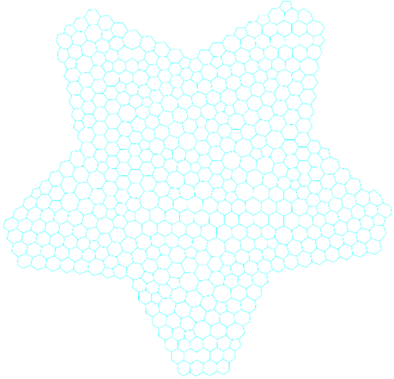
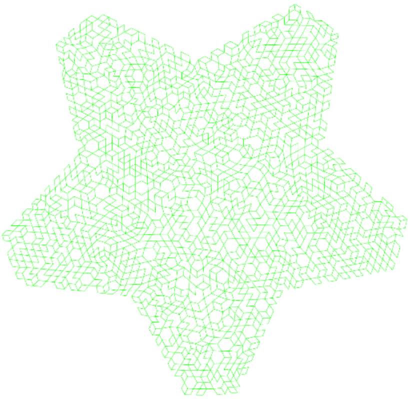


Figure 09.9
Quad – kis, 3d + FF

Figure 09.10
Tri mesh







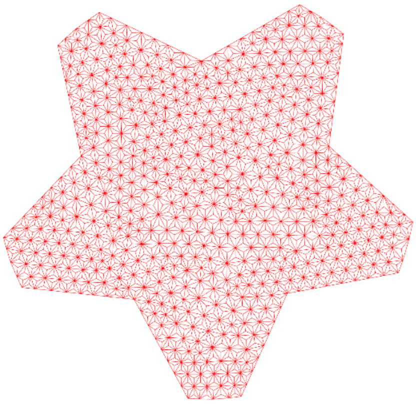


Figure 09.11: *Dual of tri mesh*

Figure 09.12: *Ambo of tri mesh*

Figure 09.13: *Kis of tri mesh*

Figure 09.14
Tri – dual, 3d + FF

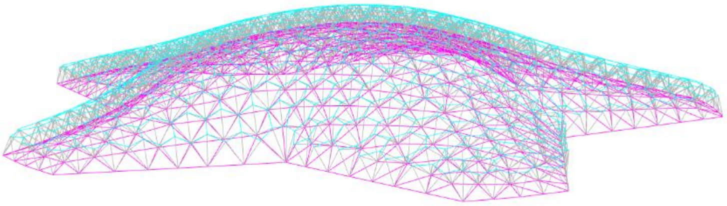


Figure 09.15
Tri – ambo, 3d + FF

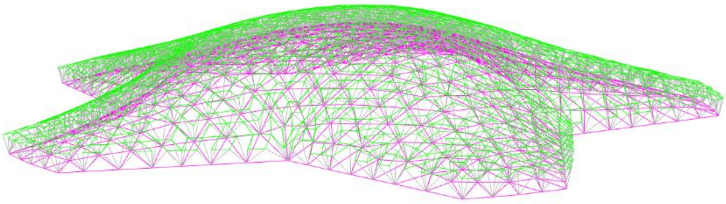
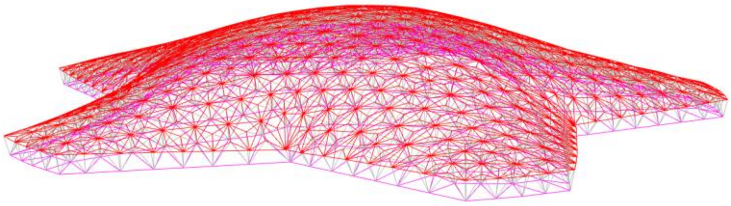


Figure 09.16
Tri – kis, 3d + FF



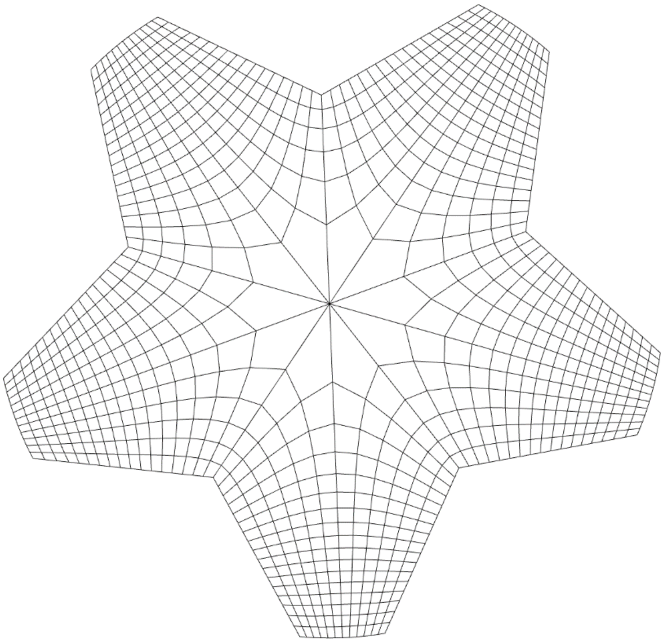


Figure 09.17
Skeleton-based quad mesh

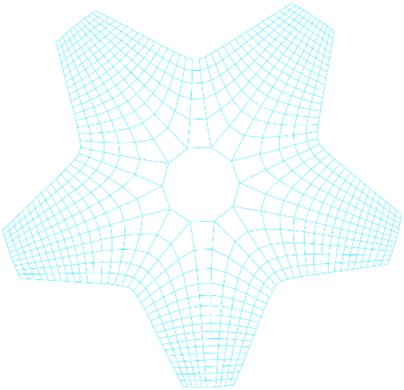


Figure 09.18. *Dual of skeleton mesh*

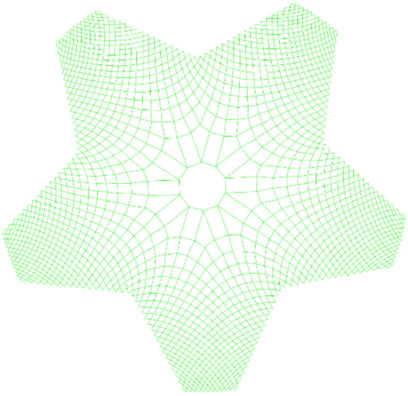


Figure 09.19. *Ambo of skeleton mesh*

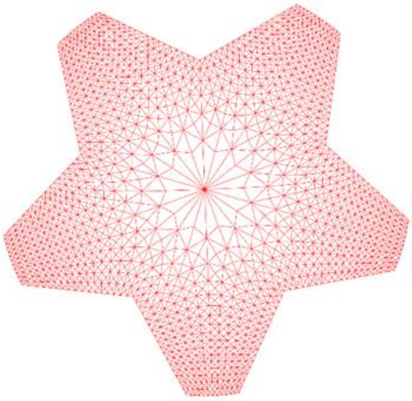


Figure 09.20. *Kis of skeleton mesh*

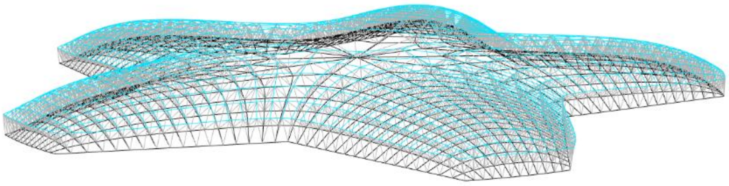


Figure 09.21
Skeleton - dual, 3d + FF

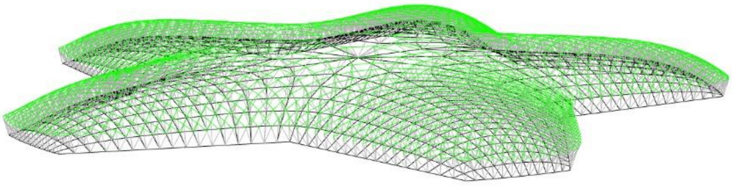


Figure 09.22
Skeleton - ambo, 3d + FF

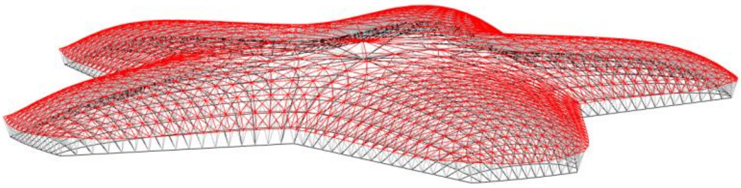


Figure 09.23
Skeleton – kis, 3d +FF

09.3 Process and details

The procedure of evaluating the input surface was done by using three separate grasshopper definitions: quad mesh definition, tri mesh definition and skeleton-based definition. Each of the definitions are the same in structure, the only difference is the meshing procedure. With three sperate definitions it was easier to both track and calculate solutions and lessen the risk of crashing or freezing of software. Utilizing the Colibri plugin for each of the three definitions, 3000 space frame configurations have been calculated, thus 9000 in total. For each grasshopper definition the "Definition specific data", as described in chapter 7, is kept constant, which enables comparison between different solutions, as they are all under the same "laboratory" conditions. All the parameters in the "Parametric specific data" category as described in chapter 7 are recorded for each calculated solution. The fitness parameters F1, structural mass, and F2, stiffness is also recorded. The following steps are repeated in each iteration:

1) Input data
2) Bottom/base layer mesh
3) Offset mesh
4) Deconstruct meshes
5) Top layer creation
6) Forming of space frame geometry model
7) Structural analysis – struct. analysis + c/s optimization
8) Recording results and corresponding parameter values

The possible parameter values or ranges, both input and output, recorded at step 8) are as follows:

Input:

- Mesh type: Skeleton, Tri, Quad
- Conway operator: Kis, Ambo, Dual
- Truss Height – 0 – 8m
- Density – 0 – 5000 # (Edge length – 0 – 12m)
- Form height – 0 – 40 m (Form finding)

Output:

- # Cross sections – 0 – 80 #
- Elastic Energy – 0 – 2000 kNm
- Mass steel - 0 – 200 kg/m2

The 3000 space frame configurations per definition are generated by specifying the following divisions of the following input parameters:

Mesh type – 1 division (either Q, T, S)
Conway operators – 3 divisions (D, A, K)
Truss height – 10 divisions – range from 0 to 8 m
Form finding height – 10 divisions – from 0 to 10 m
Number of mesh faces – 10 divisions – from 0 to 5000

By multiplying the number of divisions of each parameter the total number of configurations is obtained, 1 x 3 x 10 x 10 x 10 = 3000. In each grasshopper definition this number of divisions is the same. This all possible thanks to the Colibri plugin. It functions as an automatic iterator and data recorder. The calculation of each of the 3000 solutions although automatic, is a relatively time consuming and computationally heavy process. The time to calculate a solution primarily depends on the type of grasshopper definition used and number of faces specified for the solution.

The definition type influence is simply because of the three different meshing algorithms taking different amounts of time to generate a solution. There is also a slight influence of the generated patterns, some are more intricate and thus simply have more line geometry, considering same number of faces.

The time for the quad mesh definition to generate a solution, ranges from 5 – 30 s depending on mesh density. For the Tri mesh the calculation times range from 10 to 60 s and for the Skeleton 20 to 120 secs. Based on experience during the process, the total time for generating 3000 solutions is about 10 – 12 hours, thus 36 hours of computational time for generating 9000 solutions. In design practice this large a resolution of design space might not be needed; the computational time could be just a few minutes or hours. Especially if the design boundary conditions have The preliminary design analysis in coordination with other potential design team members could be carried out within a working day.

Furthermore, metaheuristic optimization was carried out for each of the 9 possible space frame configurations in order. The fixed parameters were, Mesh

09.4 Analysis and Discussion of Results

After generating the 9000 solutions (space frame configurations) and recording the resulting state the 11 parameters for each solution, into .csv file, this .csv file is input into Design Explorer to generate a parallel coordinate graph of the parameter results. This parallel coordinate graph is then used to analyse trends and categorized the results. (*Figure 09.24*) The analysis is based on a deductive approach of viewing the results in the parallel coordinate graph. Within Design explorer there is an ability to omit desired axis from the graph to lessen the visual clutter and focus on certain aspects, that is "zoom-in" on a set of relationships of interest. The "zoom-in" approach is also used when viewing how each of the displayed parameter values influences the other. One can select a range on the axis to be omitted. In other words, depending on what ones to research, the view of the parallel coordinate graph can be adjusted accordingly and thus form specific categories of results.

The categories were formed on the following rationale. In any constituted result research category, there are two main fitness values which are considered as optimization goals, namely, least amount of steel (F1) and highest stiffness (F2). The constituted categories are either absolute or relative. By Absolute category it is meant that the none of the five input parameter ranges are limited, while the relative categories are simply formed by narrowing further the scope of parameter ranges, based on general trends, assessed from the brute force results (investigating parallel coordinate graph containing all 9000 solutions), showing an indication where to further narrow the parameter ranges. The absolute categories can be understood as broad design spaces which contain both desirable and undesirable space frame configurations (undesirable solutions have an extremely high structural mass, or extremely high deflections/low stiffness), while relative categories represent more realistic structural configurations. All of the above was done in order to assess in what general direction should further research go in regard to parameters and possible configurations.

Thus, the specific categories constituted are:

1.) Category 1 :Absolute – full parameter ranges of all five input parameters
2.) Category 2 : Relative– form finding height parameter limited to 2% of span
3.) Category 3 : Relative – truss height parameter range limited from cca. 5 m

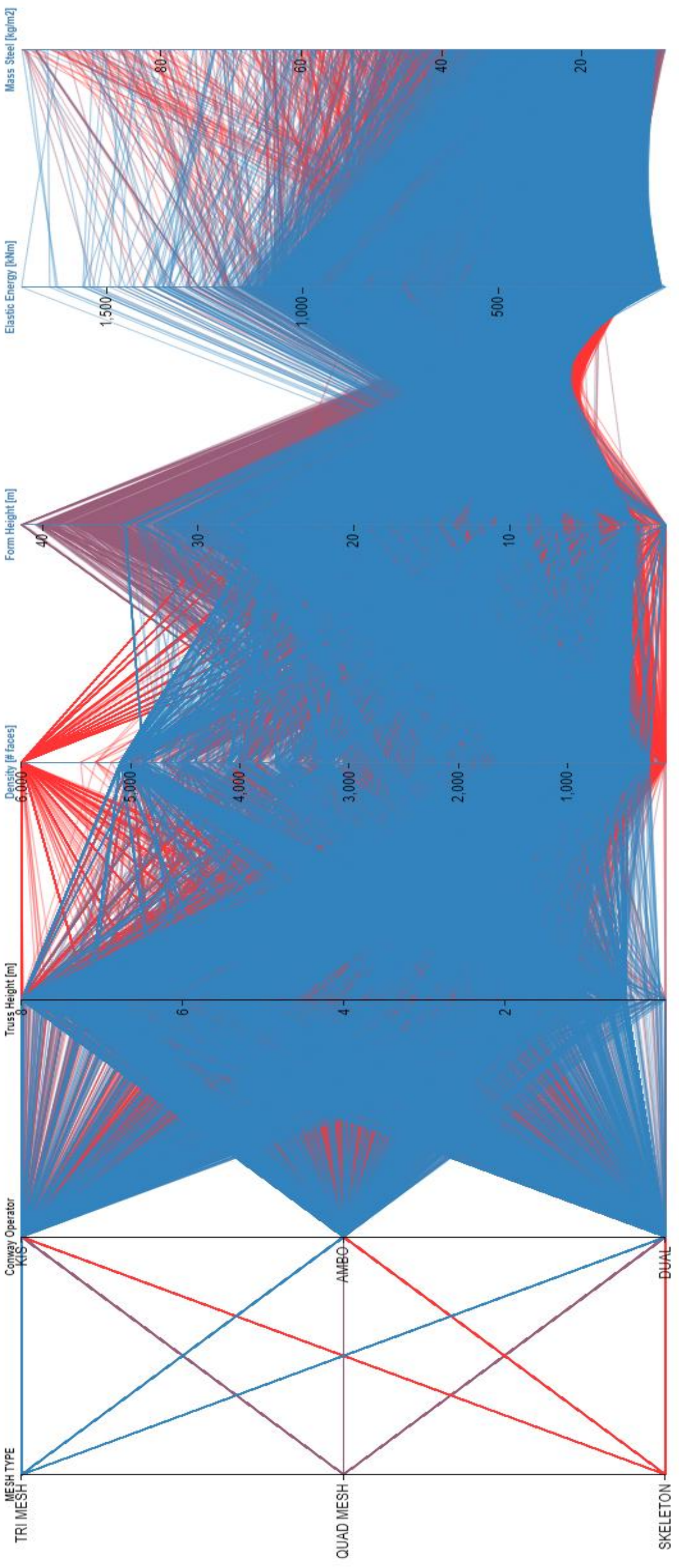


Figure 09.24
Parallel coordinate graph – 9 parameter axes

To find which combination of mesh and operator performs best for each category, the parallel coordinate graphs are adjusted accordingly. In *Figure 09.25*, the graph of all results is shown again, the red circle indicates the parameter range of interest. For the absolute category 1.), we first narrow the selection of solutions by viewing only the solutions which have the least amount of mass, afterwards the same is done for highest stiffness . This means narrowing the scope of the mass steel axis in the parallel coordinate graph indicated by the red circle (F1) and elastic energy (F2). Thus, the heaviest results and least stiff are omitted.

Category 1: Absolute, full input parameter ranges

The relevant axes set to be viewed in the graph, from left to right are as follows: Mesh type (Q/T/S), Conway operator (D/K/A), truss height (0-8 m), mesh density (0-5000 faces), form height (0 - 40 m), stiffness/elastic energy (F2), mass (F1). The colours make it easier to identify which mesh type is in question when viewing the results. Blue = Tri mesh, Purple = Quad mesh, Red = Skeleton mesh.

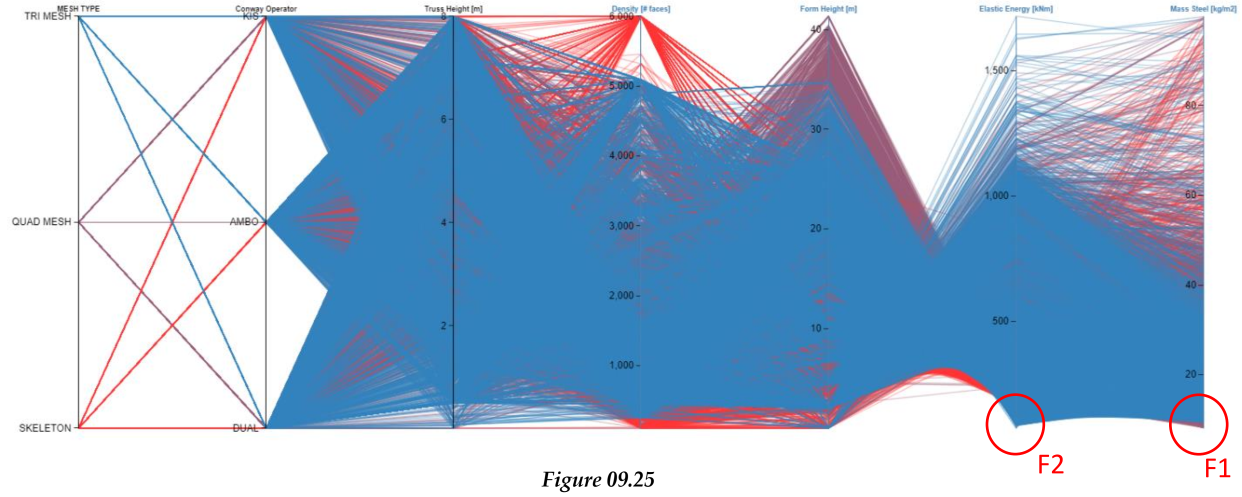
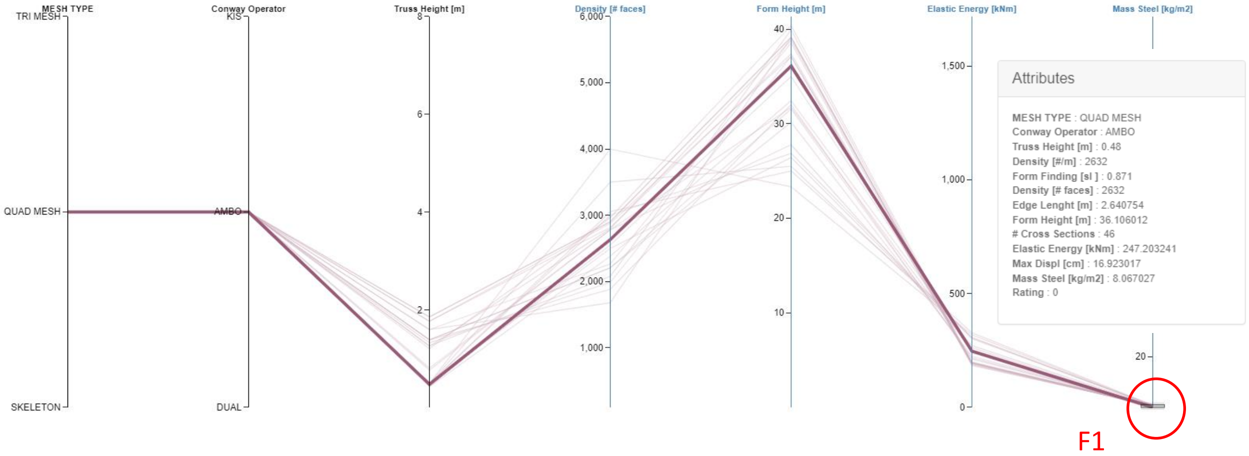


Figure 09.25
Parallel coordinate graph – 7 parameter axes

After omitting the heaviest results, the parallel coordinate graph view is thus narrowed to better asses the results. In *Figure 09.26*. The parallel coordinate graph state is shown, where we view just the results with the lowest amount of steel used.



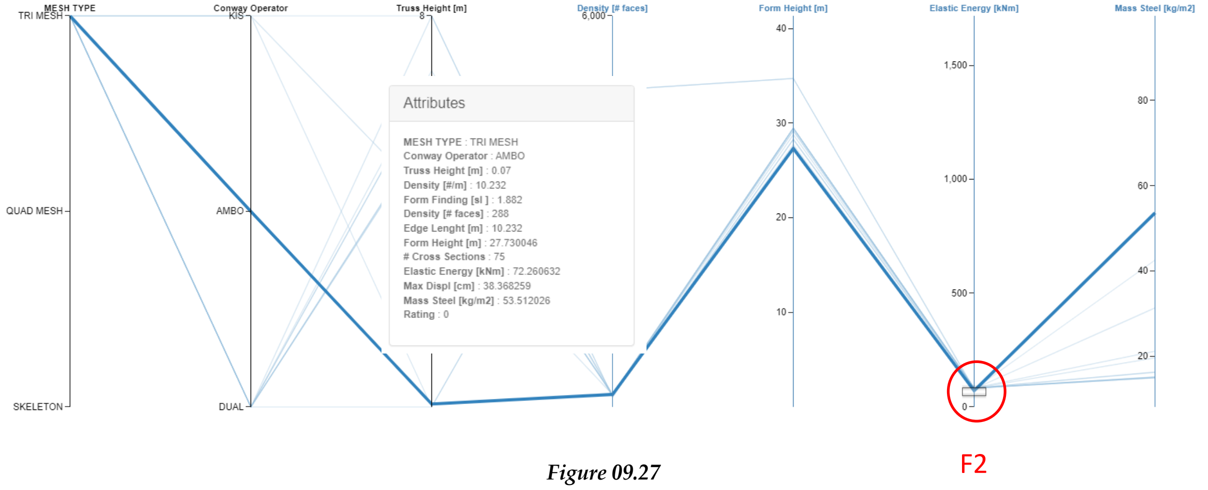
Attributes

MESH TYPE : QUAD MESH
Conway Operator : AMBO
Truss Height [m] : 0.48
Density [#/m] : 2632
Form Finding [sl] : 0.871
Density [# faces] : 2632
Edge Lenght [m] : 2.640754
Form Height [m] : 36.106012
Cross Sections : 46
Elastic Energy [kNm] : 247.203241
Max Displ [cm] : 16.923017
Mass Steel [kg/m2] : 8.067027
Rating : 0

Figure 09.26
Parallel coordinate graph – Cat. 1 – abs. – F1 – Q+A

Analysing the parallel coordinate graph in *Figure 09.26* the lowest amount of steel usage will be achieved by the combination of Quad mesh and ambo operator. Thus the "winner" in category 1 regarding **F1** (lowest steel usage) is **Q+A.** The highlighted line, accompanied with the attributes window, shows which exact space frame **Q+A** configuration achieves least mass.

To now view which combination in the absolute category wins in terms of highest stiffness, we reset the graph back to 09.25. and adjust so least elastic energy solutions are viewed only (*Figure 09.27*).



Attributes

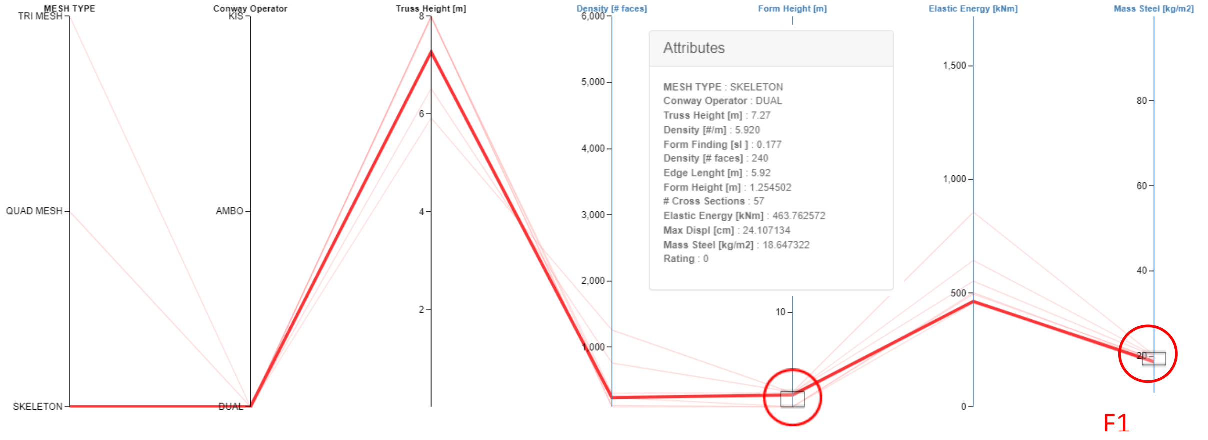
MESH TYPE : TRI MESH
Conway Operator : AMBO
Truss Height [m] : 0.07
Density [#/m] : 10.232
Form Finding [sl] : 1.882
Density [# faces] : 288
Edge Lenght [m] : 10.232
Form Height [m] : 27.730046
Cross Sections : 75
Elastic Energy [kNm] : 72.260632
Max Displ [cm] : 38.368259
Mass Steel [kg/m2] : 53.512026
Rating : 0

Figure 09.27
Parallel coordinate graph – Cat 1. – abs. – F2 – T+A

Analysing the parallel coordinate graph in *Figure 09.27* the lowest amount of elastic energy (highest stiffness) will be achieved by the combination of Tri mesh and ambo operator. Thus the "winner" in category 1 regarding **F2** (lowest elastic energy/highest stiffness) is **T+A.** The highlighted line, accompanied with the attributes window, shows which exact space frame **T+A** configuration achieves least elastic energy/highest stiffness.

Category 2: Relative, form finding height input parameter limited to 2% of span

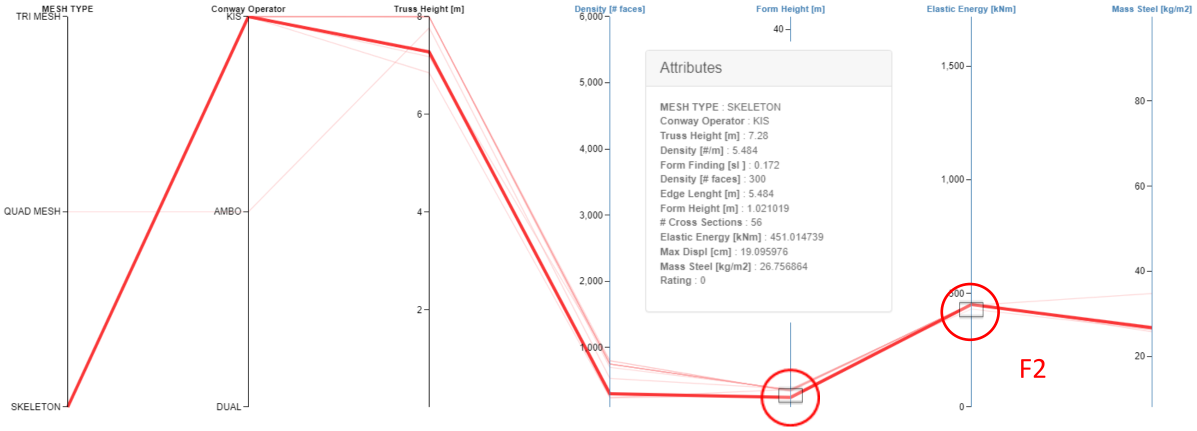
Analysing the parallel coordinate graph in *Figure 09.28* the lowest amount of steel mass will be achieved by the combination of Skeleton mesh and dual operator. Thus the "winner" in category 1 regarding **F1** (lowest steel mass) is **S+D.** The highlighted line, accompanied with the attributes window, shows which exact space frame **S+D** configuration achieves least steel mass.



Attributes

MESH TYPE : SKELETON
Conway Operator : DUAL
Truss Height [m] : 7.27
Density [#/m] : 5.920
Form Finding [sl] : 0.177
Density [# faces] : 240
Edge Lenght [m] : 5.92
Form Height [m] : 1.254502
Cross Sections : 57
Elastic Energy [kNm] : 463.762572
Max Displ [cm] : 24.107134
Mass Steel [kg/m2] : 18.647322
Rating : 0

Figure 09.28
Parallel coordinate graph – Cat 2 – rel. – F1 – S+D



Attributes

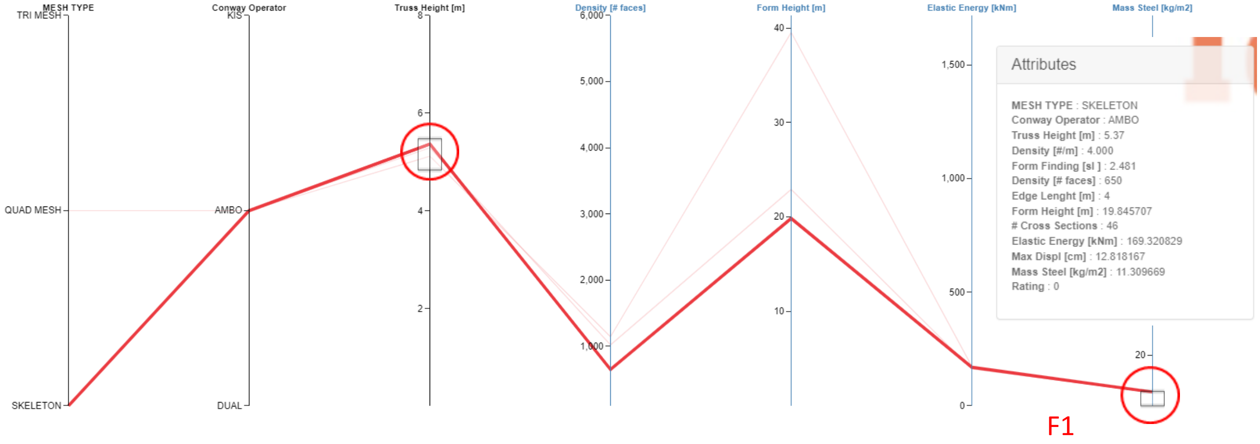
MESH TYPE : SKELETON
Conway Operator : KIS
Truss Height [m] : 7.28
Density [#/m] : 5.484
Form Finding [sl] : 0.172
Density [# faces] : 300
Edge Lenght [m] : 5.484
Form Height [m] : 1.021019
Cross Sections : 56
Elastic Energy [kNm] : 451.014739
Max Displ [cm] : 19.095976
Mass Steel [kg/m2] : 26.756864
Rating : 0

Figure 09.29
Parallel coordinate graph – Cat 2 – rel. – F2 – S+K

Analysing the parallel coordinate graph in Figure 09.29 the lowest amount of elastic energy (highest stiffness) will be achieved by the combination of Tri mesh and ambo operator. Thus the "winner" in category 1 regarding **F2** (lowest elastic energy/highest stiffness) is **S+K.** The highlighted line, accompanied with the attributes window, shows which exact space frame **S+K** configuration achieves least elastic energy/highest stiffness.

Category 3.: Relative, truss height range cca. 5 m

Analysing the parallel coordinate graph in *Figure 09.30* the lowest amount of steel mass will be achieved by the combination of Skeleton mesh and ambo operator. Thus the "winner" in category 1 regarding **F1** (lowest steel mass) is **S+A.** The highlighted line, accompanied with the attributes window, shows which exact space frame **S+A** configuration achieves least steel mass.

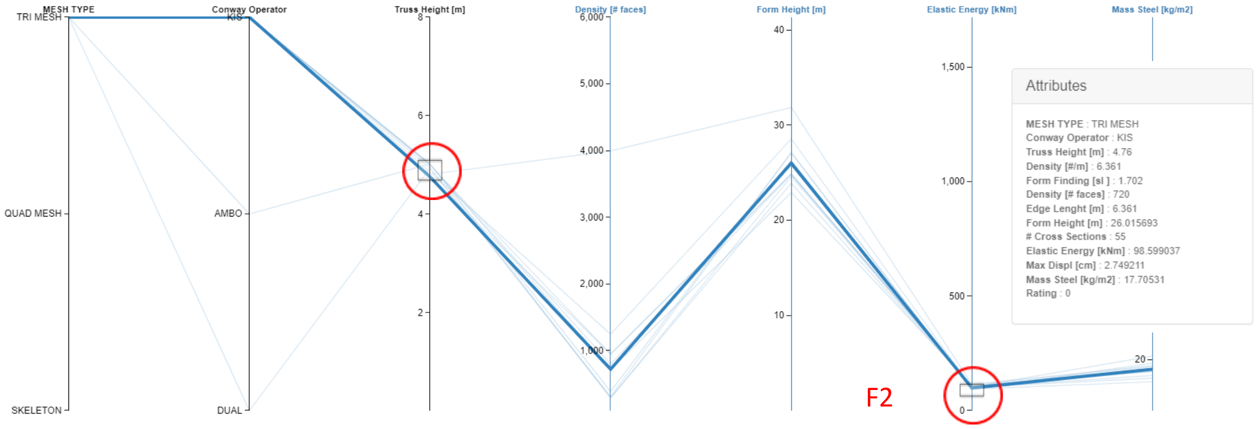


Attributes

MESH TYPE : SKELETON
Conway Operator : AMBO
Truss Height [m] : 5.37
Density [#/m] : 4.000
Form Finding [sl] : 2.481
Density [# faces] : 650
Edge Lenght [m] : 4
Form Height [m] : 19.845707
Cross Sections : 46
Elastic Energy [kNm] : 169.320829
Max Displ [cm] : 12.818167
Mass Steel [kg/m2] : 11.309669
Rating : 0

Figure 09.30
Parallel coordinate graph – Cat 3 – rel. – F1 – S+A

Analysing the parallel coordinate graph in *Figure 09.31* the lowest amount of elastic energy (highest stiffness) will be achieved by the combination of Tri mesh and kis operator. Thus the "winner" in category 1 regarding **F2** (lowest elastic energy/highest stiffness) is **T+K.** The highlighted line, accompanied with the attributes window, shows which exact space frame **T+K** configuration achieves least elastic energy/highest stiffness.



MESH TYPE : TRI MESH
Conway Operator : KIS
Truss Height [m] : 4.76
Density [#/m] : 6.361
Form Finding [sl] : 1.702
Density [# faces] : 720
Edge Lenght [m] : 6.361
Form Height [m] : 26.015693
Cross Sections : 55
Elastic Energy [kNm] : 98.599037
Max Displ [cm] : 2.749211
Mass Steel [kg/m2] : 17.70531
Rating : 0

Figure 09.31
Parallel coordinate graph – Cat 3 – rel. – F2 – T+K

The summarized results are now presented again in respect to categories:

1.) Category 1 :Absolute – full parameter ranges of all five input parameters
    - Least mass (F1) – Q+A
    - Highest stiffness (F2) – T+A
2.) Category 2 : Relative– form finding height parameter limited to 2% of span
    - Least mass (F1) – S+D
    - Highest stiffness (F2) – S+K
3.) Category 3 : Relative – truss height parameter range limited from cca. 5 m
    - Least mass (F1) – S+A
    - Highest stiffness (F2) – T+K

Further analysis of results in relation to form finding height (FF)

After, analysing the broad spectrum of brute force results, it was clear that the nine possible space frame configurations are by far mostly influenced by form finding height. Thus, the influence of form finding (FF) on F1 (steel usage) and F2 (stiffness) regarding the 3 possible mesh types and 3 Conway operators, thus nine possible configurations, is further investigated in detail, considering a more narrowed parameter scope. The parameter ranges are further narrowed, to obtain relevant results. Mesh density is from 500 to 1000 faces, truss height from 4-6 m. The categories of solutions can be constituted as:

1.) Least amount of mass (F1) for the nine possible combinations of mesh and operator (Q/T/S + D/A/K)
2.) Least amount of elastic energy (F2) for the nine possible combinations of mesh and operator (Q/T/S + D/A/K)

1.) F1: Q/T/S + D/A/K

F1: Q+D/A/K

The data in the .csv file is adjusted according to the narrowed parameter ranges, and for each mesh type three scatter plot graphs are generated and the points curve fitted to form trend lines. One for each combination of Conway operator possible for the mesh type. The graphs are divided into the two above categories, they show the mass of steel vs. the form height. It can be seen from these graphs that there is an influence of form height on the steel usage. Especially when the form height is between 1-5 m.

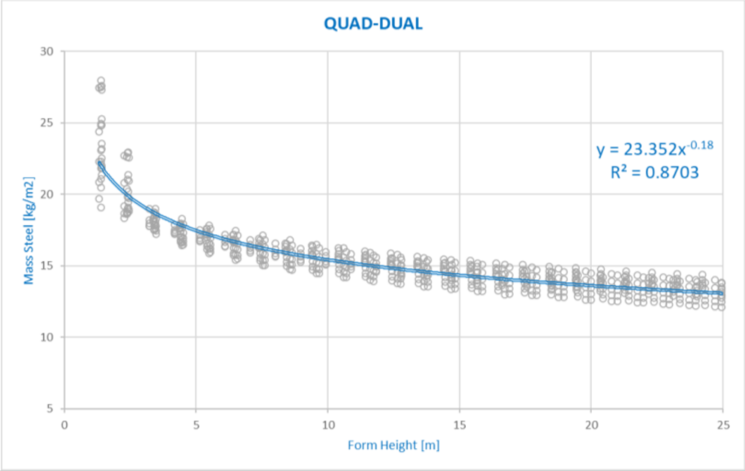


Figure 09.32
F1 vs. FF height – Q+D

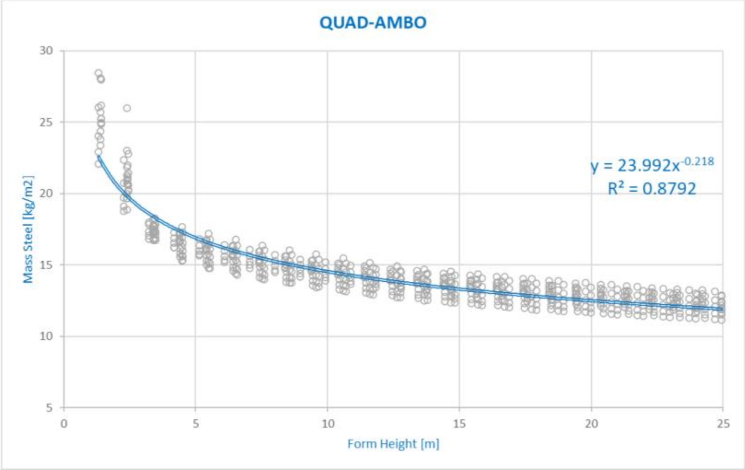


Figure 09.33
F1 vs. FF height – Q+A

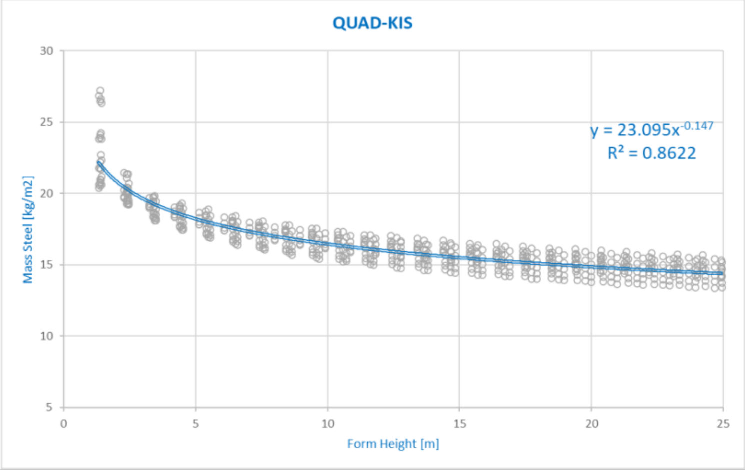


Figure 09.34
F1 vs. FF height – Q+K

F1: T+D/A/K

Like least mass results (F1) for Quad + dual/ambo/ kis, the same influence of form finding, within 1 – 5 m of form finding height is found.

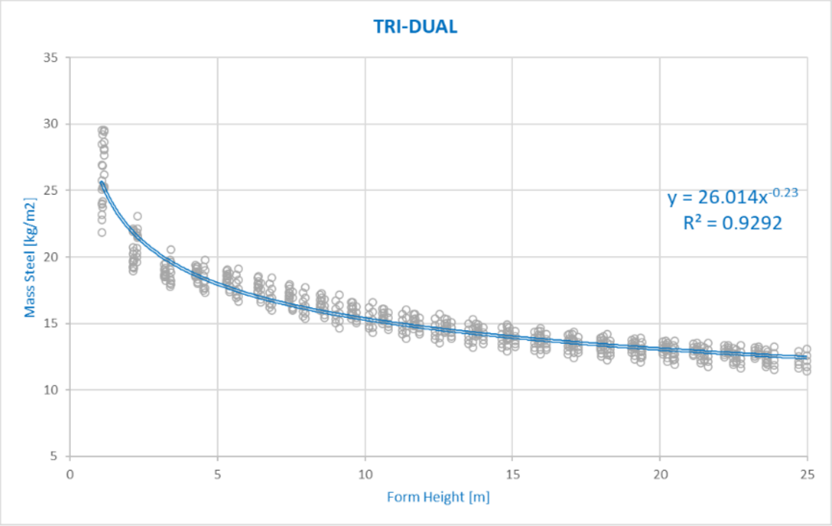


Figure 09.35
F1 vs. FF height – T+D

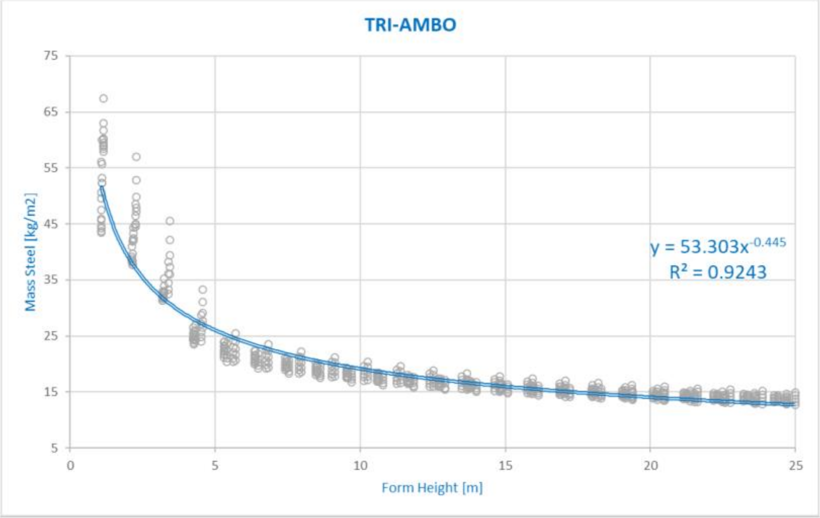


Figure 09.36
F1 vs. FF height – T+A

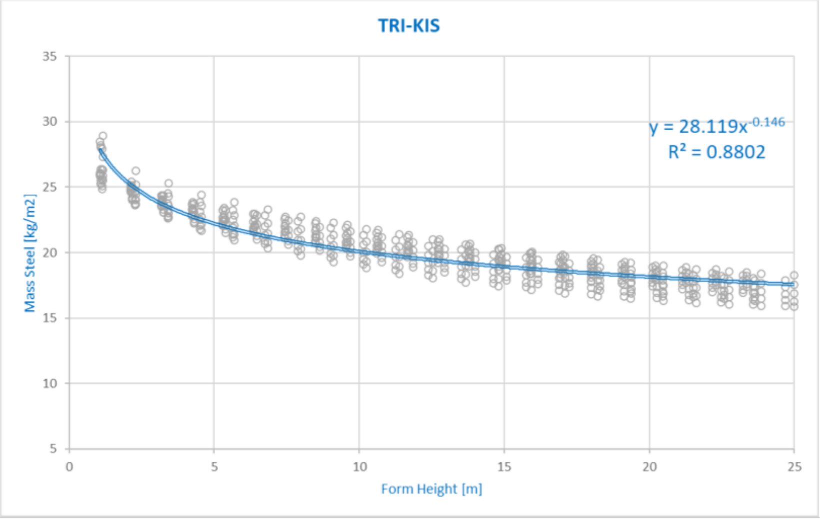


Figure 09.37
F1 vs. FF height – T+K

F1: S+D/A/K

Like least mass results (F1) for Quad and Tri + dual/ambo/kis, the same highest influence of form finding is found, within 1 – 5 m of height.

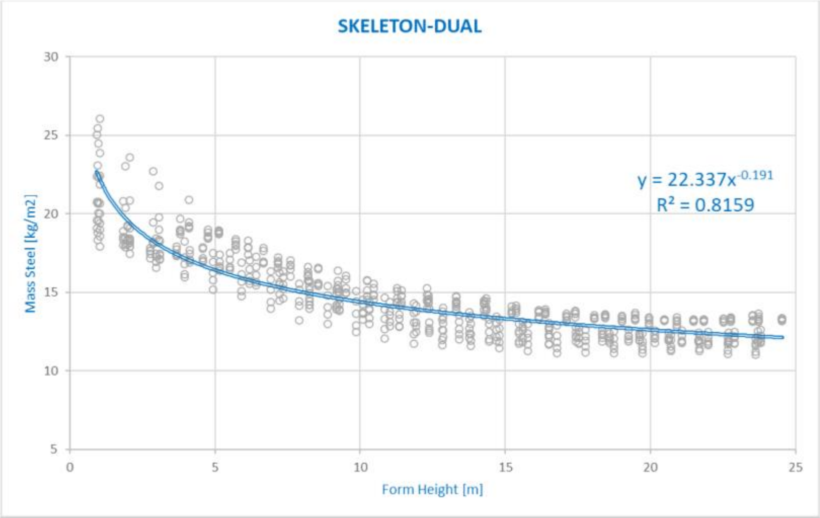


Figure 09.38
F1 vs. FF height – S+D

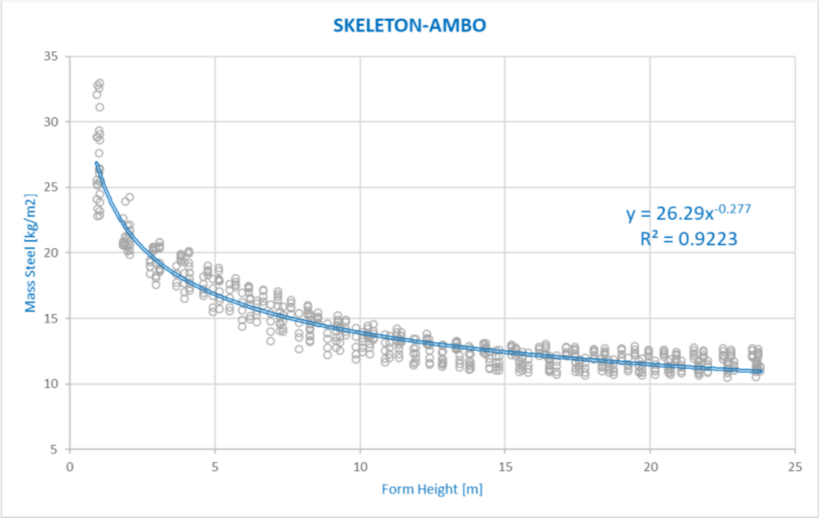


Figure 09.39
F1 vs. FF height – S + A

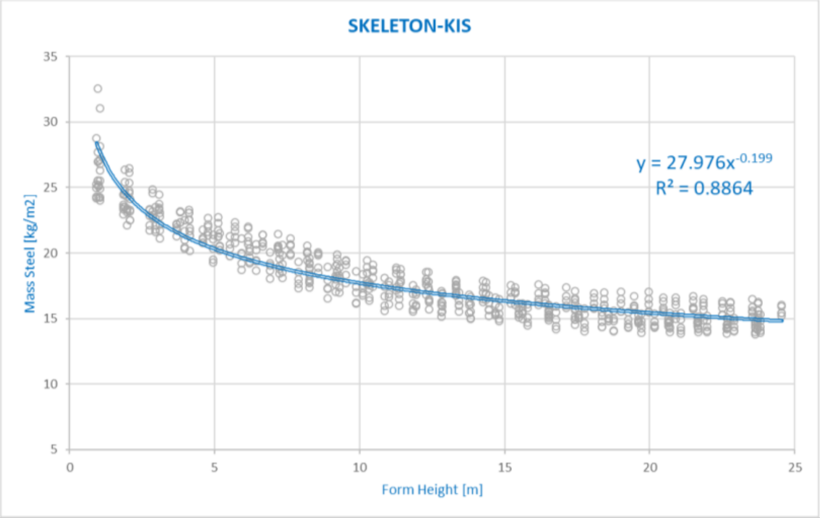


Figure 09.40
F1 vs. FF height – S+K

Cumulative graph for F1

In the cumulative graph (Figure 09.44) each line represents one of the nine diagrams presented in the last paragraph. They are colour coded in to three groups, quad group = blue, tri group = red, skeleton group = white. The line type indicates the Conway operator applied. Thus, the explanation of lines and colour codes is as follows:

Q+D (*Figure 09.32)* = Solid blue line in *Figure 09.41*
Q+A (*Figure 09.33*) = Dashed blue line in *Figure 09.41*
Q+K (*Figure 09.34*) = Dash-double dot blue line in *Figure 09.41*

T+D (*Figure 09.35*) = Solid red line in *Figure 09.41*
T+A (*Figure 09.36*) = Dashed red line in *Figure 09.41*
T+K (*Figure 09.37*) = Dash-double dot red line in *Figure 09.41*

S+D (*Figure 09.38*) = Solid white line in *Figure 09.41*
S+A (*Figure 09.39*) = Dashed white line in *Figure 09.41*
S+K (*Figure 09.40*) = Dash–double dot white line in *Figure 09.41*

Cumulative graph interpretation for F1

Analysing the cumulative graph (*Figure 09.41*), from a broad perspective it is instantly noticeable that the choice of space frame configuration is highly influential on structural mass (F1) in the first form finding height parameter value ranges of 1-5 m. The spread between the graphs shows how influential the space frame configuration choice is on structural mass. Accordingly, one can choose from the graph an appropriate space frame configuration in relation to a desired form finding height, and at different parameter ranges, different configurations have least structural mass. Furthermore, in the viewed range of 1-5 m considering least mass (F1), there are many crossing points between space frame configuration result graphs, meaning there are points where two configurations have equal masses for a certain form finding height. Viewing the graph, there are clear differences between the following configurations in the range of 3-5 m form height; T+A (dashed red line), T+K (red dash-double dot line), S+K (white dash-double dot line), while for T+D (solid red line), Q+K (blue dash-double dot line), Q+D (solid blue line), S+A (dashed white line), Q+A (dashed blue line) the steel mass per configuration differs in about +/-2 kg/m^2 . Nonetheless, at closer look, out of each of the nine graphs representing each possible space frame configuration, two extremes become apparent. Worst and best performing combination within a viewed parameter range, thus allowing ranking of solutions regarding structural mass (F1). Considering a form finding height of 3 – 5 m in the graph (where configuration choice is most noticeable), the rankings are as follows, from best to worst:

1.) S+D
2.) Q+A
3.) Q+D
4.) Q+K
5.) T+D,
6.) T+K
7.) T+A.
8.) Q+A
9.) T+A

Thus, the most efficient space frame configuration regarding least amount of mass optimality criteria (F1) in the viewed range of 3-5 m form finding height would be Skeleton dual and least efficient Tri ambo. The rankings in between best and worst solutions are not absolute, due to the overlaps between each configuration graph, and closeness of them, meaning some could switch places depending on exact form finding height viewed between 3-5 m, however, the extremes in the viewed range are shown clearly.

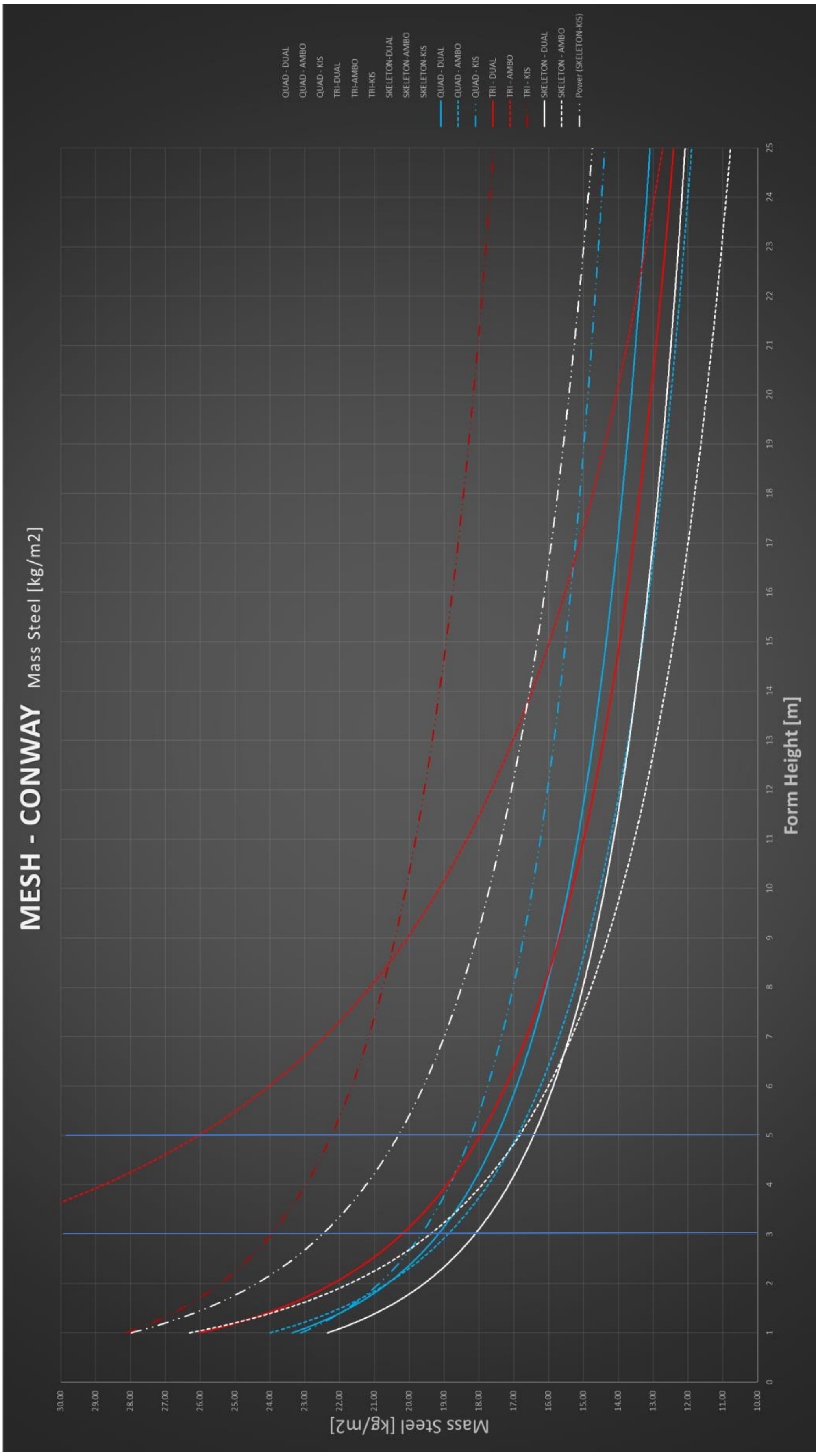


Figure 09.41
Cumulative graph of F1 vs FF height

2.) F2: Q/T/S + D/A/˙K

F2: Q+D/A/K

In this step the data is adjusted to create scatter plot graphs and trend lines regarding the combination of mesh and operator on F2, highest structural stiffness goal. Not the inverted graph, due to the inverted relationship between mass and stiffness.

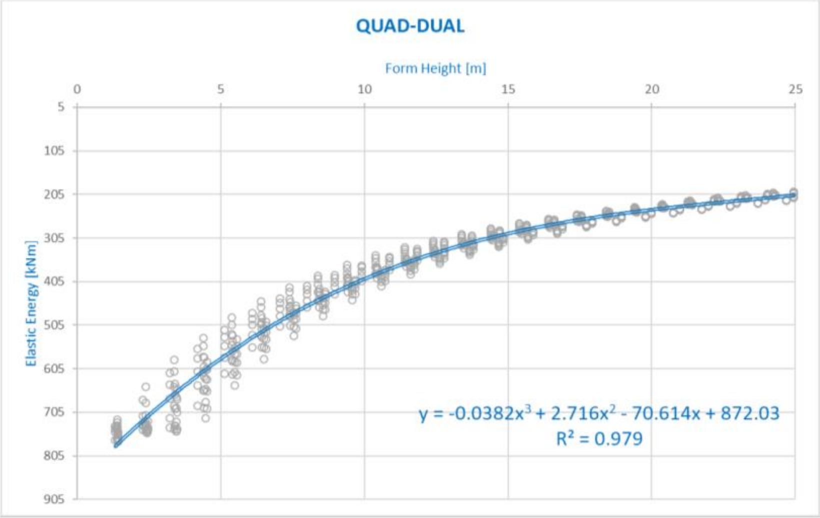


Figure 09.42
F2 vs. FF height – Q+D

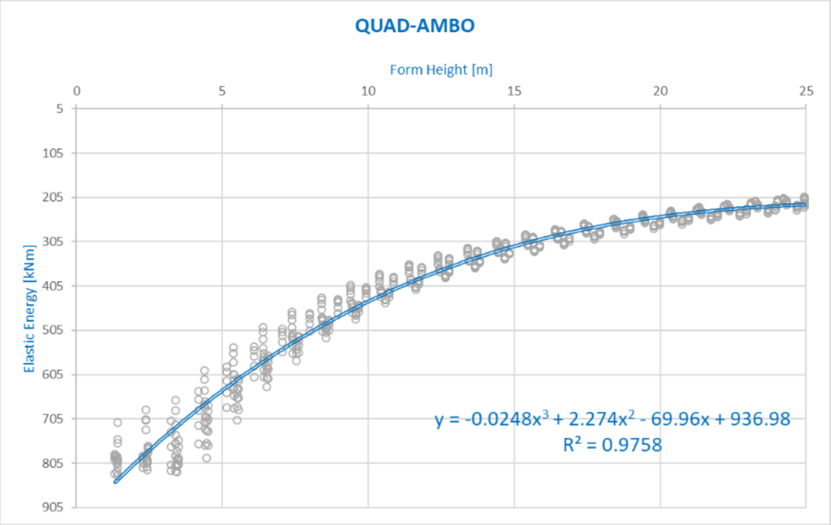


Figure 09.43
F2 vs. FF height – Q+A

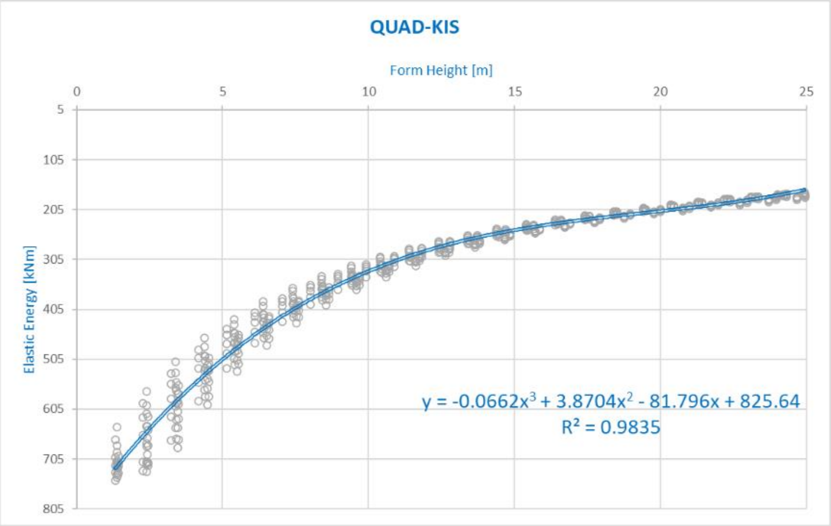


Figure 09.44
F2 vs. FF height – Q+K

F2: T+D/A/K

Like results for Quad – F1 the same influence of form finding, within 1 – 5 m of height (5-10% of span) is found.

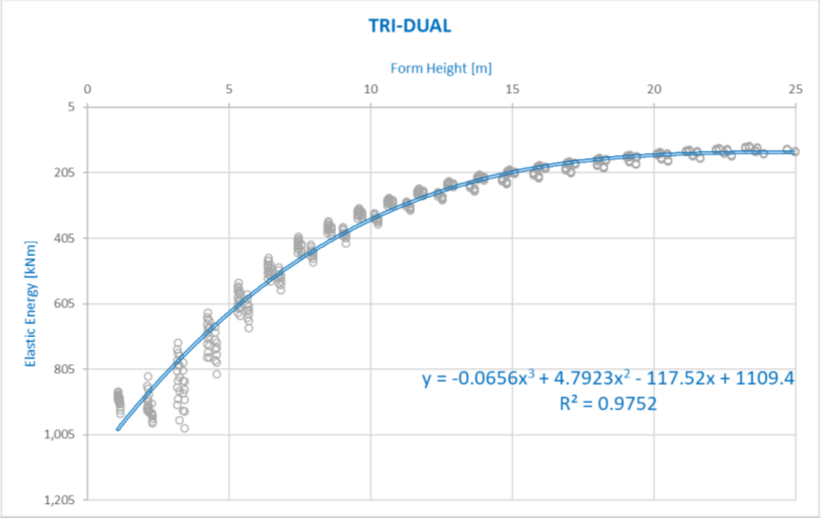


Figure 09.45
F2 vs. FF height – T+D

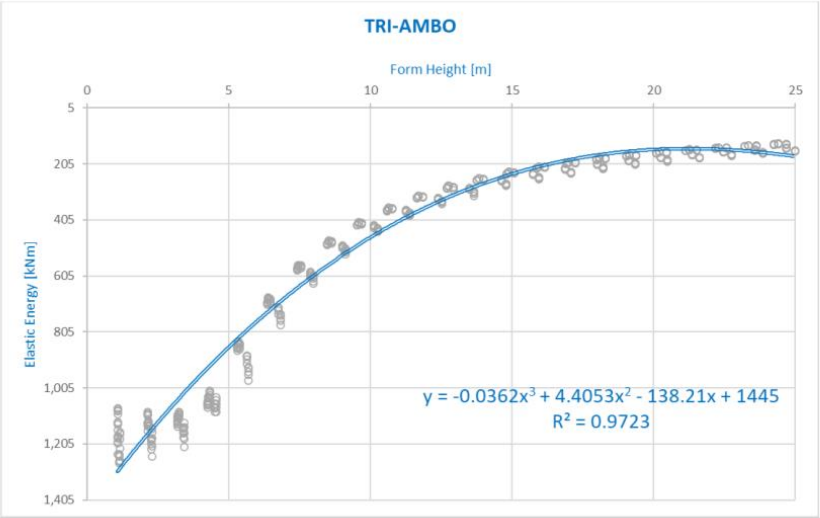


Figure 09.46
F2 vs. FF height – T+A

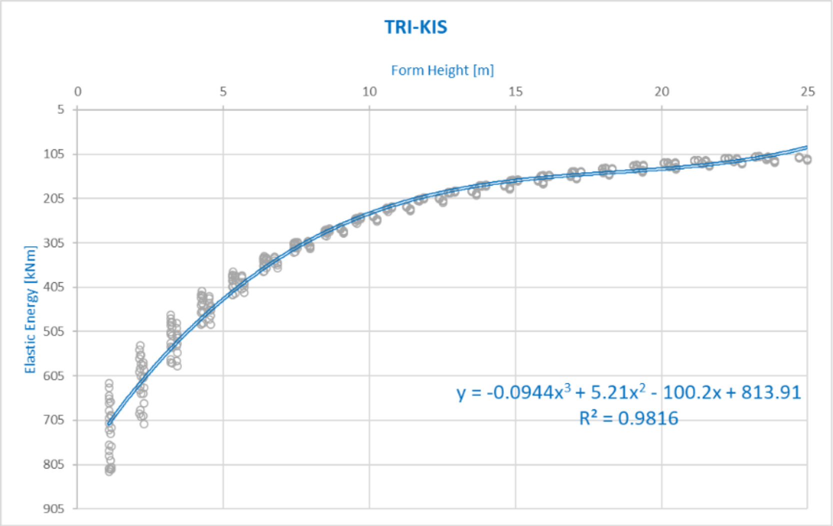


Figure 09.47
F2 vs. FF height – T+K

F2: S+D/A/K

Like results for Quad and Tri – F1 the same highest influence of form finding is found, within 1 – 5 m of height (5-10% of span).

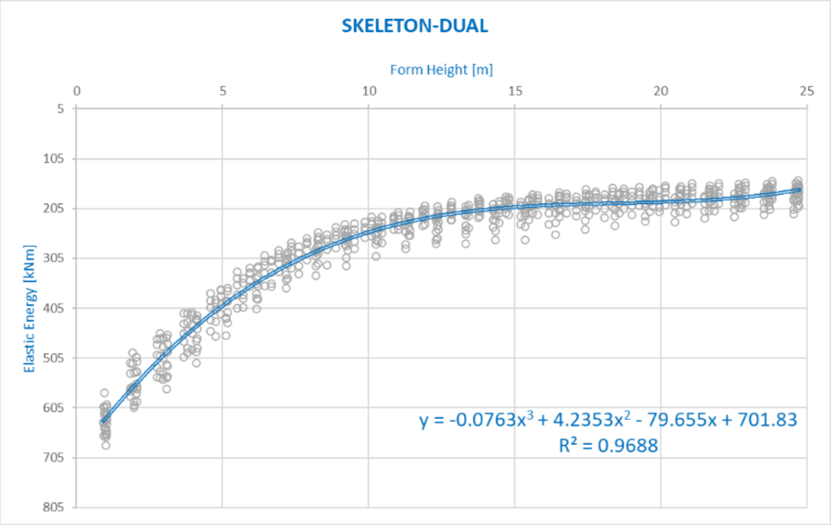


Figure 09.48
F2 vs. FF height – S+D

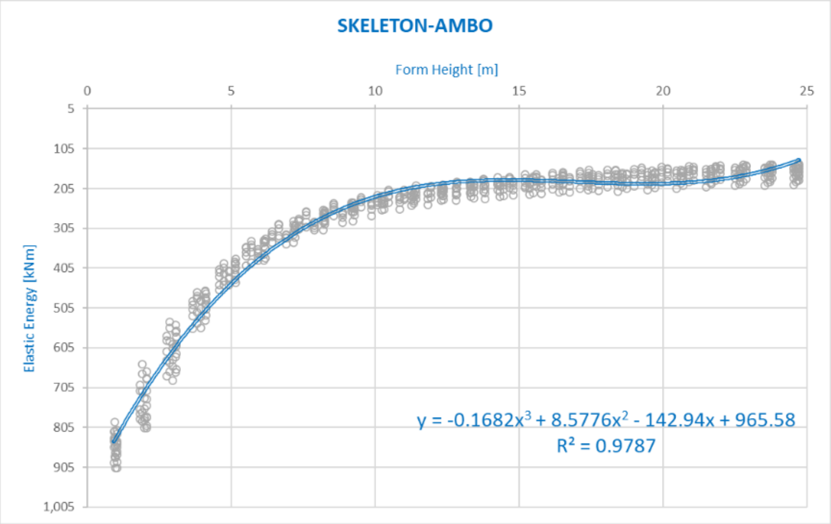


Figure 09.49
F2 vs. FF height – S + A

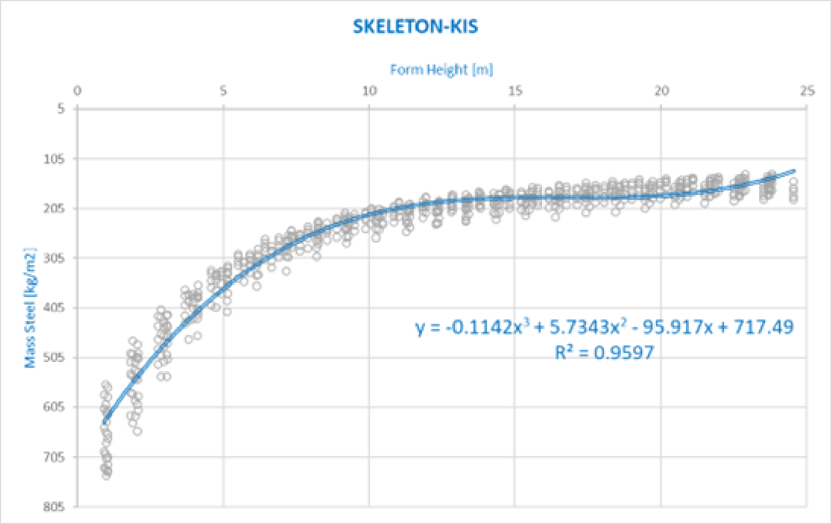


Figure 09.50
F2 vs. FF height – S+K

1.1. Cumulative graph for F2

In the cumulative graph for F2 (*Figure 09.51*) each line represents one of the nine diagrams presented in the last paragraph. They are colour coded again in to three groups, quad group = blue, tri group = red, skeleton group = white. The line type indicates the Conway operator applied. Thus, the explanation of lines and colour codes is as follows:

Q+D (*Figure 09.42*) = Solid blue line in *Figure 09.51*
Q+A (*Figure 09.43*) = Dashed blue line in *Figure 09.51*
Q+K (*Figure 09.44*) = Dash-dot double blue line in *Figure 09.51*

T+D (*Figure 09.45*) = Solid red line in *Figure 09.51*
T+A (*Figure 09.46*) = Dashed red line in *Figure 09.51*
T+K (*Figure 09.47*) = Dash-double dot red line in *Figure 09.51*

S+D (*Figure 09.48*) = Solid white line in *Figure 09.51*
S+A (*Figure 09.49*) = Dashed white line in *Figure 09.51*
S+K (*Figure 09.50*) = Dash-dot double white line in *Figure 09.51*

Cumulative graph interpretation for F2

Analysing the cumulative graph (*Figure 09.51*), from a broad perspective it is instantly noticeable that the choice of space frame configuration is highly influential on elastic energy (stiffness) in the first form finding height parameter value ranges of 1-5m. The spread between the graphs is the largest, while at around 16 m height this spread flattens out and becomes less obvious meaning less difference in terms of structural stiffness between different space frame configurations. Naturally at points where the graphs cross one another, there is no difference between the two configurations.

At closer look, out of each of the nine graphs representing each possible space frame configuration, two extremes become apparent. Worst and best performing combination within a viewed parameter range, thus allowing ranking of solutions regarding stiffness (F2). Considering a form finding height of 3 – 5 m in the graph (where configuration choice is most noticeable), the rankings are as follows, from best to worst:

1.) S+K
2.) S+D
3.) T+K
4.) S+A
5.) Q+K
6.) Q+D
7.) T+D.
8.) Q+A
9.) T+A

Thus, the most efficient space frame configuration regarding highest stiffness optimality criteria (F2 – least elastic energy) in the viewed range of 3-5 m form finding height would be Skeleton kis (S+K) and least efficient Tri ambo (T+A).

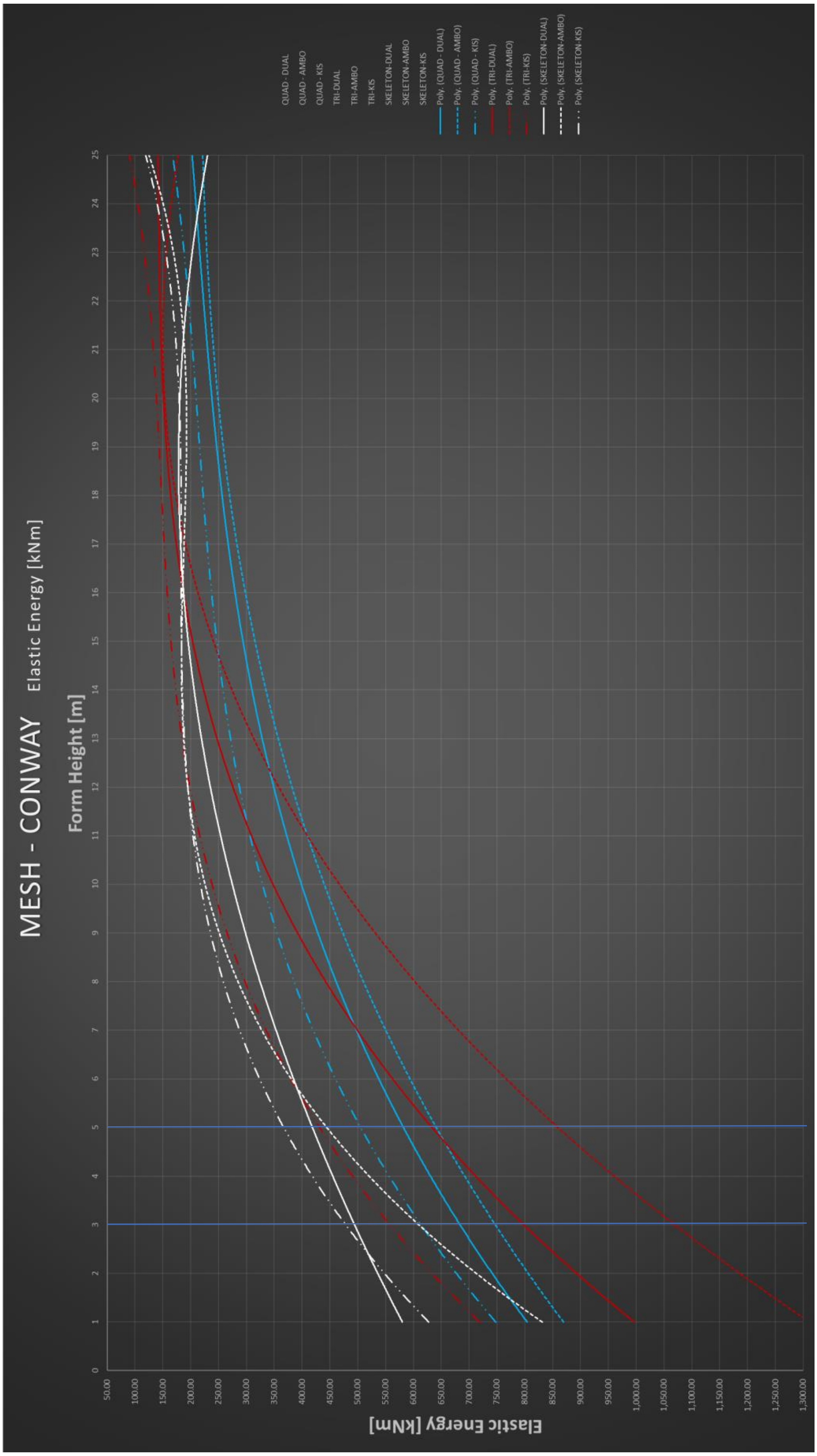


Figure 09.51

Cumulative graph of F2 vs FF heigh

Further analysis of Brute force results – Parallel coordinate graphs

By analysing the cumulative graphs in *figures 09.41* and *09.51*, which were generated by narrowing the initial broad brute force results towards more realistic parameter ranges, it was established that form finding height has an important influence on space frame configurations regarding least mass (F1) and least elastic energy (F2). Thus, trend established, further quantitative analysis of brute force results in relation to F1 is analysed in three different form finding height parameter ranges. The parameter ranges are categorized as low (cca. – 2m form finding height), medium (cca. 10 m form finding height) and high (cca. 25 m form finding height). The formed categories facilitate further understanding of trends relating to form finding height influence on optimal solution. The following 27 parallel coordinate graphs in *figures 09.52-09.78,* show for each respective space frame configuration (Base mesh + operator pair) optimal solutions considering the fitness goal of F1 (least structural mass) according to posed high, medium, and low form finding height categories. Highlighted lines in the graphs represent the solution which achieved minimal mass and show accompanying input parameter values which achieve this solution. The end goal of this part of result analysis is to rank from best to worst specific space frame configurations regarding F1, least structural mass.

Quad – D/A/K – FF-25 m

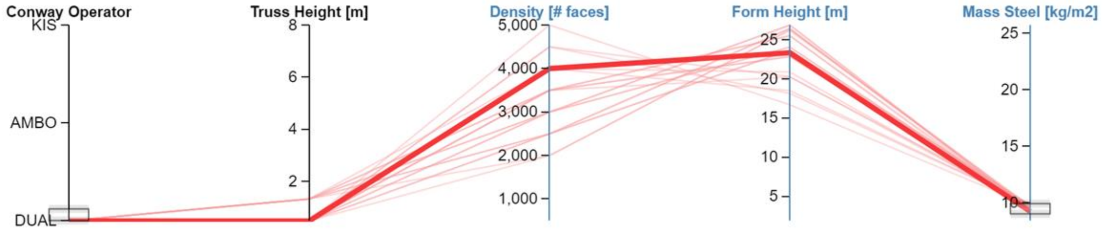


Figure 09.52 *– Q+D – brute force results – FF 25 m - F1(least amount of steel)*

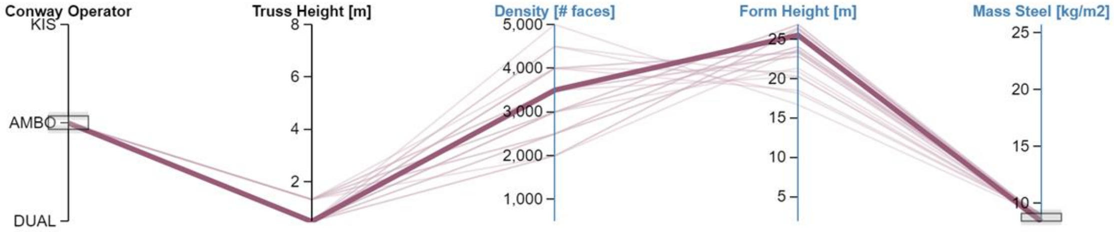


Figure 09.53 *– Q+A – brute force results – FF 25 m - F1*

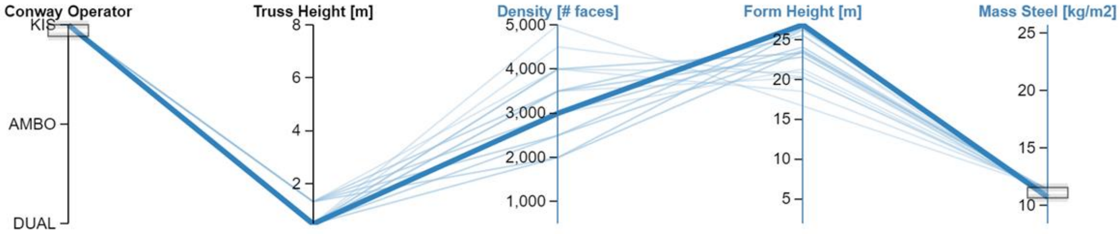


Figure 09.54 *– Q+K – brute force results – FF 25 m - F1*

Tri– D/A/K – FF-25 m

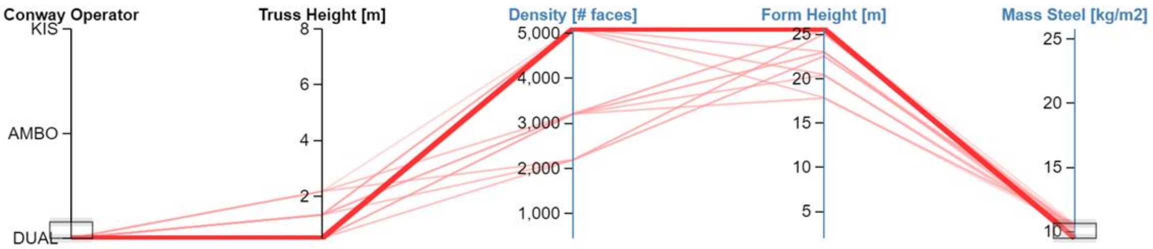


Figure 09.55 – T+D – brute force results – FF 25 m - F1

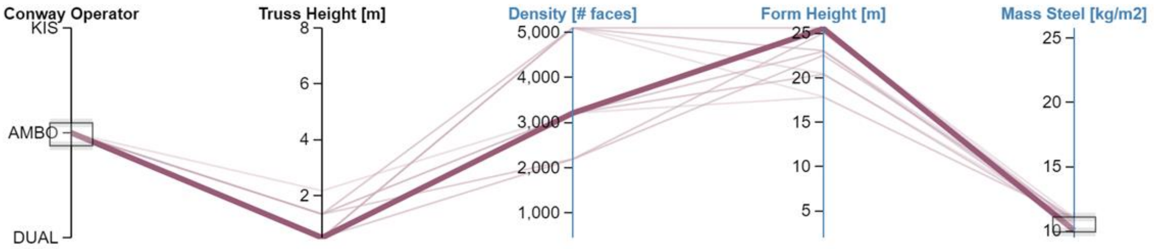


Figure 09.56 – T+A – brute force results – FF 25 m - F1

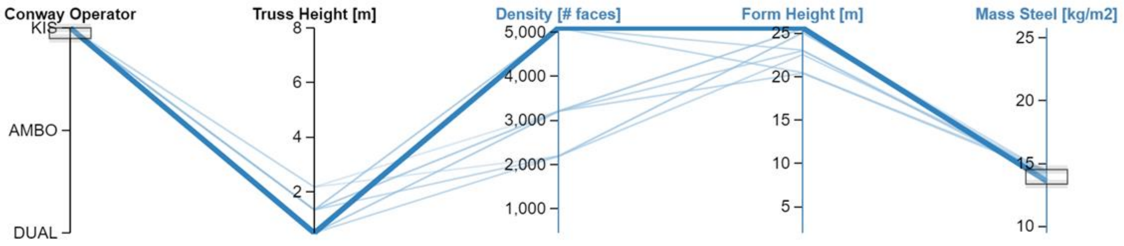


Figure 09.57 – T+K – brute force results – FF 25 m - F1

Skeleton– D/A/K – FF-25 m

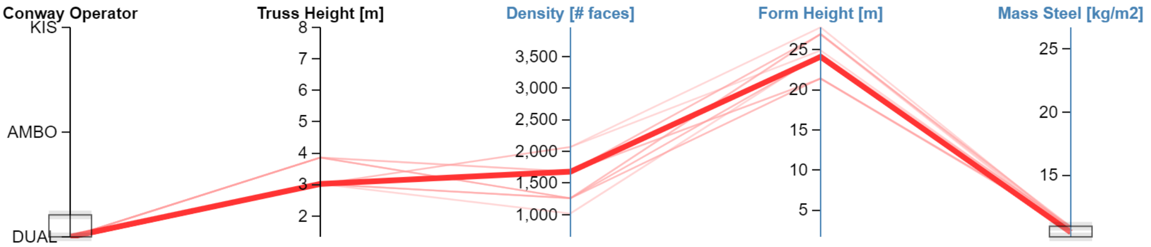


Figure 09.58 – S+D – brute force results – FF 25 m - F1

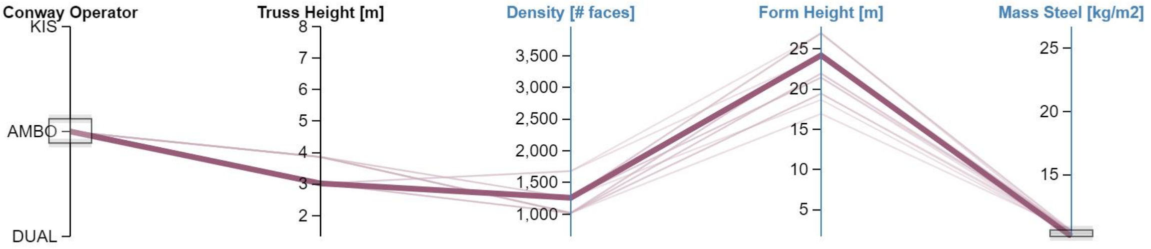


Figure 09.59 – S+A – brute force results – FF 25 m - F1

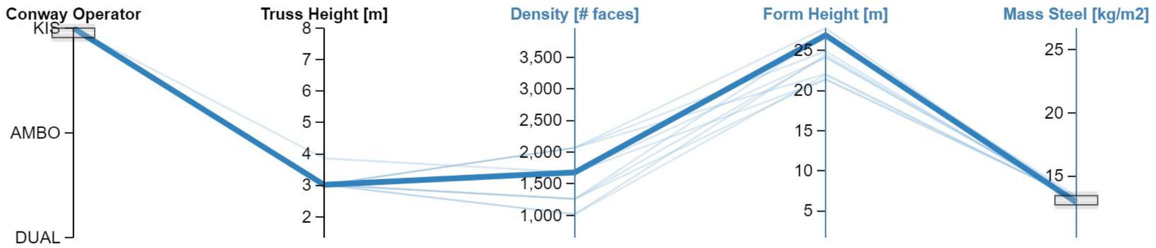


Figure 09.60 – *S+K – brute force results – FF 25 m - F1*

Quad– D/A/K – FF-10 m

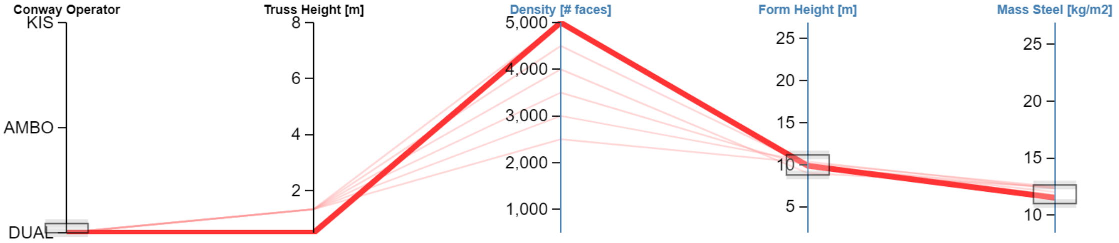


Figure 09.61 – *Q+D – brute force results – FF 10 m - F1*

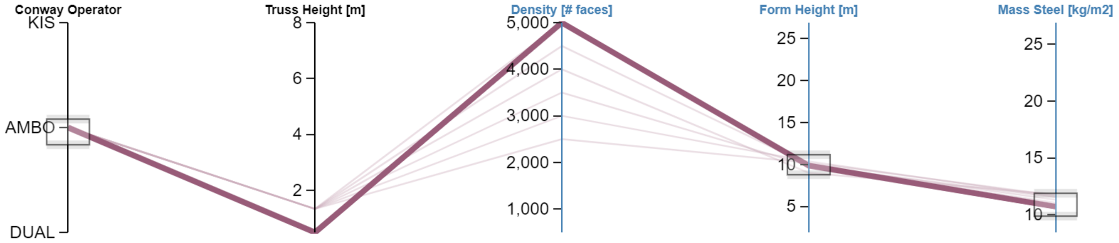


Figure 09.62 – *Q+A – brute force results – FF 10 m - F1*

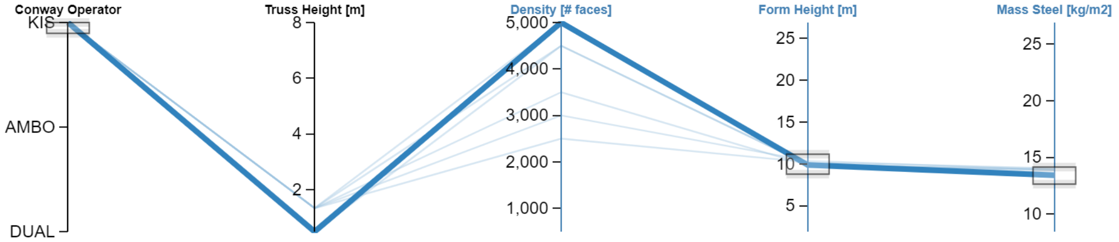


Figure 09.63 – *Q+K – brute force results – FF 10 m - F1*

Tri– D/A/K – FF-10 m

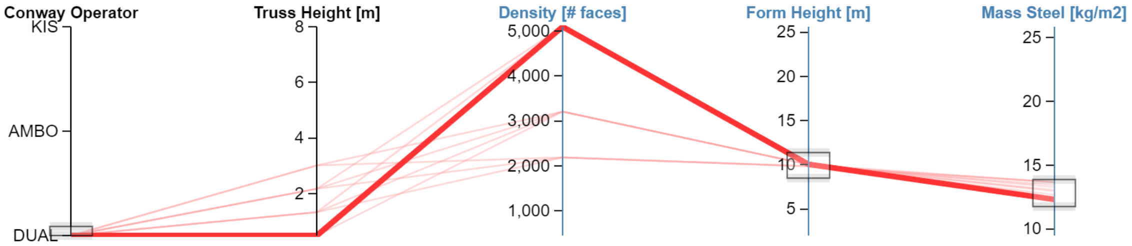


Figure 09.64 – *T+D – brute force results – FF 10 m - F1*

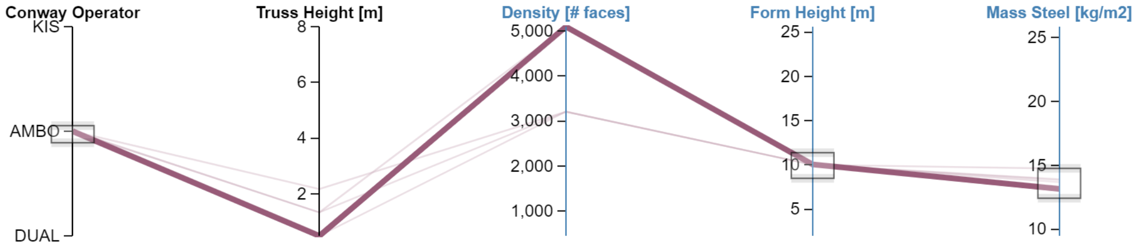


Figure 09.68 – T+A – brute force results – FF 10 m - F1

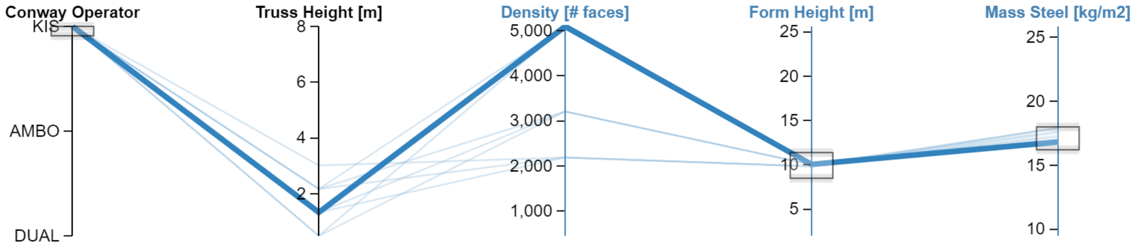


Figure 09.66 – T+K – brute force results – FF 10 m - F1

Skeleton– D/A/K – FF-10%-F1

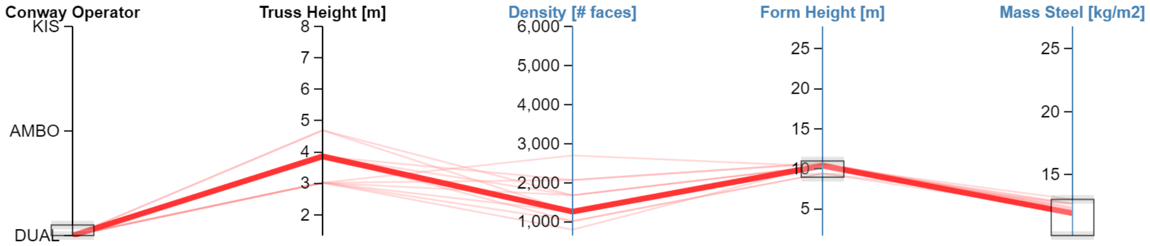


Figure 09.67 – S+D – brute force results – FF 10 m - F1

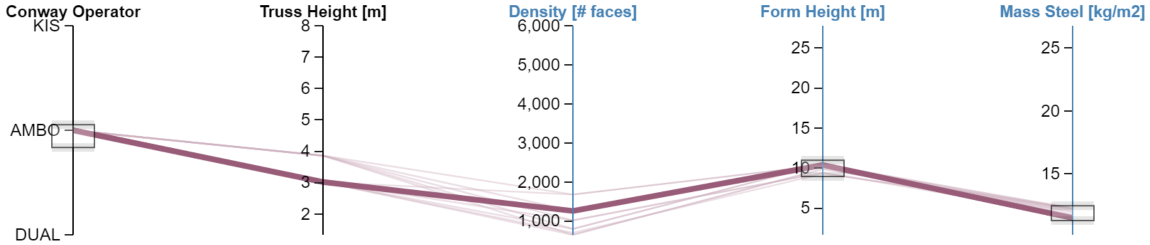


Figure 09.68 – S+A – brute force results – FF 10 m - F1

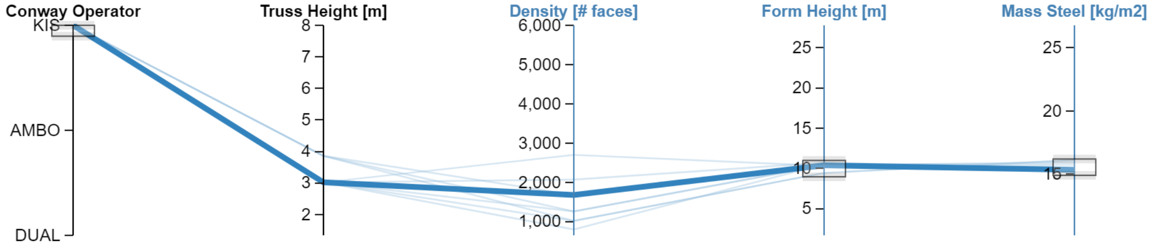


Figure 09.69 – S+K – brute force results – FF 10 m - F1

Quad– D/A/K – FF-2 m

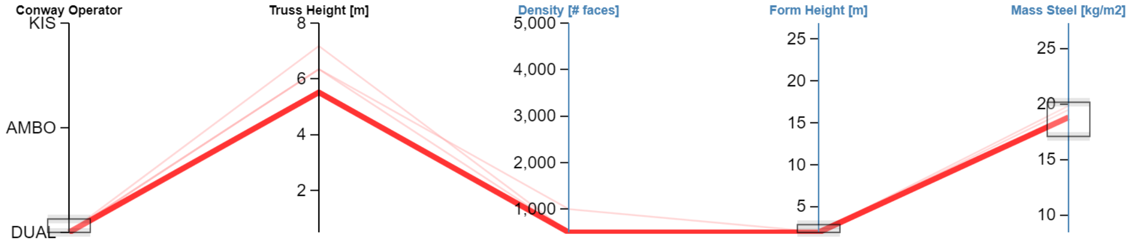


Figure 09.70 – *Q+D – brute force results – FF 2 m - F1*

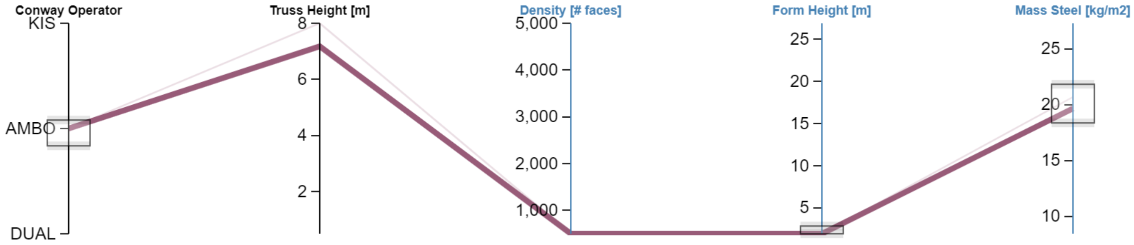


Figure 09.71 – *Q+A – brute force results – FF 2 m - F1*

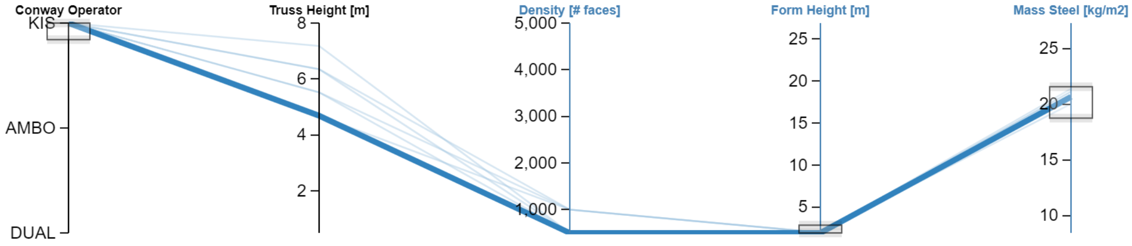


Figure 09.72 – *Q+K – brute force results – FF 2 m - F1*

Tri– D/A/K – FF-2 m

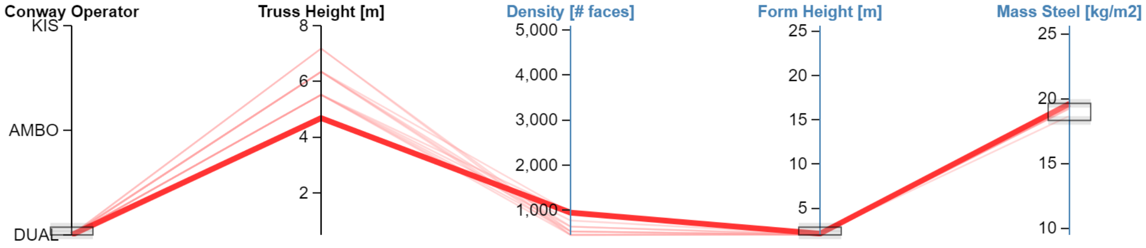


Figure 09.73 – *T+D – brute force results – FF 2 m - F1*

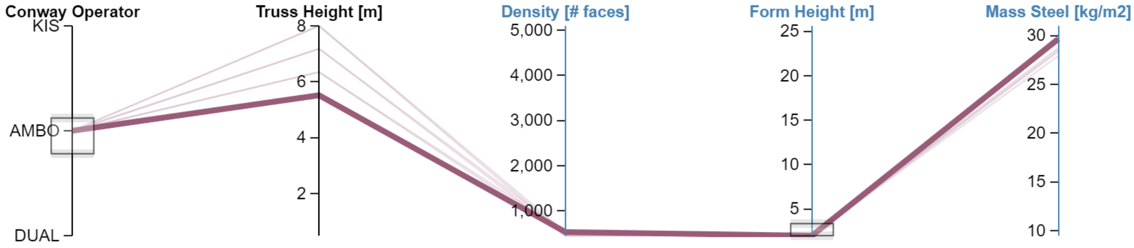


Figure 09.74 – *T+A – brute force results – FF 2 m - F1*

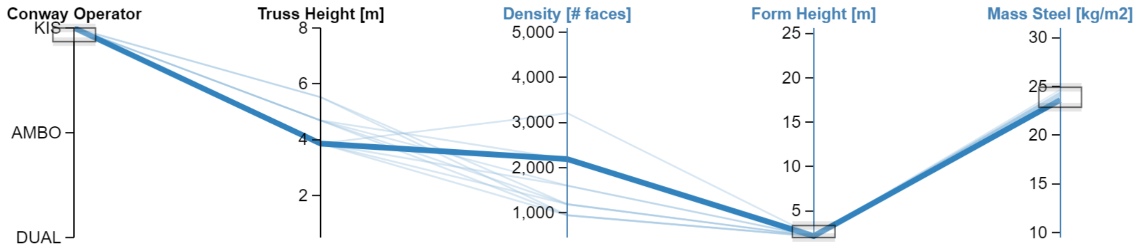


Figure 09.758 – T+K – brute force results – FF 2 m - F1

Skeleton– D/A/K – FF-2%-F1

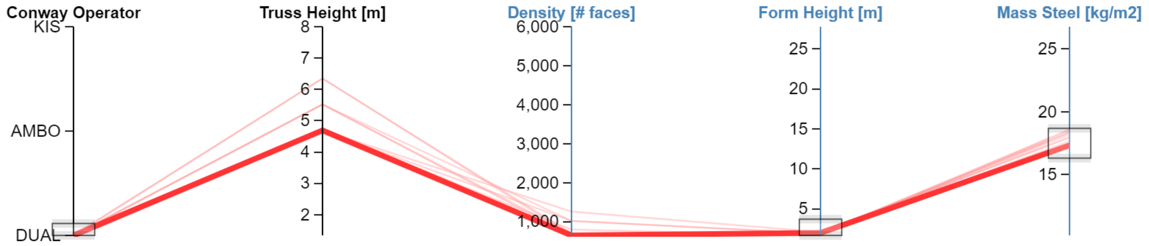


Figure 09.76 – S+D – brute force results – FF 2 m - F1

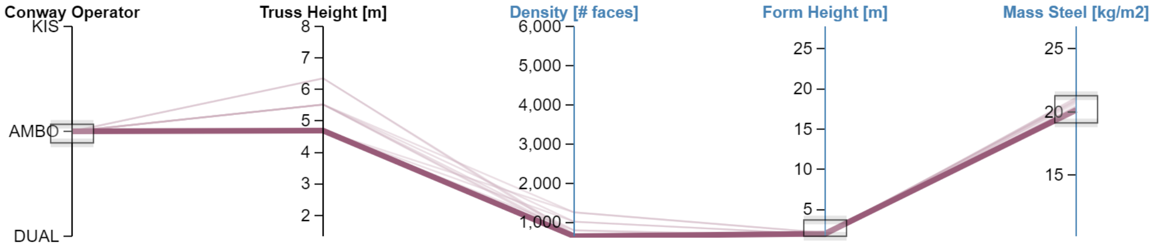


Figure 09.77 – S+A – brute force results – FF 2 m - F1

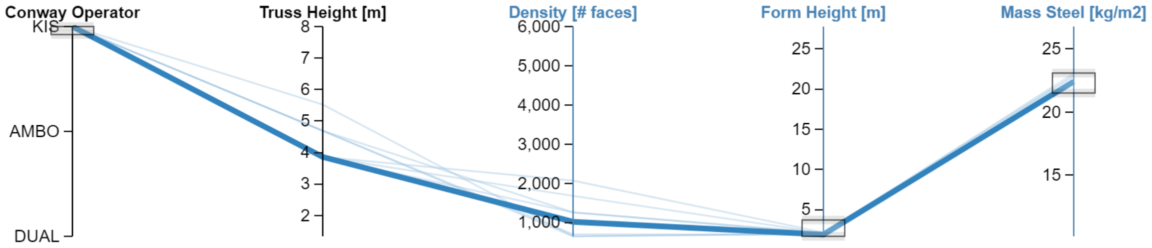


Figure 09.78 – S+K – brute force results – FF 2 m - F1

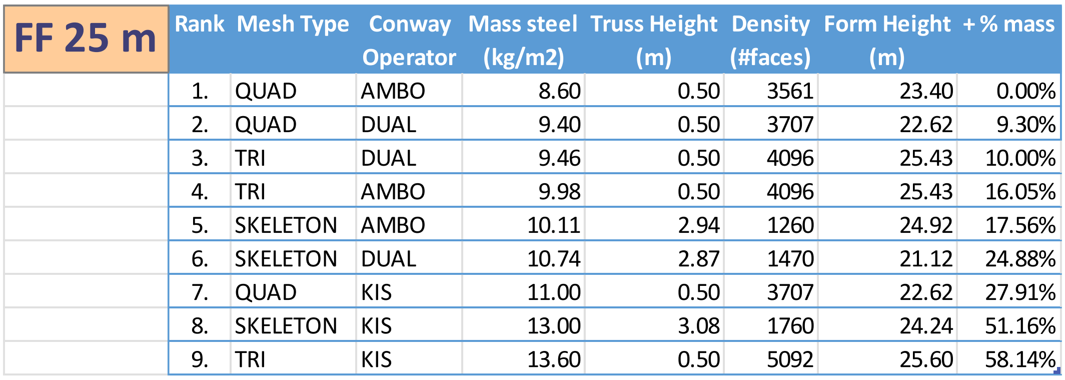
Metaheuristic optimization results

The results showed in previous graphs, *figures 09.52-09.78*., have also been checked by metaheuristic optimization, which always results in just one specific space frame configuration considered as optimal. The metaheuristic optimization was facilitated by the Galapagos native grasshopper plugin.

Metaheuristic optimization was carried out for each of the 9 possible space frame configurations considering three categories of approximate form finding height values of: 25 m, 10 m, and 2 m. The variable parameters for the optimization were truss height, mesh density while form finding height is fixed in the three mentioned categories. This form finding height limits stem from analysing the cumulative graphs (09.44 and 09.54). Increasing form finding height above 25 m of span limit has no benefits at all in relation to lowering structural mass, and the shape of the structure becomes so exaggerated that it may be deemed unfeasible or irrelevant for the design problem. Above 25 m the space frame starts to resemble more a large span arched structure than a double curved space frame structure.

The other limits of 10 m and 2 m are introduced because they represent further interesting areas of form finding height in mentioned graphs. The idea is to see how a high (25 m), medium (10 m) and low (2 m) form finding height influences the structural mass and variable parameters.

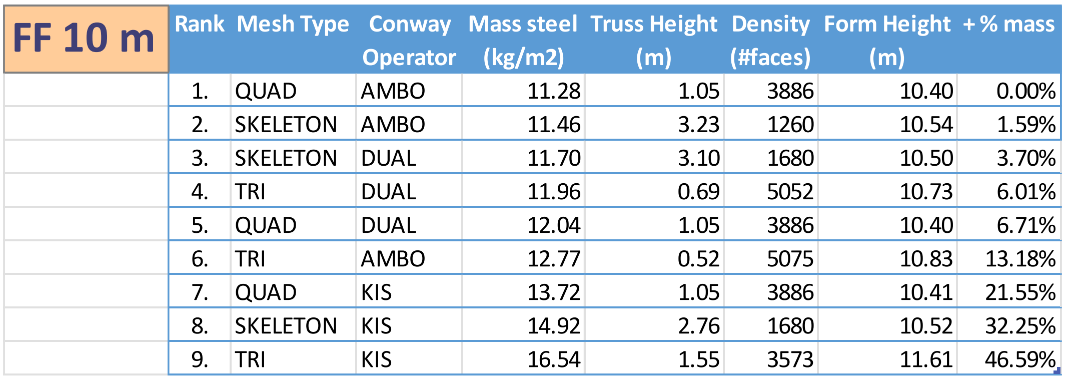
In table 09.02 the FF-25 m category is presented, Q+A is the optimal configuration in terms of steel usage, it is important to note that between the best and worst configuration there is a 58.14% difference in structural mass. Thus, in a practical design situation, considerable material savings can be achieved by choosing the appropriate space frame configuration. Furthermore, in the other two form finding height limit categories, the savings are about 47%.



FF 25 m	Rank	Mesh Type	Conway Operator	Mass steel (kg/m2)	Truss Height (m)	Density (#faces)	Form Height (m)	+ % mass
	1.	QUAD	AMBO	8.60	0.50	3561	23.40	0.00%
	2.	QUAD	DUAL	9.40	0.50	3707	22.62	9.30%
	3.	TRI	DUAL	9.46	0.50	4096	25.43	10.00%
	4.	TRI	AMBO	9.98	0.50	4096	25.43	16.05%
	5.	SKELETON	AMBO	10.11	2.94	1260	24.92	17.56%
	6.	SKELETON	DUAL	10.74	2.87	1470	21.12	24.88%
	7.	QUAD	KIS	11.00	0.50	3707	22.62	27.91%
	8.	SKELETON	KIS	13.00	3.08	1760	24.24	51.16%
	9.	TRI	KIS	13.60	0.50	5092	25.60	58.14%

Table 09.01 – Metaheuristic optimization results – FF 25 m

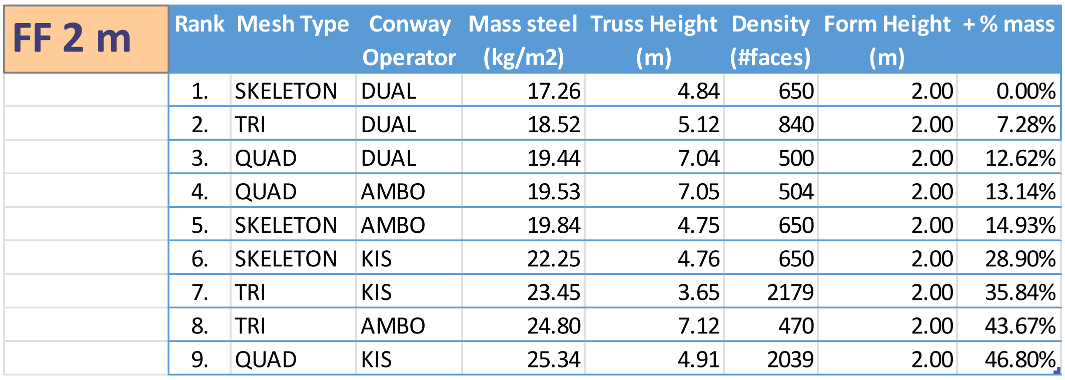
Viewing the results in Table 09.02., it is quite clear that in this category, a high form finding height is accompanied by a low truss height and high mesh density.



FF 10 m	Rank	Mesh Type	Conway Operator	Mass steel (kg/m2)	Truss Height (m)	Density (#faces)	Form Height (m)	+ % mass
	1.	QUAD	AMBO	11.28	1.05	3886	10.40	0.00%
	2.	SKELETON	AMBO	11.46	3.23	1260	10.54	1.59%
	3.	SKELETON	DUAL	11.70	3.10	1680	10.50	3.70%
	4.	TRI	DUAL	11.96	0.69	5052	10.73	6.01%
	5.	QUAD	DUAL	12.04	1.05	3886	10.40	6.71%
	6.	TRI	AMBO	12.77	0.52	5075	10.83	13.18%
	7.	QUAD	KIS	13.72	1.05	3886	10.41	21.55%
	8.	SKELETON	KIS	14.92	2.76	1680	10.52	32.25%
	9.	TRI	KIS	16.54	1.55	3573	11.61	46.59%

Table 09.02 – Metaheuristic optimization results – FF 10 m

Viewing the results in the 10 m form finding height limit category, Table 09.03, the solutions having least mass are found at higher truss heights than for 25 m limit and for certain configurations lower mesh densities.



FF 2 m	Rank	Mesh Type	Conway Operator	Mass steel (kg/m2)	Truss Height (m)	Density (#faces)	Form Height (m)	+ % mass
	1.	SKELETON	DUAL	17.26	4.84	650	2.00	0.00%
	2.	TRI	DUAL	18.52	5.12	840	2.00	7.28%
	3.	QUAD	DUAL	19.44	7.04	500	2.00	12.62%
	4.	QUAD	AMBO	19.53	7.05	504	2.00	13.14%
	5.	SKELETON	AMBO	19.84	4.75	650	2.00	14.93%
	6.	SKELETON	KIS	22.25	4.76	650	2.00	28.90%
	7.	TRI	KIS	23.45	3.65	2179	2.00	35.84%
	8.	TRI	AMBO	24.80	7.12	470	2.00	43.67%
	9.	QUAD	KIS	25.34	4.91	2039	2.00	46.80%

Table 09.03 – Metaheuristic optimization results – FF 2 m

Viewing the results in the 2 m form finding height limit category, Table 09.04, the solutions having least mass are found at even higher truss height than for 10 m and 25 m categories, while the mesh density is lower than in both categories.

09.5 Conclusion

Thus, analysing both brute force and meta heuristic optimization results, it is clear, that there is a significant difference between various space frame configurations in various parameter areas. Thus, space frame structures are influenced by their mesh topology. Furthermore, form finding height influences the structural mass of each configuration the most, and there are clear differences between them, showing how influential the choice of specific configuration is on the optimal result. It is interesting to note that even with a small form finding height, in relation to a flat space frame, considerable material savings can be achieved. The least efficient space frame would be a completely flat one. By just increasing the form finding height up to 1-3 m around 50% of material savings can be achieved. Furthermore, if one wants to favour least number of bars in the space frame, a low mesh density is beneficial. Thus, lower form finding heights might be optimal from that perspective. A lower mesh density means a higher average edge length of mesh edges. While optimality of space frame configurations is relative, depending on the posed criteria, certain space frame configurations will emerge as optimal. While this might seem trivial or obvious, with an appropriate space frame configuration choice one can optimize a space frame design, there is no “one size fits all” solution in this regard. The analysis carried out in this case study could be considered as one of the necessary steps in the overall design of space frame structures. It is highly beneficial to investigate many possible solutions within a relatively short time in the preliminary design phase, thus facilitating better quality designs, and lowering the risk of making wrong initial choices which then must be corrected in later design phases. Furthermore, considering that double curved large span space frame structures are often special projects with considerable project economy and technical challenges in execution, optimization and performance analysis is paramount for a successful design process.

To conclude the initial questions from introductory paragraph of this chapter, can be answered as follows:

Q1.) Which combination of mesh and operator, out of the nine possible, will perform best and in which category?

Metaheuristic optimization results categories

Q2.) Is there any benefit to form finding, what is the influence on F1 and F2?

Form finding influences optimal space frame configuration choice the most.

Q3.) How do the input parameter values influence the optimal solution?

Optimal solutions within form finding height categories of 25%,10% and 2%, show a certain trend. For high form finding height (25% of span length), solutions with least mass have low truss heights (0.5 – 2m) and high number of mesh faces (3000+). For medium form finding heights (10% of span length) solutions with least mass have increased truss heights (1 – 3m) and a lower number of faces (cca.1000),

	Dual (D)	Ambo (A)	Kis (K)
Quad (Q)	X	Least amount of steel (Cat 1. -F1) Occurs at high form finding heights –25% of span length	X
Tri (T)	Highest stiffness (Cat.1 -F2), occurs at high form finding heights – 25% of span length	X	Least number of different cross-sections utilized, accompanied with an average structural mass
Skeleton (S)	Least amount of steel, least number of cross-sections. Occurs at low form finding heights – 2% of span length	Least amount of steel, mesh density less than 2000 faces. Form finding height 5-10% of span	Highest stiffness, occurs at low form finding height – 2% of span length

Table 09.04

Matrix of which combination of mesh and operator have emerged as winners in categories

PART V
CONCLUSIONS

10 Conclusions and Recommendations

10.1 Introduction

The main goal of this research was to develop a parametric tool to investigate the relationship between mesh topology, Conway operators and structural behaviour of space frame structures formed based on them.

In this chapter, the results will be discussed following with conclusions and followed up with future recommendations. It is important to note that all mentioned percentages in following conclusions are a rule-of- thumb or rough estimates based on analysis done in the case study chapter, stating precise percentages does not add quality to conclusions but potentially impacts the readability.

Furthermore, the research problem as stated, or any similar research in the same sense, will always result in dealing with a multi-variable design space of the problem. With trying to expand the number of parameters, the search for the optimal design and gauging the influence of the parameters on the design, becomes less and less feasible.

Another problem was raised in the use of the word "optimal" to describe the solutions. The question of what is meant by optimal in this MSc thesis research is highly important. With optimal it was not the idea to suggest that there is one optimal solution in general, but that there exist relative optimal solutions in the context of different posed optimality criteria. In this research, four optimality criteria were considered initially, which in the opinion of the author could be considered as the four general optimality criteria for any civil engineering design. Namely, mass, stiffness, aesthetics, and fabrication. Mass and stiffness were investigated, due to the possibility of those being objective and quantifiable criteria. Aesthetics for example is hard to objectify and quantify and is often in the eye of the beholder. Fabrication as an optimality criterion was considered, however, the possible formulation of criteria would have to be quite rigorous and informed by experts within the space frame construction industry and can be a thesis topic by itself potentially.

However, even if all four of the optimality criteria could be implemented, this poses a question of how to weigh the influence of each of the criteria within the final solutions. There would have to be certain trade-offs specified, due to for example, the criteria of aesthetics and fabrication clashing together. What might be considered more aesthetic might prove much less fabricable.

Thus, the chosen optimality criteria considered in this MSc thesis research were the mass and stiffness. For large span structures, minimizing the structural mass and thus achieving both a more sustainable and economic solution, is important. Furthermore, due to the large span, stiffness is also important to limit the possible deflections. A space frame with a large span that deflects excessively is not something that should be designed in the first place.

The final conclusions, which were formed based on results of the case study, indicate more the direction towards where attention of the designer should be to optimize the design, rather than providing exactly quantified relations of parameters to achieve it. Furthermore, the tool can be used for preliminary design investigation and optimization of space frame structure configurations based on free-form input surfaces.

The parametric tool and thus the optimization process, can always be adjusted to specific project design requirements. For example, if there is a need for a minimal clearance height beneath the space frame structure, or a limit of space frame truss height due to desired aesthetical proportions in relation to total geometry of space frame, or a limited set of cross sections available on the market.

In general, any desired boundary condition can be imposed upon tool input parameters, thus narrowing the scope of the design space, and generating a new set of optimal space frame configurations.

10.2 Research question answers

In chapter 2, the main research question was stated as follows:

Given an architectural free-form irregular surface model to discretize into a steel space frame configuration, what is the optimal structural pattern configuration regarding multiple optimality criteria (mass, fabrication, aesthetics, stiffness in regards of chosen pattern) and their realistic constraints (load bearing behaviour, deflection, available types of steel cross sections)?

The posed research sub questions were as follows:

1.) *What are relevant surface tessellations/discretization's/meshing options for generating space frame structures?*
2.) *Is there any noticeable influence on structural performance of space frames regarding chosen tessellations?*
3.) *Which space frame configuration is most appropriate considering optimality criteria?*
4.) *What is the influence of the relevant parameters?*
5.) *Which structural pattern discretization strategy is the most appropriate?*
6.) *Which parameters governing the space frame structure configuration are most relevant for optimizing space frames?*

Answers to the following posed research questions are given as follows:

1.) *What are relevant surface tessellations/discretization's/meshing options for generating space frame structures?*

Answer: Based on the literature study done, the relevant tessellations, which serve to form the bottom layer of space frame, were identified as: Quadrilateral, Triangular and Skeleton based mesh, accompanied with Dual, Kis and Ambo Conway operator relations to form the space frame top layer.

2.) *Is there any noticeable influence on structural performance of space frames regarding chosen tessellations?*

Answer: Based on the case study done, the initial hunch of space frames being insensitive towards the choice of tessellation was proved to be wrong. Thus, there is a noticeable influence of tessellation and Conway operator as presented in graphs in chapter 9.

3.) *Which space frame configuration is most appropriate considering optimality criteria?*

Answer: For the researched pentagram shaped surface, the skeleton based quad mesh often proved the most appropriate. The specific combination of skeleton based quad mesh and kis operator emerged as most appropriate considering high stiffness (F2), and skeleton dual in terms of least material usage (F1)

5.) *Which structural pattern discretization strategy is the most appropriate?*

Answer: Meshing is the most appropriate discretization strategy for creating space frame structures based on free-form input surfaces. The mesh approach respects the total geometry of the inputted surface, inner and outer boundaries, and shape of the surface. Initially the discretization strategy was based on creating sub surfaces on the input surfaces using the Isosurface component in grasshopper, with which one can specify the number of U and V divisions of the surface and create subsurfaces. This approach was lacking due to this strategy not respecting the total geometry of the input surface. For example, if a trimmed surface was tried to be discretized by this strategy it would discretize the untrimmed geometry. One would then have to cull the subsurfaces which were out of the boundary of the trimmed geometry. This was more complex, less controllable and resulted in a much slower script which would in the end output incorrect geometry. Furthermore, the subsurface approach does not allow for generating meshes such as the skeleton-based mesh.

6.) *Which parameters governing the space frame structure configuration are most relevant for optimizing space frames?*

Answer: Optimization of space frame, in this case study, is understood as the minimization of structural mass (F1) or minimization of deflections (F2). The form finding height parameter influences these two fitness goals the most.

10.3 Conclusion

In this thesis research the following areas of influence were investigated to understand trends in parameter influence on optimal space frame structure configuration (top and bottom mesh topology):

1.) Type of base mesh – Quad, Tri, Skeleton-based
2.) Conway operator relations – Dual, Ambo, Kis
3.) Mesh density – 500 – 5000 faces
4.) Truss height – 0.5 m – 8 m
5.) Form finding height – 2% - 25% of span

The search for an optimal solution depends on the fitness function or goal posed. Furthermore, it depends on the number, scope and type of parameters thought of as important for the optimization of the space frame structure. With the tool developed in this thesis, the solutions can always be considered as 3 x 3 matrix (3 mesh types x 3 Conway operators) for which we evaluate goals (F1 or F2) regarding certain parameters (Mesh density, truss height, form finding height).

The focus in this research was on the two goals (fitness functions), lowest mass (F1), highest stiffness (F2) in relation to three parameters, mesh density, truss height and form finding height.

Form finding influences F1 and F2 the most. One hypothesis would be that with form finding the shape of the surface is adjusted to behave more like a shell or arch than a flat plate, thus lessening the required amount of steel and having a higher stiffness. For example, a flat space frame on a large span will behave like a plate, meaning truss height will be the main parameter for achieving adequate load bearing behaviour, and the cross sections will be large. It is important to note that even with a small form finding height (slight camber of the space frame), the structural mass can be considerably lowered, with 2% form finding height, up to 50% of material savings are achieved in relation to completely flat space frame.

To understand how space frame configurations contribute to overall structural behaviour, two approaches were used. Namely, brute force approach and meta-heuristic optimization approach. The Brute force approach was applied to generate a broad spectrum of possible results, thus visualizing the problem design space by 9000 discrete solutions. This approach facilitates understanding of which parameters are influential on optimal space frame configurations, by investigating the parallel coordinate graphs generated and generating further specific graphs such as the cumulative graphs for F1 (least mass) and F2 (highest stiffness) in relation to FF (form finding).

The Metaheuristic optimization approach was used to further investigate optimal solutions within more narrowed form finding height parameter ranges. Specifically, the influence of form finding was further investigated for nine possible configurations at three specific form finding height parameter limits. First, maximum form finding height was allowed up to 25% of the span, then 10% and finally 2% of span length. These areas were chosen after analysing the cumulative graphs, where it was clear that after a form finding height of 25% of span length, there is no more influence on lowering the structural mass. Furthermore, investigating larger form finding height than 25% of span length creates highly exaggerated arched space frames, which do not make much sense as structures. This was investigated by analysing beforementioned brute force approach results, which were generated without limiting input parameter scopes, facilitating design space discretization in full parameter range, meaning both feasible, unfeasible, desirable, and undesirable solutions were generated. Thus, after the broad picture was established, for further analysis, a narrower parameter scope was defined for metaheuristic optimization.

It was found that in the 25% form finding height limit category the space frame configurations with least mass have a small truss height (0.5 – 1 m) and high number of mesh faces (3000+). In the 10% configurations with least mass had a higher truss height than 25% category (1-2 m), and lower number of mesh faces (1000-2000). Finally in the 2% category, this trend continues further, and accordingly, truss heights are higher than in 10% category (+ 4m) and the number of mesh faces is lowest (less than 1000).

Analysing the trends of how parameters influence the optimal solution, it is thus clear, higher form finding heights, achieve material savings, lower truss height, and increase mesh density. Conversely, lowering form finding height, increases material usage (though generally the optimal masses are all lower than $20kg/m^2$ which is considered quite low for a steel structure), increases the truss height, and decreases number of mesh faces. A rationale for such structural behaviour can be explained by two phenomena. Firstly, as the form finding height from flat to double curved space frame, the structural action becomes from plate behaviour to shell-like behaviour. Exceeding 25% of span length form finding height entails more arch-like structural behaviour. Secondly, the relationship between form finding height, truss height and number of mesh faces is influenced by maintaining sensible angle ranges between bars of the space frame. All the optimal solutions found have angles between bars which are not to steep or too shallow, roughly that entails a range from 30-60 degrees.

To conclude, out of the nine possible space frame configurations that were researched, the following matrix sums up which configurations emerged as optimal regarding lowest mass (F1) and highest stiffness (F2) criteria:

	Dual (D)	Ambo (A)	Kis (K)
Quad (Q)	X	Least amount of steel (Cat 1. -F1) Occurs at high form finding heights – 25% of span length	X
Tri (T)	Highest stiffness (Cat.1 -F2), occurs at high form finding heights – 25% of span length	X	Least number of different cross-sections utilized, accompanied with an average structural mass
Skeleton (S)	Least amount of steel, least number of cross-sections. Occurs at low form finding heights – 2% of span length	Least amount of steel, mesh density less than 2000 faces. Form finding height 5-10% of span	Highest stiffness, occurs at low form finding height – 2% of span length

Table 10.01

Matrix of which combination of mesh and operator have emerged as winners in categories

Optimum	Parameters	2%	10%	25%
min. mass	mesh + operator	**Skeleton+Dual**	**Skeleton-Ambo**	**Quad+Ambo**
	truss height (m)	5	3	0.5
	nuber of faces	650	1250	2600
	mas (kg)	17,25	11.48	8.07
	el.en. (kNm)	582	352	247
max. stiffness	mesh + operator	**Skeleton+Kis**	**Skeleton-Ambo**	**Tri-Dual**
	truss height (m)	7	5.9	5.8
	nuber of faces	250	300	290
	mas (kg)	26.27	16.45	14.95
	el.en. (kNm)	431	186	82

Table 10.02
Design guideline depending on general form height categories

Furthermore, the results can also be summarized in Table 10.2 as a design guideline for space frame structures in relation to desired form height. It can be used as a rough indication of which mesh and operator combination should be chosen in regards to desired goal (mass, stiffness) in respect to desired form height (2%,10%,25%). It can thus help in narrowing down the "width" of the design space, allowing for more investigation of the chosen "depth" of the design space.

10.4 Further Recommendations

Before discussing further recommendations, I will first describe here a few problems that occurred during the development and research for this thesis. The first and foremost problem encountered was the continuous occurrence of new ideas of how to restructure both this thesis, tool logic, and focus of research. Thus, the developed tool has a large amount of input data and possibilities for researching a vast number of combinations of the related parameters. This often created some confusion, in which parameter relations and influence should be investigated, because to investigate everything that is possible with the developed tool is not feasible in the scope of this MSc thesis. Thus, gradually the scope was narrowed to be adequate for MSc thesis and resulted in the presented case study and the conclusions.

Considering the scope of research possible with the developed research tool, some recommendations can be given. The developed tool can facilitate preliminary design investigation and optimization of space frame structures based on nine possible structural configurations in a potential design setting. Depending on the hypothetical space frame project requirements, the tool can be adjusted to accommodate research accordingly. For example, in a particular design situation a flat space frame typology might be of interest (no form finding) and by simply inputting the desired flat geometry, the tool can generate similar results as in the case study, facilitating early design exploration and optimization.

Additionally, in this thesis research space frame configurations where always based on pyramid module type connectivity between top and bottom layers and hinged boundary conditions between bars. A potential direction of research could be made into investigating other possible connections between top and bottom layer accompanied with appropriate boundary conditions (nodes could be set as fixed for example) and Vierendeel type of connectivity might be investigated. With slight adjustment, the tool can also facilitate investigation into space grid typology behaviour. Furthermore, the load cases considered were self-weight of the structure + symmetric point loads in nodes representing additional loads, such as self-weight of cladding. An interesting research direction might be to see how the space frame configurations are influenced by more complex load cases. These might entail asymmetric vertical loads due to snow, horizontal loads due to wind, or if the space frame structure should support certain point loads, such as a score screen on a stadium or a walkway which is suspended on the space frame structure. The ambiguity of the tool allows for design exploration only limited by the imagination of the designer and realistic project boundary conditions.

Moreover, the parametric tool was developed within grasshopper and rhino, utilizing kangaroo plugin for form finding, karamba3d for structural analysis and cross section optimization, Galapagos for metaheuristic optimization and Colibri + Design Explorer for a brute force calculation of a vast number of space frame configurations, thus facilitating design space exploration to form conclusion about structural behaviour. Considering that grasshopper and rhino are the most common choice of parametric software, and the large number of additional plugins available, the choice for the same is clear. Even though the approach to creating the tool with rhino and grasshopper might be practical in terms of familiarity of designers with software, and consistency in software environment, due to inherent software limitations, the speed of the developed scripts is not optimal. This could be improved by applying a more custom software design approach, requiring building the functionality from ground-up and more extensive programming knowledge, thus imposing a trade-off, speed vs. ease of software implementation. The utilization of all CPU cores, parallel processing in general, combined with possible GPU utilization could greatly improve the speed of the tool.

Bibliography

[1] Eleftheria Touloupaki and Theodoros Theodosiou. Performance simulation integrated in parametric 3d modeling as a method for early-stage design optimization—a review. Energies, 10(5),2017.

[2] Dariusz Walasek and Arkadiusz Barszcz. Analysis of the adoption rate of building information modeling [bim] and its return on investment [roi]. Procedia Engineering, 172:1227–1234, 12 2017.

[3] Antiopi Koronaki, Paul Shepherd, and Mark Evernden. Layout optimization of space frame structures. September 2017. IASS Annual Symposium 2017: Interface: architecture, engineering, science; Conference date: 25-09-2017 Through 27-09-2017.

[4] Paul Shepherd and Will Pearson. Topology optimization of algorithmically generated space frames. In Proceedings of IASS Annual Symposia, number 12, pages 1–7. International Association for Shell and Spatial Structures (IASS), 2013.

[5] Robin Oval, Matthias Rippmann, Romain Mesnil, Tom Van Mele, Olivier Baverel, and Philippe Block. Topology finding of structural patterns. In AAG, 2018.

[6] Antiopi Koronaki, Paul Shepherd, and Mark Evernden. Rationalization of freeform space-frame structures: Reducing variability in the joints. International Journal of Architectural Computing, 18(1):84–99, 2020.

[7] Chilton J. (2007). Space grid structures. Taylor and Francis. Retrieved September 20 2022 from http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=297091.

[8] Tien T Lan. Space frame structures. Structural engineering handbook, 13(4), 1999.

[9] D. Veenendaal and P. Block. An overview and comparison of structural form finding methods for general networks. International Journal of Solids and Structures, 49(26):3741–3753, 2012.

[10] Popescu, M. and Oval, R. (2021, September 11). ACADIA 2021 – Knitted Growth [gitbook]. Retrieved from https://tudelft-1.gitbook.io/knittedgrowth/-MjJvUl_2s1K4MnAdtxO/schedule

[11] Hiester, H.R. & Piggott, Matthew & Farrell, P.E. & Allison, P.. (2014). Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange. Ocean Modelling. 73. 30–44. 10.1016/j.ocemod.2013.10.003.

[12] Conway, John; Burgiel, Heidi; Goodman-Strauss, Chaim (2008). "Chapter 21: Naming Archimedean and Catalan polyhedra and tilings". The Symmetries of Things. AK Peters. p. 288. ISBN 978-1-56881-220-5.

[13] George W. Hart (1998). "Conway Notation for Polyhedra". Virtual Polyhedra.
from https://www.georgehart.com/virtual-polyhedra/kepler.html

[14] Robin Oval, Romain Mesnil, Tom Van Mele, Philippe Block, and Olivier Baverel. Two-colour topology finding of quad-mesh patterns. Comput. Aided Des., 137:103030, 2021.

[15] Inês Caetano, Luís Santos, and António Leitão. Computational design in architecture: Defining parametric, generative, and algorithmic design. Frontiers of Architectural Research, 9(2):287–300, 2020

[16] Swartout, C 2016, „Case Study: An Integrated Approach to Workplace Design and Delivery“, FMJ magazine by IFMA, January/Feburary, p.21.

[17] Woodbury, R. Elements of Parametric Design. 2010, Routledge

[18] David Karle and Brian M Kelly. Parametric thinking. 2011.

[19] Clemens Preisinger. Linking structure and parametric geometry. Architectural Design, 83(2):110–113, 2013.

[20] Sarvesh P.S. Rajput and Suprabeet Datta. A review on optimization techniques used in civil engineering material and structure design. Materials Today: Proceedings, 26:1482–1491, 2020.10th International Conference of Materials Processing and Characterization.

[21] Kelley, T. R. (2010). Optimization, an important stage of engineering design. The Technology Teacher, 69(5), 18-23.

[22] Oslo Metropolitan University. (2020, May 29). Optimum design of structures. Structural Engineering Research Group (SERG). Retrieved September 25, 2022, from https://uni.oslomet.no/serg/optimum-design-of-structures/

[23] Eschenauer, H., Olhoff, N., Schnell, W. (1997). Fundamentals of structural optimization. In: Applied Structural Mechanics. Springer, Berlin, Heidelberg. https://doi-org.tudelft.idm.oclc.org/10.1007/978-3-642-59205-8_15

[24] Linfeng Mei and Qian Wang. Structural optimization in civil engineering: A literature review. Buildings, 11(2):66, 2021.

[25] Fred Glover. Future paths for integer programming and links to artificial intelligence. Computers Operations Research, 13(5):533–549, 1986. Applications of Integer Programming.

[26] Bandaru, S., & Deb, K. (2016). Metaheuristic techniques. In Decision sciences (pp. 709-766). CRC Press.

[27] Yang, X. S., Bekdaş, G., & Nigdeli, S. M. (Eds.). (2016). Metaheuristics and optimization in civil engineering. Cham: Springer International Publishing.

[28] Zavala, G.R., Nebro, A.J., Luna, F. et al. A survey of multi-objective metaheuristics applied to structural optimization. Struct Multidisc Optim 49, 537–558 (2014). https://doi-org.tudelft.idm.oclc.org/10.1007/s00158-013-0996-4

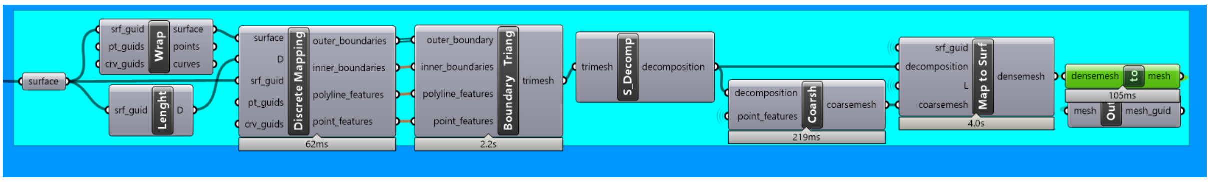
[29] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

[30] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

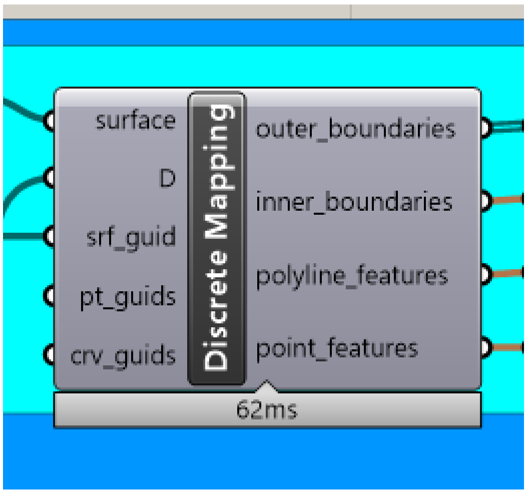
[31] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of optimization theory and applications, 45(1), 41-51.

[32] Mohammad-Taghi Vakil-Baghmisheh and Alireza Navarbaf. A modified very fast simulated annealing algorithm. 2008 International Symposium on Telecommunications, pages 61–66, 2008.

PYTHON SCRIPT – COMPAS_SINGULAR

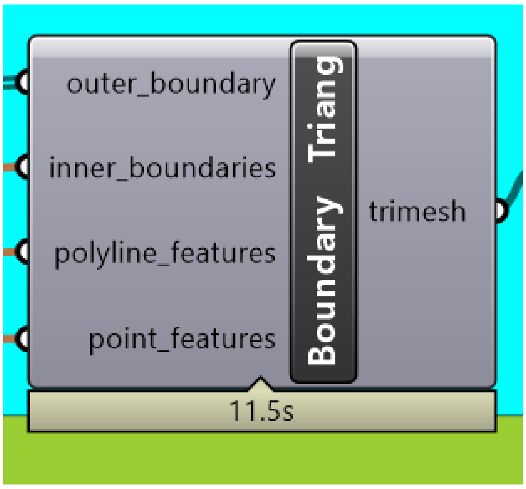


DISCRETE MAPPING



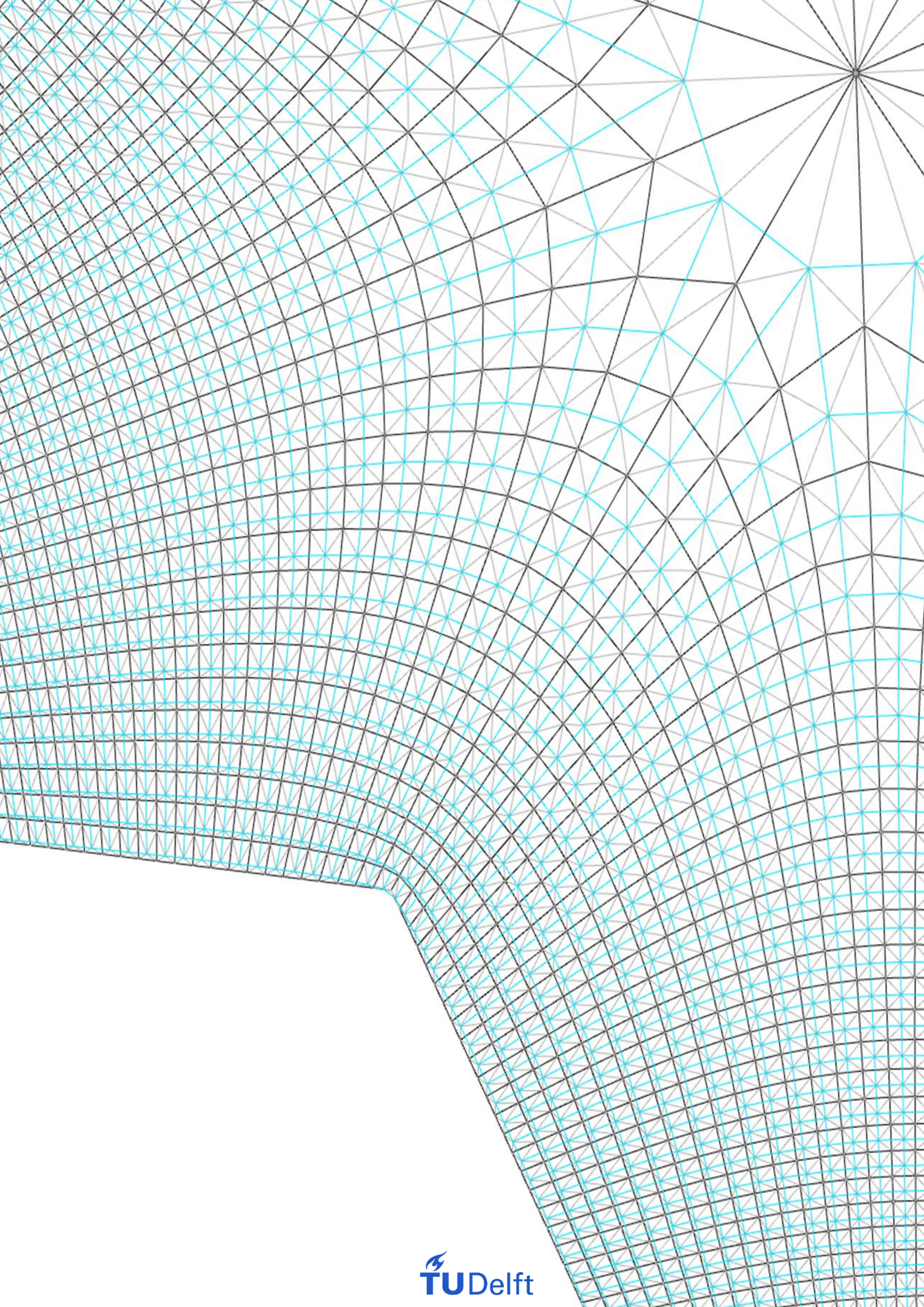
```
import rhinoscriptsyntax as rs
from compas.datastructures import network_polylines
from compas.datastructures import Network
from compas.utilities import pairwise

def point_xyz_to_uv(self, xyz):
        return rs.SurfaceClosestPoint(self, xyz)

def borders(self, border_type=0):
        curves = rs.DuplicateSurfaceBorder(self, type=border_type)
        exploded_curves = rs.ExplodeCurves(curves, delete_input=False)
        if len(exploded_curves) == 0:
            return curves
        rs.DeleteObjects(curves)
        return exploded_curves

def discrete_mapping(self, segment_length, minimum_discretisation, crv_guids=[],
pt_guids=[]):
        Aborders = []
        for btype in (1, 2):
            border = []
            for guid in borders(self, border_type=btype):
                L = rs.CurveLength(guid)
                N = max(int(L / segment_length) + 1, minimum_discretisation)
                points = []
                for point in rs.DivideCurve(guid, N):
                    points.append(list(point_xyz_to_uv(f, point)) + [0.0])
                if rs.IsCurveClosed(guid):
                    points.append(points[0])
                border.append(points)
                rs.DeleteObject(guid)
            Aborders.append(border)
        outer_boundaries = network_polylines(Network.from_lines([(u, v) for border in
Aborders[0] for u, v in pairwise(border)]))
        inner_boundaries = network_polylines(Network.from_lines([(u, v) for border in
Aborders[1] for u, v in pairwise(border)]))

        # mapping of the curve features on the surface
        curves = []
        for guid in crv_guids:
            L = rs.CurveLength(guid)
            N = max(int(L / segment_length) + 1, minimum_discretisation)
            points = []
            for point in rs.DivideCurve(guid, N):
                points.append(list(point_xyz_to_uv(f, point)) + [0.0])
            if rs.IsCurveClosed(guid):
                points.append(points[0])
            curves.append(points)
        polyline_features = network_polylines(Network.from_lines([(u, v) for curve in
curves for u, v in pairwise(curve)]))

        # mapping of the point features on the surface
        point_features = [list(point_xyz_to_uv(f, rs.PointCoordinates(guid))) + [0.0] for
guid in pt_guids]

        return outer_boundaries[0], inner_boundaries, polyline_features, point_features

a = surface
b = D
c = 5
d = crv_guids
e = pt_guids
f = srf_guid

result = discrete_mapping(f,b,c,d,e)
```

BOUNDARY TRIANGULATION



```
#import compas
#from compas.geometry import delaunay_from_points
from compas.rpc import Proxy
proxy = Proxy("compas.geometry")
```

```
delaunay = proxy.delaunay_from_points_numpy
from compas.datastructures import Mesh
from compas.geometry import subtract_vectors
from compas.geometry import length_vector
from compas.geometry import cross_vectors
from compas.datastructures import trimesh_face_circle
from compas.geometry import is_point_in_polygon_xy
from compas_rhino.artists import MeshArtist


# Triangulate the input surface.
def boundary_triangulation(outer_boundary, inner_boundaries, polyline_features=[],
point_features=[], delaunay=None):

    if not delaunay:
        delaunay = delaunay_from_points

    # generate planar Delaunay triangulation
    vertices = [pt for boundary in [outer_boundary] + inner_boundaries + polyline_features
for pt in boundary] + point_features
    faces = delaunay(vertices)

    delaunay_mesh = Mesh.from_vertices_and_faces(vertices, faces)

    # delete false faces with aligned vertices
    for fkey in list(delaunay_mesh.faces()):
        a, b, c = [delaunay_mesh.vertex_coordinates(vkey) for vkey in
delaunay_mesh.face_vertices(fkey)]
        ab = subtract_vectors(b, a)
        ac = subtract_vectors(c, a)
        if length_vector(cross_vectors(ab, ac)) == 0:
            delaunay_mesh.delete_face(fkey)

    # delete faces outisde the borders
    for fkey in list(delaunay_mesh.faces()):
        centre = trimesh_face_circle(delaunay_mesh, fkey)[0]
        if not is_point_in_polygon_xy(centre, outer_boundary) or
any([is_point_in_polygon_xy(centre, inner_boundary) for inner_boundary in
inner_boundaries]):
            delaunay_mesh.delete_face(fkey)

    return delaunay_mesh

#print outer_boundary


trimesh = boundary_triangulation(outer_boundary, inner_boundaries, polyline_features,
point_features, delaunay=delaunay)
```



TU Delft