Politecnico di Torino

Master's Degree Program in Civil Engineering

Master Thesis

Development of a double step optimization workflow for Architectural Design Optimization

Supervisors:

Prof. Giuseppe Carlo Marano (Politecnico di Torino)

Prof. Nikos D. Lagaros (National Technical University of Athens)

Ph.D. Laura Sardone

Candidate:

Vincenzo Fasciano
277848

A.y. 2021/2022

Summary

INDEX OF FIGURES

1 Introduction

1.1 Abstract

The human need to challenge physics and realize complex and spectacular structures led architects like Gaudi, Gehry, and Zaha Hadid to design free-form systems covering long spans without intermediate columns (free-span structures).

Free-span structures are usually modeled as shells, especially in the case of the design of roof covers. Shells are defined as curved external membranes representing a particular case of plates.

From the structural point of view, the shells are subjected only to membrane actions, which is advantageous since bending moment is avoided. Moreover, a particular type of shell can be defined as the gridshell, which is a grid of beams that works like a shell; therefore, all the beams are only subjected to membrane actions.

The structures mentioned above cannot be modeled with usual CAD software, but it is necessary to adopt tools to work with changing parameters. Therefore, it is fundamental to progress from the classic design methods to algorithmic drawing, in which the structure evolves as a function of some variables (defined as parameters). The shapes complexity obtained thanks to the Algorithm Aided Design approach is then subjected to structural analysis thanks to the direct link with the possibility of implementing the FEM.

However, there is a problem in the interaction between engineering and architecture processes since there is no direct connection between the structural and the architectural model. For example, suppose the user changes some parameters in the architectural design. In that case, a new structural definition is necessary: this is a critical issue since it increases the time and the steps required for obtaining the final design, especially when dealing with optimization.

The target of the thesis is to simplify this process by using different embedded tools in Grasshopper. The latter one, provided by McNeel&Associates allows the user to work with Finite Element Analysis tools like Karamba3D and Sofistik inside the environment of the software, therefore still using the algorithmic modeling approach. In terms of tensions and deflections, the evaluation of the structural results is a function of the parameters imposed by the designer, and it will be performed at different steps.

To implement a methodology that considers the factors of design (parameters) connected to the finite element modeling (FEM), with the related structural optimization, an initial model, represented by a concrete shell, has been developed. This initial design applies a double-step optimization: the initial geometry is subject to the form-finding with the add-on Kangaroo (based on the dynamic relaxation); afterward, there are subsequent phases of structural optimization with the target of minimizing the mass and the tensions inside the structural object. The second step of structural optimization uses an evolutionary-genetic solver.
This procedure is repeated in the final model. In this case, the case study is a gridshell structure with a larger span. A gridshell is a grid constituted by beams; therefore, the optimization process is easier to apply since there are many smaller elements; due to the presence of a considerable number of structural components, it is more difficult to understand clearly the distribution of the tensions.

1.2 Premise

The initial goal of the thesis was to compare the results obtained with the structural analyses of the object subjected to a double-step of optimization using two different Finite Element Model software, to check the validity of the values of stresses and displacements obtained.
However, during the experimentation, there have been some issues in the use of Sofistik, the link between Sofistik and Grasshopper works perfectly for “simple” cases, for instance for a truss structure or a small bridge.
Instead, in the case of shells with multiple elements and supports, there are problems in the computation due to a large number of elements, and also a too high number of supports generate an error in terms of the degree of freedom of the system that does not allow the analysis to proceed.

2 Parametric Architecture

The process of architectural design is usually an iterative process, where the target is to reach the best solution from the aesthetical, functional, structural, and economical point of view. However, the classic architecture with its software is not able to perform an iterative process, each element should be replaced manually.

To obtain the automatic modification of all the different elements, the parametric design with its software is introduced. In Parametric Architecture, the entire structure will be a function of some parameters, so if one or more of these parameters are modified, the whole structure will update accordingly.

There is also another important difference in Parametric Design, in fact in this case the design is not done as usual, but it is necessary to build an algorithm, which output will be the 3D representation of the designed object.

Figure 1 Parametric architecture example (IMA Architects, s.d.)

2.1.1 History of Architectural Drawing

Architects have always drawn before building, this concept represents the difference between architecture and construction. Drawings have been also the medium through which the architects could organize ideas and space. (Tedeschi, 2014)
However, the drawing has evolved in years due to innovations provided by technology.
In the beginning, the drawing was considered just as an additive process, the overlap of signs to obtain the final figure. (Tedeschi, 2014).
Therefore, there were no relations between the elements of the drawing, and it is considered not smart, but simply a graphical code to show and explain a certain idea of building.

Figure 2 Handwritten drawing (5.imimg.com, s.d.)

With the introduction of Computer Aided Drawing (CAD) software, there have not been many changes from the conceptual point of view, CAD allows the user to draw more elements in a faster way, however, there are no relations between the elements of the drawing, therefore, even CAD drawings are not smart mediums.

The conventional drawing was questioned for the first time with the introduction of form-finding in the 19th century, which aimed to investigate novel and optimized structures found through complex and associative relations between materials, shape, and structures. (Tedeschi, 2014)

These form-finding structures couldn't be drawn with the conventional approach, therefore they relied on physical models like soap films.

In the last years, form-finding became an important method to define the optimal shape for complex structures. However, form-finding at the beginning was based only on one parameter (gravity), while now there is the use of multi-parametric form-finding, in which the optimization is based also on geometry, or external loads.

The final step of drawing development during the last years is algorithmic modeling. An algorithm is a procedure used to return a solution to a question or to perform a particular task, through a finite list of basic and well-defined instructions. (Tedeschi, 2014)

When working with algorithms, the problem is divided into a list of consecutive smaller problems that can be computed easily. A simple example of an algorithm is a recipe, in which there are a series of simple steps to obtain for instance a cake.

There are some rules to respect to make an algorithm work without errors, the instruction should be clear, and should be applied to a set of input, from which the algorithm will generate the output.

Figure 3 Algorithm scheme (Tedeschi, 2014)

Consequently, the final output of an algorithm is not just a digital sign, but it depends on the input parameters, and it can change as a function of them, therefore there is a relation between the different elements of the drawing.

2.2 Characteristics of Parametric Architecture

Even if each project in the Parametric design field is very different from the others, there are common characteristics to all the computational design systems.

The first common attribute used by the software in this environment is object-orientation. Practically, the user, while working with Parametric software, creates objects like circles, cubes, spheres. Each object has its values for its main attributes, for example in the case of a line the attribute defined will be the starting point A and the endpoint B. The values related to these characteristics can be numbers (floating or integer) or functions, introducing a function, a dependency on a parameter or another object can be defined.

Another common aspect is the presence of classes and families of objects. The objects that are characterized by the same features can be organized in classes or families (for example a class of walls could be constituted by walls of a thickness of 30 cm, and thinner walls, because even if they are used for a different structural function, they are both walls)

In parametric design, there is not a set of centralized instructions, on the other hand, a method can be defined for each family, so the method is not valid for the whole project, but only for a certain class of elements (for example a surface can be defined as a function of its edges or drawing it and introducing it in the used algorithm).

Finally, the most important aspect of computational design is constituted by the parameters. The word "parameters" is a composed word with ancient Greek origin, and the meaning could be explained as the term that determines another measure. However, the term has different meanings according to the area considered, for example usually the parameter is different from the variable, because it is something assigned, so it is not unknown, moving to the Parametric CAD software, the parameter will be a variable that is used in equations, to obtain another value.

2.3 Software for Parametric Architecture

In the application of the concepts of Generative Design, it is necessary to use some tools. It is necessary to use together with the software that allows the designer to define the algorithm and a tool for its graphical representation. For instance, Revit and Dynamo can be coupled to solve the parametric design problems, or, as in the case under study, Rhinoceros 3D and Grasshopper can be used for the same purpose.

2.3.1 Rhinoceros 3D

Rhinoceros, or simply Rhino, is a 3D modeling software, very popular for manufacturers. It is based on Non-Uniform Rational Basis Splines (NURBS) geometry, and it is mainly used in fields like architecture, industrial design, jewelry, and engineering. (www.3dnatives.com)

In the beginning, this software was only a complement to the cad of AutoDesk (AutoCad), but its importance has increased during the years, especially because of the complexity of shapes and elements that can be drawn with Rhino.

The software, developed been McNeel&Associates, is categorized in the CADs, but there are some differences compared to AutoCAD or other similar tools, since Rhinoceros allows the user to build very complex shapes with high precision, therefore, while form-finding elements cannot be represented in AutoCAD, they can be illustrated in Rhino.

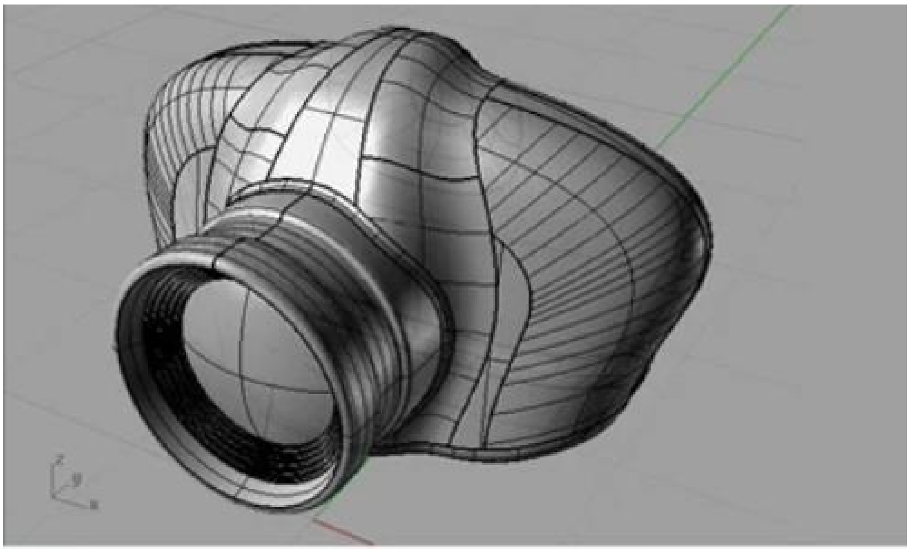

Figure 4 Camera 3d drawing in Rhino (3D Natives, s.d.)

2.3.2 Grasshopper

Grasshopper is one of the most popular and advanced algorithmic modeling tools. It is a plug-in of Rhinoceros, and its original name was “Explicit history”. This plug-in has become important both in the academic and job context in the latest years.

The advantages of Grasshopper are many, there are plug-ins for different needs, from the meshing of surfaces to structural analysis, moreover, there is a community of users who solve each other problems, and they can interact with other software, like Excel, Revit, Photoshop.

Grasshopper works always together with Rhino, to allow the user a 3d visualization of the algorithm. The objects to build the algorithms are called components, and they can represent elements, mathematical or logical functions. Practically, Grasshopper works like programming software, with a visual language, which is more intuitive than other languages, since the user does not have to code.

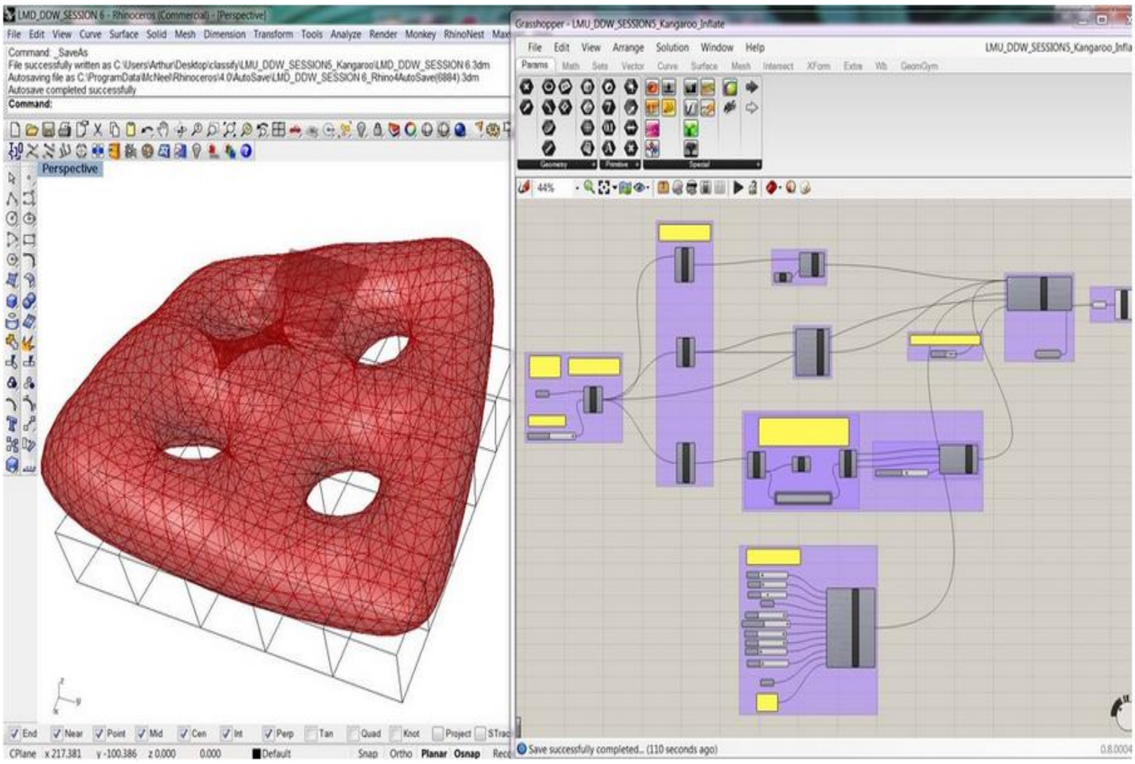

Figure 5 Example on Grasshopper (Grasshopper, s.d.)

2.4 Examples of Parametric Architecture

Since the end of the 19th century, many structures were built using the innovative approach of Parametric Architecture.

One of the first examples is supposed to be contained in the Church of Colonia Guell, located in Santa Coloma de Cervellò (Spain). This church has been designed by one of the most important architects of the last century, Antoni Gaudi, and its shape is very similar to the most known Sagrada Familia of Barcelona. But the novelty is represented by the presence of a model of strings weighted down to create complex arches and vaulted ceilings. Practically, by adjusting the position of strings he could change the shape of each arch, and consequently the shape of ceilings.

Figure 6 Santa Coloma de Cervellò strings (Researchlm, s.d.)

Moving to the 21st century, there are examples of computational design in all fields, from sports to entertainment.
Considering the entertainment area, the first example that could be included in the Sage Gateshead, located in Gateshead (UK), which is a complex of three auditoriums in which folk, jazz, and blues music is usually played, was designed by Fosters+Partners and completed in 2004. The total seat capacity is 2725 people, and the structure covers an area of 20000 m^2. In 2005 this auditorium has won different awards for its acoustics, quality, and complexity of design.

Figure 7 Sage Gateshead (RIBA Architecture.com, s.d.)

Another example can be found in Baku (Azerbaijan), and it is the Heydar Alijev Centre, which is a center mainly used for nation's culture programs and was dedicated to the former president Heydar Alijev. It was designed by the office Zaha Hadid Architects, and it is very different from the city landscape for its curvilinear shape, opposed to the rigid lines of Russian architecture. The building covers an area of 57000 m^2, and it is surrounded by a public park, it was finished in 2012, five years after the call for tender won.

Figure 8 Heydar Aliyev Cultural Centre (Arcquitectura Viva, s.d.)

Moving to the sports infrastructures, the generative design is applied also for stadiums, in different ways. For example, in the case of the Mercedes Benz Stadium, located in Atlanta (USA), the petals that compose the cover of the stadium are made with parametric design, these petals with the push of a button will rotate and reposition themselves to have the football pitch covered in case of rain, then the cover can be opened again. The whole process of opening or closing these petals lasts 7-8 minutes. The stadium hosts soccer and a football team, it has a total capacity of 71000 supporters, and according to the architect Bill Johnson, the roof is inspired by the Pantheon. This revolutionary stadium was opened in 2017, and it cost about 1,5 billion dollars.

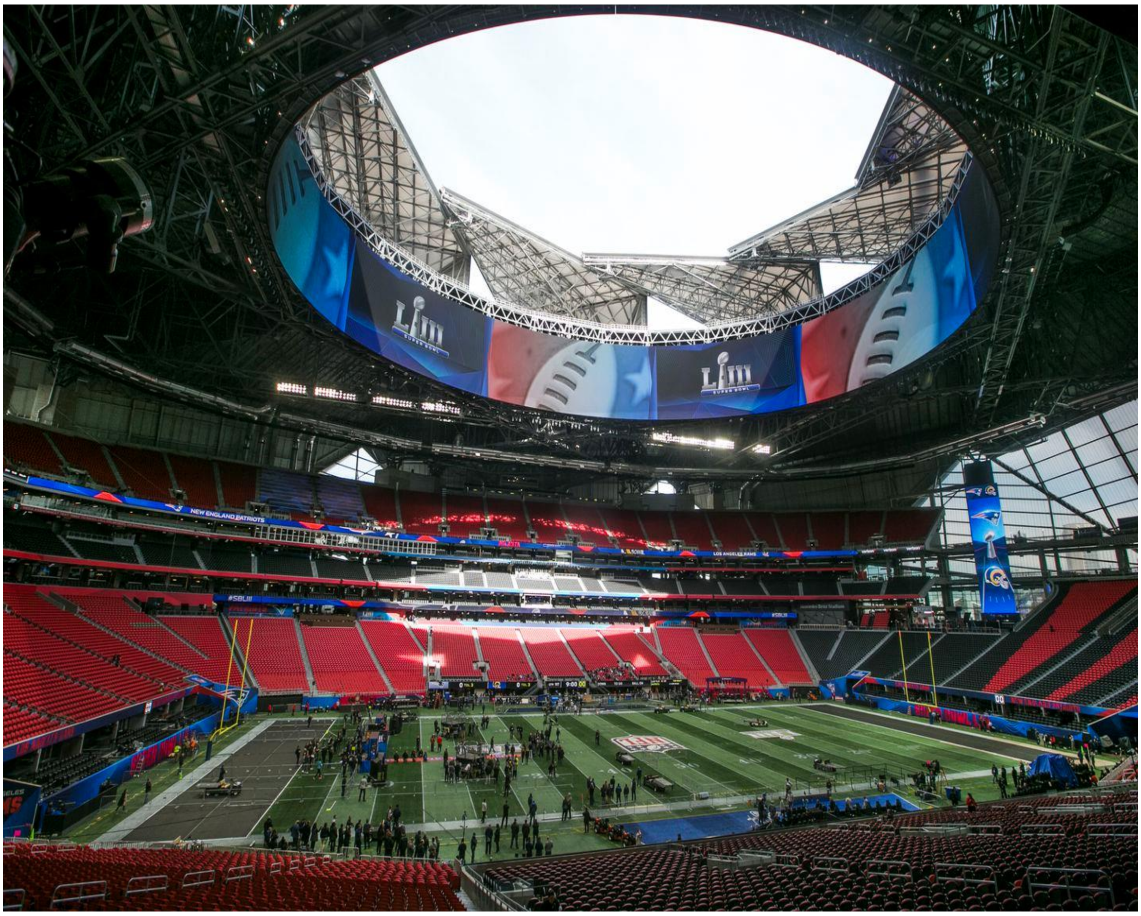

Figure 9 Mercedes-Benz Stadium (The Boston Globe, s.d.)

Another example is the Al Janoub Stadium, located in Al Wakrah (Qatar), which opened in May 2019, it's been the second stadium, between the ones of Fifa World Cup 2022, to be completed. The design was made by Zaha Hadid Architects, inspired by a traditional Arab sailing boat called dau, and this stadium can be closed completely to protect the supporters and players from the hot climate, typical of the middle east region. Its capacity is 40000 people, but after the World Cup, it will be reduced to 20000.

Figure 10 Al Janoub Stadium (ArchDaily, s.d.)

Parametric architecture is also applied to train stations, an example is the Station of Liege-Guillemins, which is the main station of Liege, in Belgium. It is one of the busiest train stations in the Wallonia Region, but it's also an important railway hub for high-speed trains.

This station was designed at the beginning of the 21st century by the engineer and architect Santiago Calatrava. He aimed to increase the permeability by creating a building without façades and relying on the roof to have an identity of the structure. The roof is constituted by a canopy that covers five platforms, and about 145 m, and it is connected to columns made of concrete. The project won also an Award for Excellence in its field, from the European Concrete Society Network.

Figure 11 Station of Liege-Guillemins (G2Rail, s.d.)

3 Shell structures

In architecture and civil engineering, a shell structure is a curved external membrane that can be used as roof cover in constructions.
Usually, this membrane is made of reinforced concrete, it is characterized by a reduced thickness, and its main features are continuity and curvature.
Moreover, shell structures are defined as form-passive structural systems, and a form-passive structural system does not significatively change its shape with the variation of load conditions. (Adriaenssens, et al., 2014)

3.1 Membrane Shell Theory

The membrane shell theory assumes that the equilibrium in the shell is achieved due to the presence of in-plane membrane tensions that resist all the applied loads without any bending moment. (Jawad, 1994)
However, this theory gives precise results considering a distributed load, while for the concentrated load it is necessary to consider the bending actions.
Some hypotheses must be considered, for instance, the shell is supposed homogeneous and isotropic, the thickness of the shell is relatively small compared to the radius of curvature, the bending strains are negligible and the deflection of the shell due to the applied loads is small. (Jawad, 1994)
There are different equations defined, the first equations can be obtained through the summation of the forces parallel to the tangent of the meridian (the shell is defined as a part of the meridian obtain rotating a plane curve about one axis), the second equation is obtained summing forces in the direction of the parallel circles, and the last equation is obtained considering the tangential stresses in the direction of the parallel circles.

3.2 Shell shapes

Shell structures have an important role in architecture and engineering, because of their possibility to be extended on a large span without any columns, and the particular shapes which can be obtained.

Different shell geometries can be designed, with relative advantages and drawbacks. In the definition of shell geometry, there are a lot of inspirations, for example, a shell could be inspired by nature, or simply obtained through the form-finding process as functions of the applied load.

There have been different examples of shell structures in time, the first one is considered the Hooke chain, which is based on the inversion of the chain shape to obtain an arch that is only subjected to compression.

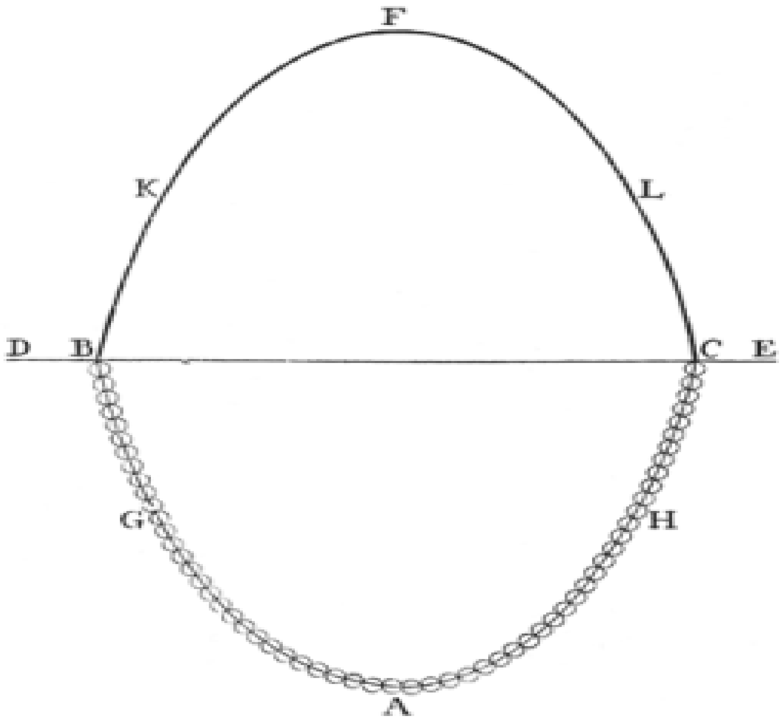

Figure 12 Hooke's arch (Jawad, 1994)

In the field of construction, the shells are widely used since the past centuries, there are many monuments and churches built with arches, domes, and vaults, with the presence of some openings which were important for light distribution inside the building. These shells are characterized by high strength in compression, while they are weaker in tension. These domes were only calculated to be able to withstand their weight, snow for instance was not considered by designers in the past.

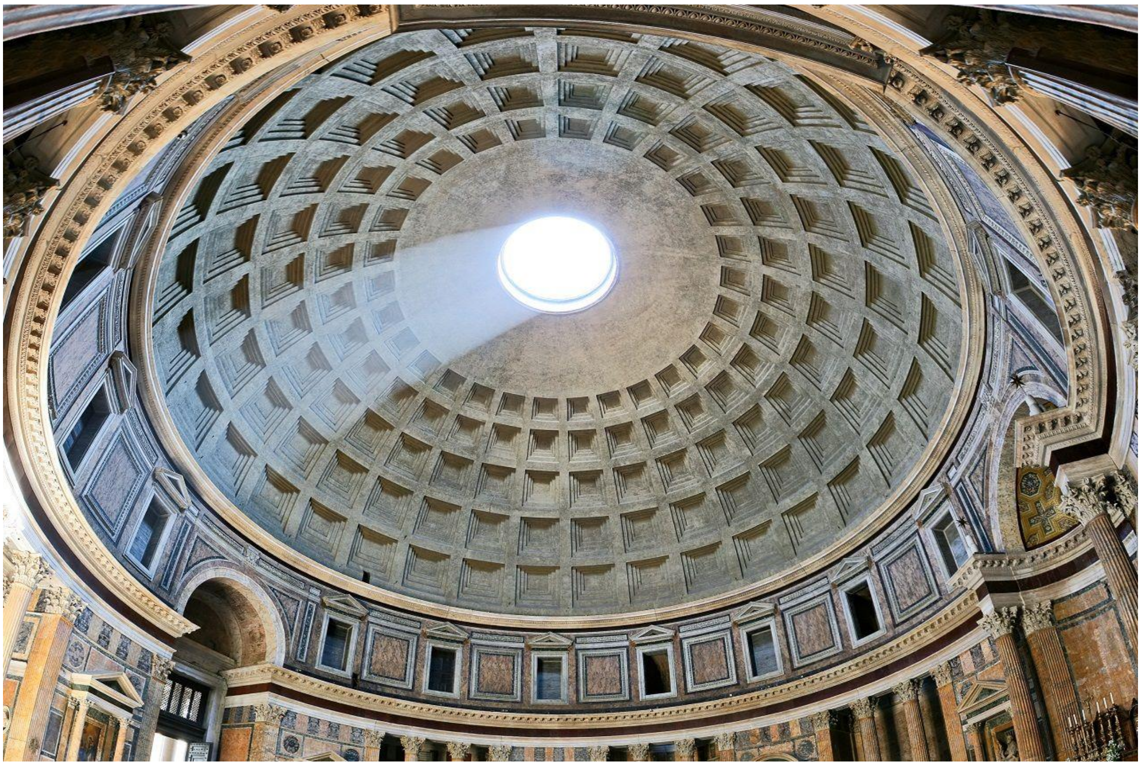

Figure 13 Pantheon Vault (Vistanet.it, s.d.)

Moving towards our days, the designers found out that shells can be applied even to more complex structures, this is also due to the versatility of concrete.

3.3 Gridshells

A shell is a construction that carries the loads through membrane forces and/or in-plane stresses, while usually considering a beam, it carries the load through its bending and/or shear forces. (Mork & Dyvik, 2015)
Therefore, the mechanical behavior of shells is very different from the monodimensional objects, it could be considered even way more complex because a lot of in-plane and out-of-plane stresses should be considered.
However, while the shell is a surface, the gridshell is a grid that acts as a shell, so each element composing this grid can be considered as only subjected to membrane or axial actions, which means that, if the result after calculation provides a high bending moment, the structure is not working as a shell, therefore there might be mistakes in the form-finding.

The reason for the introduction of gridshells is mainly related to better use of the spaces, with the reduction of the structural elements necessary. Also, the fact that gridshells can be curved allows the designer to obtain a more characteristic,

However, the gridshells cannot have any shape, in fact, according to the assumed shapes there are different forces and actions to consider, therefore it is crucial to have a collaboration between the architect and the engineer already during the design stage, to reduce the time and the iterations necessary to obtain the final shape.

Some examples of shapes can be defined, and they can be distinguished in good and bad examples, in general, to have a shell behavior there should not be flat areas, in fact in the case flat areas are present, the structure goes back to the traditional beam or flexural behavior.

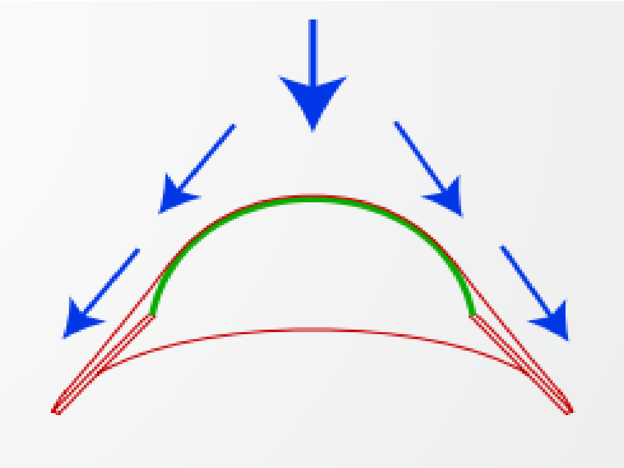

Figure 14 Tall arc (Gridshell manual, Authors: Steinar Hillersøy Dyvik, John Haddal Mork)

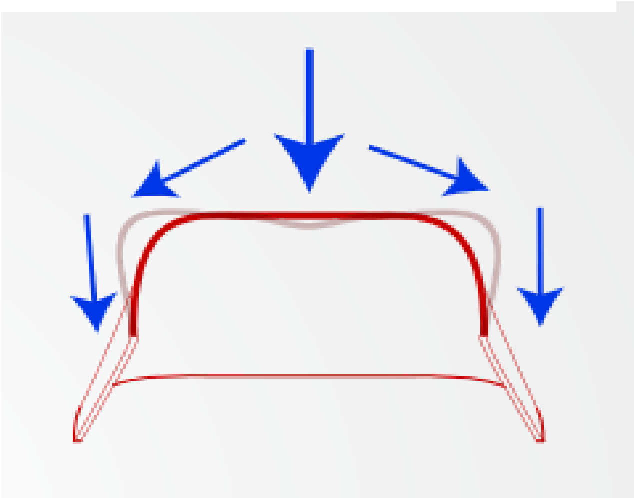

Figure 15 Flat top (Gridshell manual, Authors: Steinar Hillersøy Dyvik, John Haddal Mork)

3.3.1 Reference curve of the gridshell

Also, different curves of gridshells can be defined, and while the circle and the parabel are easier to draw and realize, the best reference curve is the catenary, which is the curve obtained due to gravity considering supports only at the start and end of it.

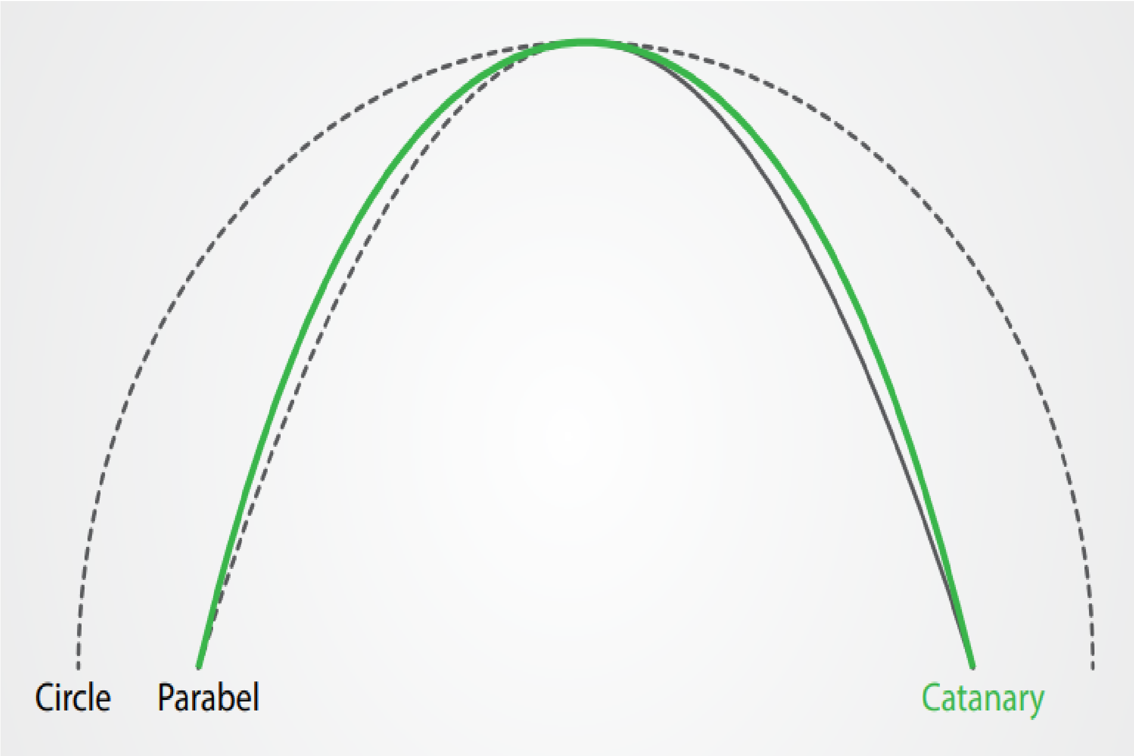

Figure 16 Reference curves (Mork & Dyvik, 2015)

3.3.2 Grid layout

Another important aspect is the grid layout, two main possibilities can be considered, an orthogonal or a diagonal grid. In general, the diagonal pattern is used only for short spans, because the study of the bending behavior of this grid becomes very complex.

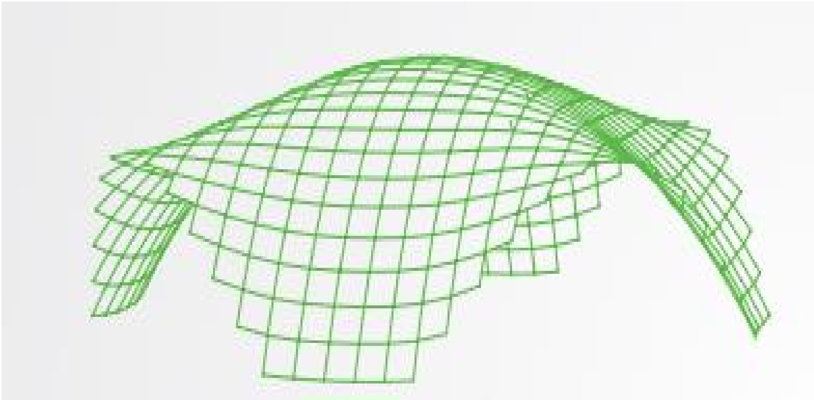

Figure 17 Orthogonal grid (Gridshell manual, Authors: Steinar Hillersøy Dyvik, John Haddal Mork)

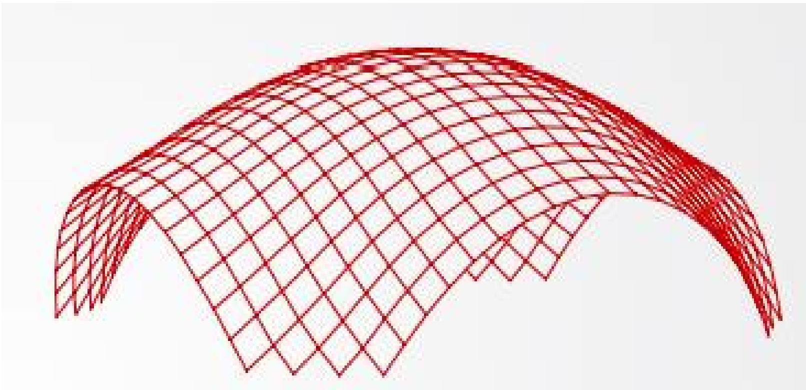

Figure 18 Diagonal grid (Mork & Dyvik, 2015)

The choice of the grid pattern is a function also of the shape of the object, for example in the case in which the slope between a point of the grid and the support is high, it is better to use a diagonal grid because it can handle a higher curvature.

Another important issue is that the size of the components of gridshell should be fixed, in this way it is easier to do a parametric design, since the number of necessary parameters which are reduced, and it is easier to study the mechanical behavior too.

3.3.3 Examples of Gridshells

In the world, there are several examples of gridshells, used for different purposes and characterized by different materials.

Most of the time the gridshell is used to design roofs or cover structures. The first object that can be considered is present in the Weald and Downlands Museum, and it has the function of a conservation center. It is the first timber gridshell built in the United Kingdom, and before its construction, to analyze the possible behavior of gridshell, and select the right type of timber, a scaled model was built.

Figure 19 Weald and Downland gridshell skeleton (Structurae, s.d.)

This structure has been covered with polycarbonate and wood panels to reach the final result.

Figure 20 Weald and Downland gridshell (Structurae, s.d.)

Another example of gridshell very different from the previous one is the Carioca Wave, located at the entry of the "Casa Shopping" commercial center in Rio de Janeiro, Brazil. It is a gridshell designed by Nir Sivan Architects, and it has been built as a self-standing structure, therefore there are no additional columns. In this case, the gridshell elements are made of steel, then the empty areas are characterized by the presence of glass panels.

Figure 21 Carioca wave (Nir Sivan Architects, s.d.)

Other examples can be found seeing the railway station around the world, as the Canary Wharf railway station in London, this is a multistorey metropolitan station immersed in the river Thames, or the Elbbrucken station, which is also an underground station, located in Hamburg (Germany).

The Canary Wharf station was opened in 1999, and it’s the busiest underground station outside the center of London (located in the Zone 2 of the United Kingdom capital). There are 43 million passengers per year.

The Elbrukken station is a public transit terminal station in Hamburg (Germany), it has two elevated stops, one for the underground and one for the suburban train. It was opened in December 2018, with a little delay compared to the initial program due to problems in the excavations, the final cost was about 145 million euros.

Figure 22 Canary Wharf London (Schmid Schrauben hainfield, s.d.)

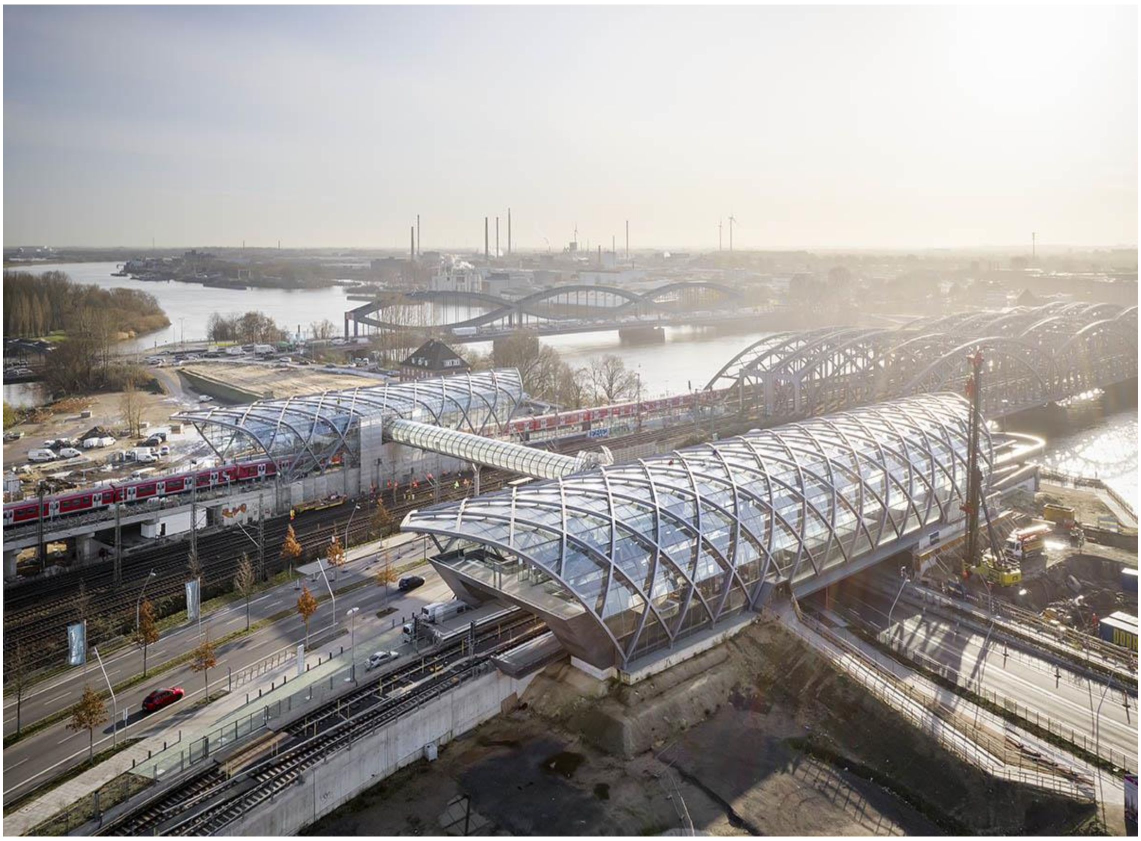

Figure 23 Elbbrucken Station (World Architecture Community, s.d.)

Other examples can be found in exposition areas, for instance in Milan at Fiera Rho, or to cover some commercial areas, such as the Zlote Tarasy Golden Hall in Warsaw, Poland.

The cover of Fiera Rho is one the biggest examples of freeform architecture with steel and glass in the world, it was designed by the architect Massimiliano Fuksas, and it covers a gallery of about 35000 m^2, also due to the presence of 180 parametric columns to sustain the whole cover.

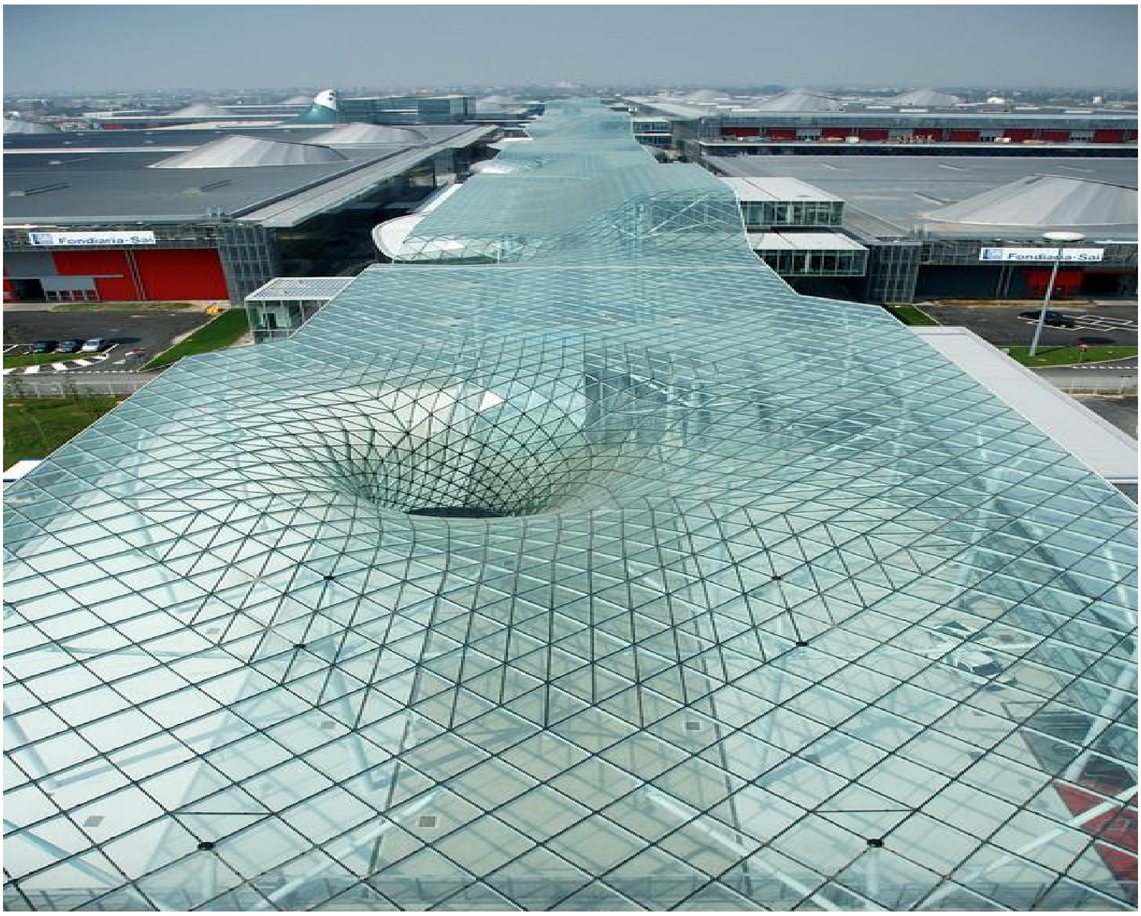

Figure 24 Fiera Rho gallery roof (Divisare, s.d.)

Zlote Tarasy gridshell, has been built to cover the area outside of the Zlote Tarasy (Golden Terrace) mall, and it's been designed by The Jerde Partnership together with Arup, it can be considered like a cloth draped over seven spheres (Anon., s.d.)It covers an area of 10000 m^2.

Figure 25 Zloty Tarasy gridshell (Wagner biro steel and grass, s.d.)

4 Structural Optimization

The optimization process allows the user to achieve the best results with the most reduced use of material, therefore it is based on the concept of effectiveness of the elements adopted during the design stage.

The necessity of the reduction of the use of materials is mainly related to economic factors, the target of designers is to obtain a performing structure with a lowered cost.

The design stage is usually divided into four steps that include the formulation of functional requirements, the conceptual design stage, optimization, and detailing. (Kirsch, 1993)

The general formulation of an optimization process is:

$$\text{Minimize } f_0(x)$$
$$\text{Subject to } f_i(z) \leq 0, i \in (1, \dots, m)$$
$$h_i(z) = 0, i \in (1, \dots, p)$$
$$||z - x||_2 \leq R$$

(Boyd & Vandenberghe, 2004)

where $f_i(z) \leq 0$ is the inequality constraint, and $h_i(z) = 0$ is the equality constraint.

Among these four stages, an iterative procedure, based on the results of the structural check, is applied, ending with the best solution possible, which will be the configuration in which the elements will work at 85-90% of their maximum possibilities according to the current structural code. To simplify and accelerate this procedure, some automated optimization procedures can be applied.

The available optimization methods are divided into two main categories, the Analytical methods, and the Numerical methods.

Analytical methods are usually employing the mathematical theory of calculus, in studies of optimal layouts or geometrical forms of simple structural elements (Kirsch, Structural Optimization). These methods are preferred for studying single elements; therefore, they cannot be applied to the integral structure.

Numerical methods are based on mathematical programming, and practice using numerical methods an optimized solution can be obtained automatically through an

iterative method, implemented starting from an attempted solution based on theory and experience until the optimum is reached.
The numerical methods have some advantages, there is a reduction in design time, a larger number of variables and constraints can be considered, and the possibility to obtain even unexpected results.
On the other hand, there are also some drawbacks, the quality of the result is based on the starting model, therefore an inaccurate starting model can lead to a solution characterized by no physical meaning; moreover, it is not guaranteed that the global optimum design will be obtained.

4.1 History

In the 1960s there are the first evidence of the adoption of structural optimization. In this period there were three main categories of optimization methods based, respectively, on the least weight of highly idealized frames, the weight-strength analysis, and the plastic analysis.
However, analytical optimization was developed for the first time in 1890, with the work of Maxwell, followed by Mitchell in 1904. Their works provided theoretical lower bounds on the weight of trusses. (Kirsch, 1993)
Furthermore, there were already some applications of structural optimization in the design of aircraft during World War II, followed by the development of the first numerical methods in the 15th century, which happened together with the release of the first computers.
However, as stated before, only in 1960 there is the first comprehensive statement of the use of mathematical programming, provided by Schmit, which constituted a milestone in the structural optimization history. Only since the 1980s, optimization become a broadly applied philosophy in the engineering field.
Mathematical programming techniques were considered effective tools for civil, aeronautical, and aerospace applications.
In the final years of the 17th century, another approach was introduced, called Optimality Criteria, presented by Prager and Venkayya respectively in the analytical and numerical methods. These techniques were easier to understand and apply, nevertheless, OC became the most effective design too.

However, while MP had a strong theoretical basis, OC had no clear theoretical support, even if it offered a solution for many design cases.

During the years OC and MP have been developed, but in recent years optimization became an important topic for research in civil engineering, leading to different approaches to its use, which are also possible with the improvement in the software and programming skills.

4.2 Design variables

A structural system is described by a set of data divided into assigned quantities, that usually coincide with the parameters defined during the automated design process and do not change the optimization process, and unknown quantities will constitute the design variables.

The design variables can represent mechanical or physical properties of the material, the topology of the structure, the geometry of the structure, or the cross-sections of the considered elements. They can be divided, from the mathematical point of view, into continuous and discrete design variables.

Material design variables are discrete since conventional materials are usually characterized by discrete properties. If there is only a limited number of materials that can be used it is more appropriate to perform the optimization for each material and compare the results.

The topology of the structure can be optimized automatically, using continuous design variables, however, there is the possibility that using a continuous set of data the global optima is not reached.

Geometrical design variables can represent the coordinates of a point in a frame.

The cross-sectional design variables are usually represented by the dimensions of the cross-sections, the moment of inertia, or the cross-sectional area of a certain element.

4.3 Constraints

Not all the designs are adequate for structural optimization, the ones that meet all the requirements are defined as a feasible design. The restrictions that must be satisfied to produce a feasible design are called constraints. (Kirsch, 1993)

The constraints can be divided into constraints imposed on the design variables reducing their range (called technological constraints), and constraints that derive from behavior requirements (called behavior constraints).
From a mathematical point of view, all the constraints are expressed through a certain number of inequalities and a certain number of equalities.
The constraints can be linear or non-linear functions of the design variables. (Kirsch, 1993)
The most typical constraints regard the displacements and the stresses, however usually there are also side constraints to consider.

4.4 Objective function

Usually, there are many feasible designs, it is important to find the best one by defining a function of the variables to compare the different design alternatives. The objective function is the function whose least value is sought in an optimization procedure. It is usually a non-linear function and may represent the weight, the cost of the structure, or other criteria. (Kirsch, 1993)
The selection of the objective function is one of the most important tasks in the design process, and its mathematical formulation can be very difficult.
In general, the objective function represents the most important property of a design, therefore usually it is represented by the weight.
Another approach could be to consider as the objective function the cost function, composed by the cost of materials, transportation, fabrication, and other costs such as maintenance.
In some design problems, a multicriteria objective function can also be used, however, working with multicriteria functions is complex and should be avoided.

4.5 Optimization methods

The methods regarding the mathematical programming can be used to solve a wide variety of problems including complex structural systems subject to different failure modes in each load condition, general design variables representing cross-sectional dimensions, geometry, and topology of a structure, various constraints on the structural behavior, a general objective function representing the weight or the cost of the structure. (Kirsch, 1993)

The MP methods will find the optima where there is a unique global optimum, however, there are problems in which there are also local relative optima, this happens due to the nature of the constraints and the objective function. In this case, to avoid any mistake the optimization should be repeated many times until there is enough confidence that the optima obtained are the global ones.

Different methods can be used, however, none of these methods can solve all the optimization problems, and they are not very effective in the case of large structures.

4.6 Optimization concepts

The first important concept is the definition of unconstrained and constrained minima.

A point X* is a relative minimum of the function f(X) if there is a region containing X* in its interior, such that $f(X^*) \leq f(X)$. (Kirsch, 1993)

At the optimum, the differential change in the objective function f(X), in terms of differential change in X must vanish, this is valid both in the unconstrained case, but also in the case in which a function is subjected to equality constraints. (Kirsch, 1993)

4.6.1 Unconstrained minimization

In optimization, the target is to obtain the minimization of the function.

Considering the unconstrained minimization, there are two cases:

- **Minimization along a line**, the target is to find a minimum of a function with a single variable, which is a problem at the base of many techniques. There are different methods to solve this issue, for example, the golden section method and the polynomial fitting method. In general, in both cases the function is considered as a unimodal function, therefore, if a minimum exists, it is unique in the considered interval.
- **Minimization of functions of several variables**, in this case, most optimization methods can obtain only the local minimum, unless the considered function is convex. The methods used for unconstrained minimization, which is an argument

still subjected to research, are divided into direct search methods, gradient methods, and Newton and Quasi-Newton methods.

4.6.2 Constrained Minimization

Considering constrained minimization problems, they can be solved using:

- **Linear programming,** it is an MP method, its special characteristic is that all constraints and objective functions are expressed in linear terms of the variables. (Kirsch, 1993)
  Only a small part of the structural design problems can be formulated as Linear Programming, however, this method is widely used since the exact global optima are reached in a finite number of steps, computer programs of LP are better, and also some practical non-linear problems can be simplified to linear problems and solved with Linear Programming algorithms.
- **Non-linear programming,** also in this case no single non-linear programming method can solve all constrained optimization problems, the effectiveness of the optimization method depends on the algorithm and the software. (Kirsch, 1993)
  When choosing a method three important factors should be considered, the efficiency of the method in large scale problems, the reliability of the method, and the ease of use. There is no right method, but it depends on the problem under study. However, most of the methods are based on four steps including the determination of the set of active constraints in the design, selection of a search direction in the design space, calculation of how far to go in that direction, and convergence check. Constrained optimization is a field of active research of new methods, however, the methods used are divided into direct and indirect. While the direct method deals with the constrained formulation as it is, the indirect method converts the problem into an unconstrained problem.

4.7 Form finding

Form finding is a process applied to define the surface configuration of the considered shell structure, considering the constraints and the effect of gravity action.

A form-passive structure is characterized by the capability of maintaining its shape when subjected to varying loads. These shells are only subjected to membrane actions; therefore, these elements are not subjected to bending. There are different form-finding numerical methods, which can be used to define the final shape of the shell, the Transient Stiffness Method, the Force Density Method, and the Dynamic Relaxation Method.

4.7.1 Transient Stiffness Method

The transient stiffness method is based on the theory which considers the linear dependence of displacements upon forces applied to the structure, called the small displacement theory. (Lewis, 2012)

The first step requires the discretization of the surface, with the definition of the points.

The form-finding starts assuming a certain surface configuration and an imposed tension matrix, characterized by the pre-tensioning forces. Solving the tensional matrix in the x, y, and z-direction, the resultant of internal forces would be different from 0, so the system will be not in equilibrium. The displacement will be computed as the product between the inverse matrix of stiffness and the resultant of the internal forces matrix.

$$(\delta) = K^{-1}(R)$$

Nevertheless, there is a problem with the resultant of internal forces, if the values are too large, also the displacements will be, therefore, it is necessary to apply an iterative procedure to obtain the best solution possible, characterized by the smallest values of displacements.

The method is called the "Transient stiffness method", because, during the iterations, the stiffness matrix changes, even if sometimes it is better to keep the stiffness constant for a few iterations.

4.7.2 Force Density Method

The "Force Density Method" uses a surface, that is discretized as a system of branches. (Lewis, 2012)

A simple branch is composed of four elements connected to a central node, the target is to obtain the equilibrium of tensile actions in both directions, but the system under study is not linear.

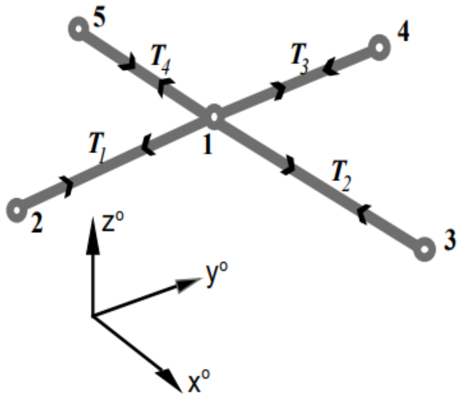

Figure 26 A branch of elements (Lewis, 2012)

To obtain a linear system it is necessary to introduce a quantity, defined as density qm, obtained by dividing the tensions T_m by the lengths L_m, which will represent the new unknowns of the system.

$$q_m = \frac{T_m}{L_m}$$

Known the density values, the x and y coordinates of the node can be calculated.
In the case of more complex problems, it is necessary to introduce the connectivity matrix, to compute the lengths.
The internal reactions in each point are computed with the product between the force densities and the lengths, and the summation of the internal reactions in each direction should be in equilibrium with the load.

4.7.3 Dynamic relaxation method

The "Dynamic Relaxation Method" is different from the previous two methods since it is not dependent on the stiffness matrix. This method considers lumped (or concentrated) masses applied at the nodes of the surface. (Lewis, 2012)
Also, in this case, there is an iterative process applied with the reduction of the out-of-balance forces, until they are close to zero.

Originally this method was characterized by viscous dampers, therefore the whole system was governed by the equation of motion for a discretized system.

$$P_{ji} = [\sum K\delta]_{ji} + M_{ji}\ddot{\delta}_{ji} + C\dot{\delta}_{ji}$$

Where P_{ij} represents the external load, K the stiffness matrix, M the mass matrix, and C the coefficient of viscous damping. These coefficients are multiplied respectively times the displacement, the velocity, and the acceleration.

A more recent approach is based on kinetic damping. In this case, the viscous damping coefficient is supposed to equal zero, and the system of oscillating masses is brought to rest after a certain number of iterations.

4.7.4 Benchmark 1: Form Finding of a 1D element

The first step to understanding the approach used by Kangaroo for form-finding is to consider a simple one-dimensional geometry, for instance, a line. The target of this procedure is to understand what the best shape is assumed by the line because of gravity. Therefore, using grasshopper a line is defined, then it is divided into different segments, then using the List Item component and the Cull index component, it is possible to ensure that the first and last points are not subjected to the gravity force, so they are fixed.

The gravity load is applied through a unit z vector applied to the component Point Load, making sure the load is pointed downwards.

Moving to the Kangaroo components, the Kangaroo anchor should be applied to the output of the List Item component, and the Length component is connected to the points. In this way, there is a dependence on the strength of the material and the length of segments.

Then the Load, the Length, and the Anchor components are connected to the Kangaroo solver, to get the result.

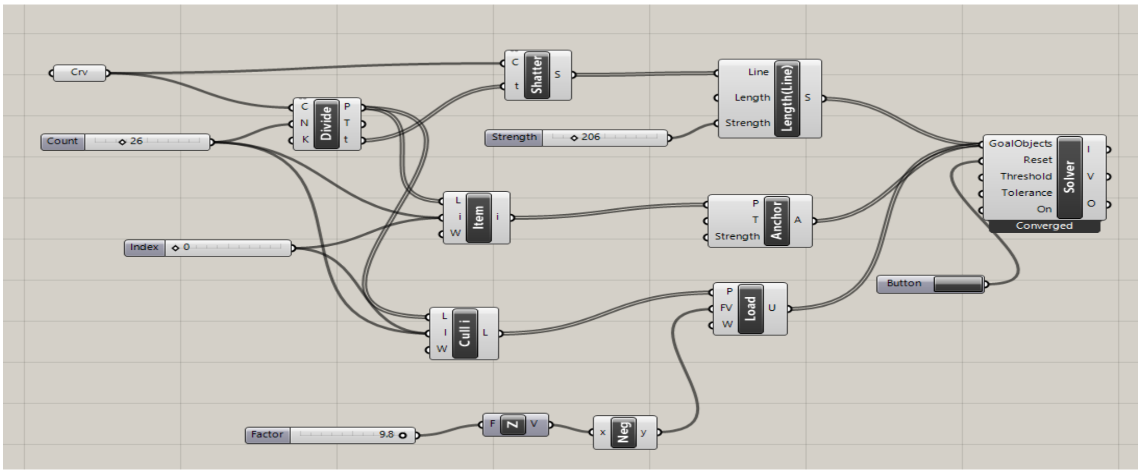

Figure 27 Form finding definition of a line

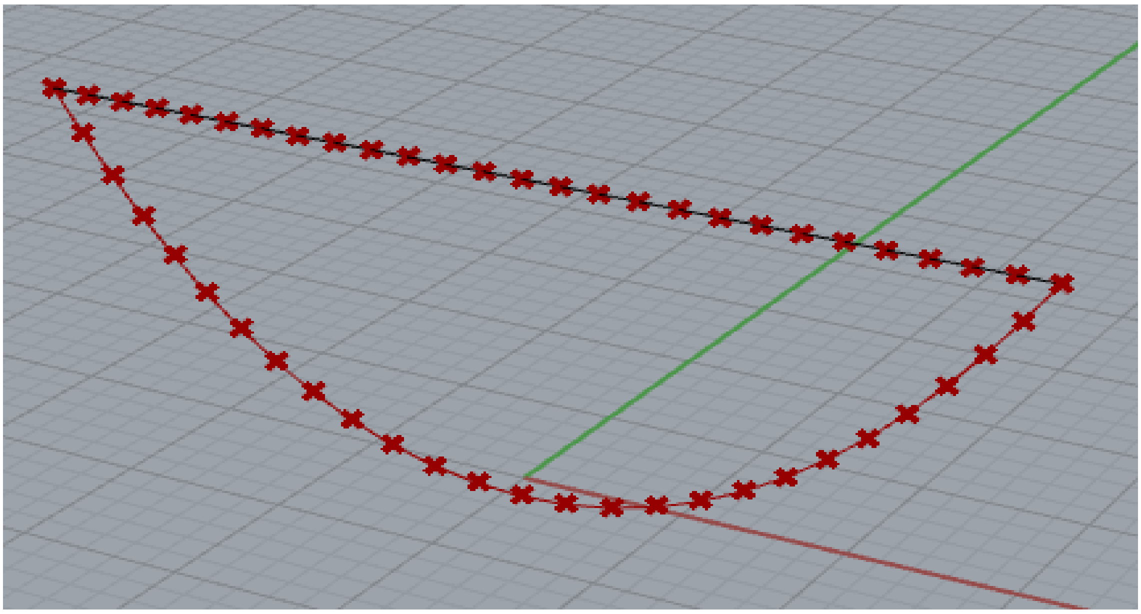

Figure 28 Graphical representation of the form-finding

Changing the strength of the material, it is possible to appreciate a variation of the shape obtained. As the strength increases, the curvature decreases, since the material will be stiffer, so less subjected to deformation.

4.7.5 Benchmark 2: Form Finding of a rectangular surface

In the second example, there is the application of the form-finding procedure starting from a surface drawn down in Rhino.

The surface is imported in Grasshopper with the relative component, then a mesh is defined, splitting the surface in a certain number of segments in U and V direction, defined with number sliders.

Also, in this case, the target is to define the line length in Kangaroo, to do that, the edges of the surface should be obtained with the Meshed Edges component, where E1 output represents the outer lines, and E2 the inner lines.

Afterward, it is necessary to anchor the corners with the component from the Kangaroo plug-in, but first, they should be obtained with the Meshed Corners component.

Finally, the point load should be applied, but first, we want to make sure that all the segments are carrying the load, by using the component Deconstruct Mesh, the force is applied to the vertices. In the end, the component Grab will be added, and all the outputs will be connected to the Solver.

The Grab component allows the user by pressing Caps lock to move the structure with the mouse.

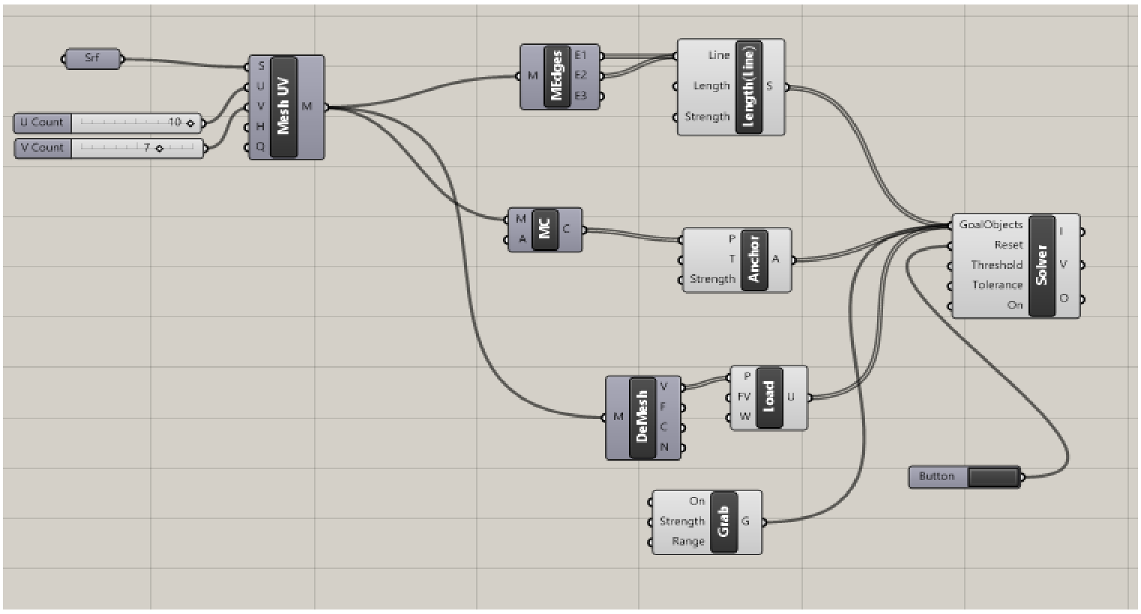

Figure 29 Form finding definition of a surface

After the running of the Kangaroo solver, the best shape of the tensile structure is finally obtained.

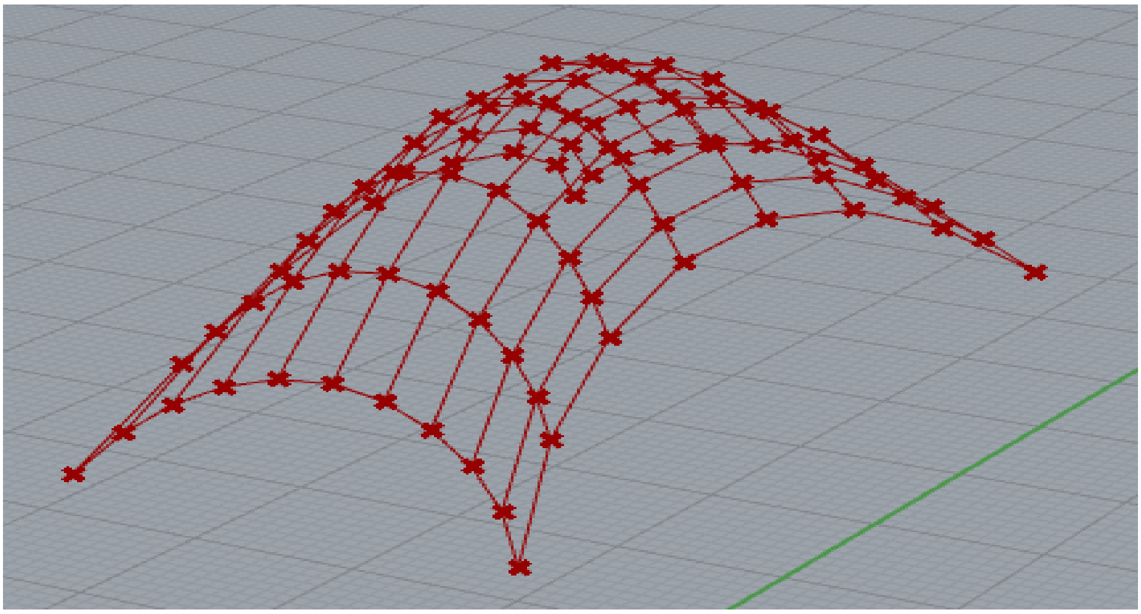

Figure 30 Graphical representation of the form-finding for a surface

4.8 Evolutionary multi-objective optimization

A multi-objective optimization problem involves several objective functions which can be minimized or maximized. The multi-objective optimization can be characterized by the presence of a variable number of constraints, passing to constrained multi-objective optimization, as in the case study.

One of the main differences between single-objective and multi-objective optimization is that in multi-objective optimization the objective functions constitute a multidimensional space called objective space (Deb, 2001)

In mathematical terms, a multi-objective optimization problem can be formulated as:

$$min_{x \in X}(f_1(x), f_2(x), \ldots . f_k(x))$$

Where the integer number k ≥2 is the number of objectives, and the set X is the feasible set of decision vectors. The feasible set is typically defined by some constraint functions. In addition, the objective function is often defined as:

$$f: X \to R^k$$

$$x \to \begin{pmatrix} f_1(x) \\ \cdot \\ \cdot \\ f_k(x) \end{pmatrix}$$

If some objective function is to be maximized, it is equivalent to minimize its negative or its inverse. Denoting $Y \subseteq R^k$, the image of X; $x^* \in X$ a feasible solution or feasible decision; and $z^* = f(x^*) \in R^k$ an objective vector or an outcome.

The optimal solutions in multi-objective optimization can be defined with the concepts of partial ordering and dominance.

A certain solution is dominant concerning another one if it is not worse in all objectives, and if it is slightly better in at least one objective.

The points not-dominated, part of the objective space, form a front called non-domination front, and all the points that are in this front are defined as the Pareto-optimal points or Pareto-optimal solutions.

In multi-objective optimization, there does not typically exist a feasible solution that minimizes all objective functions simultaneously. Therefore, attention is paid to Pareto optimal solutions; that is, solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives. In mathematical terms, a feasible solution $x_1 \in X$ is said to (Pareto) dominate another solution $x_2 \in X$:

- $\forall$ i $\in (1, \ldots, k), f_i(x_1) \leq f_i(x_2)$
- $\exists$ i $\in (1, \ldots, k), f_i(x_1) < f_i(x_2)$

  A solution $x^* \in X$ corresponding to the outcome $f(x^*)$ is called Pareto optimal if there does not exist another solution that dominates it. The set of Pareto optimal outcomes denoted X^*, is often called the Pareto front, Pareto frontier, or Pareto boundary.

Evolutionary multi-objective optimization algorithms follow two different goals:

- Find a set of solutions laying on the Pareto front
- Find a set of solutions diverse enough to represent the entire range of Pareto front

In the case of the study, the Pareto-optimal solutions are obtained directly with Octopus, a Grasshopper add-on.

5 FEM-Rhino Interaction

Rhinoceros is one of the most used software for parametric architecture, together with Dynamo. Practically this program allows the user to use a simplified coding language to obtain an algorithmic model the code used is a visual code, in which the commands are boxes characterized by pins for the connection of the different objects.
But Rhino is only a graphical software, therefore it cannot compute the stresses and displacements in the modeled structure, therefore it will be necessary the connection with some Finite Element Analysis (FEA) software, in this case, Karamba3D and Sofistik.
These FEA programs have a plugin that can be included in Rhino to obtain the results needed.

5.1 Finite Element Analysis

The finite element analysis is the simulation, with the help of adequate software, of a physical phenomenon. The method that is used for this analysis is the Finite Element Method (FEM), therefore the software used for this analysis is defined usually as FEM software or FEA software.
The basic idea in the finite element method is to find the solution to a complex problem by replacing it with a simpler one. Because of this approach, the solution that will be obtained is an approximated solution rather than an exact solution. (Rao, 2011)
The advantage of FEA is the speed in the computation of the results, in terms of stresses and internal actions, due to one or more physical phenomena (for example wind, dead load, snow, etc.). This velocity in the elaboration allows the engineer to do multiple simulations to define the best cross-sections and reduce the costs, applying a process of optimization.
This process was more difficult in the past since all the calculations were done by hand, or with the help of Excel, so for each edit that was done in the structure, all the elements should have been computed again, and this process required a lot of time and effort.

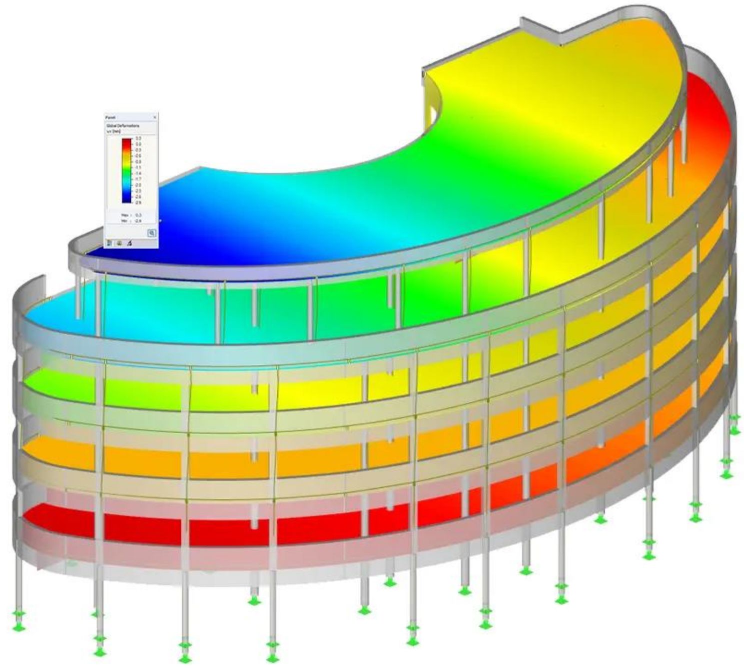

Figure 31 FE Analysis example (Dlubal, s.d.)

5.1.1 Principle of Finite Element Method

The principle on which the Finite Element Method is based is not physical, so it is not connected to the cause as the usual hand calculation, instead, it is based on a purely mathematical concept. Finite Element Analysis can be defined as a numerical method to solve partial differential equations in two or three variables.

In an FEA software, the object is divided into finite elements, which are relatively small elements with fixed dimensions, then for each of these elements, a mesh is applied, composed by even smaller elements forming a grid, finally the parts between two points of the mesh are calculated based on interpolation (which can be linear, quadratic, cubic). These small, interconnected subregions, called finite elements, are used to represent complex geometrical shapes. In each element, the most suitable approximate solution is assumed, and the conditions of an overall equilibrium of the structure are derived. (Rao, 2011)

Therefore, the finite element method does not always give the right solution the solution is a function of the input data, for instance, if wrong load conditions or wrong cross-sections are applied in the program, the result will be wrong. However, there is also the dependency from the mesh applied, because a function of the mesh refinement a different interpolation is applied, which leads to a different computational time and a different accuracy of the final solution. After defining all these possible errors, FEA should be used properly, and the results should never be taken as facts, therefore it is important to define if they have a sense from the physical point of view.

5.1.2 History

Even though the name Finite Element Method is quite recent, the concept was used in the past. Ancient mathematicians found the circumference of a circle approximating it by the perimeter of a polygon. (Rao, 2011)

Depending on the number of sides of the polygon the accuracy varies, in fact considering inscribed polygons, it is possible to see that as the number of sides increases the polygon will tend to the circle, therefore it can be approximated to a circle with a reduced error.

Shelback (1851) was the first one to discretize a circular surface into many triangles to find a differential equation to express that surface. Since the 1950s, some engineers in the aircraft industry worked on the development of approximate methods to prevent the stress-induced in the wing. In the same period, Argyris and Kelsey presented some papers containing some of the finite element ideas to solve problems of Structural Engineering. (Rao, 2011)

The name finite element was coined by Clough in 1960. In the beginning, the method was mainly based on physical concepts, however, with the definition of a mathematical basis, the method was spread and improved all around the world, also thanks to computer invention.

Afterward, FEM became a useful tool both for mathematicians, to solve linear and non-linear complex equations, and for engineers, to solve practical problems.

5.1.3 Applications

The Finite Element Method is used not only in the subject of structural mechanics, but it has been applied to solve several engineering problems, such as heat conduction, fluid dynamics, seepage flow, and electric and magnetic fields.
This method can be applied to different areas of engineering since it can be used for obtaining the solution of ordinary and partial differential equations. (Rao, 2011)
Therefore, the method can be used in structural engineering for the static analysis of elements such as truss, beams, shells, etc., in aerospace engineering for the static analysis of aircraft wings, in geomechanics for the analysis of excavations, retaining walls, and tunnels, and for instance in nuclear engineering an application could be the analysis of nuclear pressure vessels.

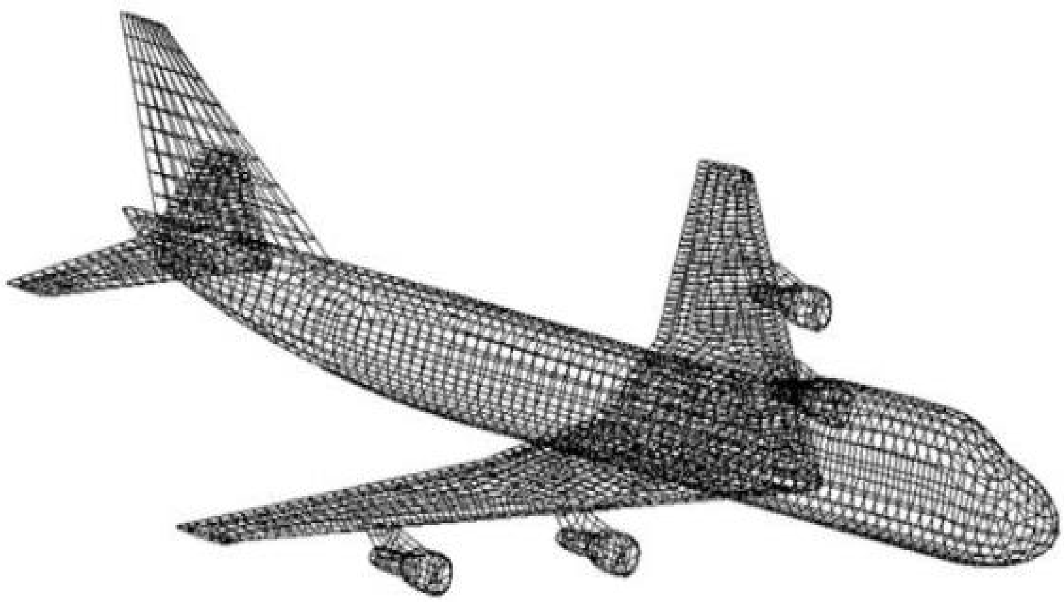

Figure 32 Discretization of an aircraft (Elisa.fi, s.d.)

However, even if the FEM is used in different applications, the steps of the method include always a discretization (a subdivision of a structure into discrete elements), the choice of the proper displacement model, obtaining the stiffness matrices and load vectors for each element, assemble element equations to obtain global equilibrium, and finally compute the results (displacements, strains, and stresses). (The Finite Element Method in Engineering – Singiresu S.Rao)

5.2 Benchmark on Connection Sofistik-Grasshopper

The target was to find a connection between the FEM software and the algorithmic modeling program, for that purpose, there is a plugin for Grasshopper provided by Sofistik, which allows working in the canvas of the parametric design software.
In this plug-in, different components allow us to define a text file or different text files containing all the structural information necessary for doing the computation with the FEA.

5.2.1 Pratt Truss

One of the most used truss systems is used as an example, the Pratt truss. This truss was first developed in 1844, but only with a combination of steel and timber elements, later it was made also totally of steel. This system is very spread in structures because, considering the vertical or dead loads as the most important and frequent ones, the diagonals will be subjected to tensile stresses, therefore they would work in their best condition, instead the vertical elements will be compressed.

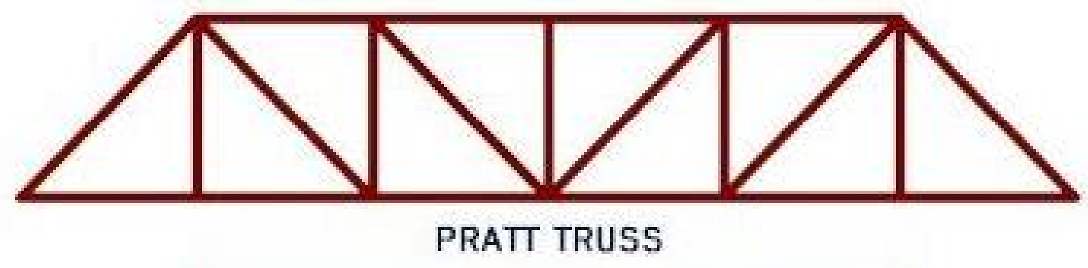

Figure 33 Pratt truss (Quora, s.d.)

This kind of structure was used in the past also for relatively long-span bridges, like the Governor's Bridge, a bridge located in Howie (Maryland).
It is a single-lane bridge, built-in 1912, it has been made completely of steel, and it connects the two sides of the river Patuxent, with a span of about 35 m.

Figure 34 Governor's Bridge (Quora, s.d.)

It is important to have diagonal members in tension because, for instance in steel, the strength in tension is larger than the strength in compression, also in tension, there are no problems as buckling, a phenomenon in which, with very high compressive actions, the element tends to deform, and it loses part of its strength.

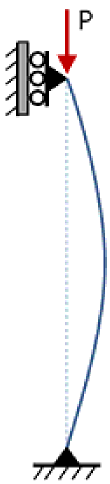

Figure 35 Flexural buckling due to compression (MechaniCalc, s.d.)

5.2.2 Geometry definition

The geometry is modeled completely in Grasshopper, so it will be parametric. The parameters that can vary in this structure are the length of the truss, the height of the vertical elements, and the number of segments.

The first step to build the geometry is to define a line between the starting point and the endpoint, then this line is subdivided into a certain number of segments, defining some points, that finally are moved upwards at the required height.

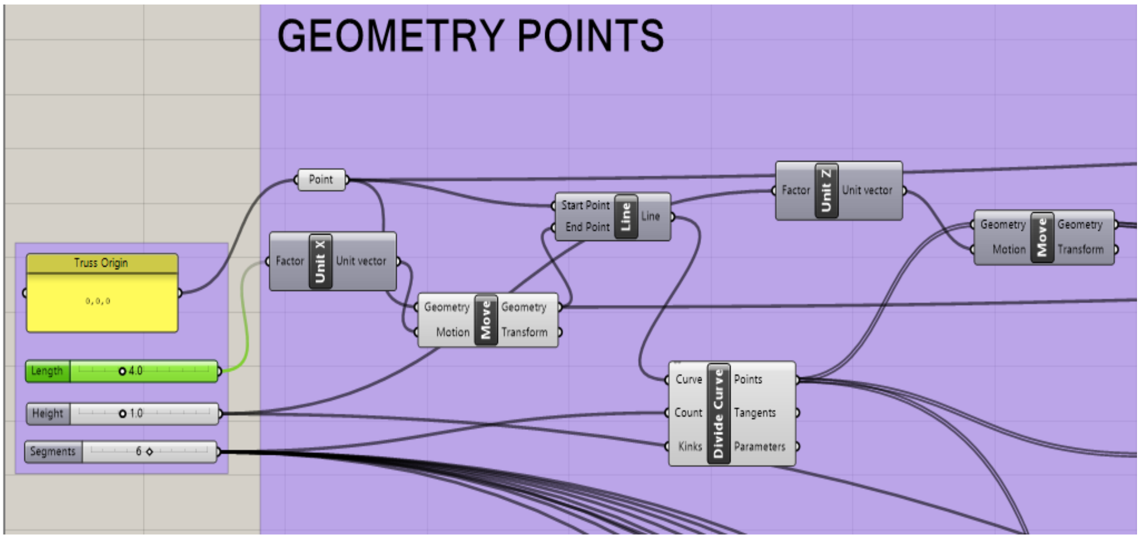

Figure 36 Geometry definition on Grasshopper

Once the points are defined, the elements are introduced, divided into horizontal elements, which include the top and bottom chords, vertical elements, and diagonals.

To generate these elements the line component is used, but the most important issue is to define the correct points to connect, this is achieved with the use of the Sublist component, which is a component that allows the user to obtain a reduced list starting from the total one. Between the inputs of this component, there is also the domain, which can be defined with the Construct domain component.

An example of code for computing these elements is represented with the diagonals, considering that their inclination is different on the external elements and internal ones. In the picture below, there is the definition of the diagonals on the left side only, then everything can be mirrored with the use of the Reverse list command, to obtain the same result on the other side.

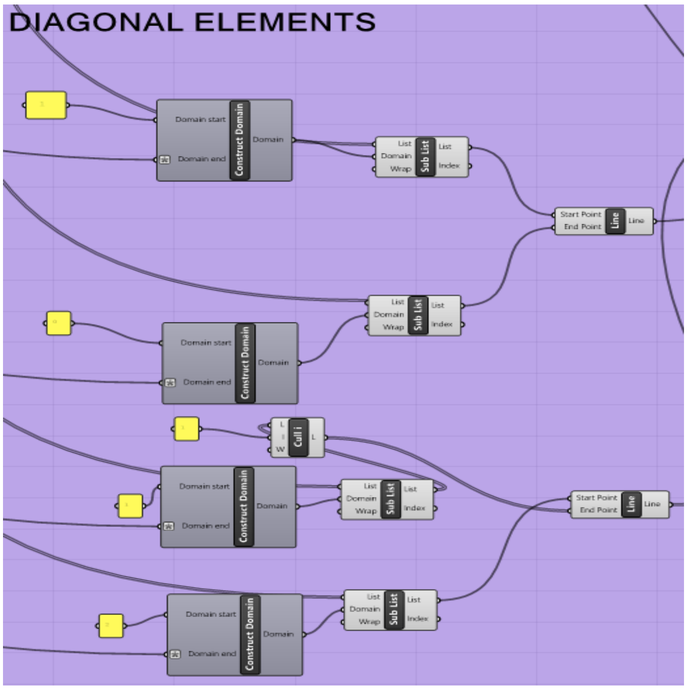

Figure 37 Diagonal elements definition

After all these steps the skeleton of the structure has been built, there will be a geometry constituted only byline elements.

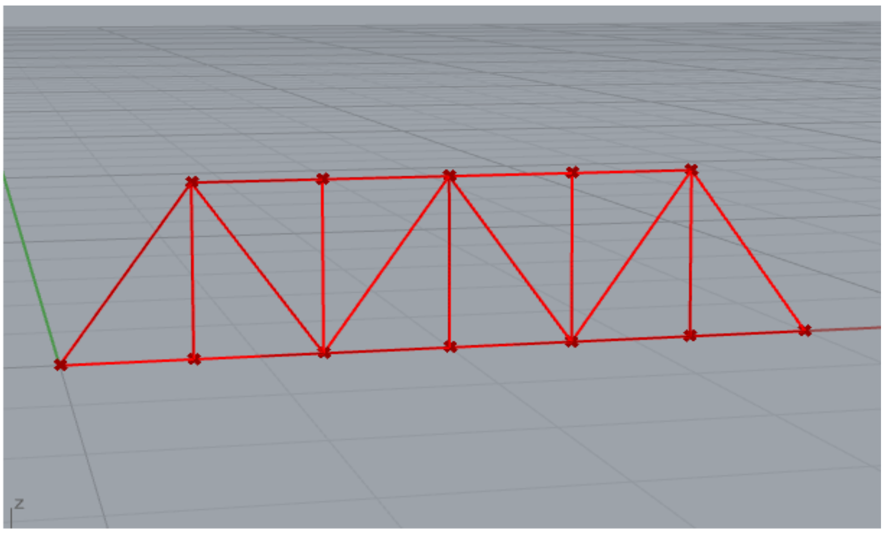

Figure 38 Final geometry

5.2.3 Design code and material

The previous was not different from the usual steps of projects in grasshopper, instead of moving to the connection with the Finite Element Analysis software, the first thing that can be defined is the reference design code to consider in the checks, which is the Eurocode 1993-2005. Then another important step required the choice of the material, which is a type 11 steel of class S235.

All this information is included in the Grasshopper canvas utilizing an Aqua component, to which a panel with all the data is connected, to obtain a text file.

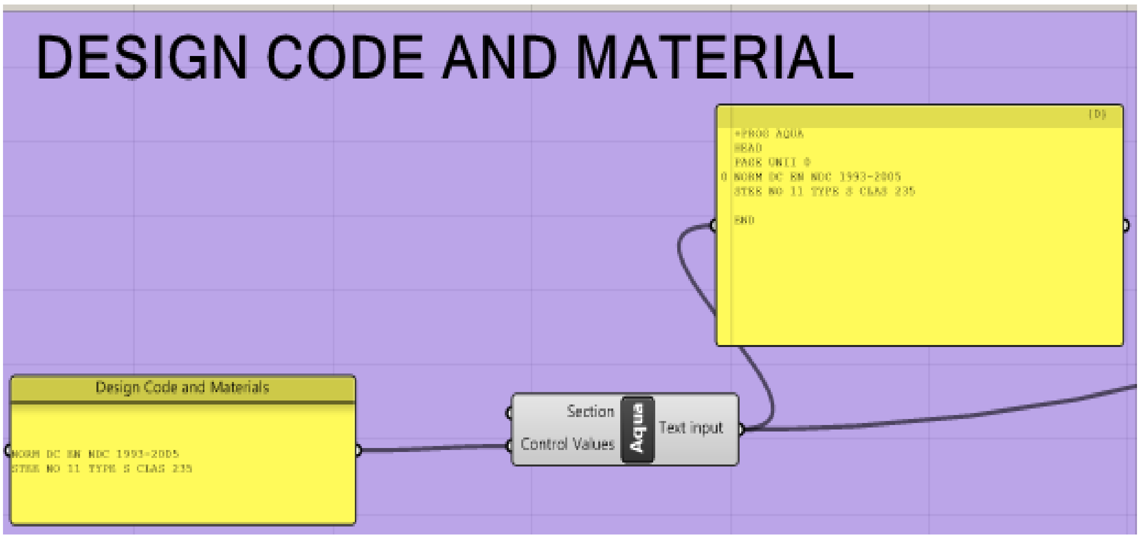

Figure 39 Design code and material definition

5.2.4 Sections

After the definition of the code for the verifications and the material, it is important to select and name the cross-sections that will be applied to the different elements.

This process is done using two different components in the Sofistik plugin present in Grasshopper, respectively “Section attributes” and “Read Section Library”, in the first one the name of the section and the material can be linked, the second component, instead, is important to select the cross-section between the ones provided by the software, which are usually standard cross-sections.

In the case under study, different cross-sections were considered for horizontal elements, vertical members, and diagonals. Furthermore, finally, everything is connected to an Aqua component to generate a text input.

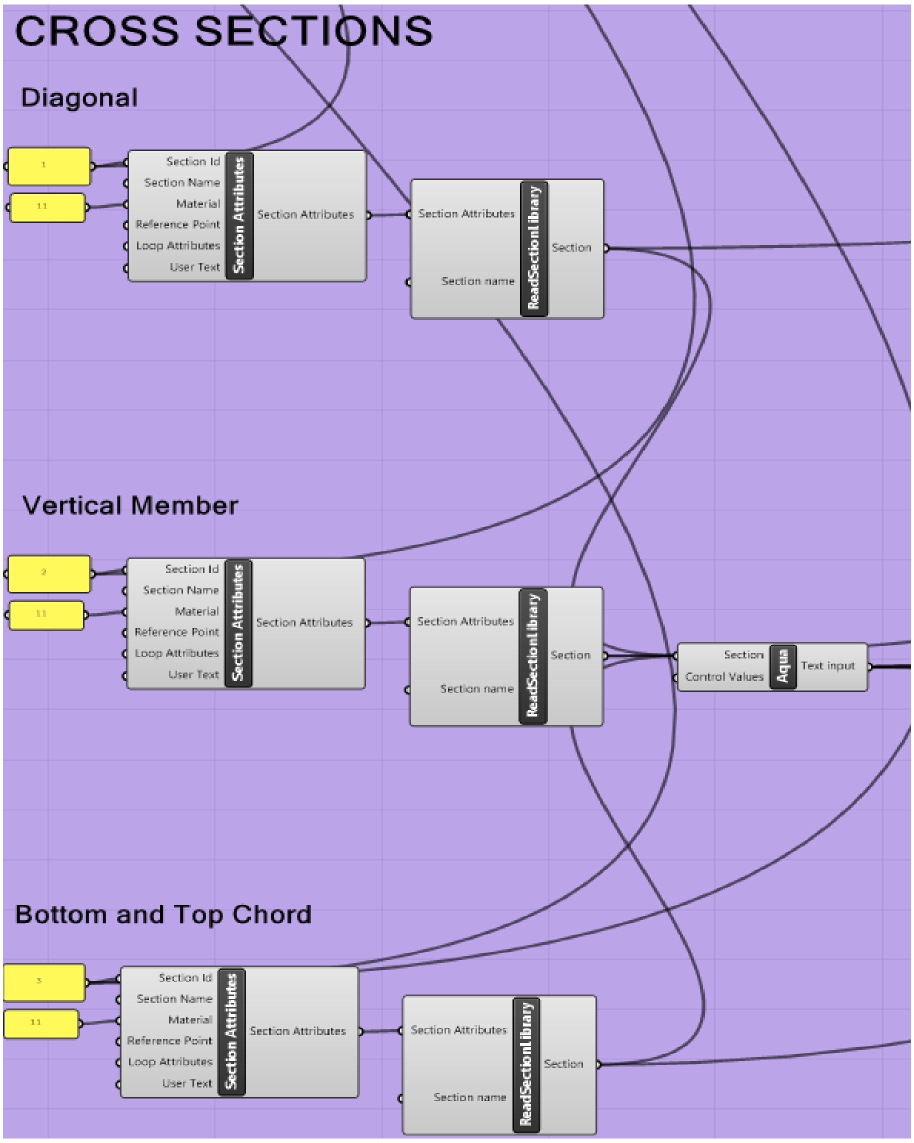

Figure 40 Cross-sections definition

5.2.5 Structural elements

Once all the sections are defined, they should be applied to the grasshopper curves that have been obtained from the definition of geometry. To achieve this result, the Structural Line component is used, and in this component, it is possible to connect the curve, the section, and the element type.

The curve will be taken from the previous geometry, the sections are obtained by connecting the panel with the number of sections defined before, and the element type is introduced with a panel, and it can be a truss element, which is a unidirectional element with zero bending stiffness, or a beam element, characterized by a bending moment together with the axial forces too.

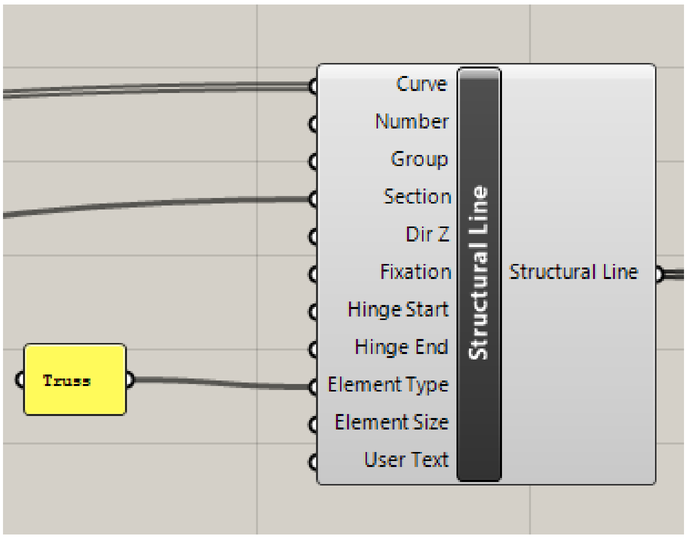

Figure 41 Structural lines definition

Moreover, to have all the structural data, it is important to define the restraints. The restraints are defined in Grasshopper with the introduction of the Structural point component, to which it is possible to connect the selected point for the application of the constraint and the fixation. In the case of fixation, there are text codes written with a panel that can be used, in the case of a fixed constraint, it is "PPMX", while if the target is to obtain a hinge it becomes "PYPZMX", in this way only the movement in the x-direction is allowed.

In the considered case, the beam is simply supported, therefore there will be a fixed node and a hinged node.

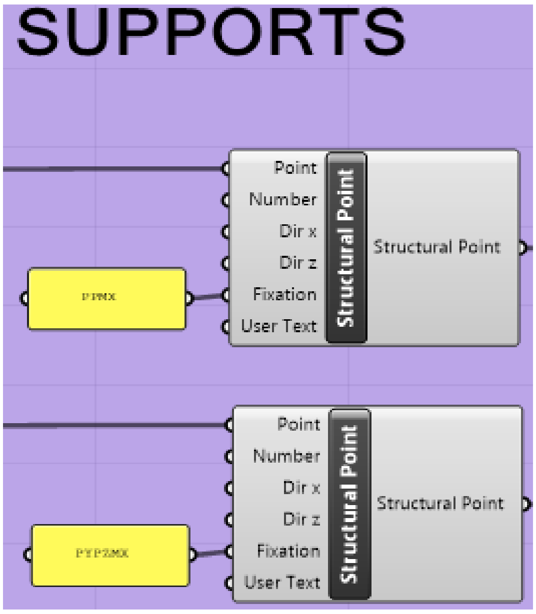

Figure 42 Supports definition

In the end, all the structural data is connected to the Sofimshc component, which generates the text input for the calculation.

5.2.6 Loads

The last input data to introduce is the load, which can be a point load, or a distributed load over an area or a line. In the studied case, the load is concentrated in the nodes, to obtain this result the Point load component is used, and the inputs of this component are the hosting point and the applied force, moment, or displacement. The force to apply is defined with a vector in the Z direction with the length equal to the applied load.

Another component, called Load case Attributes, is introduced to define the type of load applied and the id, this information is introduced through a panel.

Finally, all this data is connected to the Sofiload component, which generates the text input for Sofistik calculation.

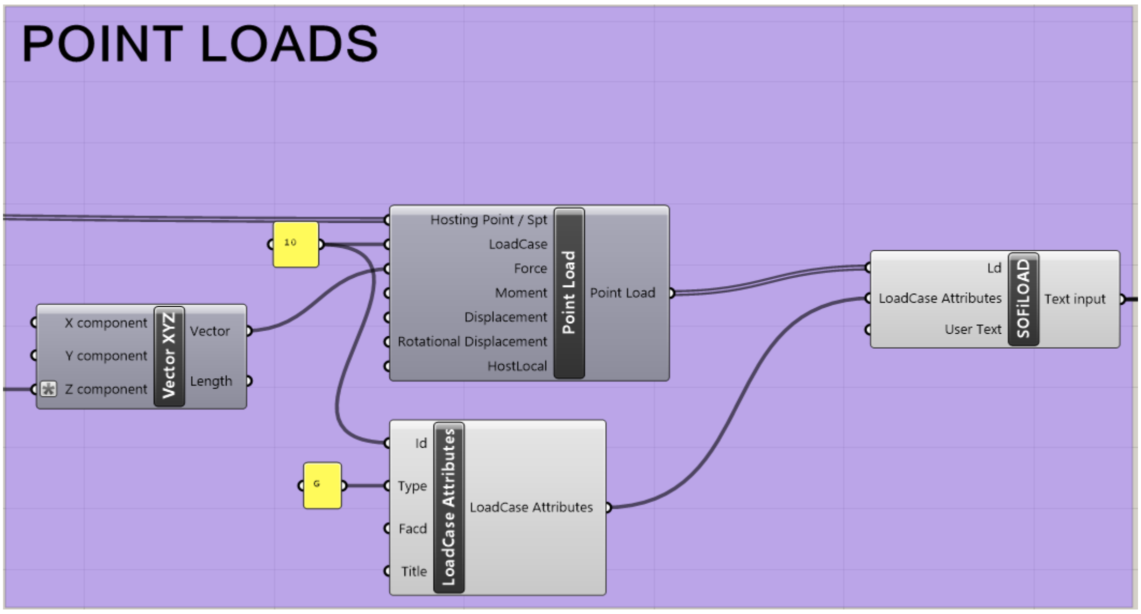

Figure 43 Loads definition

All the elements introduced can be visualized in the Rhino interface.

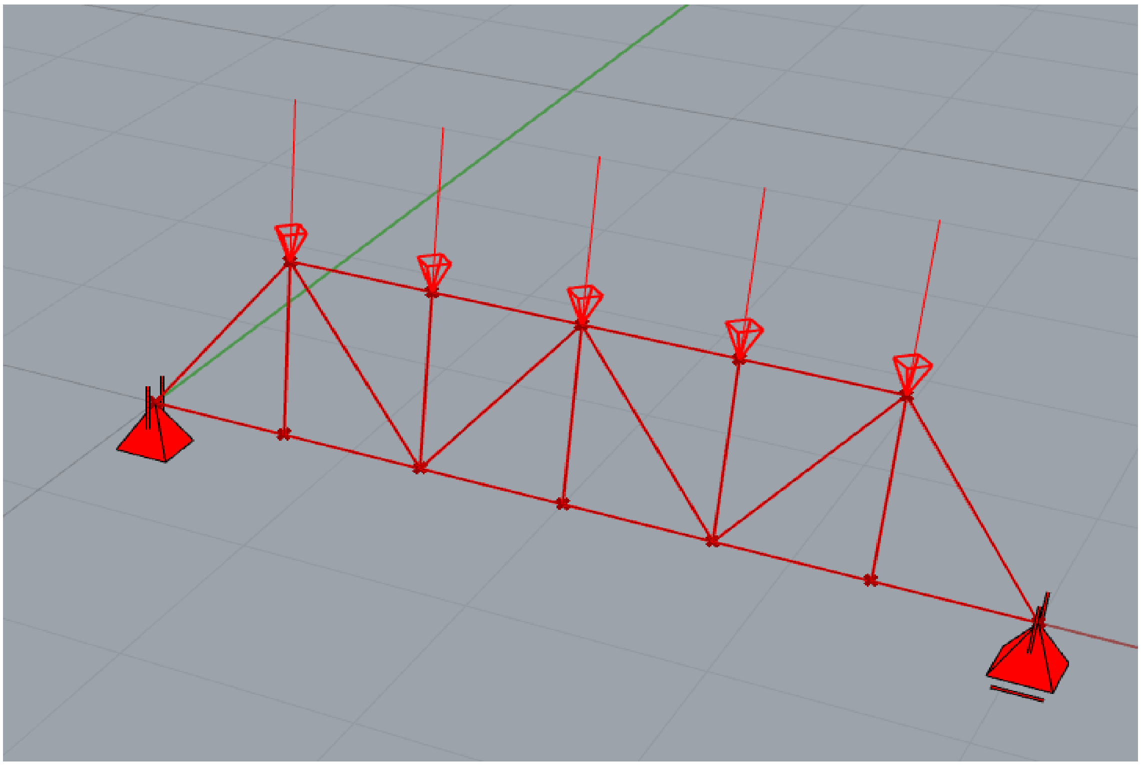

Figure 44 Loads graphical representation

5.2.7 Analysis and Results

The final step is to perform the analysis in Sofistik, which can be done in two different ways. The first approach requires the introduction of the Advance Solution Engine component, and the Sofistik Project component.

The ASE component has the function to load all the load cases considered.

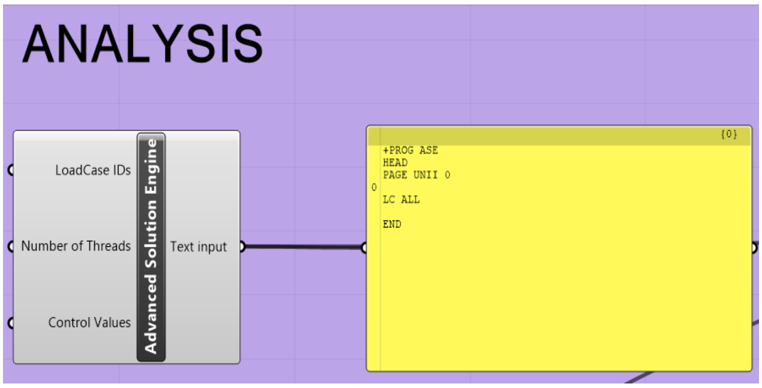

Figure 45 Analysis component

Instead, the Sofistik Project component is used for computing the results by clicking on the button "Calculate", and after introducing all the Sofistik data that was defined before and merging them following the right order. First, there should be the Design Code and Material, then the Cross-Sections, the Sofimshc data (including structural lines and points), the Point Loads, and finally the Analysis.

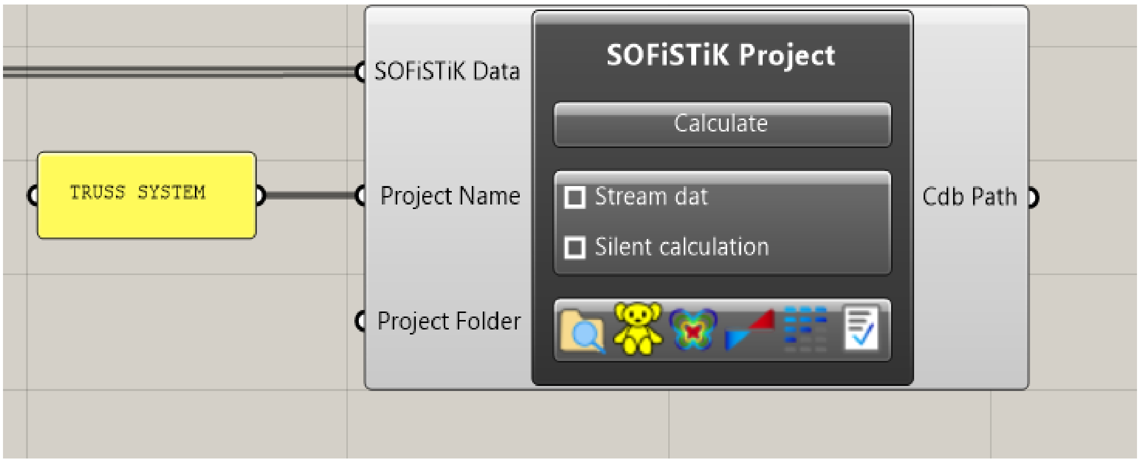

Figure 46 Sofistik project component

The results in terms of deformation and axial forces are computed.

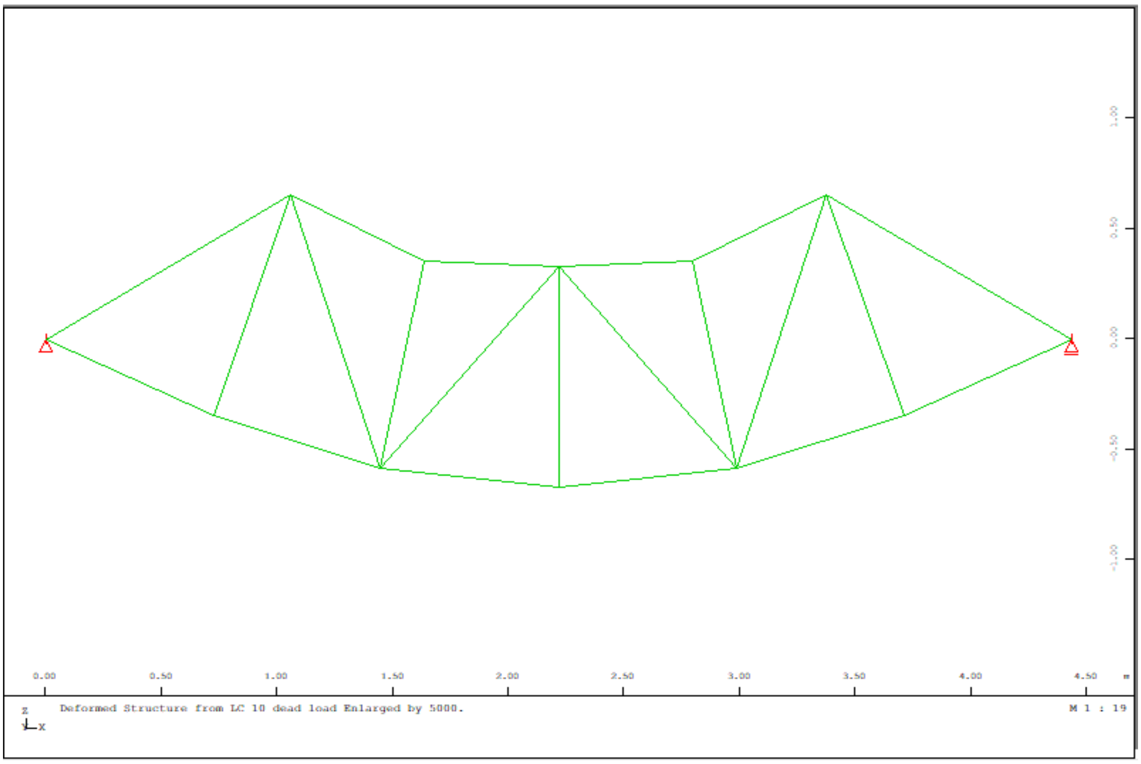

Figure 47 Deflection result

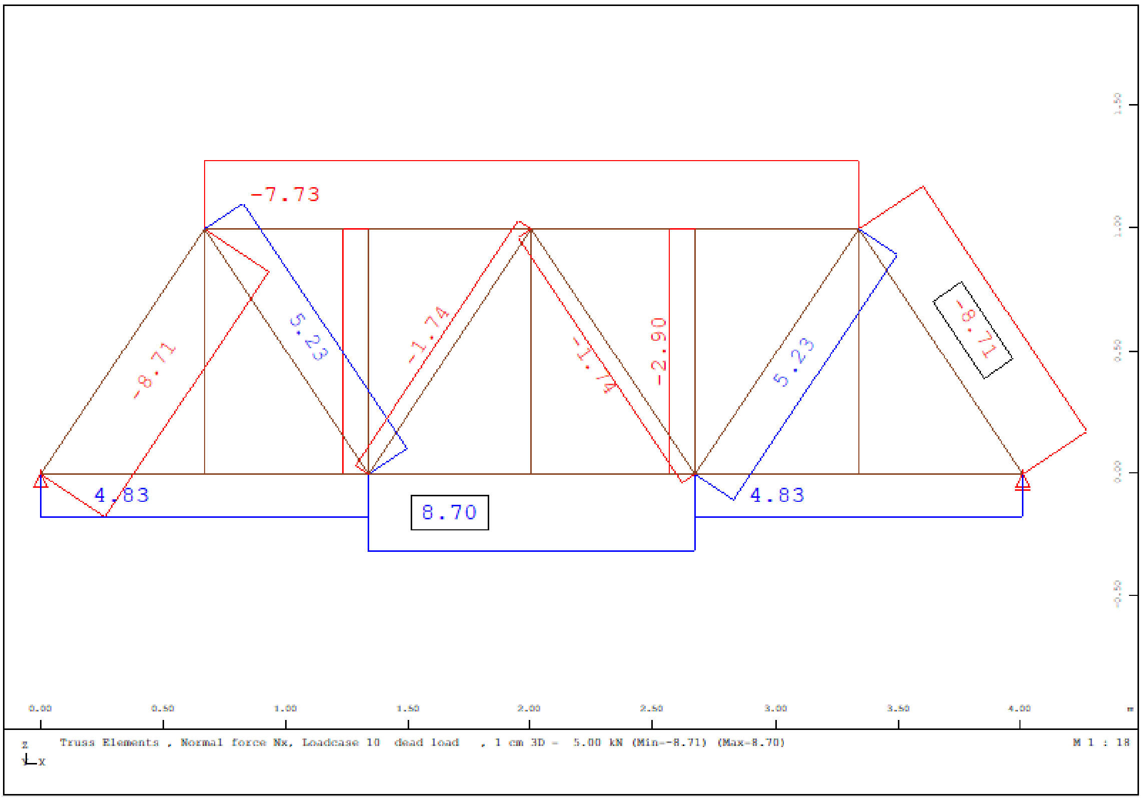

Figure 48 Axial forces results

The second approach, to obtain the same results requires defining text files for cross-sections, loads, and geometry, through the Text Input component.

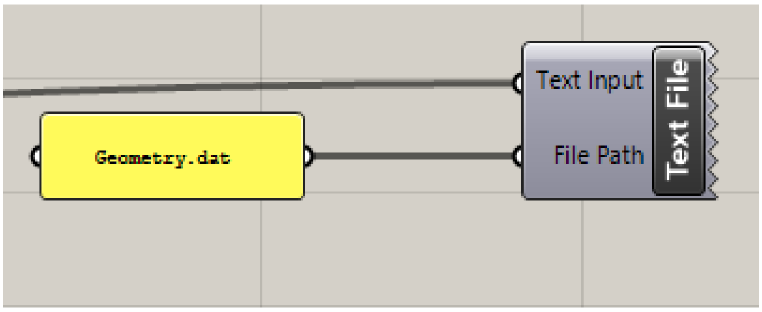

Figure 49 Text files definition

Then this data is included in the Sofistik software, where also the combinations and the type of analysis (in this case linear static) are included, and the analysis is performed again, obtaining the same result.

6 Case of study: Centre Pompidou Metz

To apply the process of interaction between a parametric design software, in this case, Grasshopper, and a Finite Element Analysis software called Sofistik, it is necessary to consider an object which has been already designed using a generative design approach. The object selected is the roof of the Centre Pompidou of Metz, a cultural center located in Metz (France).

Figure 50 Centre Pompidou of Metz (Arcquitectura Viva, s.d.)

6.1 General information

The Centre Pompidou of Metz is the first Centre Pompidou built outside of Paris, which was completed and opened to the public in 2010.

It is located in a strategic position, because, even if it is not in the center of Metz, it is very close to the Metz train station.

Each floor of this cultural center covers about 12500 m^2, and it was built to host the collection of the Musée National d’Arte Moderne.

However, in addition to exhibition areas that cover most of the area of the Centre Pompidou, there are also areas in which seasonal festivals are hosted, including the projection of movies, live shows, and lectures.

The project, which was designed by the Japanese architect Shigeru Ban alongside the French Jan de Gastines and Gumuchdjian Architects of London, had a cost of more than 80 million euros and was financed by the French government, European Union, Lorraine region, and Moselle department.

6.1.1 Historical background

In 2003, the Pompidou Centre of Paris and the City of Metz announced their idea of building a new Pompidou Centre in Metz. Afterward, there was a competition between architects coming from every place in the world to work on this ambitious project, and the competition was won by Shigeru Architects.

The final project was released by the Japanese office in 2005, and it was based on the same idea used by Shigeru in designing of Japanese Pavilion of EXPO 2000 in Hamburg (Germany), with a “paper tube structure”.

Figure 51 Japanese Pavillion EXPO2000 (Structurae, s.d.)

The construction started in 2006 and ended in 2010, but during these periods some showcases were built, such as the “Maison du Projet”, where there were scaled models of the museum, and they attracted a lot of visitors.

6.1.2 The Museum

The target of the architects Shingeru Ban, and Jean de Gastines was to obtain a building that “has a direct, sensory relationship with the surroundings”.

Under the gridshell roof, there are three rectangular cantilevered tubes, long 80 m and characterized by a width of 15 m, each tube has a gallery inside with no columns. The absence of columns and the walls made completely of glass, let the light enter to involve the user in a particular experience in each gallery.

Figure 52 Gallery of Centre Pompidou-Metz (Aasarchitecture, s.d.)

Beneath the galleries, there is an area for temporary exhibitions, which is directly connected to the garden with a glass door.

From the forum, it is possible to enter the auditorium, which can host about 200 spectators during lectures or live shows.

6.1.3 The Roof

Even if the whole museum is characterized by challenges for designers, the most complex part of the structure is represented by the roof, which is characterized by a shape inspired by the Chinese hat.

Figure 53 Chinese hat (Bamboogrovephonto.blogspot.com, s.d.)

The actual roof of Centre Pompidou of Metz is made of glue-laminated timber beams, and it was built by the German firm Holzbau Amann.

The innovation in the production of the beams is the use of CNC machining, where CNC stands for "Computer Numeric Control". Practically, the instructions to execute cuts to the timber beams are obtained utilizing the software.

To obtain these results, the geometry of each beam is defined using CAD software, then with CAM (Computer-Aided Machine), the information is translated into machine

language. Moreover, this software allows the introduction of the precise coordinates in the 3 directions to know where to do the cut in the most precise way.
These beams were assembled to obtain small hexagons elements with a side length of about 3 m, forming a larger hexagon, which is the actual shape of the roof.
The choice of the hexagon for the shape of the roof is related to the fact that it is considered a symbol of France because the country's geographical shape is very similar to the regular six-sided polygon.
The definition of the perfect shape and slope was obtained through the form-finding software, then the preparation and installation of this cover took more than a year.

Figure 54 Gridshell of Centre Pompidou Metz (Peri Vietnam, s.d.)

On the top, there is a steel spire of a height of 77 m that is connected to the whole wood frame. Finally, the roof is completed by adding a fiberglass and carbon textile canopy to cover the grid, to have a better aesthetic result, and to guarantee the protection of wood and the spaces under the cover from atmospheric phenomena.

6.2 Simplified Case

Before the study of a gridshell, a simplified case can be developed. A circular surface is considered, in which there are some holes. This object will work as a covering, and the holes will be the bases of its columns.

This circular surface with holes will be considered as a unique body, therefore, after being subjected to a form-finding, it will work as a shell.

The target of the development of this model is to understand the phases of optimization and the results (in terms of tensions and deflections) in a more efficient way.

6.2.1 Geometry definition

The first step is the definition of the circle, which will represent the external boundary of the surface that will be created, and here there is the definition of the first parameter of the project, the radius of the circle in fact will be variable since it is connected to a number slider.

The circle is moved upwards of 14 m, which represents the average height of the columns, to complete this process a vector in the Z direction is defined, characterized by the module equal to 14.

Afterward, it is necessary to define the position of the holes that will represent the base of the columns in the final object. These holes are positioned in an axisymmetric way, therefore they are dependent on the radius, but also, they are a function of the number of subdivisions of the circle, since there is one hole for each radius, thus if the number of radii increases, the number of holes are enlarged too.

Finally, the surface is defined by applying a solid difference between the boundary surface of the whole circle and the boundary surface of the holes.

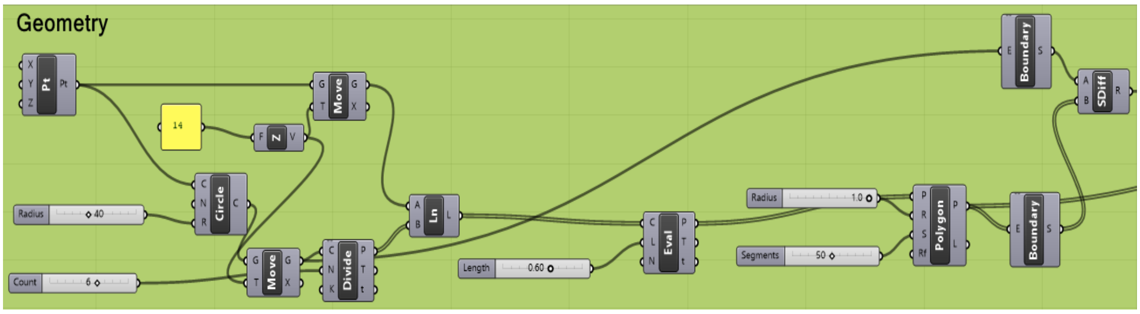

Figure 55 Solid difference Grasshopper

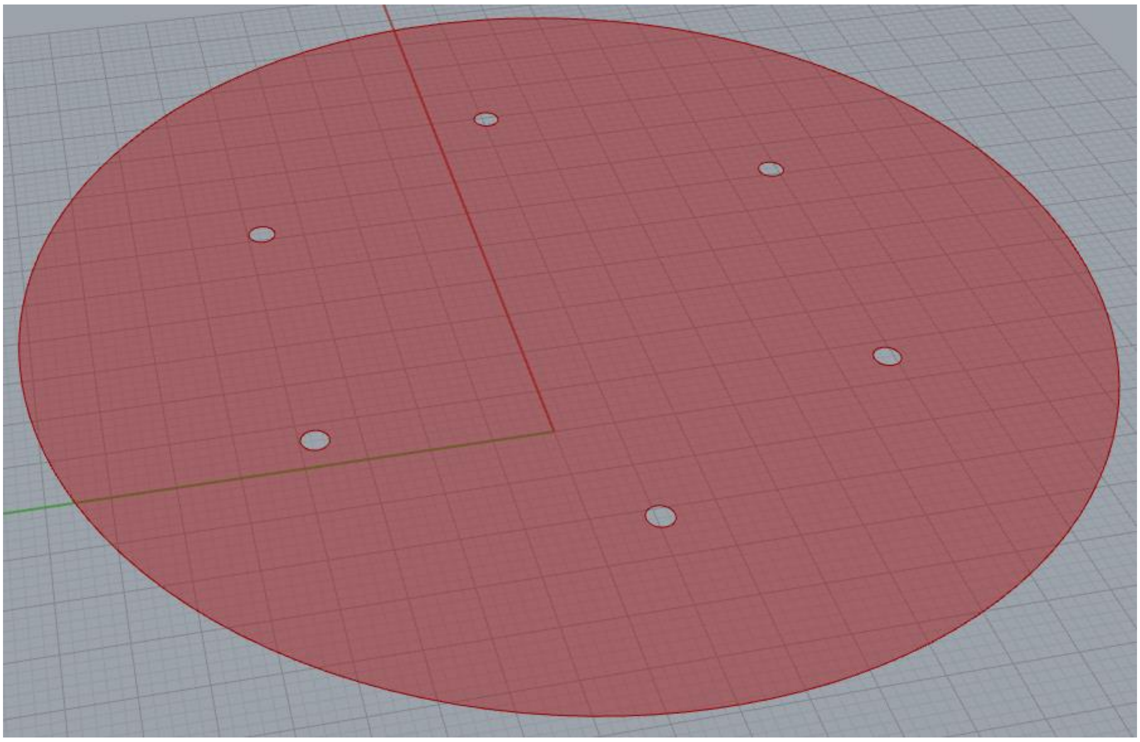

Figure 56 Solid difference Rhino

6.2.2 Form finding with Kangaroo

Once the final surface is defined, it is necessary to apply the first step of optimization, which is form-finding.

To develop this phase Kangaroo2 can be used, which is a tool embedded in Grasshopper to evaluate the shape of a surface or a line in the conditions of dynamic relaxation.

The first step requires the definition of the mesh of the BRep under study, using the Mesh BRep component in Grasshopper, this is necessary since Kangaroo can work only with meshes and lines.

The Solver component of Kangaroo requires the input of some Goal Objects to provide the solution. These Goal Objects are represented by the Length, the Anchor points, and the Load. To define the length the edges of the mesh should be extracted and input in the Kangaroo Length component.

The most complex part is the definition of the anchor points. Ideally, the canopy should be anchored in each point of the edges of the holes, furthermore, these points should be lowered to Z=0, since they represent the base of the columns. To define the required points, the Grasshopper component “Mesh Curve intersection” can be used, these points are then connected to the “Anchor” component in the geometry input. Afterward, to bring them at 0, a vector with a module of -14 is inputted in the move components and the points obtained will be to the targets of the anchor points.

Finally, to consider the effect of the own weight of the structure in the form-finding, the “Load” component is introduced.

Two more goal objects can be defined, the first component is “Show”, and it allows the user to see the new shape of the mesh after the form-finding, moreover, there is the “Grab” component, that allows the designer to modify the shape of the object manually on Rhinoceros.

The result obtained shows a surface that gets to 0 in the anchor points, while it moves up in the center forming a convexity. In the extremes, instead, the structure goes up like a flower petal before it blooms.

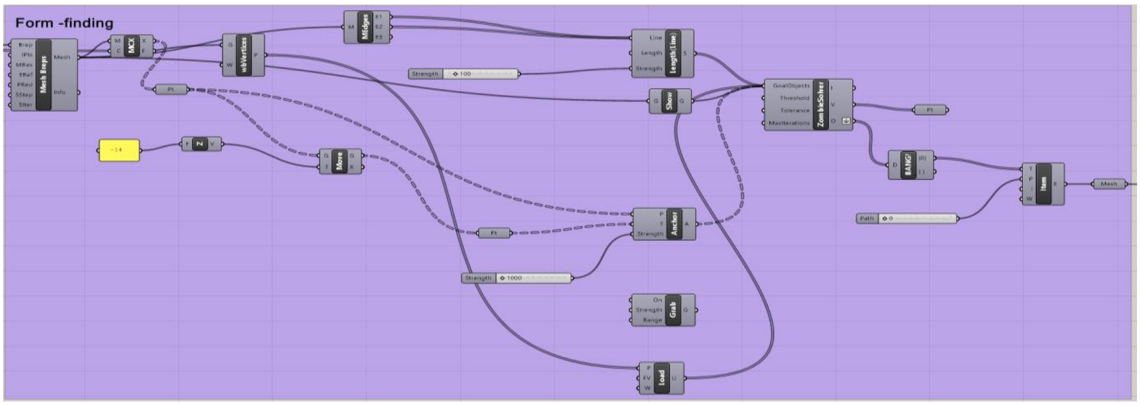

Figure 57 Form-finding definition

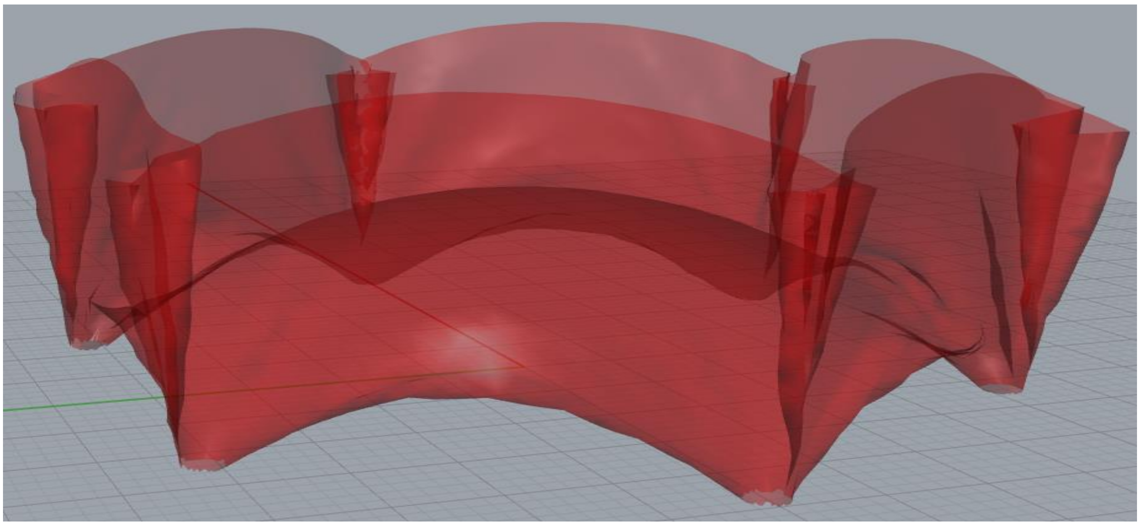

Figure 58 Form finding result

6.2.3 Finite Element Analysis with Karamba3D

The surface obtained with form-finding is subject to a structural analysis on a finite element analysis tool (Karamba3d), to evaluate the tensions in the shell.

The output of the Kangaroo add-on is a data tree, however, to study the object in Karamba it is necessary to have a mesh, which can be obtained as the output of the component "Explode tree".

The following step requires the definition of the cross-section and the material, in the case under study the shell is characterized by a constant section made of concrete of a class C25/30.

Once the cross-section and material are determined, it is crucial to obtain the shell from the available mesh. The outputs of the component "MeshToShell" are the elements that constitute the shell and the points of connection between these elements.

The points are important results since the next phase requires the definition of supports, the structure is constrained at z=0 in correspondence of the points at the base of the columns, therefore it is necessary to sort the points-based of the Z coordinate to extract the indices of the points that will be anchored. This is possible using the Grasshopper components "Sort list" and "Sublist". The supports applied in each of the points considered will be fixed supports, therefore no movements will be allowed.

The last feature that is fundamental for the analysis is the load. Two load cases will be applied to the shell: the own weight or gravity load, and the snow.

The snow action is applied as a constant mesh load, that will be represented by a series of vectors in the global Z-direction, directed downwards and characterized by a module of 3 KN/m^2; it is a high value of snow, since the canopy is supposed to be designed in a zone close to the mountain, and therefore subjected to heavy snowfall events.

All the features defined are then input in the component "Assemble model", which will generate the final model, which is analyzed with the relative component.

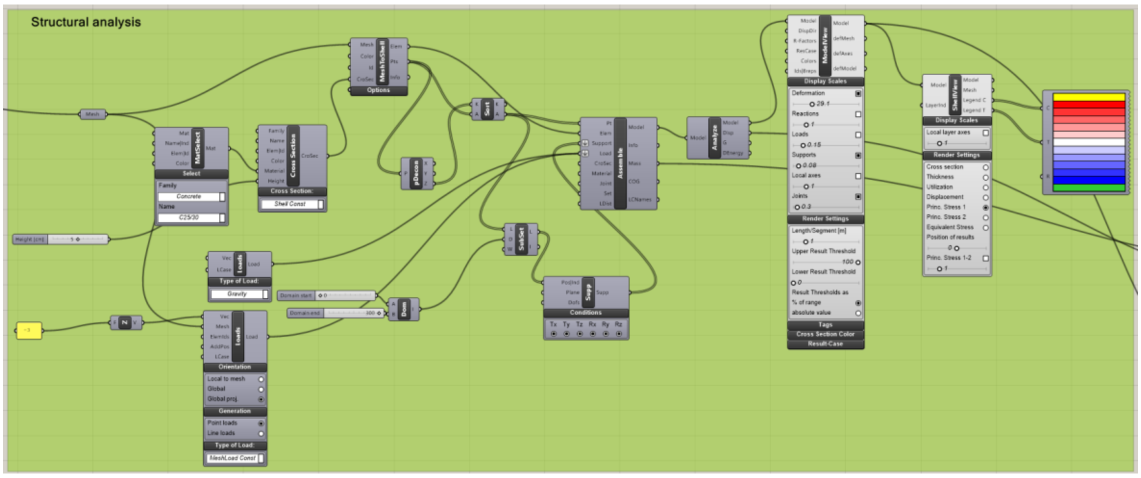

Figure 59 Structural analysis with Karamba

6.2.4 Results from Finite Element Analysis

The results of the analysis are connected to the component “ModelView”, which allows us to see graphically all the input parameters (for instance loads and supports), but also the outputs (deformations and reactions).

Instead, to evaluate the stresses, it is fundamental to connect the model to another component labeled “ShellView”, which, together with the legend, allows the view of the graphical and numerical results.

Among the results, there are the tensions that are subdivided in principal and equivalent, but also the utilization (in percentage), and the deformation can be seen.

The results are shown in Rhinoceros with the shades of colors that vary between blue which represents the highest tensile value, and red which characterizes the largest compression.

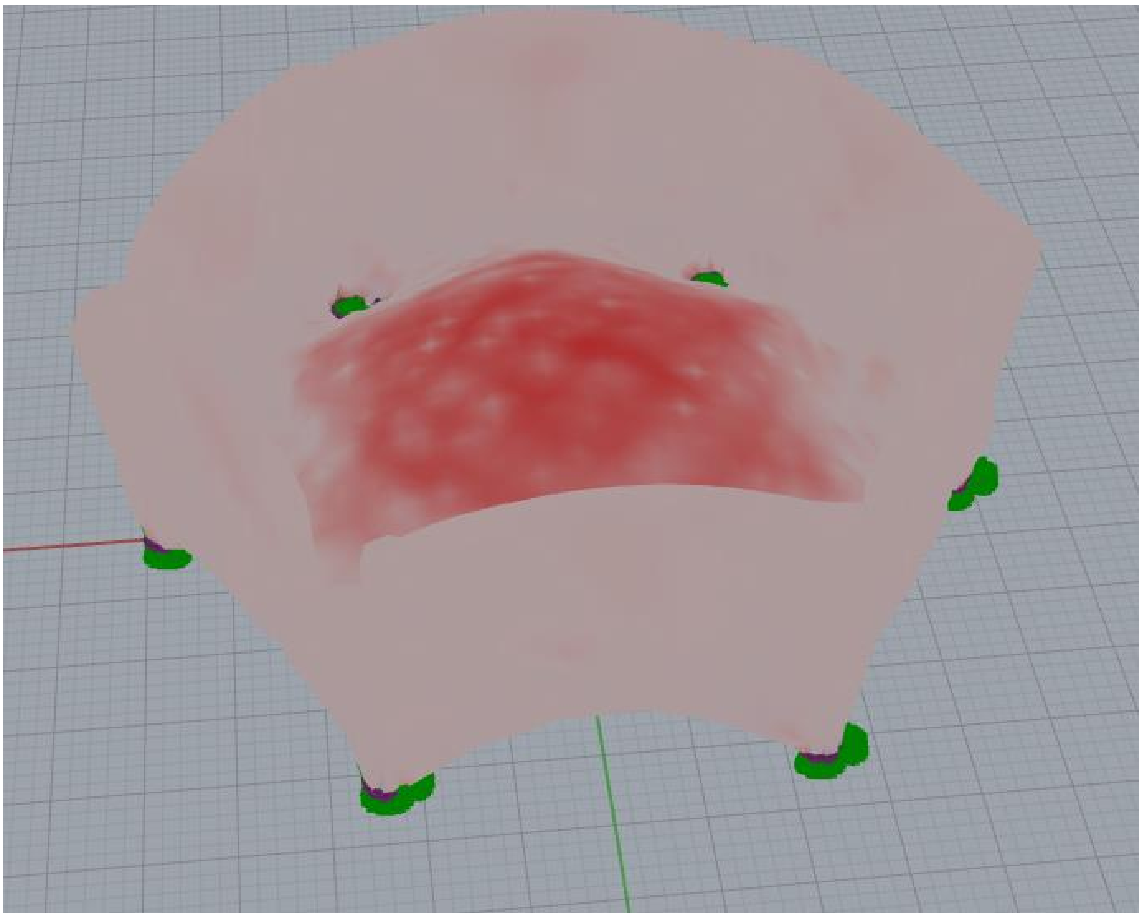

Figure 60 Structural analysis results

As it is visible, the central area of the object is subjected to high compression values, therefore the shell will work as an arch, while the external sides will be subjected to low compression or reduced tensile stresses.

This is acceptable behavior in terms of tensions since the shell is made of concrete, which is a material with a high strength to compressive actions, and low resistance to tension.

6.2.5 Structural Optimization

Once the structural analysis has been completed, optimization should be performed, in particular, a constrained multi-objective optimization.

The multi-objective optimization allows the designer to obtain the ideal solution, allowing the variation of some parameters fixed, to obtain the reduction of the objective functions. In the case under study, the parameters considered are the number of columns and the radius of the column base. The objectives instead are, to reduce the mass, to reduce the maximum tension (the starting value is about 1,44 KN/cm^2), while the displacement can increase, however, it should be lower than L/500, which is the maximum displacement to avoid problems at the Serviceability Limit State for floors and covers.

In the case study, the objective function is defined with the following formulation:

$$\text{Minimize } F(x) = \begin{Bmatrix} f_1(x) \\ f_2(x) \end{Bmatrix}$$

$$\text{where } f_1(x) = mass,\ f_2(x) = \sigma_{id}\ (stress)$$

$$\text{Subject to } g_i(x) \leq \delta_{max}$$

$$\text{Where } \delta_{max}\text{=max displacement in z direction (9,6 cm)}$$

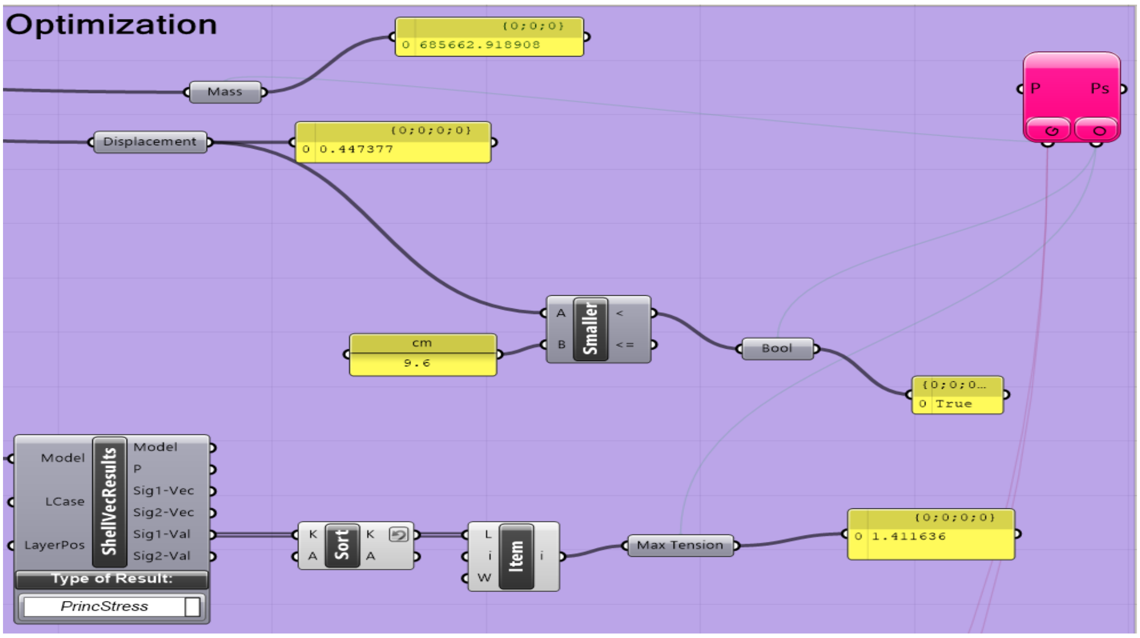

Figure 61 Structural optimization shell

The optimization is performed with the aid of Octopus, a Grasshopper plug-in, considering 300 generations, allowing the diversification of the parameters, and considering only the Pareto Front results.

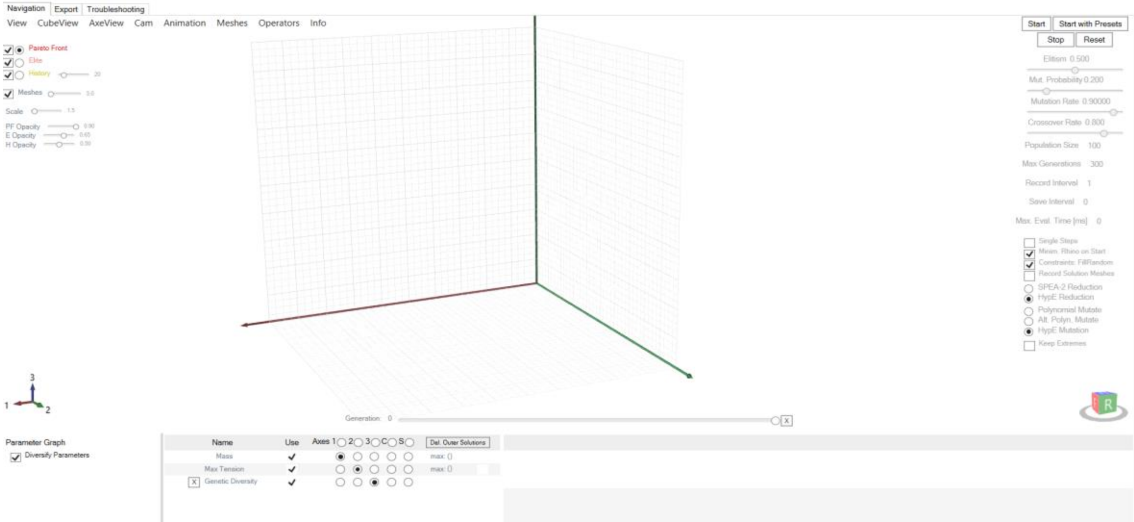

Figure 62 Octopus settings

After the optimization process has been completed, there will be some points that will form the final chart of the optimization. For each point, there will be a different mass, max compression, and displacement.

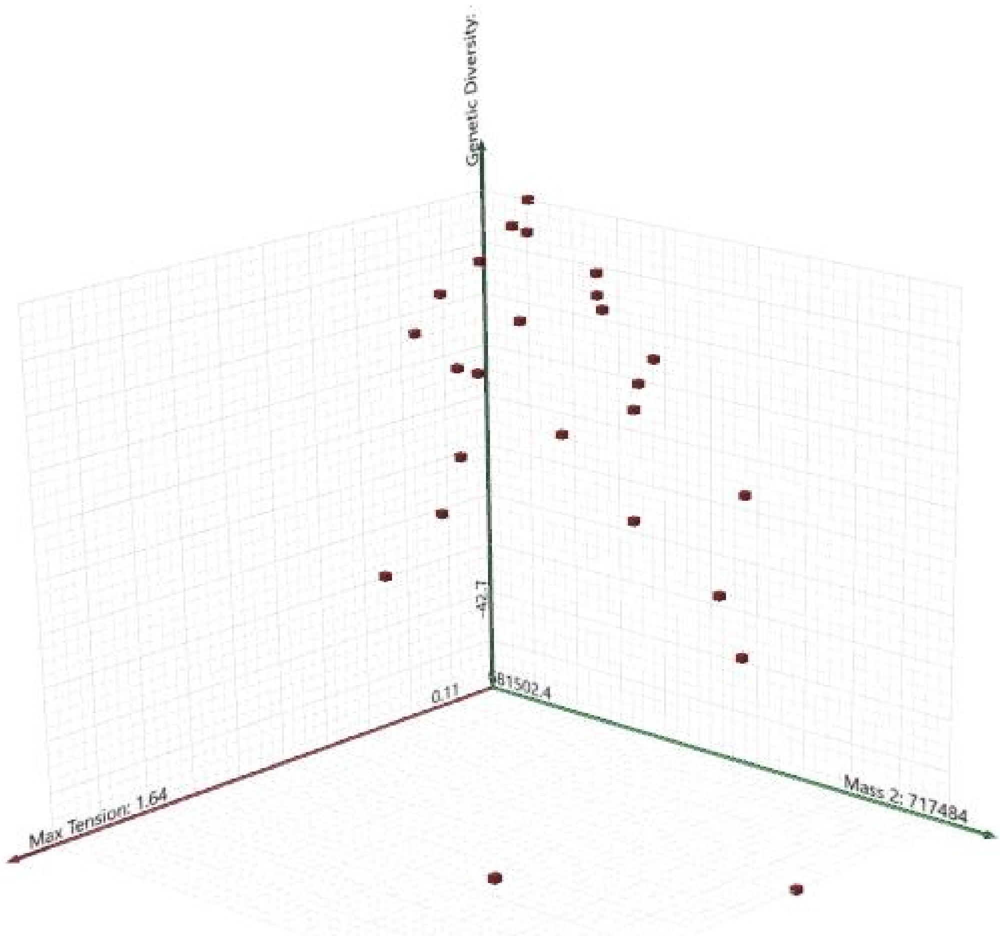

Figure 63 Octopus result

The results can be compared to the pre-optimization values, this is done by considering 3 points of the chart, corresponding to three solutions of the optimization process.

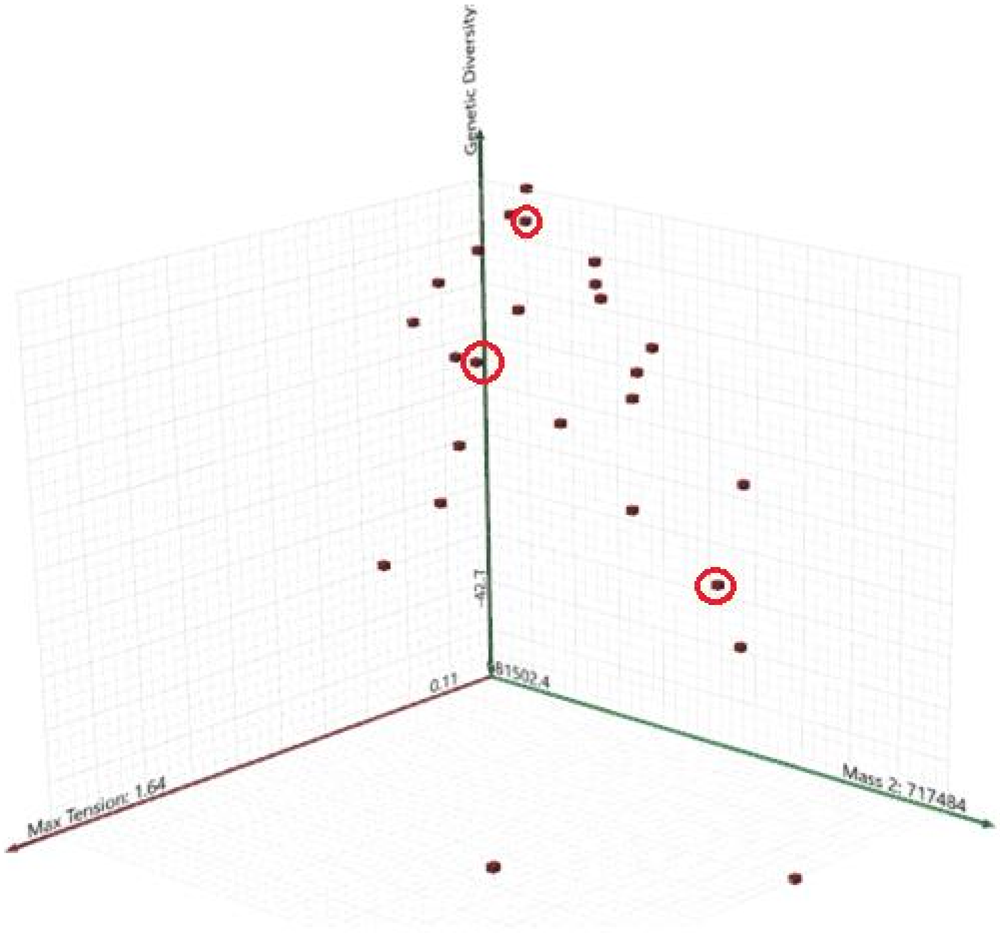

Figure 64 Selected points Octopus

Case	Mass [Kg]		Displacement [cm]		Max Compression [KN/m2]	
	Value	% of reduction	Value	% of increase	Value	% of reduction
PRE-OPT	705595,55		0,4474		1,4159	
OPT1	685722,03	2,82%	0,4586	2,49%	0,1429	89,91%
OPT2	686351,58	2,73%	0,7790	74,12%	0,3576	74,74%
OPT3	704361,92	0,17%	1,0573	136,32%	0,3211	77,32%

Figure 65 Table of comparison of Octopus results

From the results obtained, it is possible to understand that there is a reduction in the stress data much higher compared to the reduction of mass.

6.2.6 Comparison between Sofistik and Karamba

To analyze the shell with Karamba, the mesh is transformed into a BRep, then there is the definition of the design code and the material used, the structural area, the structural points (which will be the supports), and the load.

These components are connected to the components "Sofimshc" and "Sofiload", all the inputs are merged and connected to the Sofistik project.

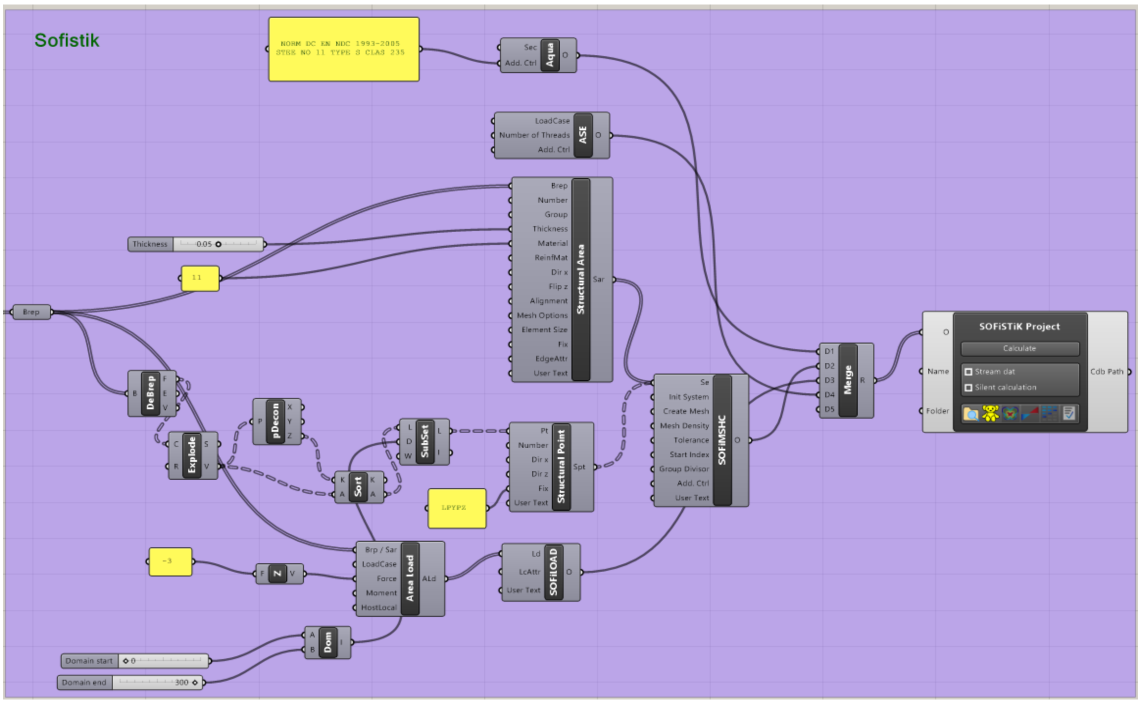

Figure 66 Analysis with Sofistik

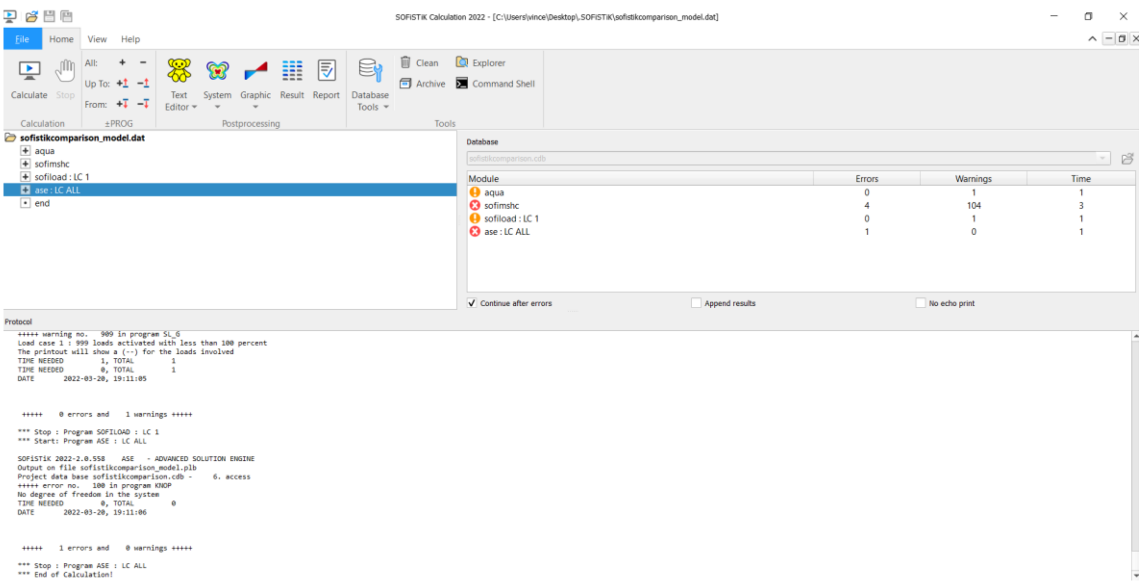

Figure 67 Results Sofistik

However, the calculation doesn't provide a result, there is an error in the Sofimshc calculation, probably because the structural area to consider should be flat, and also in the final calculation, where it says "No DOFs in the system", which is probably due to the presence of a large number of fixed points.

6.3 Geometry definition

The project that will be developed is based on the geometry of the "Centre Pompidou of Metz". Therefore, the target is to obtain a geometry close to the real one, even if there are some differences between the original project and the object that will be built in this thesis.

The roof of the Centre Pompidou of Metz is sustained partially by parametric columns, and it is fixed on the other sides of the museum building. Therefore, this geometry is very complex to study, and it is not only related to parametric architecture.

The case of study is focused only on the roof, which will be independent of the building, and supported by a larger number of columns.

6.3.1 Roof Pattern

The roof has a hexagonal shape, and this large hexagon, with the side's length of about 90 m, is divided into smaller hexagons that compose the pattern.

The first approach applied was to draw the roof edges using the component Polygon, applying a number slider for the radius and the number of sides. Afterward, this polygon is divided into smaller hexagons, to do this the first step is to obtain the centers of the small hexagons by using the component Divide Surface, then draw the hexagons using again the polygon component, together with the relative sliders.

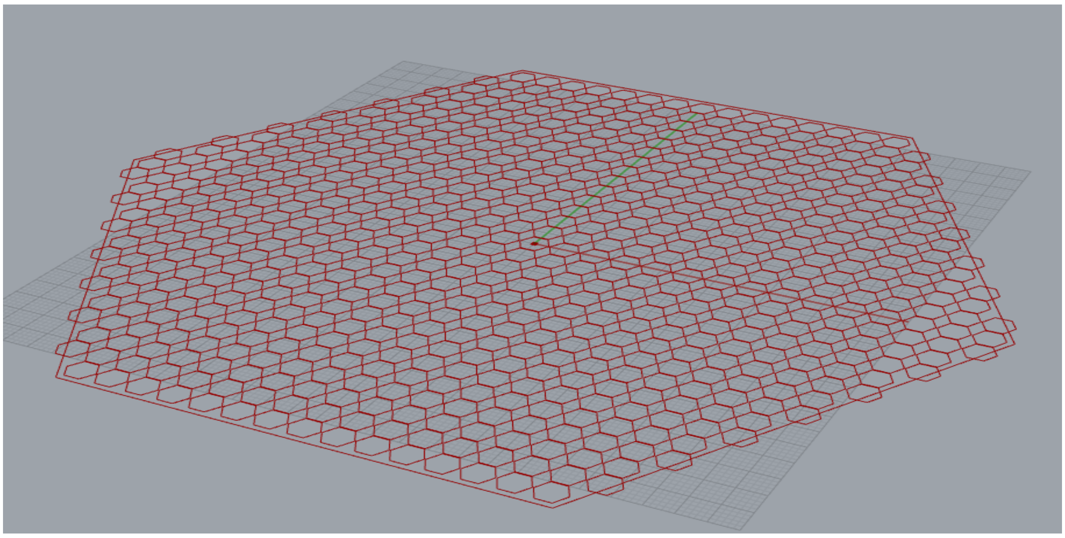

Figure 68 First attempt pattern Rhino

All the previous passages are resumed in the following Grasshopper visual code.

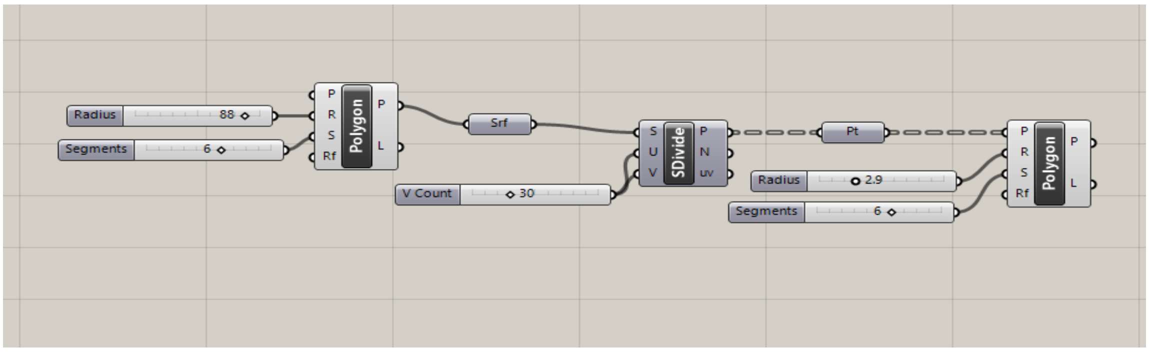

Figure 69 First attempt pattern Grasshopper

However, this approach is not correct, since there is no continuity in the beams constituting the roof pattern. This is not acceptable from the physical and structural point of view, because it will not be possible to transfer the weight to the columns.

To solve this problem, a different approach is adopted, trying to obtain the small hexagons that constitute the roof, through the intersection of the beams starting from each of the six sides that constitute the polygon under study.

The first step to reach this target requires exploding the polygon with the relative component, to split the hexagon polyline into different lines representing the sides.

Afterward, the sides are subdivided into different segments using the command Divide Curve, it is important to select the single curve with the help of the Subset component.

Then the points obtained on each side are connected between each other using means of lines, and the following result is obtained.

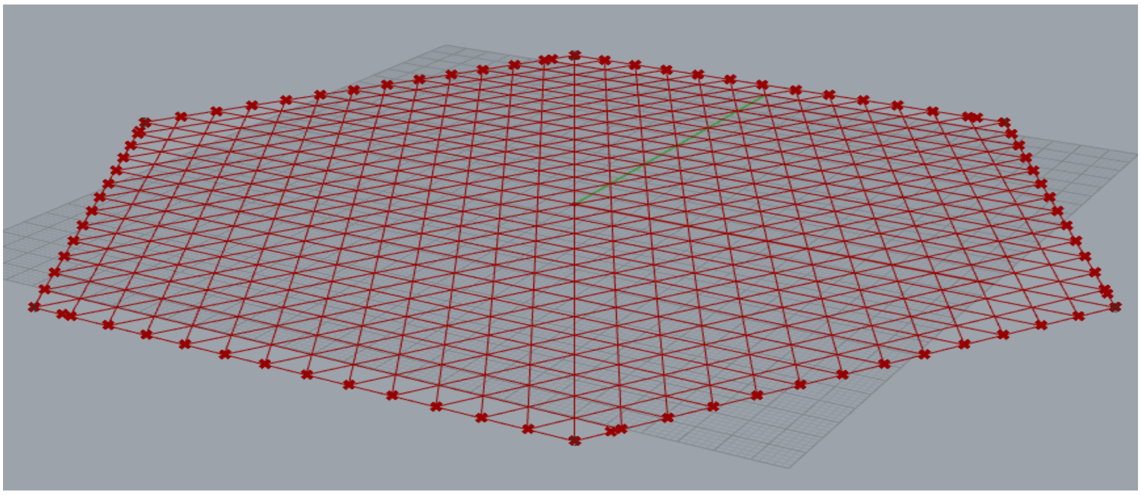

Figure 70 Final pattern Rhino

The result obtained does not precisely represent the real pattern yet, since the hexagons are not visible yet, to solve this problem it is necessary to have one line drawn and the following not present, and so on, and this target can be reached by introducing a Boolean component.

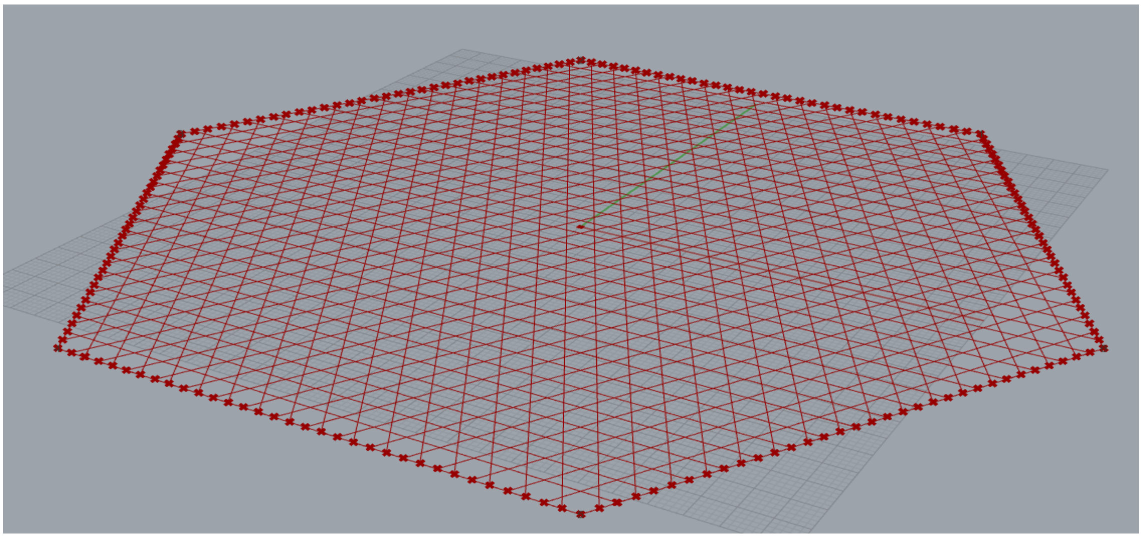

Figure 71 Increase number of subdivision

Also, the roof pattern is easily controlled with a parameter representing the number of subdivisions of each side, increasing which more and smaller hexagons can be obtained, on the contrary with the reduction of this parameter the exact opposite result is found.

All the previous steps are included in the following code.

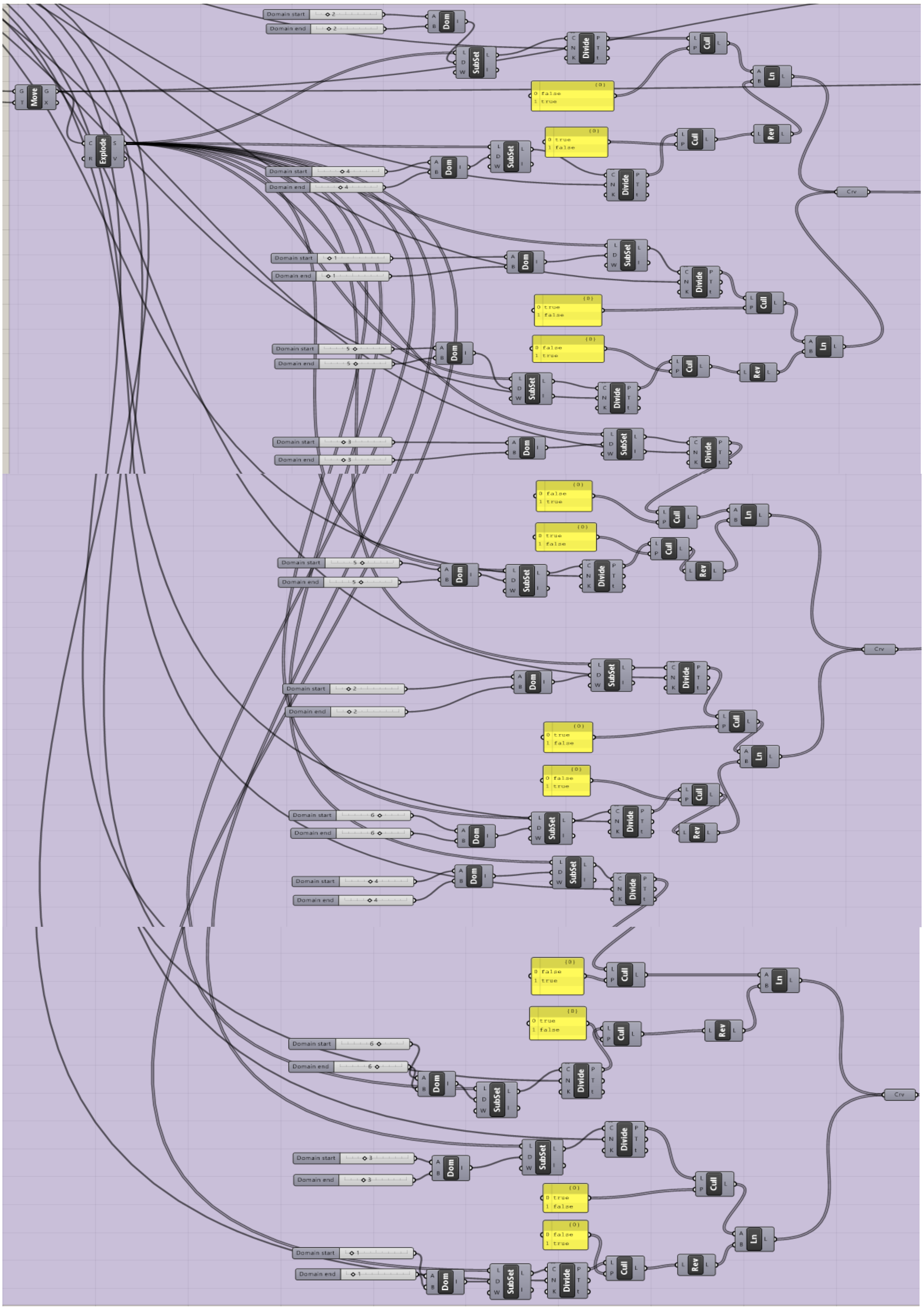

Figure 72 Final pattern Grasshopper

6.4 Study of the final object

Once the pattern of the roof has been defined, it is important to pass from a flat cover to the final object, which will be arched and composed also by columns, in a similar manner to the simplified example.

Obtaining this particular shape, while maintaining the pattern defined, is a complex operation, therefore three different methods have been developed to achieve the best solution possible, both from the aesthetic point of view and the functional point of view.

6.4.1 Geometry projection approach

The first approach that was developed is based on considering the object initially as a shell, to apply form-finding using Kangaroo.

Practically, it is applied the same approach of the simplified case, defining a BRep, characterized by the difference of the filled polygon and the polygon with holes in correspondence of the radii into which the object is subdivided.

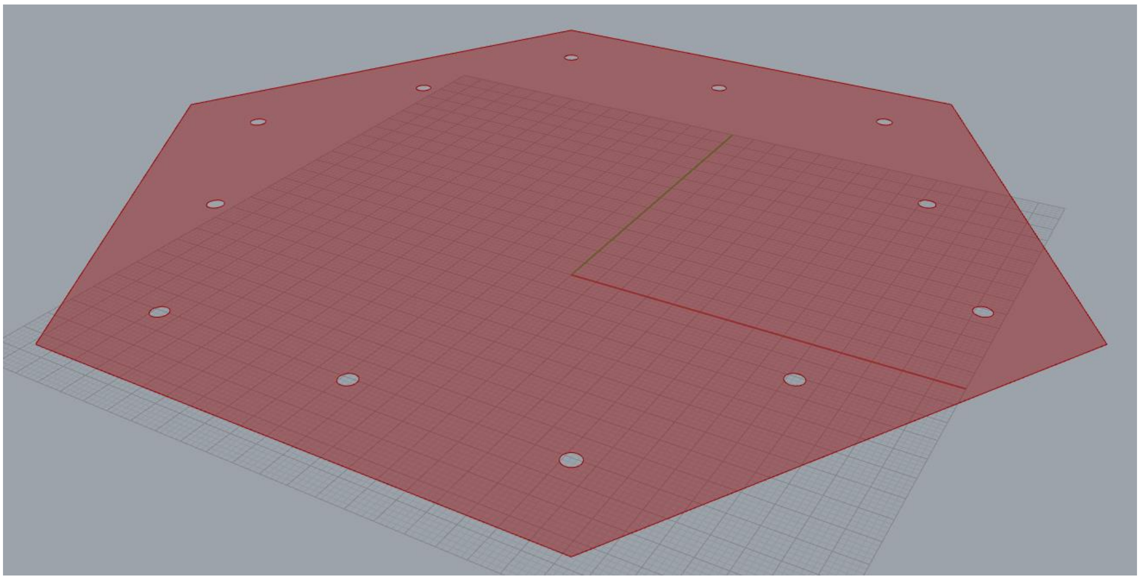

Figure 73 Solid difference polygon

Afterward, the defined solid is transformed into a mesh, the points of the circumferences of the circles become the supports, the gravity load is defined, and all these components are input in the Kangaroo Zombie Solver, from which the form-finding result is obtained.

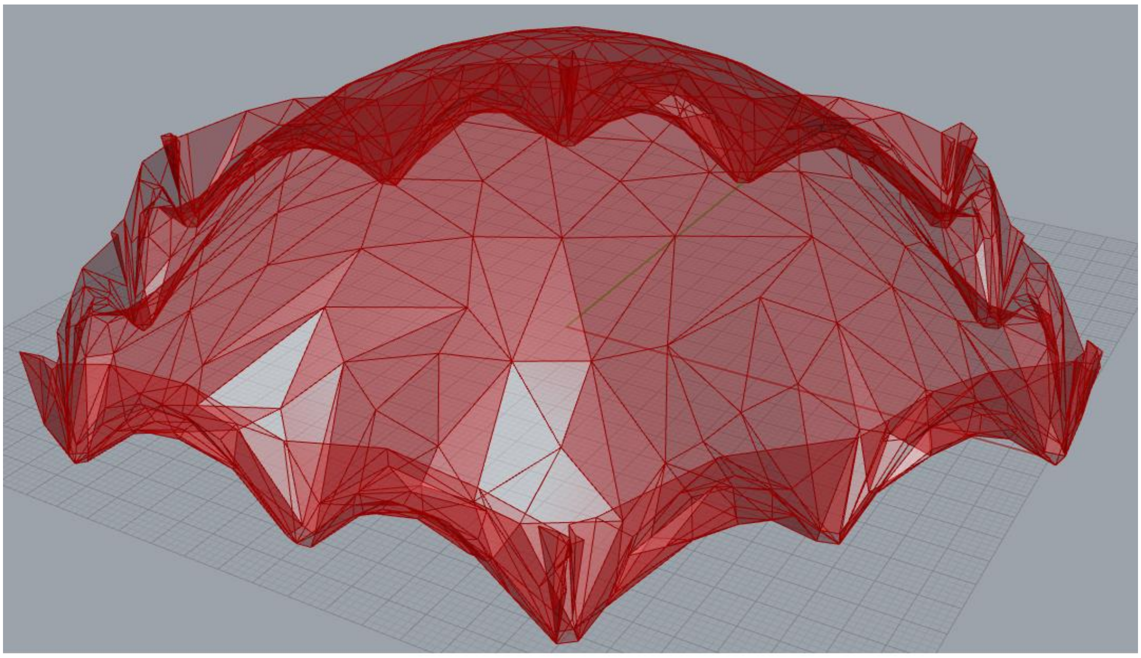

Figure 74 Brep Form-finding result

Finally, the geometry pattern defined previously is projected on the BRep obtained, after the transformation of the mesh outputted by the Solver, through the Grasshopper component “Project”.

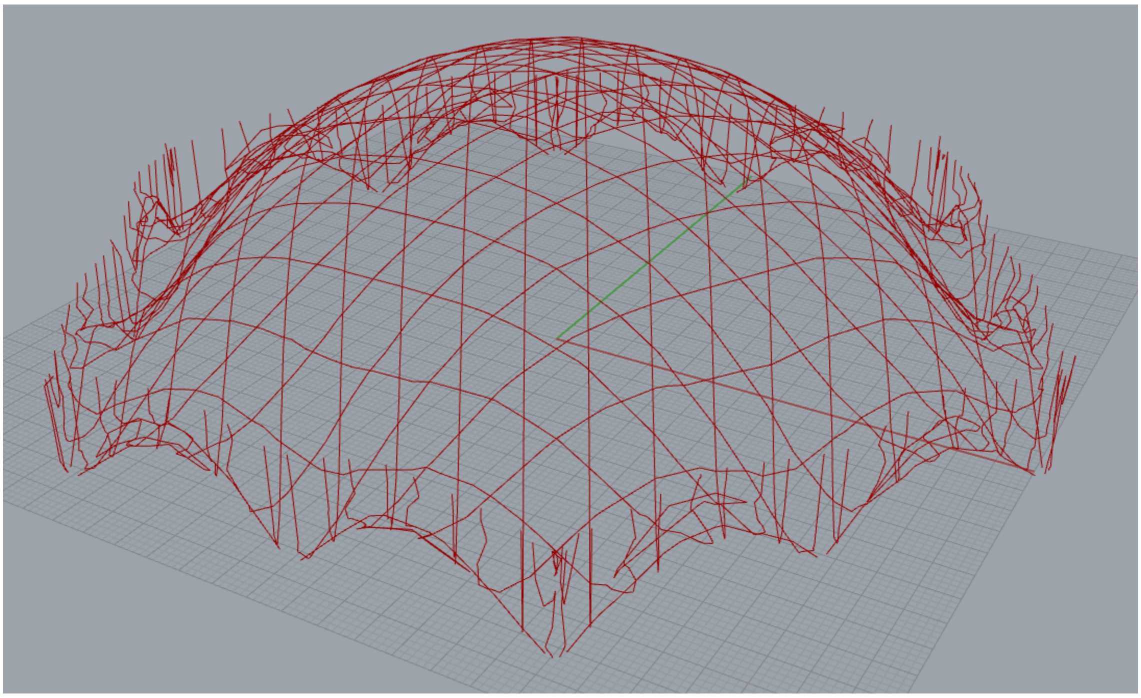

Figure 75 Grid projection

The result is not acceptable, since not only the original grid has been modified and it's not recognizable anymore, but also all the elements located on the outside of the positions of the columns are disconnected.

Passing to the structural analysis, this means that once the elements are transformed into structural lines (beam or truss elements), they will be infinitely rigid, therefore the calculation is not possible and Karamba cannot output any result in terms of stresses, deformations, and displacements.

6.4.2 Geometry into mesh approach

The second approach used to solve the problem of studying a gridshell with a complex shape on a Finite Element Model software would be to transform the geometry, defined in the previous step into a mesh, and apply it to it directly form-finding.

6.4.2.1 Form finding of the geometry mesh

The geometry already defined can be transformed into a mesh. To do this operation, first of all, there is the definition of the holes, which will be considered as polygons with 50 sides, this allows better control of the tool on Grasshopper instead of using circular holes. Meanwhile, the surface of the object is created, and the curves defined in the previous stages composing the pattern are joined using the Grasshopper component "Merge".

Afterward, the component "Surface split" is used, and the inputs of this component are the defined surface and all the curves. The centers of the fragments outputted, are connected to the component "Point on curve", where also the column bases are inputted, the relationship obtained is connected to the mathematical operator "Larger than", taking into account only values larger than one. These values will represent the pattern introduced in the "Dispatch" component, where also the list length of the fragments is connected.

In this way it is possible to obtain two results, the first one labeled as "A", represents the mesh only inside the holes, the "B" output instead will contain the mesh of the object except for the holes, which is the necessary result for this example.

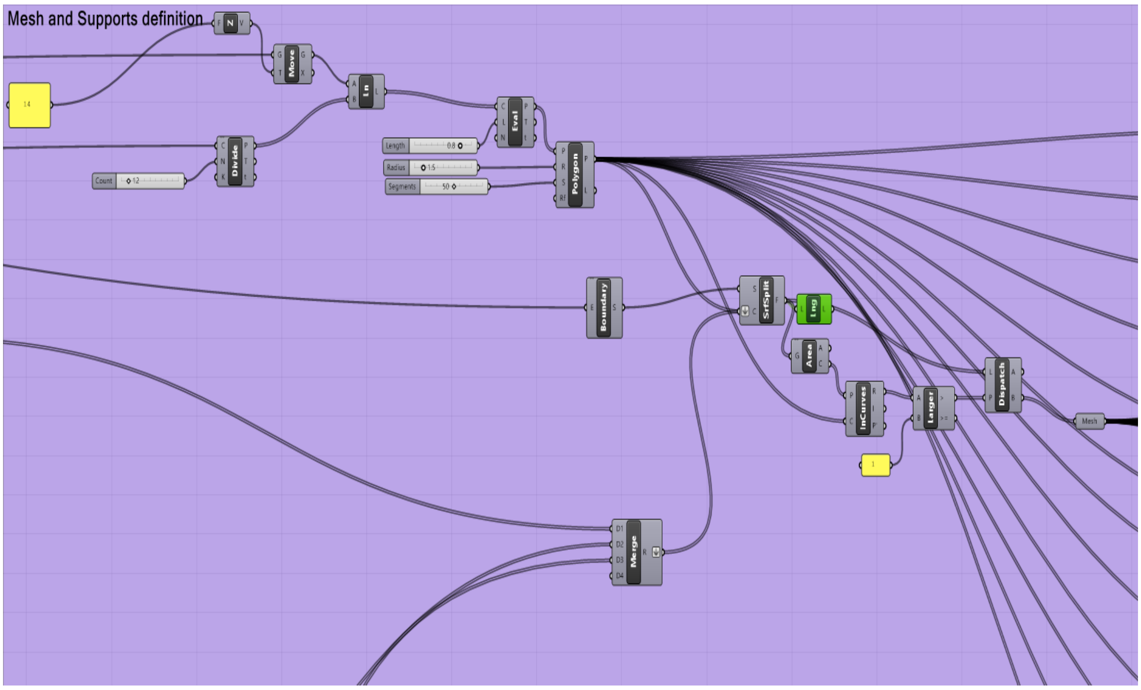

Figure 76 Transformation pattern in mesh

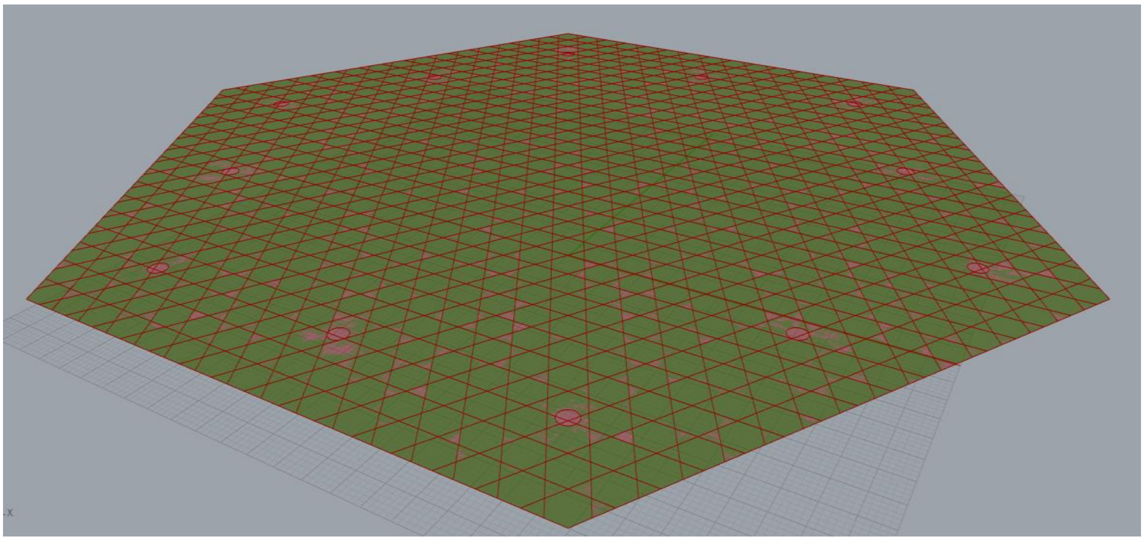

Figure 77 Mesh obtained Rhino

Once the mesh is defined, the Zombie Solver needs three important input data: the anchor points, the length, and the load.

The anchor points definition is not elementary, it is used the "Mesh-curve intersection" component, however not all the points are obtained using this command one time, therefore it is necessary to repeat the intersection for each hole, and then merge all the points.

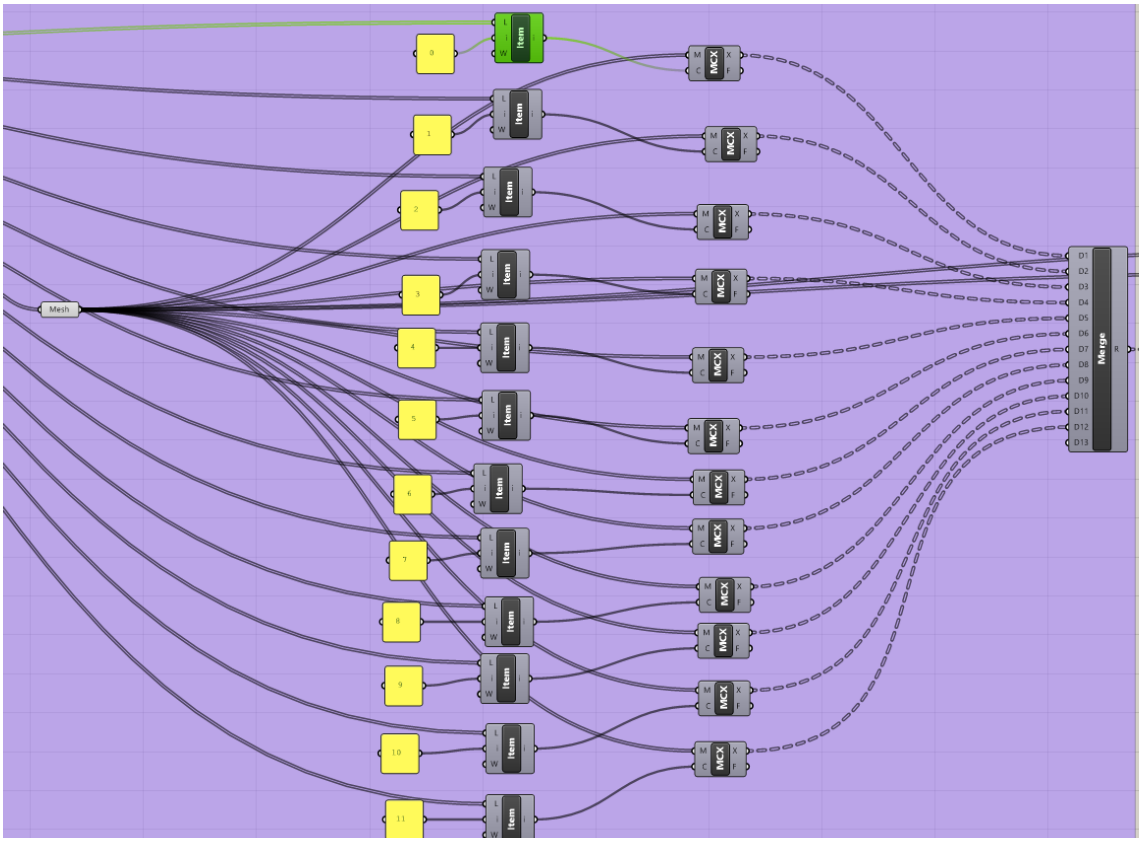

Figure 78 Mesh-Curve intersection

Once the supports are defined, also the length and the load are connected to the Solver and the result of the form-finding is obtained.

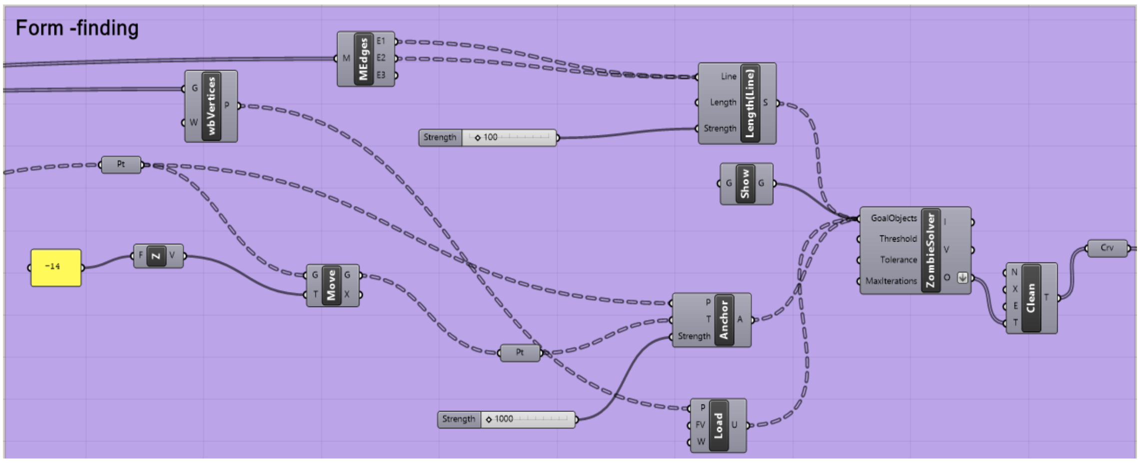

Figure 79 Form finding transformed mesh

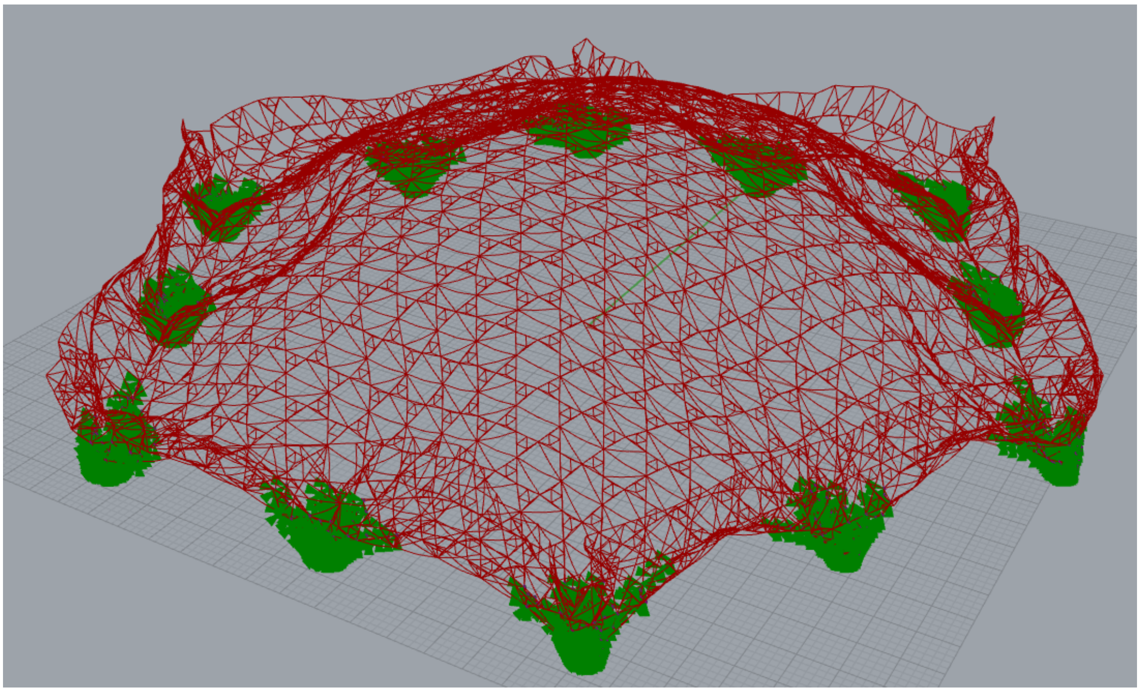

Figure 80 Form finding result

As it is possible to observe, in the form-finding procedure the pattern is not completely maintained.

6.4.2.2 Structural Analysis

Once the object is obtained from the form-finding procedure, it is necessary to evaluate the structure with FEA software.

In this case, there is not a shell with a constant thickness, but there is a series of small elements which will be modeled as beam or truss.

Firstly, the lines are transformed in beams with the Karamba3D component "LinetoBeam", afterward the cross-section and the material are defined. The material defined is structural steel S235, since it allows a good behavior both in tension and compression, this can be useful in the case under study, since also due to the wind action (which was not taken into account in this study) the distribution of stresses can change, there could be used also some innovative materials like bamboo, however, the standards and codes regarding these materials are not sufficient to develop such a complex structure in safety.

The considered cross-section is a rectangular closed cross-section because it is supposed that the connection of these elements is compared to HEA or HEB elements, however, this cross-section can be subjected to buckling since for most of the dimensions the section will be in Class 3, therefore subjected to local instability. To avoid this phenomenon, the compression stresses should be reduced, this is possible through the optimization process, however, the snow action will generate a constant compression on the roof.

b x a	s	Weight	Area	Inertia		Strength modulus		Inertial radius	
				Jx	Jy	Wx	Wy	ix	iy
mm	mm	kg/m	cm2	cm4	cm4	cm3	cm3	cm	cm
100 x 20	2	3,64	4,64	48,7	3,39	9,74	3,39	3,24	0,85
	3	5,37	6,84	69,77	4,52	13,95	4,52	3,19	0,81

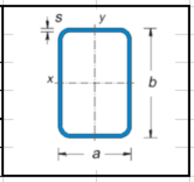

Figure 81 Steel section properties

After the definition of the cross-sections and material, the supports are defined, there will be 2797 points at z=0 which will be fixed.

Finally, the loads are applied, considering the dead load of the structure and the snow action.

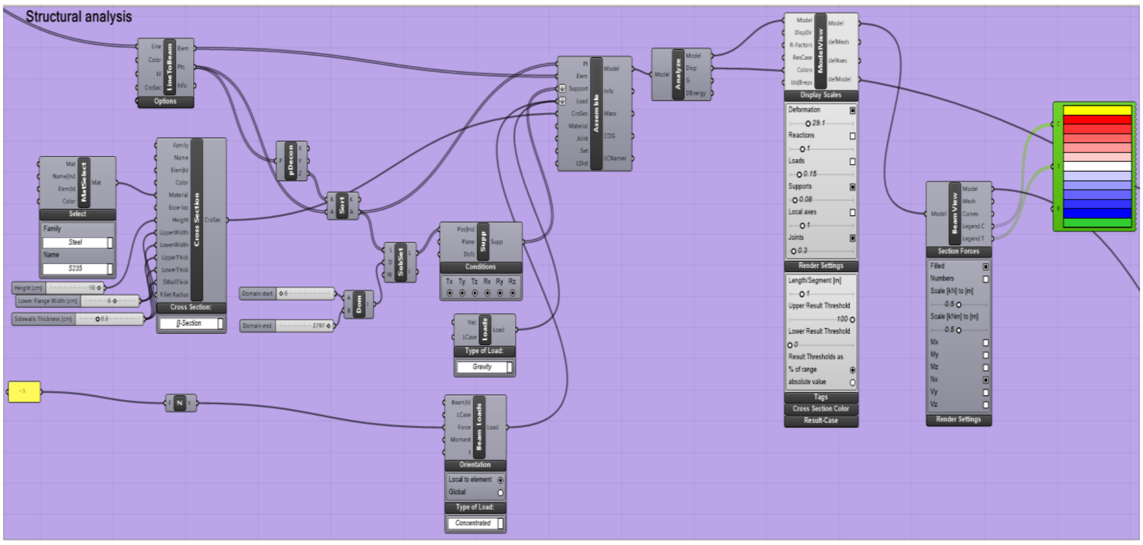

Figure 82 Structural Analysis with Karamba

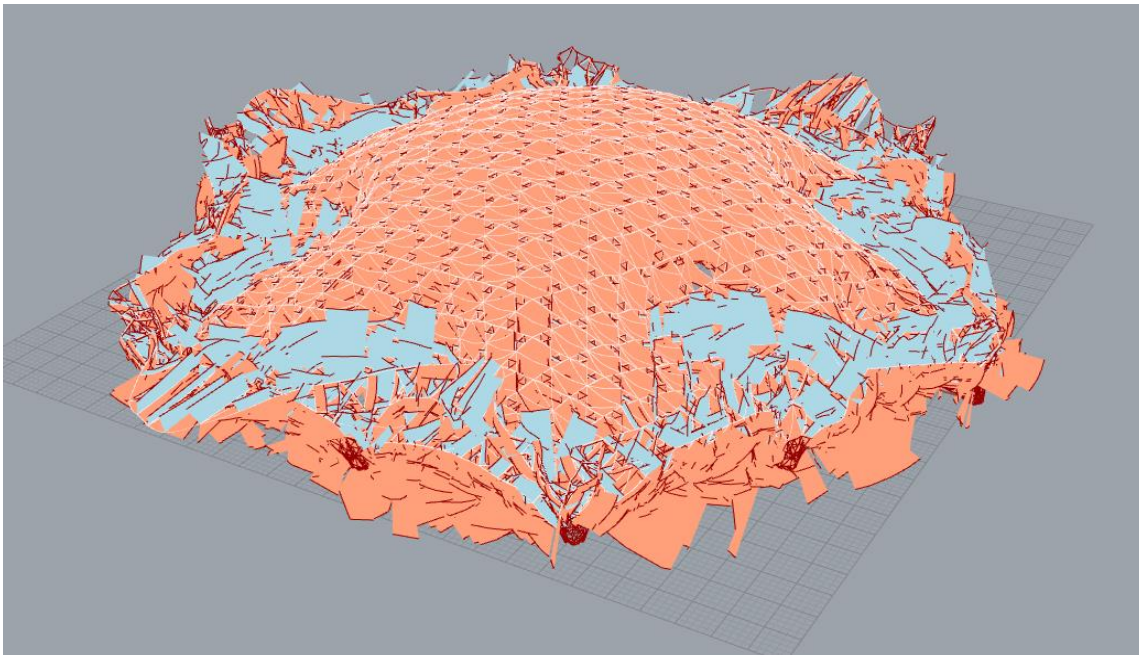

Figure 83 Result of Karamba

As in the example of the shell with constant thickness, also in this case the beams close to the center of the object will be mainly compressed, while the external beams are subjected to tensile stresses, following the arch distribution of tensions.

6.4.2.3 Structural Optimization

The structural optimization, in this case, has been developed considering only the radius of the column as a parameter, and the maximum compressive strength and displacement as objective functions.

This is because the number of columns is not independent, since in the use of the command "Mesh Curve intersection", it is necessary to consider the columns one by one to obtain the points of intersection, this implies that the number of circles cannot vary automatically.

$$\text{Minimize } F(x) = \begin{Bmatrix} f_1(x) \\ f_2(x) \end{Bmatrix}$$

$$\text{where } f_1(x) = mass,\ f_2(x) = \sigma_{id}\ (stress)$$

$$\text{Subject to } g_i(x) \leq \delta_{max}$$

Where δ_{max}=max displacement in z direction (10 cm)

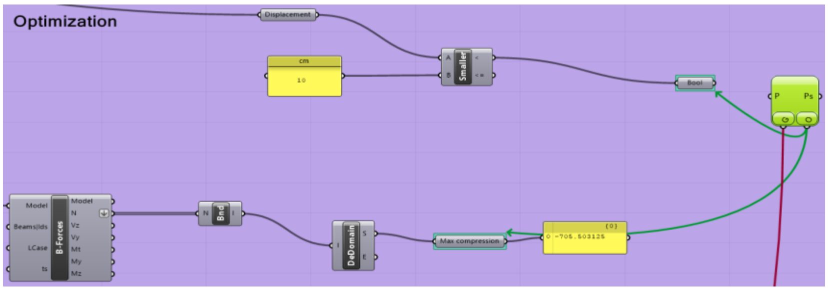

Figure 84 Optimization gridshell

6.4.2.4 Comparison between Sofistik and Karamba

In this case, the comparison between the two tools is not possible since the analysis with Sofistik cannot be launched, there are errors in Rhino (due to the large use of CPU and RAM) probably due to a large number of elements.

6.4.3 Large deformation analysis approach

Another approach that can be used to develop the structural analysis of the object is by obtaining the shape with the “Large Deformation Analysis” component of Karamba.

The procedure is similar to the previous one in the first steps, there is the definition of the geometry pattern and the transformation of this pattern in a mesh with the help of the “Dispatch” component.

Afterward, it is needed to assemble the model with the Karamba3d relative component, and the input data is composed of the structural elements, the cross-sections definition, the material, the load, and the supports.

The structural elements are obtained with the component “LineToBeam”, where the input is the pattern geometry, and the outputs will be beam elements.

The cross-section and material are the same as the previous case a rectangular hollow cross-section is considered, and the material is structural steel of class S235.

The load is defined as in the previous example considering gravity loads, and the applied vertical loads of $3KN/m^2$ represent the snow action.

The main problem is related to the definition of the points where the supports can be applied. To reach this target some attractor points are defined manually in the Rhino environment, then using the “Closest points” component, together with the “ Cull index “ component, the points of the bases of the columns are obtained, and they are connected to the support.

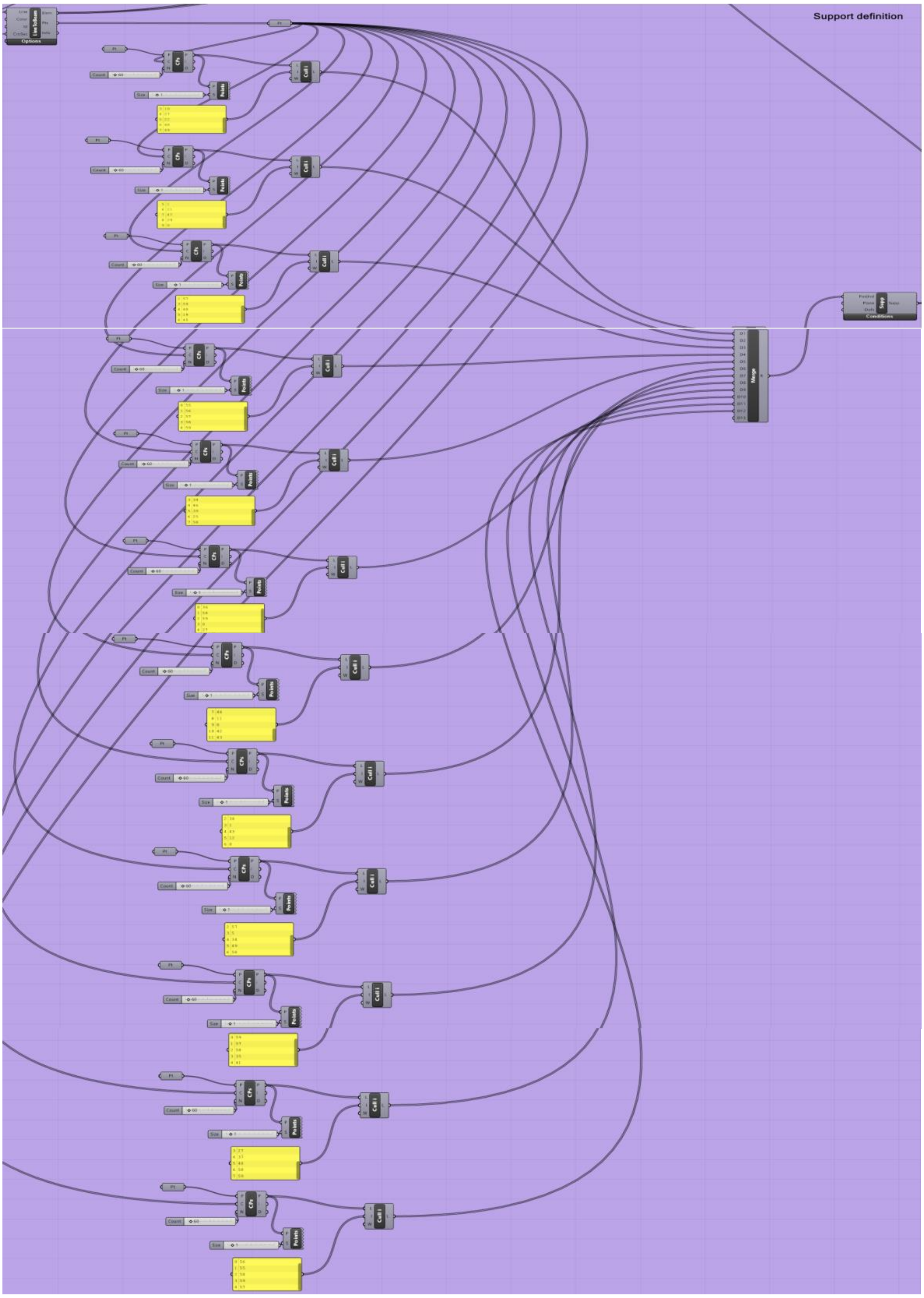

Figure 85 Fixed points definition

Then the component "Large deformation Analysis" is used, and this time the pattern is not ruined, but it maintains the original hexagons.

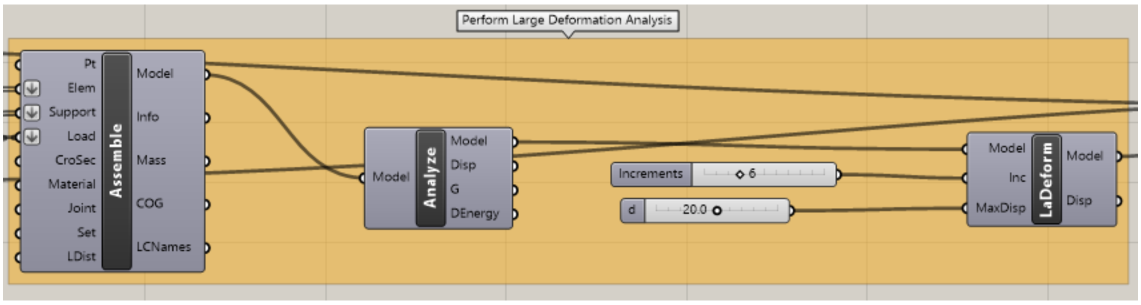

Figure 86 Large deformation analysis

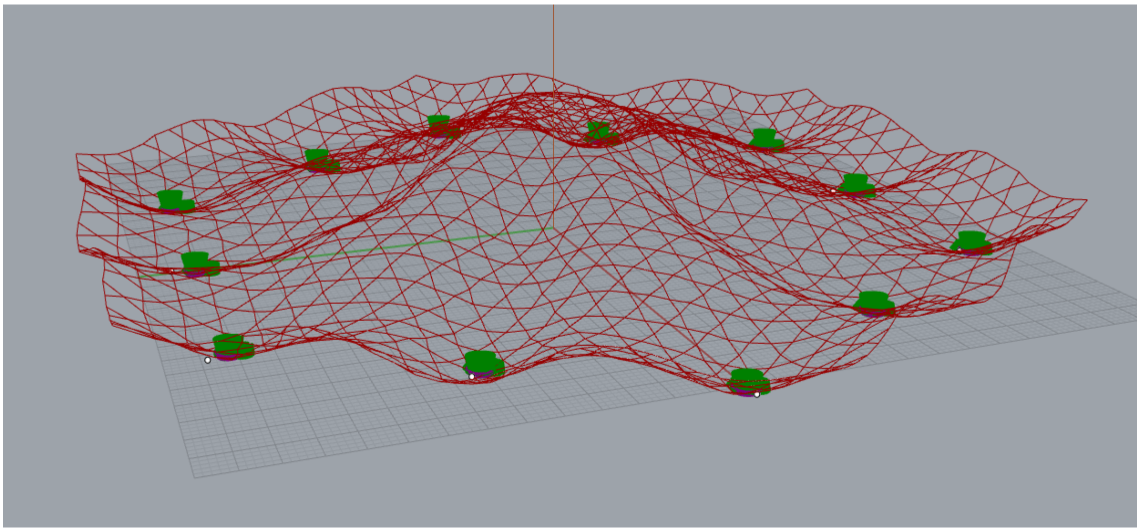

Figure 87 Large deformation analysis result

Finally, the object obtained is disassembled using the "Disassemble model" and "Disassemble element" components, it is reassembled reintroducing the same supports, load, cross-section, and material, and the model is analyzed to see the results in terms of tensions.

It is possible to see graphically that, in this case, the elements are subdivided between compressed elements (highlighted with the red color) and elements subjected to tensile stresses (highlighted with the light blue color), however in this case there is not a general trend in the distribution of tensions, and the sign of the axial force is a function of the concavity of each beam.

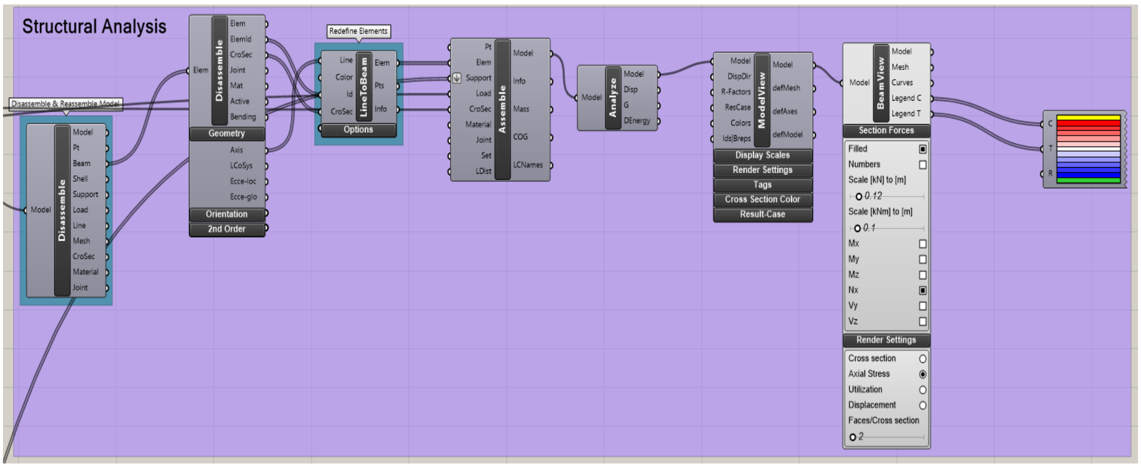

Figure 88 Karamba analysis of Large deformation analysis object

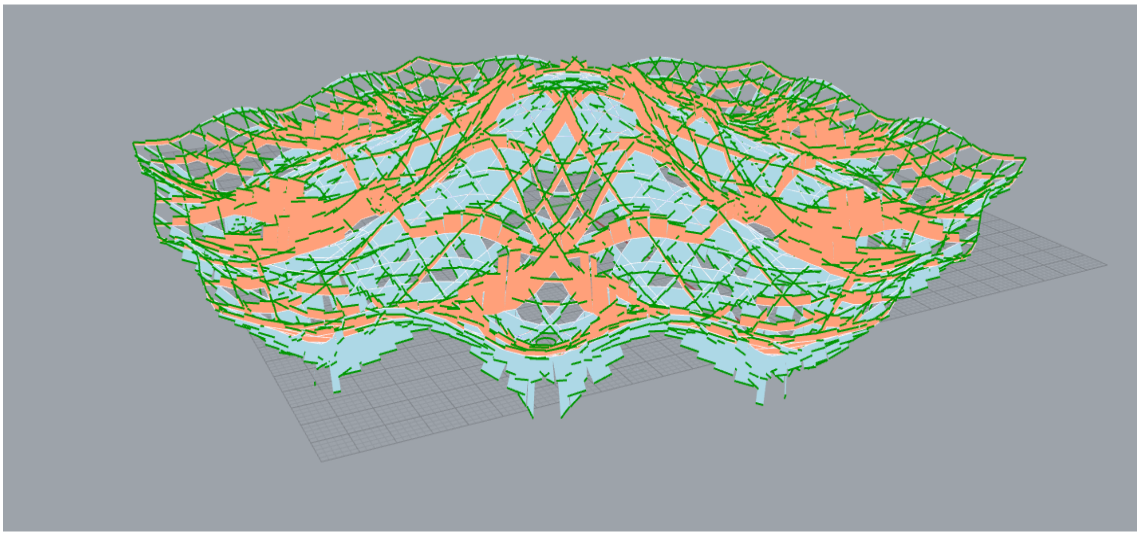

Figure 89 Result of analysis of large deformation analysis object

In this example there is a relevant issue, structural optimization cannot be applied, since the definition of the supports is dependent on points defined on Rhino. These points could be even defined in Grasshopper without problem, however, there still would be a dependence on some fixed points, therefore the number and the diameter of holes could not be modified automatically as it happens during the optimization procedure.

7 Conclusions

The thesis has been developed in different stages, starting from the study of a theoretical part including the subjects that were not present in the MSc syllabus such as parametric design and structural optimization, but also the revision of shell structures, which have been a topic in the structural mechanics' courses.

The second part of the work has been focused on the application of the concepts of parametric design using Grasshopper, together with the study of the possibility of interconnection between the algorithmic design software and the Finite Element Model tools such as Sofistik and Karamba3D.

Regarding this interconnection, some benchmarks with simple cases have been developed, followed by the two main cases under study, the shell, and the gridshell.

The link between Grasshopper and Karamba3D is working perfectly, there are components of Karamba present on the Rhinoceros tool which allow not only the definition of the supports, load conditions, materials, cross-sections, and structural elements but also the analysis and graphical visualization of the results in terms of stresses and displacements, moreover, these processes require a relatively low computational time too. However, some problems related to numerical instability have been detected, for instance, during the calculation of tensile stresses, some values, albeit slightly, were negative instead of positive.

In the case of Sofistik, the connection is working properly in simple cases, while for complex cases, for example, in the case of the shell, the analysis is not working due to a large number of elements, or an error is found in the support definition related to the use of a high number of fixed points.

Moreover, Sofistik requires a knowledge of its programming language to define different materials and different cross-sections. In particular, the cross-sections have to be included with a text file or with a json file.

After connecting the FEM software and Grasshopper, SO has been applied to the case studies.

The optimization is divided into two steps, firstly the form-finding procedure has been implemented to the objects under study using Kangaroo2, a Grasshopper add-on; the latter optimization was a multi-objective constrained optimization, which has been developed using the Octopus add-on.

The form-finding process is relatively fast from the computational time point of view, however, in the case of the gridshell, the pattern is disrupted in the result from Kangaroo. Another issue can be found running the form-finding solver: the solution needs to be recalculated every time the file is opened and every time a single parameter is changed extracted by the SO solver. Moreover, the object cannot be moved to another file but must be always connected to the Kangaroo solver.

In the second stage of optimization, the target was to reduce the mass and the maximum stress allowing the increase of the number and the radius of the columns. The problem of this process is the computational time since with high-performance computers the optimization process requires about one week. This high computational time is a strong drawback, since every time it is needed to modify an input (for example the load, the dimensions of the shell), the results are computationally expensive.

7.1 Future works

- Guarantee a direct interconnection between Rhinoceros and Sofistik, with the introduction of components that allow the definition of the elements without the need of text files, but also the visualization of results graphically in the Rhino environment.
- Development of interconnection between Grasshopper and Sofistik to allow the user to work with a larger number of elements, or more complex structural typologies.
- Development and implementation of updated solvers (Octopus was released in 2018) to apply the multi-objective constrained structural optimization directly in the Grasshopper canvas, with the help of a precise component.
- Implementation of components that allow the generation of grids based on a complex curvature and geometry (mesh).

8 References

Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C., 2014. Shell Structures for Architecture. s.l.:s.n.

Boyd, S. & Vandenberghe, L., 2004. *Convex Optimization.* s.l.:s.n.

Deb, K., 2001. Multiobjective Optimization Using Evolutionary Algorithms. s.l.:s.n.

Jawad, M. H., 1994. Theory and Design of Plate and Shell Structures. s.l.:s.n.

Kirsch, U., 1993. Structural Optimization: Fundamentals and Applications. s.l.:s.n.

Lewis, W. J., 2012. Computational form-finding methods for fabric structures. s.l.:s.n.

Mork, J. H. & Dyvik, S. H., 2015. Gridshell Manual. s.l.:s.n.

Rao, S. S., 2011. The Finite Element Method in Engineering. s.l.:s.n.

Tedeschi, A., 2014. AAD Algorithm- Aided Design. s.l.:s.n.

Wassim, J., 2013. Parametric Design for Architecture. s.l.:s.n.

3D Natives, s.d. [Online]

Available at: https://www.3dnatives.com/

5.imimg.com, s.d. [Online]

Available at: http://5.imimg.com

Aasarchitecture, s.d. [Online]

Available at: https://aasarchitecture.com/

Adriaenssens, S., Block, P., Veenendaal, D. & Williams, C., 2014. Shell Structures for Architecture. s.l.:s.n.

ArchDaily, s.d. [Online]

Available at: https://www.archdaily.com/

Arcquitectura Viva, s.d. [Online]

Available at: https://arquitecturaviva.com/

Bamboogrovephonto.blogspot.com, s.d. [Online]

Available at: http://bamboogrovephoto.blogspot.com/

Divisare, s.d. [Online]

Available at: https://divisare.com/

Dlubal, s.d. [Online]

Available at: https://www.dlubal.com/

Elisa.fi, s.d. [Online]

Available at: https://elisa.fi

G2Rail, s.d. [Online]

Available at: https://www.g2rail.com/

Grasshopper, s.d. [Online]

Available at: https://www.grasshopper3d.com/

IMA Architects, s.d. [Online]

Available at: https://ima-architects.co.uk/

MechaniCalc, s.d. [Online]

Available at: https://mechanicalc.com/

Nir Sivan Architects, s.d. [Online]

Available at: http://www.nirsivan.com/

Peri Vietnam, s.d. [Online]

Available at: https://www.peri.com.vn/

Quora, s.d. [Online]

Available at: https://it.quora.com/

Rao, S. S., 2011. The Finite Element Method in Engineering. s.l.:s.n.

Researchlm, s.d. [Online]

Available at: https://researchlm.wordpress.com/

RIBA Architecture.com, s.d. [Online]

Available at: https://www.architecture.com/

Schmid Schrauben hainfield, s.d. [Online]

Available at: https://www.schmid-screw.com/

Structurae, s.d. [Online]

Available at: https://structurae.net/

The Boston Globe, s.d. [Online]

Available at: https://www.bostonglobe.com/

Vistanet.it, s.d. [Online]

Available at: https://www.vistanet.it/

Wagner biro steel and grass, s.d. [Online]

Available at: https://wb-sg.com/

Wassim, J., 2013. Parametric Design for Architecture. s.l.:s.n.

World Architecture Community, s.d. [Online]

Available at: https://worldarchitecture.org/

9 Acknowledgments

I would like to express my deepest appreciation to Politecnico di Torino and the National Technical University of Athens for allowing me to develop this thesis in Athens.
I would also like to extend my deepest gratitude to Professor Giuseppe Carlo Marano, Professor Nikos Lagaros, and Doctor Laura Sardone for their guidance during the thesis elaboration, working on innovative and different topics such as structural optimization and parametric design would not have been possible without their support.
I would like to thank all the colleagues and Professors I have met during my studies, and mainly all the friends I met during my six years of university. I wish to thank Francesco, Vito, Giuseppe, and Davide because even if I left Politecnico di Bari our bond is still present. I'm also grateful to Luca DC and Luca DV, meeting them in the first months of my experience in Turin was crucial, especially at the moment when I was in a crisis for the distance from home. Moreover, I would like also to thank Fanis, Giorgos, and Panos, who helped me live in Athens in the Greek way.
I'm extremely grateful to the people who have always been there and who have never let me down, my mother Antonella, my father Giovanni, my sister Alessia, and my lifelong friends Francesco, Francesco, Fulvio, Marco, Roberto, and Emanuel.
I have to say thank you also to the two basketball teams I played in these three years, Pallacanestro Andria and Polisportiva San Donato because they were important to let me disconnect my mind from studies sometimes.
Finally, many thanks to my relatives who also supported me during this journey. In particular, I want to dedicate this achievement to Nonno Franco, since I know he's looking after me from wherever he is right now.