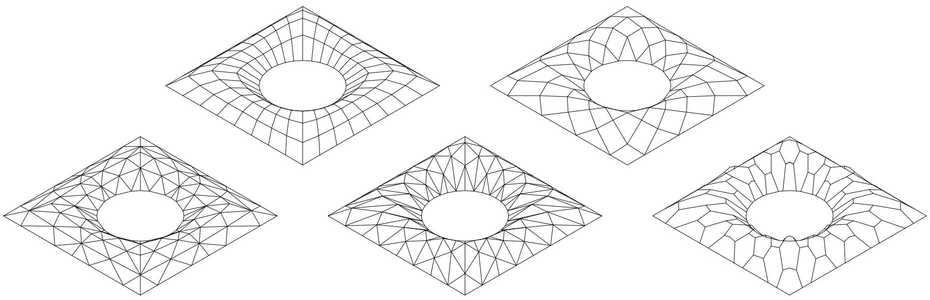
Design framework for optimized gridshell structures
– and their potential applications



Master Thesis by Rick Persson
Chalmers University of Technology
Department of Architecture and Civil Engineering
Supervisor: Kengo Skorick
Examiner: Jonas Lundberg

Design framework for optimized gridshell structures
– and their potential applications

Author: Rick Persson
Contact: rickp@student.chalmers.se

Chalmers University of Technology
Department of Architecture and Civil Engineering
Master Thesis in Architecture and Urban Design
MPARC ACEX35 Material Turn 2023

Supervisor: Kengo Skorick
Examiner: Jonas Lundberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Abstract

This master thesis investigates the design process of gridshell structures constructed from discretized elements, with a focus on optimizing the design through the exploration of key considerations regarding geometry, structure, and other potential challenges. The research results in a design handbook that provides valuable insight for architects and engineers, that would like to design a gridshell. Additionally, the thesis explores the potential of gridshell technology beyond typical applications with a concluding demonstration of the contents through the development of a bridge prototype that demonstrates the possibilities of gridshell technology.

Purpose & aim

The purpose of this master thesis is to collect design logic regarding gridshells, which are otherwise scattered and mostly very specific, under a common roof that is geometry. The aim is to create a comprehensive, and easy-to-read introduction to gridshell design. Furthermore, the thesis aims at expanding on the field of application of gridshell technology, through speculation and hypothesizing on a prototype outside the common field of application.

Student background

The author of this thesis has a background in the "Arkitektur och teknik" program, which is a combined architecture and civil engineering program offered by the Department of Architecture and Civil Engineering at Chalmers in Gothenburg, Sweden.

Since the fall of 2006, the program has provided students with the opportunity to pursue a combined degree in architecture and civil engineering. This initiative arose from the industry's demand for architects with technical knowledge and engineers with a design-oriented approach. The program aims to bridge the gap between these two fields and comprises a total of 300 higher education credits, equivalent to 300 ECTS points. The bachelor's degree accounts for 180 credits. Upon completing the bachelor's degree, students have the option to apply for a master's degree in Architecture, Civil Engineering, or both.

This thesis was written after the author completed the bachelor's examination and accumulated sufficient credits from both of the master's programs (Architecture and Urban Design, MSc and Structural Engineering and Building Technology, MSc) for a double master's degree.

2017-2020	BA, Architecture and Civil Engineering *Chalmers University of Technology*
2020-2021	MSc, Structural Engineering and Building Technology *Chalmers University of Technology*
2021-2022	MSc, Architecture and Urban Design *Chalmers University of Technology*
2022-2023	MSc, Structural Engineering and Building Technology MSc, Architecture and Urban Design *Chalmers University of Technology*
2023 (spring) *2023 (Autumn)*	MSc, Architecture and Urban Design (Master thesis) MSc, Structural Engineering and Building Technology (Master thesis) *Chalmers University of Technology*

Reading instructions

This master thesis is thought of as a handbook or a collection of general design ideas, and strategies that can aid in the design of gridshell structures. Think of this work as a compilation, from which you can take what's useful and apply it according to your design goals.

The thesis starts with a general description of gridshells, followed by a brief overview of the history of gridshells. A flowchart is then introduced that aids to guide the reader through the various considerations that go into the designing of gridshell structures. The order of the flowchart is not universally applicable to all design scenarios but rather outlines a general workflow for design steps. The primary design steps, which are topology, form-finding, and discrete elements can be thought of as chapters or design checkpoints. To aid the reader in furthering their design, some complimentary thoughts on materiality, connections, and structural behavior are additions to the design logic of the different gridshell types. The designs are done with Rhino and Grasshopper, but the thesis aims to keep the logic and procedures behind the designs applicable to other programs. Theorizing and discussion of the wider applications of gridshell technology are conducted through the design process of a pedestrian bridge, which aims to demonstrate some of the design procedures as well as a potential use case of the technology.

Table of contents

What are gridshells?

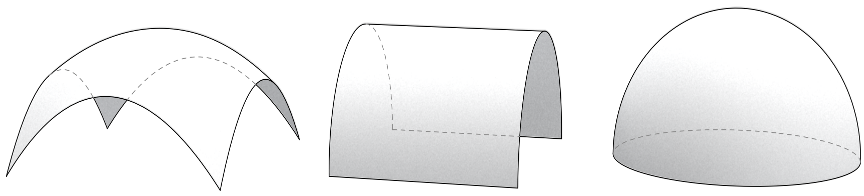


Figure 1. Hyperbolic paraboloid, vault, and dome, all of which are funicular forms.

A shell structure is a type of structural element characterized by its geometry and thinness relative to its other dimensions. Shells are mostly constructed in funicular forms, meaning that their load bearing comes from compression or tension. The term "funicular" refers to the fact that these structures rely on a balance of forces that pass through the form's shape, similar to the force equilibrium of a hanging chain. The curve that forms is a natural shape that distributes the load evenly along the entire chain.

Funicular forms are highly efficient and can span large distances without the need for intermediate supports. Funicular elements are often incorporated in the design of bridges, tunnels, and other infrastructure projects where a strong, stable structure is required. Additionally, they can be highly aesthetically pleasing, as the natural curves of funicular forms can create beautiful and unique architectural designs.

Common forms of shell structures include domes, vaults, and hyperbolic paraboloids, of which there are endless variations. A gridshell is a discrete segmented shell, a segmented shell is a shell that is divided into multiple segments or sections, and each segment is separate and distinct from the others. The term "discrete" emphasizes that each segment is clearly defined and separate from the others, as opposed to a continuous shell where there are no clear divisions between segments. Gridshell is synonymous with lattice and reticulated-shell structures, which refers to the systematic subdivision of the surface and shells where the node handles the complexity. (Leung, 2019)

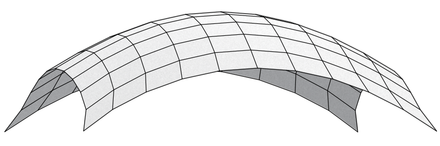


Figure 2. Discrete segmented shell with planar elements (straight segments).

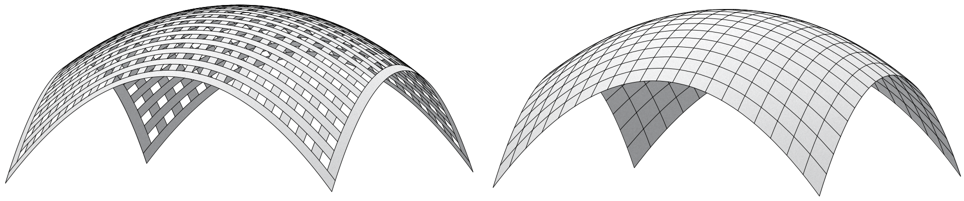


Figure 3. Unstrained and strained gridshell.

Although the term gridshell refers to discrete shell structures, it is often used to describe double curvature structures and free-form shapes more generally, as they tend to be constructed from discrete elements.

Generally speaking, gridshells can be classified into two sub-types: strained or unstrained. An unstrained gridshell consists of segments or elements that are relatively short and pass from node to node, these elements are often planar (straight) and are unstrained initially until assembled as a whole. Strained grid-shells consist of segments or elements that are longer and overlap across nodes, these elements are initially strained when assembled, and the strain is evenly dis-tributed across the whole structure when the elements are connected into a grid. Strained gridshells are limited by materials and are primarily constructed through the usage of initially straight and flexible timber laths that are bent into the desired shape. Unstrained gridshells on the other hand have a wider application range as the structure is more favorable to a wider range of materials. (Leung, 2019)

Gridshells can be used for a variety of building structures including, roofing, build-ing extensions, and facades. They are particularly well-suited for buildings with irregular shapes or complex geometries, as they can be designed to conform to virtually any shape. Additionally, gridshells are suited for sustainable design, as they are typically lightweight, easy to assemble, and can be made from renew-able or recycled materials. Shell structures are highly efficient and can span large distances with little material or create certain desired aesthetics that are hard to achieve with other structural systems. They afford flexibility in design for archi-tects and engineers, with a technology that applies to a wide range of materials. The focus in this framework will be on unstrained gridshell structures, due to them being the more versatile of the two gridshell types.

Background of gridshells in architecture and engineering

Figure 4. British Museum Great Court is one of the most prominent examples of a gridshell in steel by Buro Happold with Fosters + partners.

The history of shell structures in architecture dates back to the early 20th century and is known for its potential to provide formal yet poetic design while efficiently distributing loads. Unlike form-active structures such as cables or membranes, the shape or contours of shell structures do not change significantly under varying loads. Despite the numerous structural forms that can be achieved through shell design, differences in efficiency arise due to limitations in material choices and other context-sensitive parameters.

Frei Otto is commonly credited with inventing the strained timber gridshell and his 1976 Mannheim Multihalle is considered a prime example of this type of innovation. However, the first gridshell wasn't built by Otto.

Figure 5. Vladimir G. Shukhov.

The history of gridshell design and construction dates to the late 19th century when the Russian steel industry was expanding. To reduce the cost of custom-casting molds and joint connections, architects had to incorporate replicable elements into their designs, and engineers had to consider the most efficient way of joining prefabricated building components. Vladimir G. Shukhov (1853-1939), a premier structural engineer in Russia, was at the forefront of this movement. He used mathematical analysis to design and build roofs that required minimal materials and labor. (Leung, 2019)

Figure 6. The Roof of the Upper Trading Rows in Moscow and the central Dome portion (1890).

In 1890, in collaboration with the architect Aleksandr Pomerantsev, Shukhov engineered the iron and glass roof for the Upper Trading Rows in Moscow. The multiple barrel vaults of the department store were constructed from iron and glass, with radial diagonal cross-ties allowing for a light structure with minimum deflection.

Figure 7. The world's first doubly curved steel gridshell exhibition hall under construction in Vyksa near Nizhny Novgorod, for the All-Russia Industrial (1896).

In 1896, he was contracted to build four large gridshells as exhibition halls for the All-Russia Industrial and Art Exhibition in Nizhny, Novgorod, which became known as "roofs without trusses". Shukhov's most innovative structures at the exhibition were his designs incorporating double curvature surfaces, which he patented in 1895. Subsequently, Shukhov and his firm Bari applied his design methods to numerous structures throughout Russia such as factory buildings, warehouses, water towers, and more. His works remained largely unknown outside of Russia. (Leung, 2019)

Frei Otto (1925–2015) intended to study architecture at the Technical University of Berlin in 1943 but was conscripted into the Luftwaffe as a pilot instead. The war's atrocities of the second world war, especially aerial views of burning cities, deeply impacted his worldview. In 1945. While serving as a foot soldier, he was captured by the French and spent two years in captivity as a prisoner of war in Chartres where he served as the camp architect. This experience taught him to design with scarcity in mind, and formulated his philosophical approach to architecture in response to the Nazi regime's "architecture of killing". Otto aimed to build with a "lightness against brutality" instead of the massive Nazi monuments.

Figure 8. Frei Otto (1925–2015).

Frei Otto's passion for piloting planes inspired his later grid shell designs. He studied at TU Berlin in 1948 and spent a year in the US, where he learned about suspended grid roofs from Eero Saarinen and Fred Severud. Otto wrote his dissertation and first book, "The Suspended Roof: Form and Structure", on this subject. He designed gridshells using statics and analysis of gravity, focusing on form-finding over form-making. Otto founded the research group "Biology and Building" at TU Berlin in 1961, promoting the integration of design with ecological systems.

Otto was a proponent of "Adaptable Architecture" in response to rapid urbanization and opposed the era of concrete bunker architecture. He believed in using less material, concrete, and energy, and desired a roof covered in greenery that would blend harmoniously with the landscape. (Leung, 2019)
Although Frei Otto initially focused on steel structures with tensile designs, his interest shifted toward timber gridshells. Otto put his experience and ideas into practice with his first gridshell, made from slender timber laths, at the German Exhibition Building in Essen in 1962. In 1972, Otto and Ove Arup built a renowned free-flowing structure for the Munich Olympic Games. While some describe it as a gridshell, it is a hanging-grid structure, also categorized under tensile structures, which like gridshells are funicular forms. (Rigamonti, 2022)

Figure 9. Tensile structure for Munich Olympic Stadium (1972).

Figure 10. Mannheim Multihalle (1975).

His most significant achievement was the Multihalle at the 1975 German Federal Garden Exhibition in Mannheim, the largest self-supporting timber gridshell at the time. Despite its impact, relatively few timber gridshells have been built since. (Leung, 2019)

Frei Otto valued physical models in his design process and believed that relying solely on computer calculations was foolish in architecture. Despite their simplicity, he credited physical models for providing a more accurate approximation of reality in the design of the Mannheim Multihalle. A wire mesh model was first created to establish the basic form, followed by a more detailed hanging-chain model to determine the most efficient geometry for the roof. Although Antoni Gaudí had previously made similar models for the Sagrada Família, Otto claims that his works were inspired by logic, and collaborations and influenced by his biological approach to architecture. (Leung, 2019)

Many gridshells have been built since their introduction in the late 1800s and their evolution during the 1900s. There are well-known gridshells today that have been constructed in primarily metal, timber as well as concrete. While gridshells in timber are often tied to academic research and pavilion applications, there are a few that have been built for structural purposes. These gridshells are often categorized as strained ones and are mostly part of roofing structures.

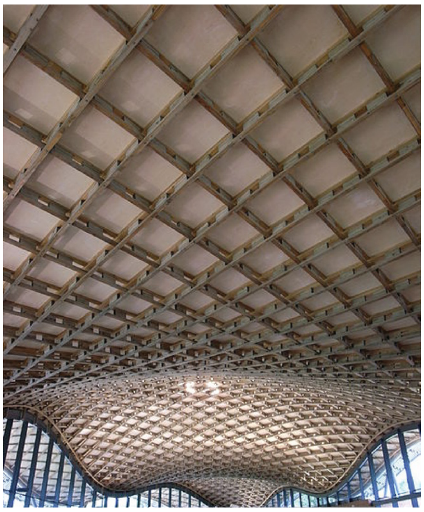


Figure 11. Savill Garden, Surrey, Great Britain (2006).

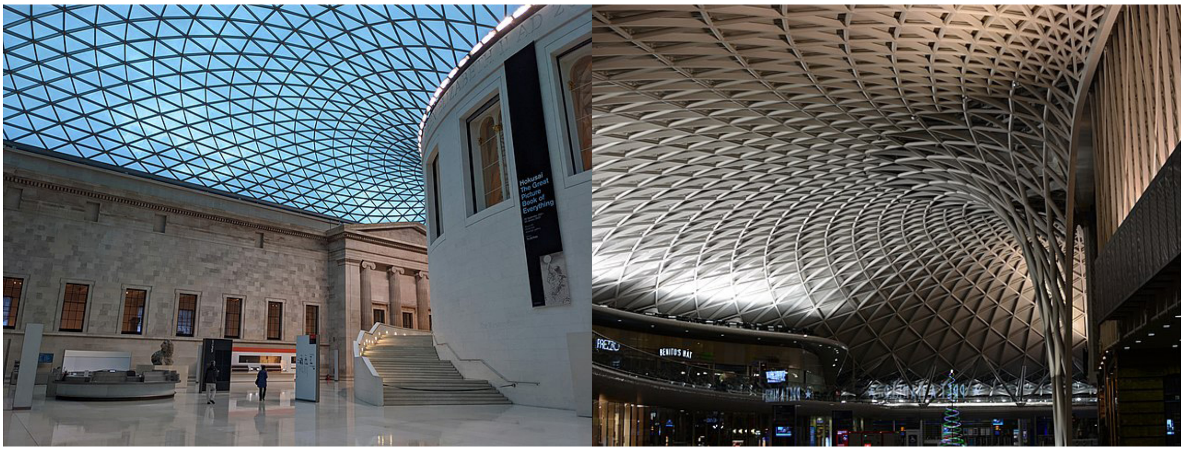


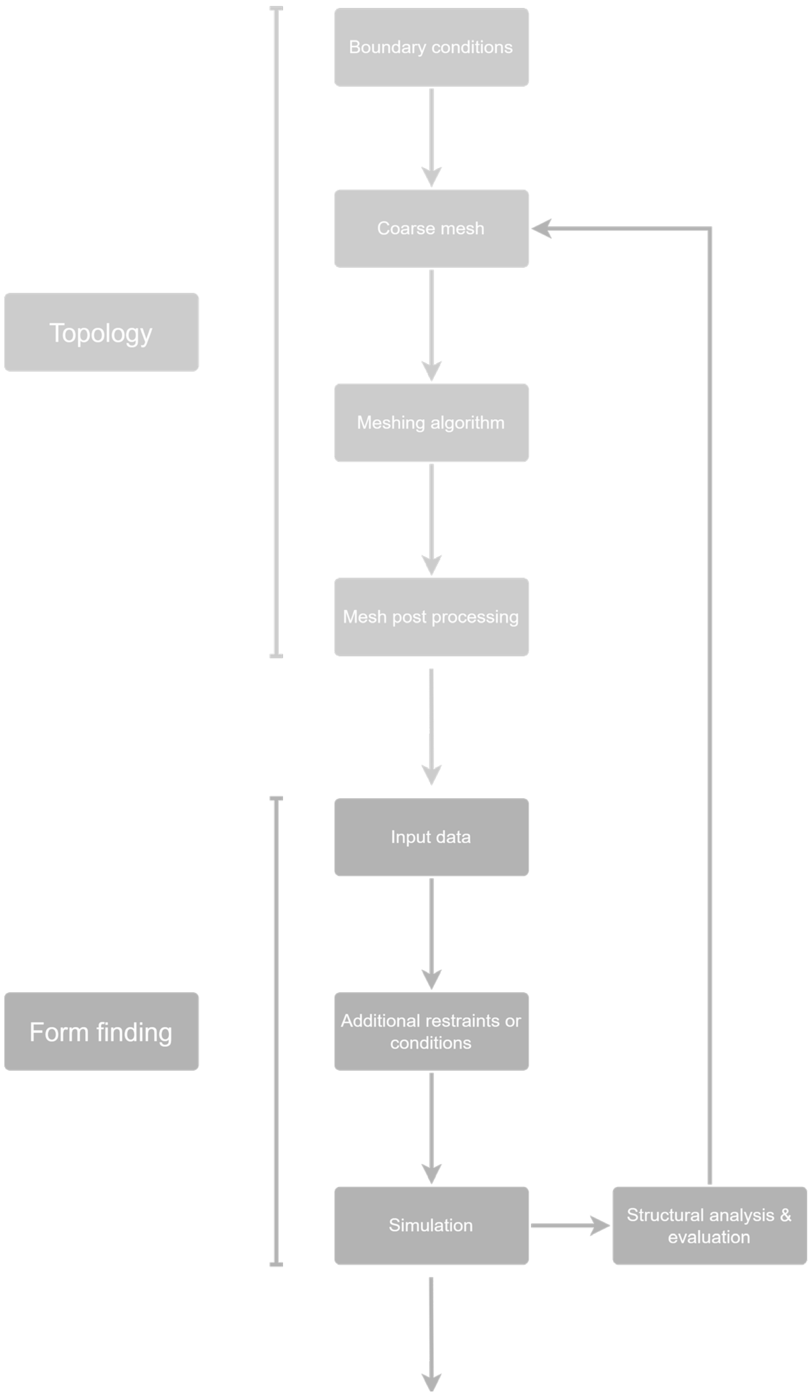
Figure 12. British Museum Great Court, London, Great Britain (2000) & King's Cross Station, London, Great Britain (2012).

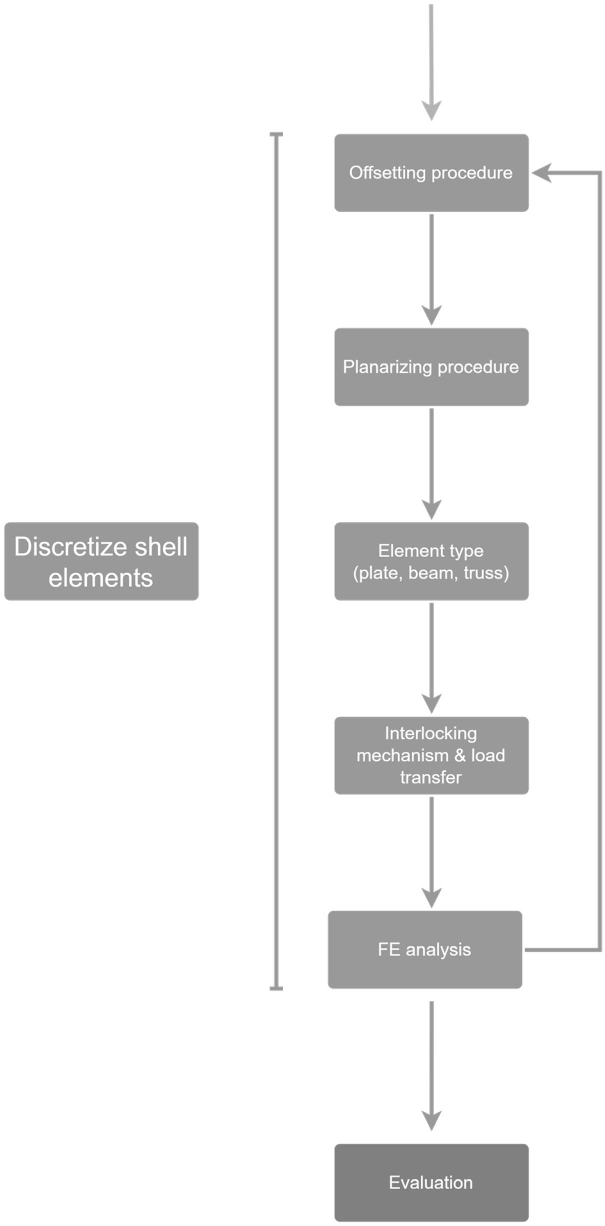
The more known gridshells are those related to steel and glass construction acting as building extensions, roofings, and facades. These gridshells tend to fall into the unstrained category, as that allows for smaller elements to be prefabricated and assembled on-site, furthermore, metal and steel connections can simplify some geometric problems. Steel has unmatched relative strength as a construction material which allows for large spans with less material and fewer structural elements. Large glazing areas or large spans are often desired when choosing gridshells in steel as a structural system.

Concrete shells are rarely discretized into segments, as casting doesn't necessitate that, resulting in a smooth structure, rather than segmented. These shells are constructed through the use of complex formwork and reinforcements.

Figure 13. L'Oceanografic, Valencia, Spain (2003).

Design flowchart




Discretize shell elements
Offsetting procedure
Planarizing procedure
Element type (plate, beam, truss)
Interlocking mechanism & load transfer
FE analysis
Evaluation

Gridshell topology

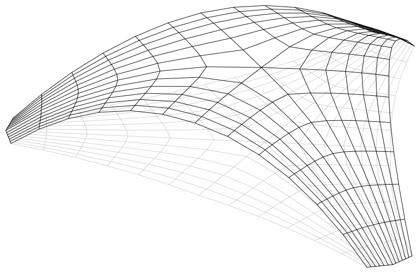


Figure 14. Illustrated gridshell topology and its respective flat topology.

The topology is an essential aspect of a gridshell structure, as it determines the overall stability and strength of the structure, it refers to the pattern and configuration of the interwoven elements that make up the gridshell. A well-designed topology ensures that the structure can withstand the forces of nature and external loads, such as wind and snow, without collapsing. It also ensures that the structure can distribute these loads evenly across its entire surface, reducing the risk of failure at any one point. In addition to providing stability and strength, a properly designed topology can also enhance the aesthetic appeal of a gridshell. The interwoven elements can create interesting patterns and shapes that can add visual interest to the structure.

For example, a gridshell can have a regular or repeating pattern of beams or plates, or it can have a more complex, irregular pattern that is designed to optimize structural performance or to create a specific aesthetic effect. The topology can also include variations in the size, shape, and orientation of the elements, as well as the use of different materials or connections. The choice of topology is determined by the design goal, the loads the structure will have to bear, and the material properties.

Overall, the topology is a crucial aspect of gridshell design and must be carefully considered during the planning and construction process to ensure that the structure is both functional and visually pleasing.

There are several ways to design the topology for gridshells, including:

Manual design: This method involves manually creating the topology of a gridshell by sketching out the different members and connections. This method can be time-consuming, but it allows for a high degree of control and creativity.

Genetic algorithms: Evolutionary algorithms use an iterative process to generate a wide range of potential topologies, which are then evaluated based on certain design criteria such as structural efficiency, material usage, and cost. This method can be used to generate many different topologies, but it can be computationally intensive.

Parametric modeling: This method involves using parametric modeling software such as Grasshopper for Rhino to create a model of the gridshell, and then using a set of parameters to control the topology of the gridshell. This method can be used to quickly generate a wide range of topologies, but it can be difficult to control the outcome.

Heuristic algorithms: Heuristic algorithms are based on problem-solving techniques that use a combination of trial and error, and rules of thumb, to find a solution. These algorithms are good at finding a good solution quickly, but they may not always find the best solution.

Machine learning: Machine learning algorithms can be trained to generate topologies based on a set of design criteria and examples. This method can be used to generate topologies that are not easily predictable by humans, but it can be difficult to understand and control the outcome.

Each of these methods has its advantages and disadvantages, and the choice of method will depend on the specific project and the design goals. The design procedure for topology design described in this thesis will combine aspects from manual design, and parametric modeling with some general rule-of-thumb approaches such as aiding load to the supports, consideration of singularities, mesh sizing, and alignment.

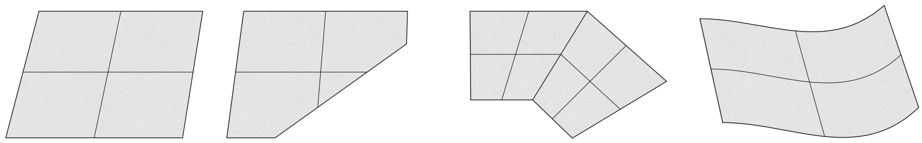


Figure 15. Untrimmed surface, trimmed surface, poly-surface, and NURBS surface with iso curves.

Starting from the boundary conditions we can develop a skeleton topology, or a coarse mesh (interchangeable), which aims to guide the desired meshing or subdivision algorithm. This is easily done by hand, by drawing the lines and lofting them together, but can also be done parametrically from line inputs. The input for most meshing algorithms are surfaces, which are then subdivided into a polygonal pattern. The surface types that are often encountered in programs such as Rhino include untrimmed surfaces, trimmed surfaces, poly-surfaces, and NURBS surfaces.

Untrimmed surfaces are the simplest type of surfaces and are constructed from 4 points, quadrilateral shape. For example, surfaces from 3 points will have the underlying geometric shape of a 4-point quadrilateral shape that has been trimmed.

Trimmed surfaces are surfaces that are trimmed or split by curves or other surfaces, this allows for more complex shapes that cannot be created with a single surface. Important to understand is that a trimmed surface has its untrimmed surfaces underlying it, defining its geometric shape which is the shape that is considered when applying meshing algorithms, they are therefore to be avoided to apply meshing procedures.

Poly-surfaces consist of two or more joined surfaces, trimmed or untrimmed, and act as such respectively.

NURBS (Non-Uniform Rational B-Spline) surfaces are a type of mathematical surface used in computer graphics and modeling. They are created using a series of control points and curves that define the shape of the surface. NURBS surfaces allow for a high degree of flexibility and control over the final shape of the surface without needing to be trimmed.

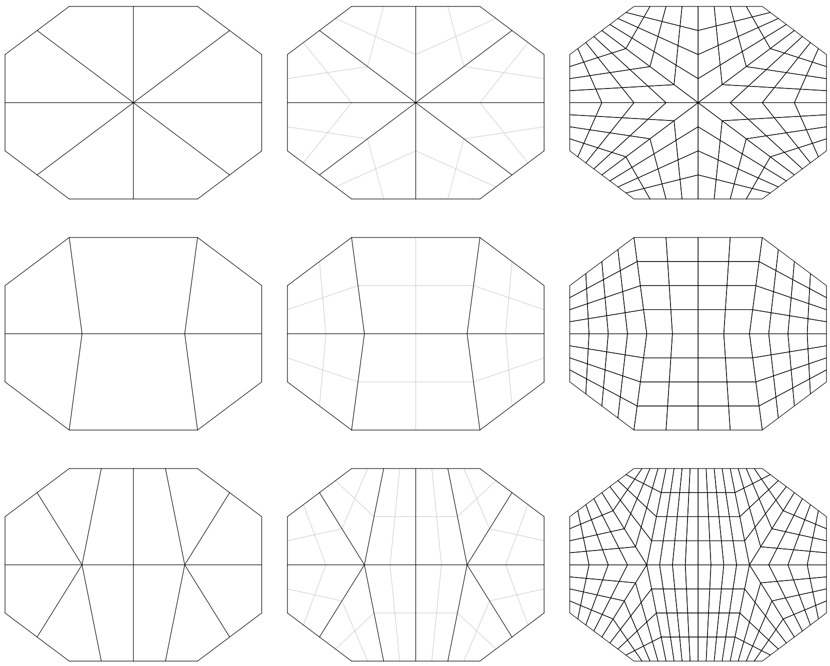


Figure 16. The first column shows three different coarse mesh configurations, the second column shows the iso curves of the surfaces and the third column shows the same meshing refinement applied to each of the coarse meshes.

Because most of the meshing algorithms require surfaces, it is a good starting point to have the coarse consisting of individual surfaces that are created by lofting (with direction in mind). In Figure 16 we can see how the Isocurves, in grey, on surfaces, can act as a guide, as the meshing algorithm will follow the underlying UV structure. In Figure 16 all the surfaces have the same UV subdivision to illustrate the relation between the flat topologies and the coarse meshes. The surfaces can also be joined and translated to meshes if the algorithm for the subdivision or refinement would require that.

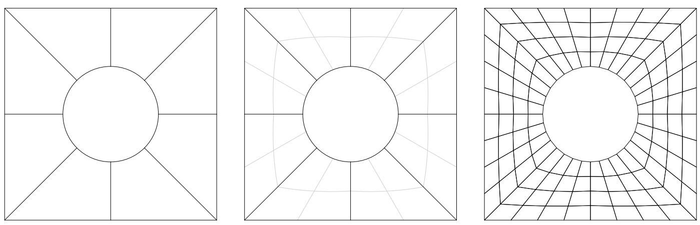


Figure 17. From left to right: NURBS surfaces created by lofting lines and curves, highlighted iso curves of the constructed surfaces, quad-subdivision of surfaces.

NURBS surfaces, which are created by lofting curves, are preferred whenever there is a curvature or an opening that must be considered for in the coarse mesh, or when the surface needed cannot be constructed from 4 points.

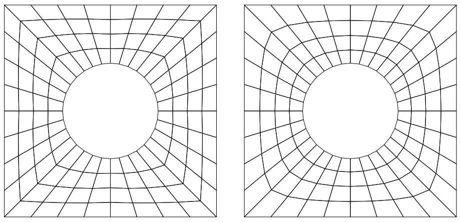


Figure 18. Mesh smoothing procedure applied to the flat topology.

As the meshing algorithms follow the underlying isocurves and UV structure of the surface, the outcome may not always be as smooth as desired. Before form-finding with the flat topology, it is sometimes desirable to smooth the mesh or relax it, if the form-found topology, or flat topology has inconsistencies or undesirable sharpness. This can be done through mesh smoothening procedures or by pre-tensioning the flat topology, without subjecting it to gravity.

As shown in Figure 19, the topology to the left is directly form-found from the flat topology, with every element having the same stiffness, allowing a minimum amount of stretching (edge lengthening) of the elements. This will result in a topology that is close to the initial flat topology, which is often desirable but can cause inconsistencies when the flat topology is suboptimal for the form. In the figure, the topology to the left has pointy naked edges, and the form is less dome-like relative to the topology to the right, derived from an equivalent pre-tensioned flat topology.

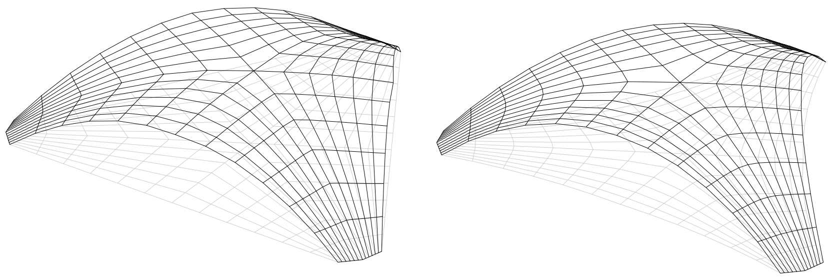


Figure 19. From left to right: topology from initial flat topology, topology from pre-tensioned flat topology.

Mesh types and considerations

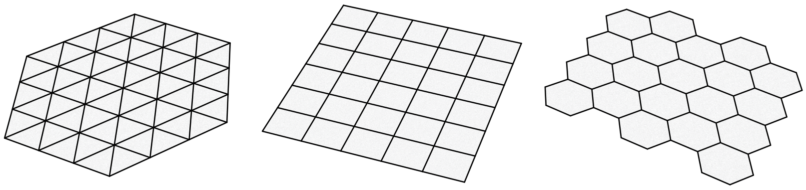


Figure 20. Common meshing algorithms are illustrated on simple surfaces.

The most common mesh procedures include triangular, quadrilateral, or hexagonal elements, and sometimes a combination of them. The regularity of the topology is important from a structural perspective and for aesthetics, statics, and fabrication. Choosing a specific mesh type will result in structural and geometric effects and each has its advantages and disadvantages.

One of the most desired traits in any type of gridshell is planar elements, Gridshells are usually thin, lightweight structures that rely on the interlocking of individual elements to provide stability and support. If the elements are not planar, they will not be able to interlock effectively, and the structure may be unstable. Furthermore, planar elements simplify the analysis of the structure by reducing the number of degrees of freedom. This makes it easier to predict the behavior of the structure under different loads and to optimize the design for specific requirements.

In addition, both planar elements and elements of the same thickness can simplify the manufacturing process and reduce the cost of production. Planar elements can be manufactured with standard cutting and forming techniques, whereas curved elements may require more specialized processes. Similarly, having elements of the same thickness helps ensure that the same cutting and forming techniques can be used for all elements, without the need for special considerations for thicker or thinner elements. This not only reduces the cost of production but also helps ensure that the elements are consistently shaped and sized, which is important for maintaining the stability and structural integrity of the gridshell.

In Figure 21 and 22, simple meshing algorithms are applied to the example coarse meshes from the previous section.

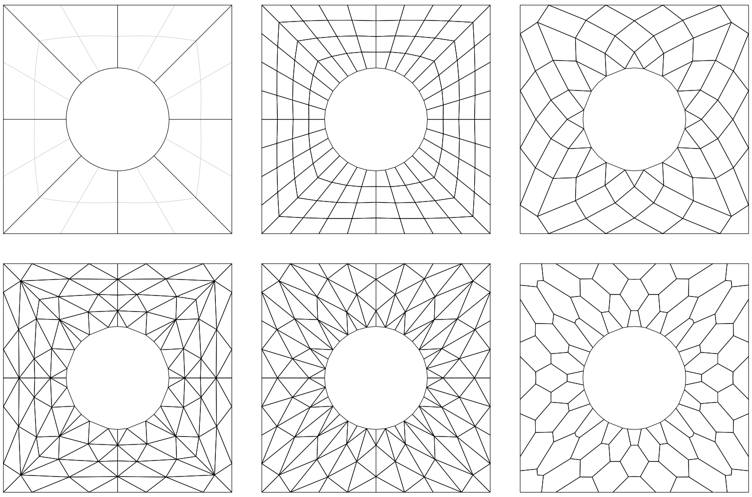


Figure 21. From left to right: coarse mesh with isocurves. Quad-subdivision. Diamond quadsubdivision. Triangular subdivision from quads. Triangular subdivision from quads from the opposite corners. Hexagonal subdivision.

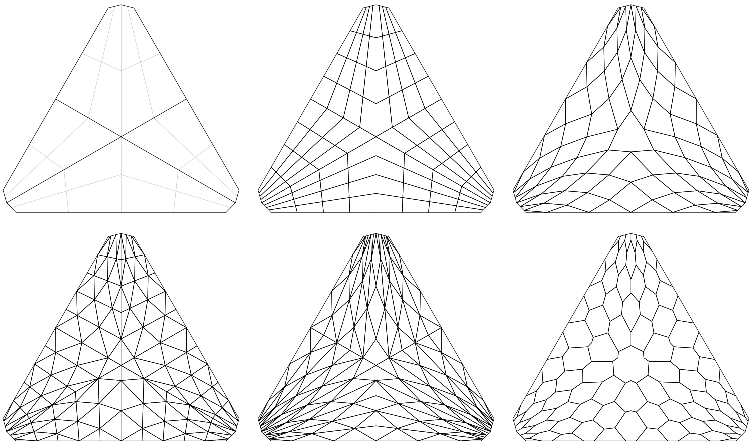


Figure 22. Same meshing procedures as in Figure 21 with a different coarse mesh.

Form-finding

There are three main design approaches, or geometries, used to form a shell structure, as explained by Sigrid Adriaenssens and co-authors in their book "Shell Structures for Architecture."

Freeform Shells - also referred to as free-curved or sculptural shells, are created without considering the structural efficiency of the form. Computer-aided modeling tools are used to shape such shells digitally, typically by higher-degree polynomials such as patches of Non-Uniform Rational Basis Splines or NURBS. The form of a freeform gridshell is often conceptualized as an aesthetic shape sculpted by the designer, as a spatial articulation of the interior program, or for phenomenological effects in and around the shell. After the form is generated, a grid of elements is geometrically arranged across the surface of the shell.

Mathematical Shells - are constructed through analytical functions rather than considering their structural efficiency. These shells are typically generated using lower-degree polynomials such as hyperboloids, ellipsoids, hyperbolic or elliptic paraboloids, or trigonometric and hyperbolic functions like the catenary. Analytically defined geometries are commonly preferred in the construction industry because of their convenience in describing the shape of a shell.

Form-Found Shells - a type of shell that is derived from the process of form-finding. This method involves controlling parameters to find an optimal geometry for a structure in static equilibrium with its self-weight. These shells may be derived from physical models or hanging shapes, as seen in the funicular structures of Antoni Gaudí, Frei Otto, and Heinz Isler. Alternatively, digital models can be used for form-finding by either simulating the physical model numerically or computing imaginary properties parametrically. The form-found shells are known for their structural efficiency and are considered one of the most innovative design methods in architecture and engineering.

Examples of these shells include the roof of the Fiera di Milano, the Berlin Hippo House Gridshell, and any of the works of Frei Otto, respectively. (Leung, 2019)

The focus of this thesis is on form-found shells, derived from physics-based simulation. While the focus is primarily on form-found shells, the procedures and techniques used in their analysis and design are also applicable to Freeform Shells.

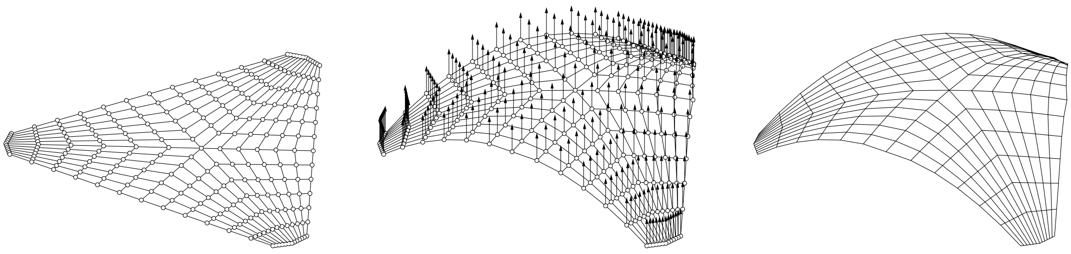


Figure 23. From left to right: flat topology mesh with highlighted vertices. A negative load is appliedevenly across vertices, pulling the Mesh upwards. Form-found topology from static equilibrium with the load, and the anchored corners.

Physics-based form finding is a computational design method that utilizes principles of physics to generate and optimize structural forms. This method involves simulating physical forces acting on a flexible structure, and iteratively adjusting the form until it reaches a state of equilibrium. Grasshopper and Kangaroo are two software programs that are commonly used in physics-based form-finding. Grasshopper is a visual programming language and Kangaroo is a physics engine plugin for Grasshopper. Together, they allow designers to simulate various physical forces on a digital model, such as gravity, tension, compression, and bending, and adjust the model's shape and form in response to these forces.

The process of physics-based form finding involves several steps. First, the designer creates a digital model of the structure using Grasshopper or Rhino (in this case, a flat topology). The model is then subjected to various physical forces using Kangaroo, which simulates the behavior of materials under these forces. The designer can then observe how the structure reacts to the forces and adjust the model's shape and form accordingly. During this iterative process, the designer can also set specific constraints and goals for the model, such as minimizing material usage, maximizing strength and stability, or achieving a desired aesthetic effect. By continuously adjusting the form of the model in response to these constraints and goals, the designer can arrive at an optimal structure that balances all these factors.

Physics-based form finding offers several advantages over traditional design methods. By simulating physical forces, designers can create structures that are more efficient, stable, and structurally sound. It also allows for the creation of complex and innovative forms that would be difficult or impossible to achieve using traditional design methods. Furthermore, this method can be used to optimize existing structures, improving their strength and durability without compromising their original design intent.

Below are some form-finding examples shown for the two different flat topologies from the previous section. Note that the mesh faces are not planar (apart from the triangular ones) and that no mesh smoothing or relaxing procedures have been applied beforehand. Furthermore, the stiffness is constant across all elements, but the load is varying to illustrate the form for similar heights. For the pavilion form, notice how the form varies between mesh types, mainly from the stiffness of the geometry.

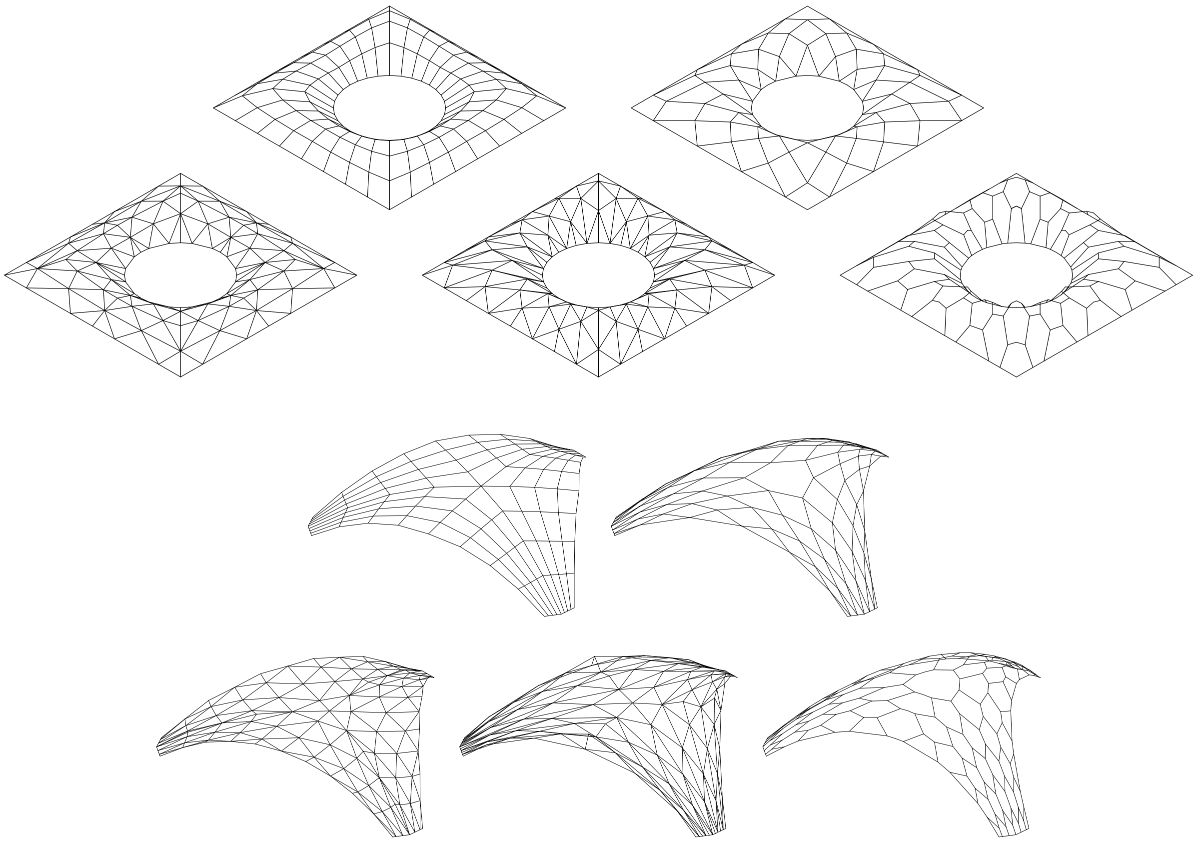


Figure 24. Physics-based form-found mesh topologies from the flat topologies of the mesh types section.

These form-found topologies are by no means perfect, especially the ones consisting of triangular elements, as their topology is too far from the "optimal" form. This is where mesh relaxing, re-meshing, or re-drawing of the coarse mesh and experience comes into play, as an understanding of the structural behavior under loading can improve the topology. Nevertheless, they aim to illustrate the translation from the topology to a physics-based found form.

Form-finding constraints

When designing with physics-based form finding, constraining the form is sometimes necessary for the design goal, and can be helpful at other times. Below are some constraints that the designer can keep in mind other than loads and boundary conditions.

Anchoring XYZ
Anchoring points or mesh vertices in only the Z-direction is particularly useful when boundary conditions or supports are not pre-determined, as it will allow the form to find an optimal structure that supports can be designed for. Keep in mind that at least some points of the boundary should be anchored in XYZ in such a case. Anchoring in XY-direction can be useful for certain design goals, where for example the form must cover a certain area or match the initial topology.

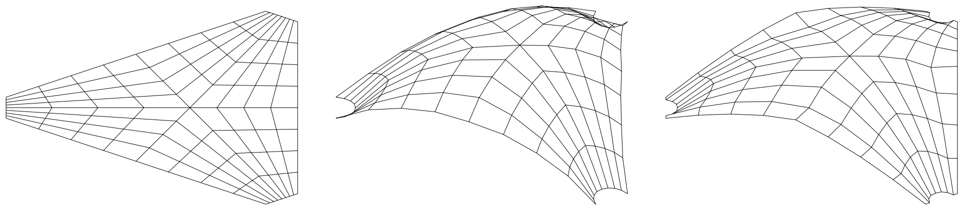


Figure 25. From left to right: initial topology. Form-found mesh topology with endpoints of the supported edge anchored in the XYZ direction with the remaining support vertices anchored in the Z direction. The same form with additional anchorage of the non-supported edges (naked edges) anchored in XY direction.

Constraining edge lengths
Constraining edge lengths relative to those of the initial flat topology is the equivalent of adding tension or compression to the element. Adjusting the stiffness is essential for controlling the form and allowing edge stretching is sometimes a necessity for allowing the form to converge when accommodating constraints, particularly planarizing constraints. Easing on the stiffness will inevitably exaggerate the form which then has to be balanced with the loads. Pointy parts can often be smoothed out by shortening edge lengths, which will in turn pull the topology together and down, which can be balanced with increased load (equivalent to adding pre-tensioning). (Piker, 2021) (Piker, 2014)

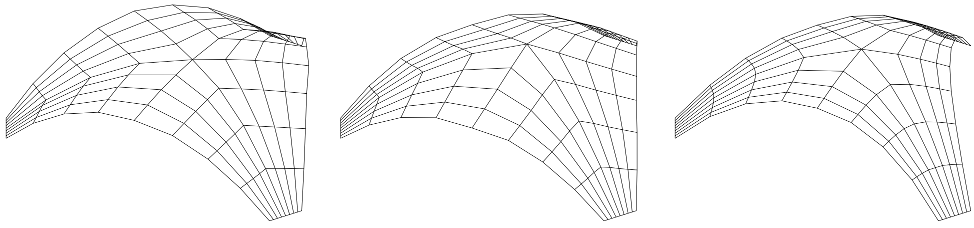


Figure 26. From left to right: form-found mesh topology from the flat topology above. Mesh topology with planarity constraint, introducing a pointy center. Mesh topology with shortened edge lengths (pre-tensioning), smoothening the mesh topology.

Co-planar or planarize

Co-planar allows a select number of vertices that span a surface or a mesh face to be planarized, this allows for a bit more freedom as the designer is able to planarize ngons (polygon of more than 4 vertices), as mesh faces consists of triangles or quadrilaterals. Planarize aims to make the input mesh faces planar (torsion free).

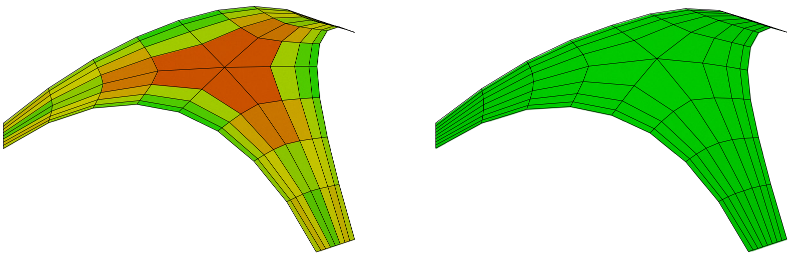


Figure 27. From left to right: form-found mesh topology and its respective planarized topology. Red to green indicates planarity deviation per face as shortest by diagonals divided by average diagonal length.

Conicalize

A quadrilateral mesh with a conical vertex layout with planar faces has an offset at a constant distance that is planar to the input and a corresponding torsion-free beam layout. This means that the beams created between the mesh outlines are planar and there is no twist in either beams or nodes. This is useful as previously mentioned for fabrication purposes and is further elaborated on in the next section. The details of conicalizing are described in 'The focal geometry of circular and conical meshes' by Pottman & Wallner (2008).

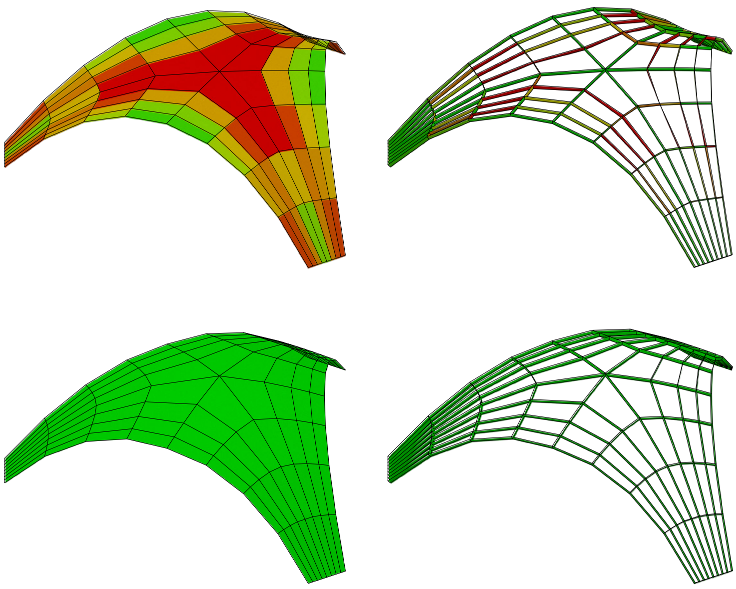


Figure 28. A form-found mesh topology and its conicalized (and planarized) form with their respective beam layouts, with red to green indicating planarity deviation per face.

Although conical meshes may seem convenient, it is not possible for all forms. A conical mesh has its faces neighboring each vertex tangent to a common cone, with closely related circular meshes having the vertices of each face sharing a common circle. Circular meshes also have a torsion-free beam layout, but at a constant distance between vertices as opposed to faces of conical meshes. (Piker, 2021) The latter is generally less useful as the beams vary in height, which is further described in the offsetting section. To construct a conical mesh from planar quadrilaterals, each vertex must satisfy the condition $\omega 1 + \omega 3 = \omega 2 + \omega 4$, where ωi are the interior angles to cyclically ordered neighboring vertices (Piker, 2022). This condition can be hard to fulfill for forms where the curvature is changing from concave to convex, and vice-versa (it is not possible for all forms).

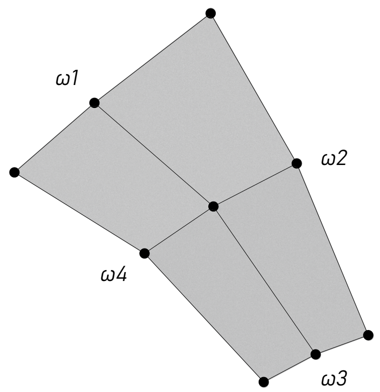


Figure 29. Mesh faces with highlighted cyclically ordered vertices, with ωi being the angle between a vertex and its neighboring vertices.

Restraining points

If for example the initial flat topology is rough and does not follow the intended supports, one can restrain mesh vertices to a nearby curve, surface, or plane of the support. Which is useful when allowing the supports to find an optimal form.

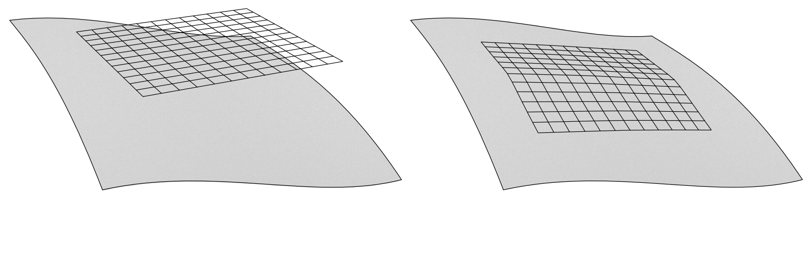


Figure 30. Mesh vertices restrained to a surface during form finding, with additional planarity constraint

Torsion-free beam layout

Assume we now have a mesh for a structure that follows a doubly curved surface that we would like to build, like the quadrilateral mesh with an opening that could resemble a roofing structure, from previous examples. The structure could be built from beam or plate elements, but regardless of the choice, beam elements would be present, either as sole elements or as edges of the plate elements.

One way to go about the construction would be to simply extrude the lines vertically, this would ensure that the beam elements or edges of the plate elements would be planar and intersect in a line at each node.

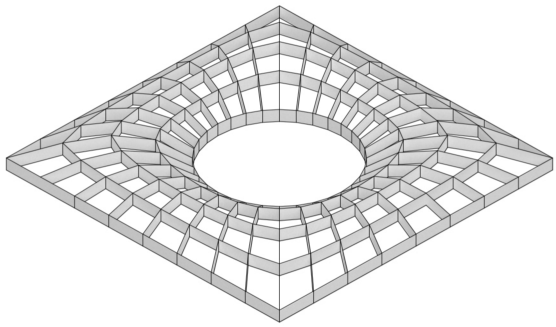


Figure 31. Beam layout from vertical extrusion of mesh edges.

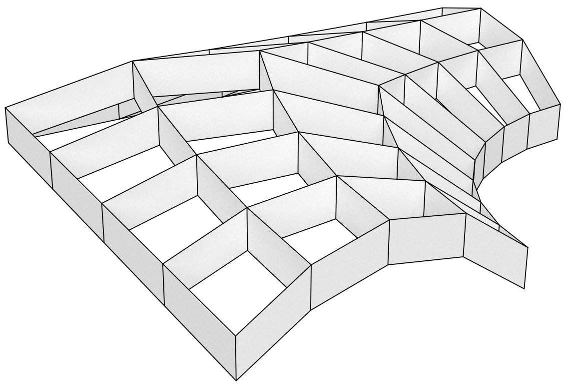


Figure 32. Isolated corner of the beam layout.

However, if we would like the elements be more perpendicular to the topology and the form. This could be due to structural considerations, such as improving load transfer, or aesthetic and fabrication reasons relating to the skewness of the beams, which becomes more pronounced as the form curves and becomes more vertical. To construct perpendicular elements to the surface, we then must turn to offsetting procedures. (Piker, 2019)

Mesh offsetting generally works by moving mesh vertices along their normal direction and a mesh is constructed from those. Generally speaking, without optimization, the offset will not have a constant distance as the offset mesh vertices do not lie in a common plane with the input mesh vertices (elaborated on further in the next section). As a result, the beams will be twisting, which as mentioned earlier will have structural drawbacks as well as fabrication complications. (Piker, 2019)

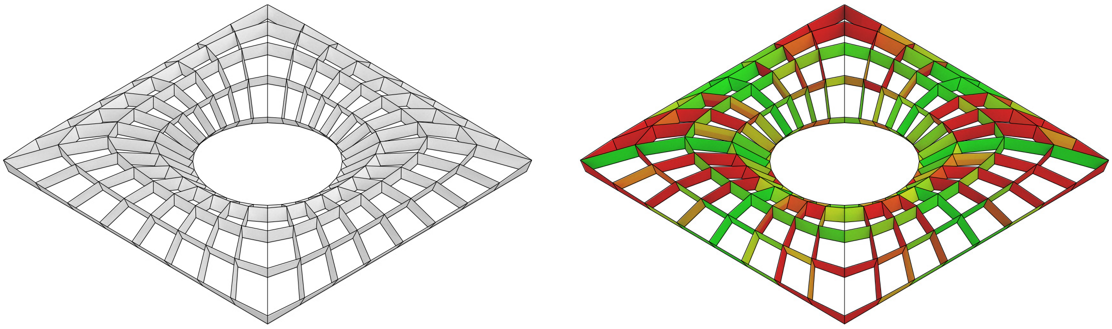


Figure 33. Beam layout from mesh offset with planarity deviation per face.

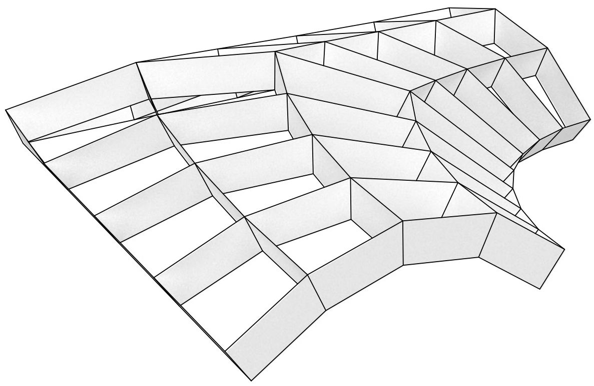


Figure 34. Isolated corner of beam layout from mesh offset

On the contrary, if we were to extrude the mesh lines along the edge normals, the resulting planar elements would lack a shared line where the beams could intersect. This can be seen in the figures below, which simply aim to highlight the inherent twist that the nodes would need to resolve in order for the beams to extend and meet in a linear fashion, emphasizing the necessity for optimization.

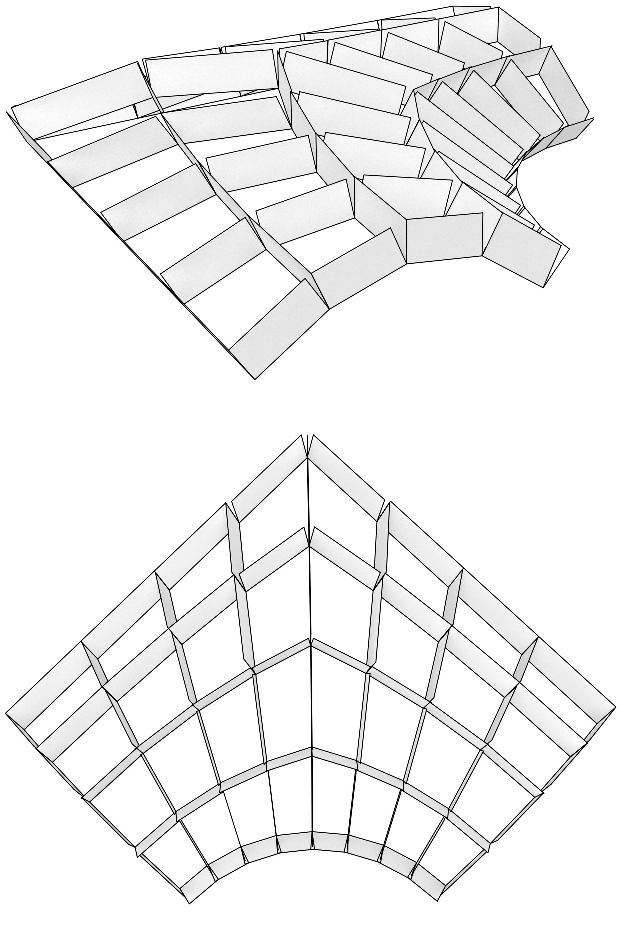


Figure 35. Beam layout from mesh edge normal extrusion of mesh edges.

Offsetting

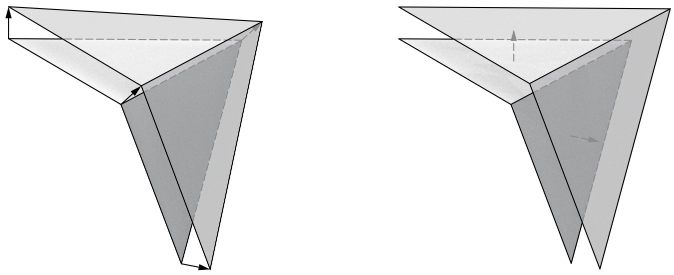


Figure 36. Vertex offset and face offset.

In architectural design, face-offsetting is utilized to create layered surfaces. The process of face-offsetting creates a new face by producing parallel face planes that are a set distance away from the original surface's face planes. This process is useful in architecture as it provides thickness or layering to architectural elements.

The specific details of the mesh, including intersection angles and face planes, play a crucial role in the appearance and performance of the final structure, making it unsuitable to modify the mesh when creating an offset mesh. Additionally, materials have a thickness, but initial concept designs are often done using two-dimensional digital surfaces. To make these designs more tangible, face-offsetting is employed to add depth and thickness. However, this operation presents a challenge in that it does not maintain the geometric properties of the original mesh, particularly when more than three planes intersect at a single point under conditions that are desired, namely, preserving the number of vertices and a constant distance offset (faces parallel to the input).

The problem with offsetting a mesh where more than three planes meet in a vertex with a constant distance is that it leads to the loss of the combinatorial structure of the original mesh. In a triangular mesh, six planes meet generically in a vertex, and offsetting such a vertex typically results in the vertex "splitting" into multiple new vertices, each of which has three incident faces. This results in the loss of important geometric properties, such as node coincidence where more than three planes meet. (Ross, et al., 2015)

Offsetting a mesh with programs like Rhino or Grasshopper, offsets the mesh in the vertex normal direction by a distance, allowing for the offset to have the same amount of vertices, but at the cost of varying distance, meaning that any mesh with a valence greater than 3 will generally have an offset that is not planar, or parallel to the input mesh.

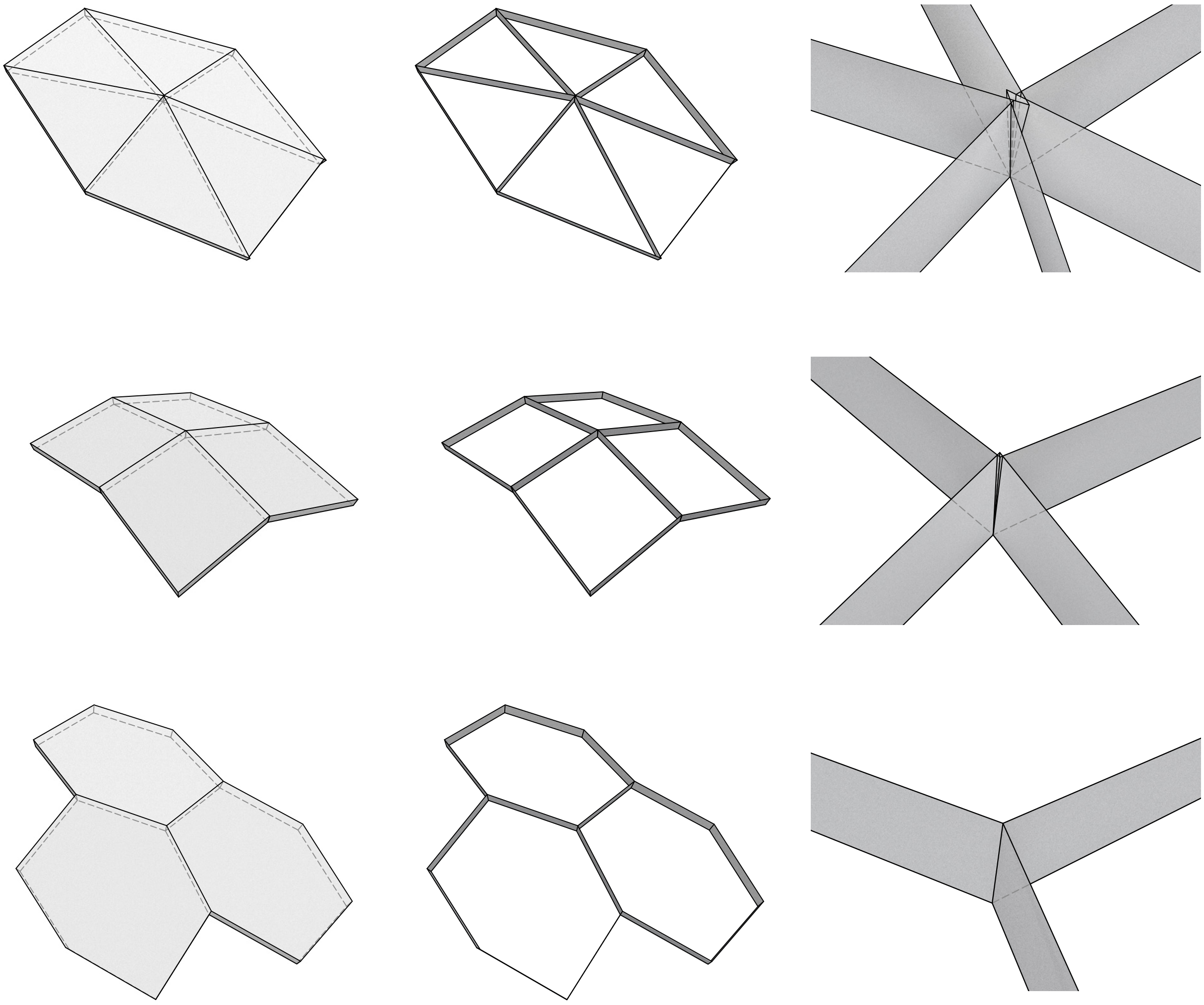


Figure 37. Planar triangular, quadrilateral, and hexagonal offset faces and their vertex interaction, keeping faces parallel to the input at a constant distance, but introducing vertex splitting when more than three planes meet.

On the other hand, face-offsetting retains faces parallel to the input, but introduces new vertices, when more than three faces meet in a vertex.

There are lots of workarounds for this geometric problem some of which include: connections that can resolve the difference in planes between elements, alternative meeting solutions resembling reciprocal frames, or optimization procedures that compromise either element height or the vertex intersection.

Discrete plate elements

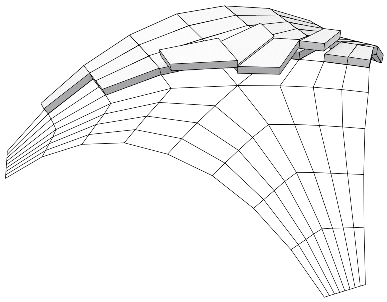


Figure 38. Quadrilateral tessellation on form-found mesh topology

Tessellation is the process of covering a plane or a surface with a repeated pattern of geometric shapes, without any overlaps or gaps. The shapes used in tessellation can be regular or irregular polygons, such as triangles, squares, hexagons, or other shapes, which can be thought of as tiling. For gridshell design, the topology or form found topology governs the underlying pattern of the tessellation or tiling procedure, which simply aims to translate the 2-dimensional form-found mesh of the topology into a 3-dimensional solid representation of the elements.

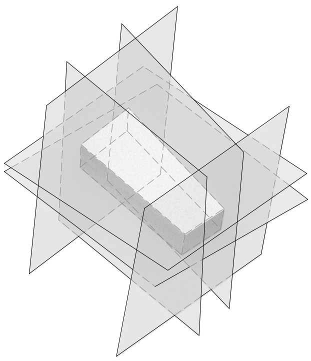


Figure 39. Singular quadrilateral element with respective surface planes.

When creating planar elements with a constant thickness, it is important to understand the implications of offsetting procedures, as mentioned in the previous section, a mesh with more than three faces surrounding any vertex generally does not have an offset solution with the same number of vertices and faces parallel to the input at a constant distance. For the procedures that follow, numerous methods of tackling these limitations will be presented and their implications. For these, the form found -topology or -mesh (used interchangeably) for a gridshell pavilion will act as a basis for which the methods assume.

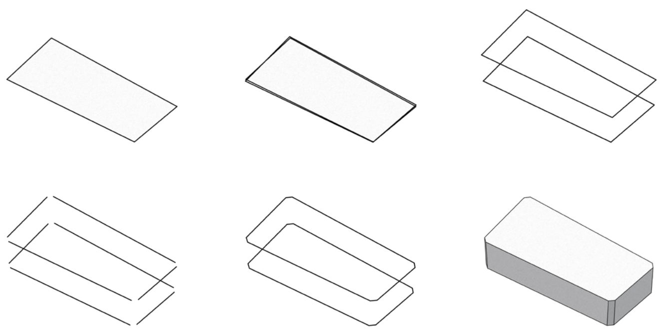


Figure 40. From left to right: edges of a singular mesh element, from the form-found mesh topology. Planarized mesh face. Edge polylines of the mesh face, from the input mesh and its planar offset. Edge polylines with removed corners. Rebuilt polylines. Solid elements, constructed from chamfered polyline edges.

From the form found mesh, planarize the mesh directly, this can be quite computationally heavy as constraints are tougher and it may require lots of iterations, keeping in mind reasonable tolerance for planarity. On the planarized mesh, apply a vertex offset that has a constant distance offset but does not restrain node coincidence, for example, Ngons (Grasshopper plugin) OffsetPlanar, and not a mesh offset. The output in this case is the polyline edges.

From the polyline edges (lines), shorten or trim them a distance sufficient for the aim or enough so that there are no potential intersections (overlapping of polyline curves). The trimmed distance can vary between the mesh and its offset (to create a cone-like meeting point for the elements), but should be constant for each element, to ensure that the created additional edge faces are planar. The endpoints of the trimmed polylines can be joined by lines so that the elements can then either be constructed from surface fitting (lofting) the polylines of the faces or the edges.

When the elements are constructed in this manner, all their faces will be planar, including those at the meeting point between elements, but they will not align perfectly, as the basis for their construction is overlapping polyline curves.

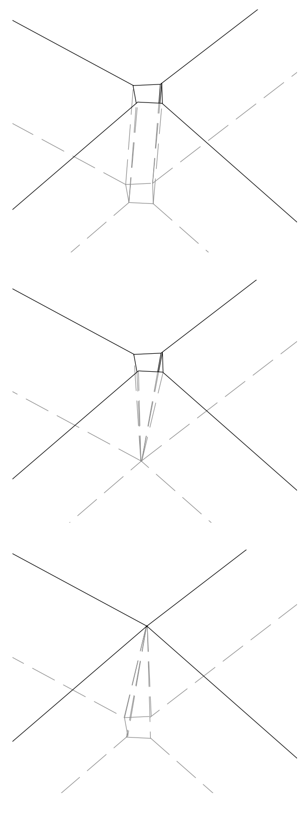


Figure 41. Possible solutions for the vertex interaction where more than 3 elements meet in a vertex, important to note is that the offset mesh needs the opening, but not the form found mesh.

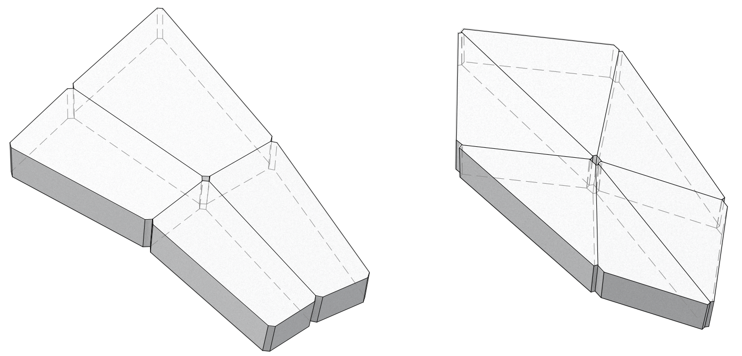


Figure 42. Quadrilateral and triangular elements with the same tessellation procedure.

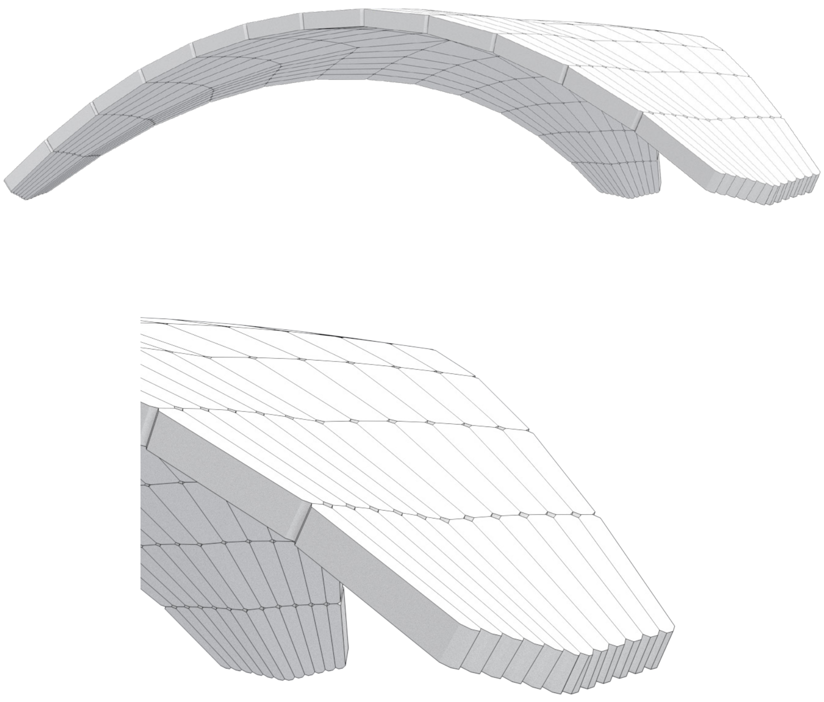


Figure 43. Gridshell from the tessellation procedure.

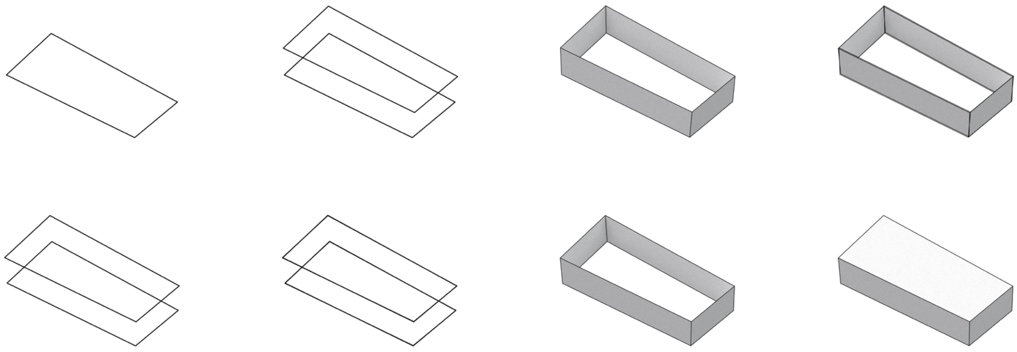


Figure 44. From left to right: edges of a singular mesh element. Edge polylines from mesh offset and its input. Lofting between the edge polylines to create beam elements. Planarized beams. Edge polylines (or face edges) of the planarized beams. Projected edge polylines to the average plane of the vertices. Solid elements, constructed from the polyline edges.

This other procedure aims to produce simple block-like elements that can be produced with as few cuts as possible, the result reminisces of masonry vaults and has a certain aesthetic where parts of the element's edges are more or less not in alignment with one another.

From the mesh edges (polylines) extrude them in their normal direction, or alternately offset the mesh (vertex offset, restraining node coincidence) and extract the polylines from there. Between the two polylines apply a surface fitting (loft), this is the basis of the edges for the gridshell elements. Planarize the edge elements through the desired method, for example through a physics engine form-finding with constraints.

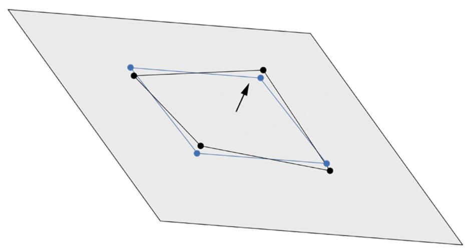


Figure 45. Illustration of plane fitting through the vertices of the non-planar mesh face and the projection of the polyline of the edges (blue) onto the average plane of those vertices (black).

Now that the edge surfaces or meshes are planar, they will act as a basis from which we construct the faces of the element. The polyline of the edge surfaces is then projected onto the average plane of the vertices of the polyline. Figure 17 illustrates how one of the two polyline curves is projected onto the average plane of the vertices. The new polyline curves are the outlines of the top and bottom of each element, from them, the beams can be constructed, which will remain planar (Vestartas, 2018)

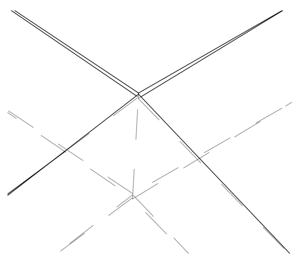


Figure 46 Node interaction.

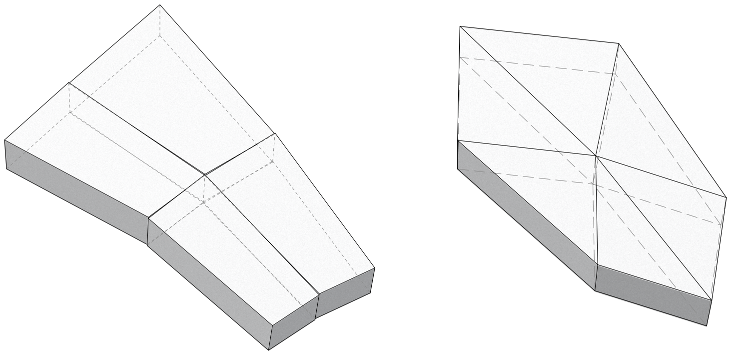


Figure 47. Quadrilateral and triangular elements with the same tessellation procedure.

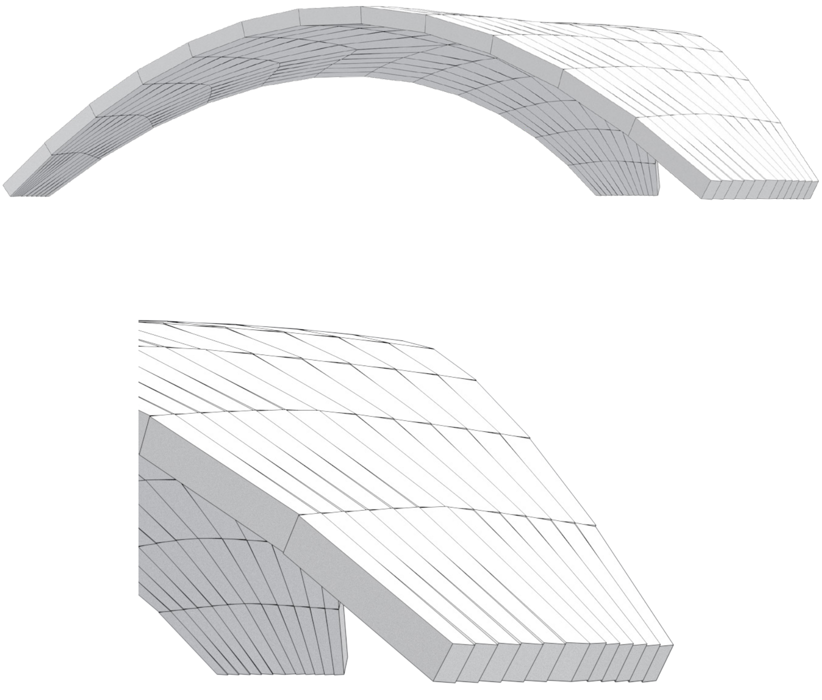


Figure 48. Gridshell from the tessellation procedure.

The result is a gridshell where elements can meet in a “line” rather than a point, by allowing individual face vertices. When doing this for triangular elements, they will have aligned edges, as the average plane of the vertices will be the same as the face of the triangular element, as there is only one plane that can be constructed from three vertices. For triangular elements, the same result will be had by simply building the elements from the planarized edges.

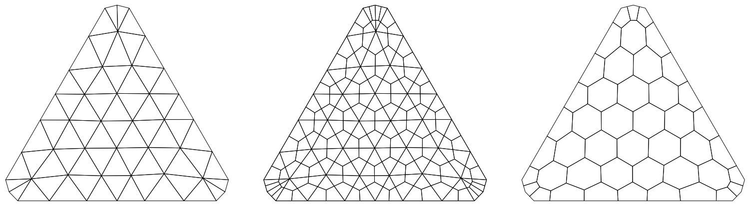


Figure 49. Flat topology design procedure for hexagons, from left to right: near an equilateral triangular subdivision, mesh dual of the subdivision, and the result

The two previous procedures are aimed at topologies with greater valence than 3, for hexagons, as the tessellation process does not require some workaround in terms of the node interaction. The design is more straightforward forward although hexagons are harder to work with when it comes to the topology design. For generating the flat topology from a coarse mesh, a good starting point would be from equilateral triangles as they can be translated to equilateral hexagons by the mesh dual. The mesh dual is formed by connecting the centroids of adjacent polygons (mesh faces) in the original mesh to form new polygons and a so-called ngon mesh (mesh constituting polygons with more than 4 sides). The resulting mesh is also known as the "dual mesh" or "dual graph." To the form-finding, co-planarity is added to restrain the vertices of each hexagon or mesh face to the same plane, and the resulting topology is then offset in the vertex normal direction of the intersection of the bisecting planes between each of the surrounding faces (Piker, 2019). For this offsetting procedure, there are specific plugins and components such as planar offset from the "ngon" plugin for Grasshopper, as mentioned earlier. This will yield a torsion-free beam layout (planar edges of each element), with a constant distance to the input.

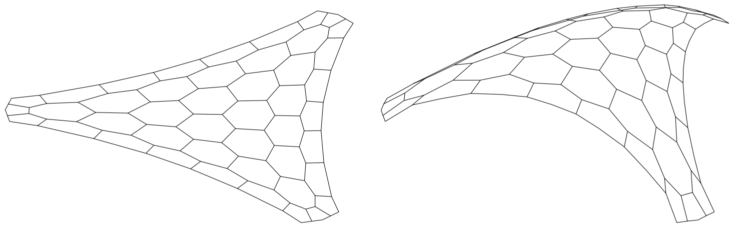


Figure 50. Pre-tensioned flat topology of hexagons and the form-found topology.

Note that it can be hard to retain the hexagonal form of certain elements during form finding because the form finding is physics-based, and hexagons lack inherent bracing. This often results in deformed hexagons, especially when the form finding must consider co-planarity or other demanding constraints.

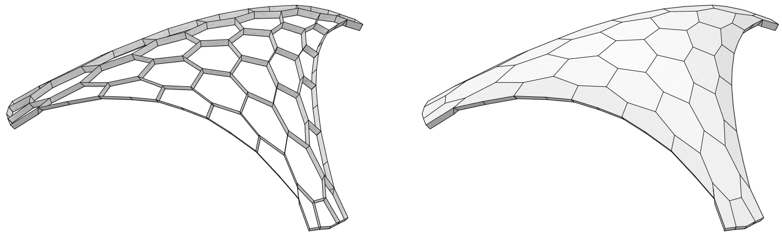


Figure 51. Torsion-free beam layout, and plate elements constructed from them.

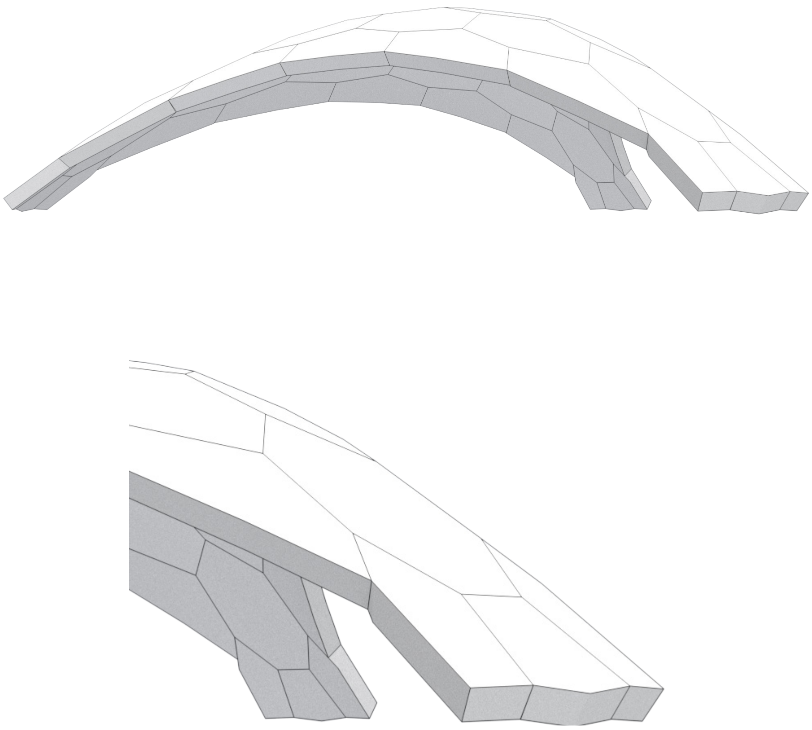


Figure 52. Gridshell from the tessellation procedure.

Discrete beam elements

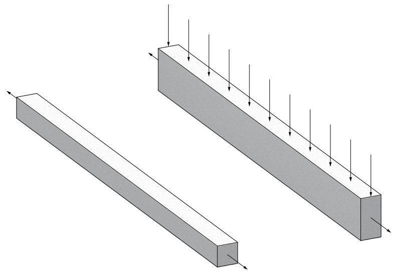


Figure 53. Axially loaded bar, and axially loaded beam with transverse load

When designing beam or truss elements, one must be more considerate of the topology, as the translation of forces will act a lot differently between plate elements and bar and beams respectively.

In structural mechanics, a bar (or truss, as in a frame of trusses) is an elongated body that translates loads through axial forces only. When designing systems or frames of trusses, all loads are translated to tension and compression, which a well-designed truss system is highly efficient for. A beam element is much like a bar element, with the difference being that the beam can translate transverse loads, through its bending stiffness. That being said, it should be considered that the elements at hand are beams if the topology does not resemble a system of trusses, meaning that the system is braced, and loads are transferred through nodes between elements.
When considering meshing or subdivision algorithms, triangular meshes are a favorable option for their geometry-based bracing of the system. Furthermore, triangles are always planar and can be constructed from quadrilaterals which are generally easier to work with topology-wise. For quadrilaterals and ngons, bracing of the system is part of the optimization, when working with bar or beam elements. To brace the system will practically mean adding diagonal members, in essence, this means triangulating the topology.

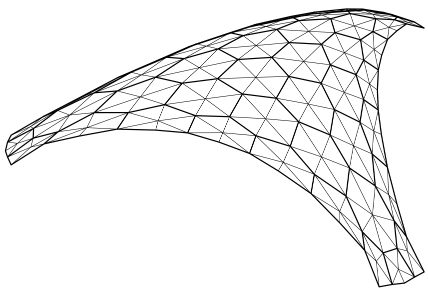


Figure 54. Bracing of hexagonal gridshell

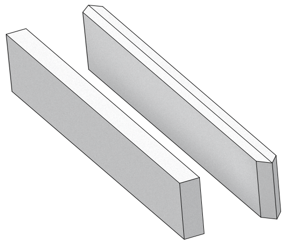


Figure 55. Regular and kink-angled beam.

Whether beams or bars are chosen as the element to go forth with, one has to consider the cross-section of those elements. From here on, bars and beams will be referred to simply as beam elements. The most common approach for a beam layout is to have a shared beam between two faces, for these beams to have a common meeting in a single line with a constant thickness, the beams must have a kink angle. The kink angle is the result of accounting for the two planes of the opposite faces of the beam element. The angled beam is useful in the regard that it can have flat rigid panels or coverings laying on top of the frame of the beams. For regular beams, there is a need for costly elements or procedures that would have the panels inserted between the beams, or on top of them in a folded type of fashion as described in 'Gridshells without kink angle between beams and cladding panels' (Tellier, et al., 2018). This is not to say that one requires less fabrication time or costs, but rather to highlight the choice that must be considered, in terms of context and design goals.

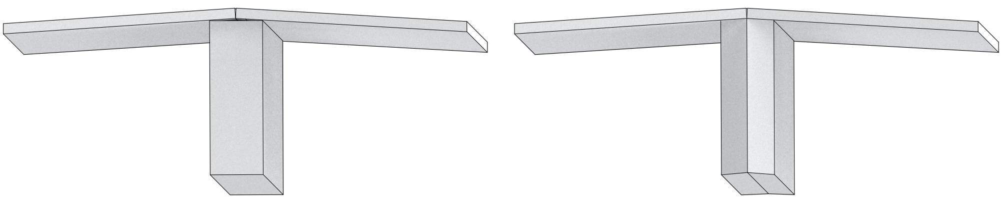


Figure 56. Interaction between the beam and kink-angled beam with planar cover

On the following pages, design procedures are outlined for "one common line" connections, favorable for timber construction, and two beam procedures more favorable for metal or steel structures. These procedures assume a torsion-free beam layout from the form found mesh.

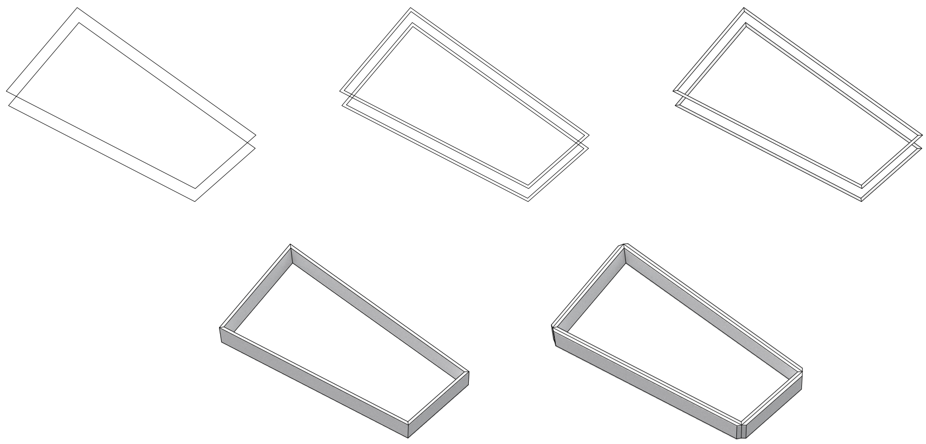


Figure 57. Kink-angled beam procedure, from left to right: polyline edges of the mesh and its planar offset. The inward offset of the polylines and the input polylines. Connecting the vertices of the corners. From these lines, the beam elements are constructed by lofting. A frame of beam elements with the respective neighboring elements.

The following procedure is done for a quadrilateral conical mesh but is also applicable to meshes of planar hexagons and will result in kink-angled beams with a constant thickness, that meet in a single line.

From the mesh lines of the input and the offset mesh, offset the lines inwards by their planes and loft them together. Now that the top and bottom for half a beam element are created, extract the lines of the edges, and loft them together. Either cap them or join them with the top and bottom to create half of a beam. The beams can be seen as shared beams when joined with the opposing faces beam or as beams for each respective modular frame.

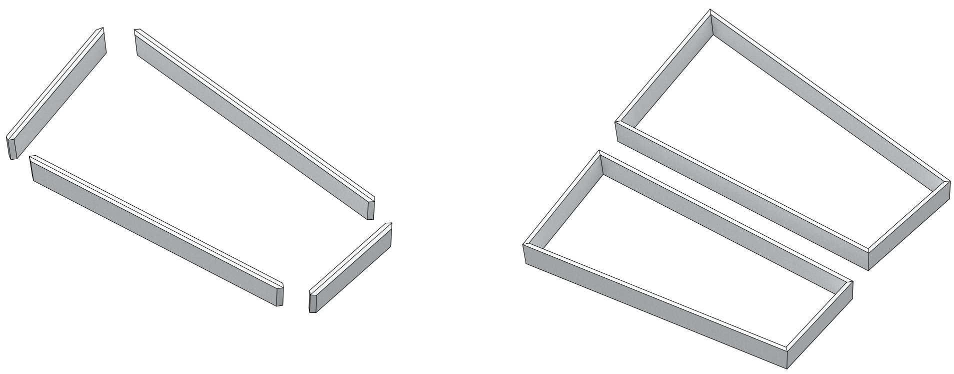


Figure 58. Kink-angled beams joined to singular beams and joined as a modular frame

Designing the beams as parts of modular frames can be beneficial in terms of the amount of work needed on site. It is suitable for timber gridshells and allows the modules to be bolted together, making for a simple connection. Steel and metal gridshells most likely need the beam elements to be welded together.

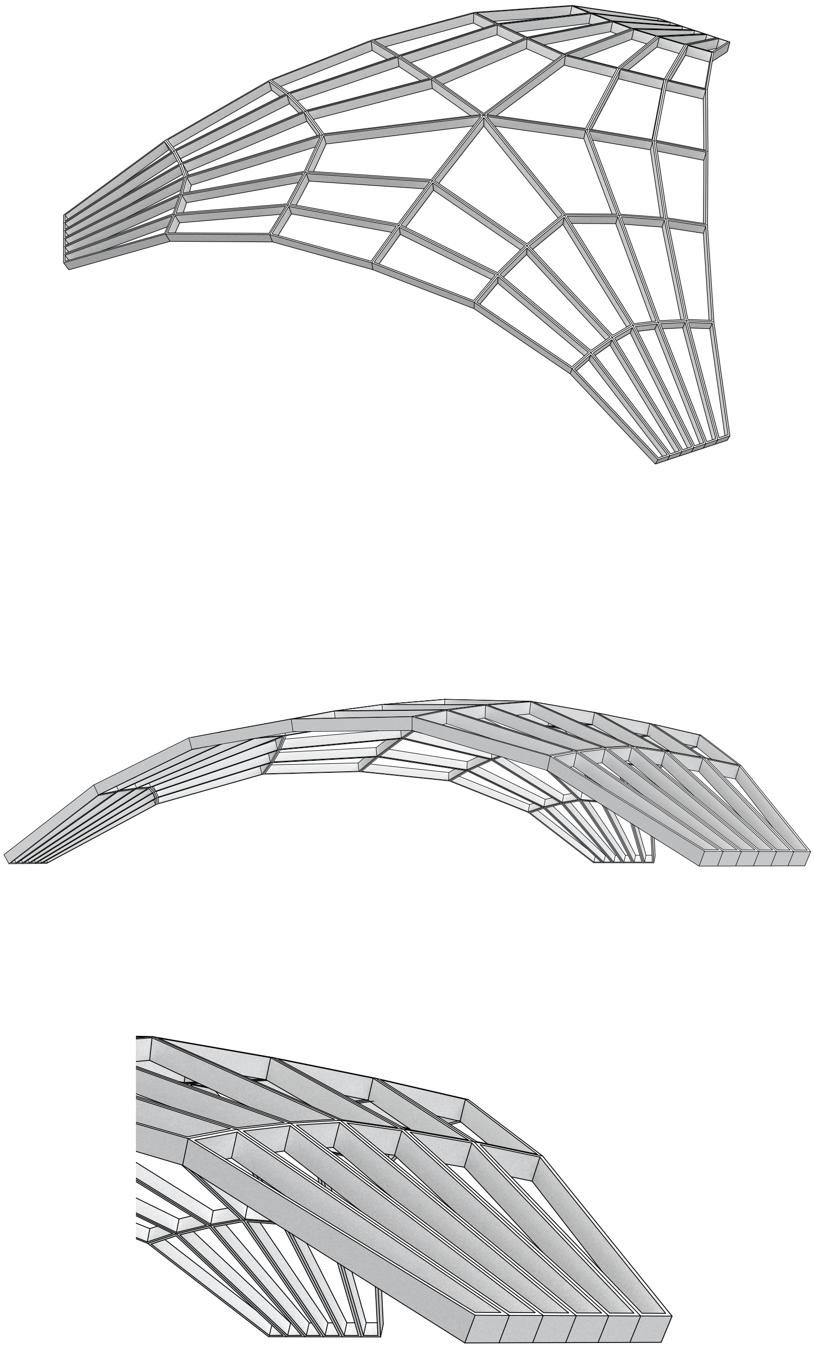


Figure 59. Gridshell from the kink-angled beam procedure.

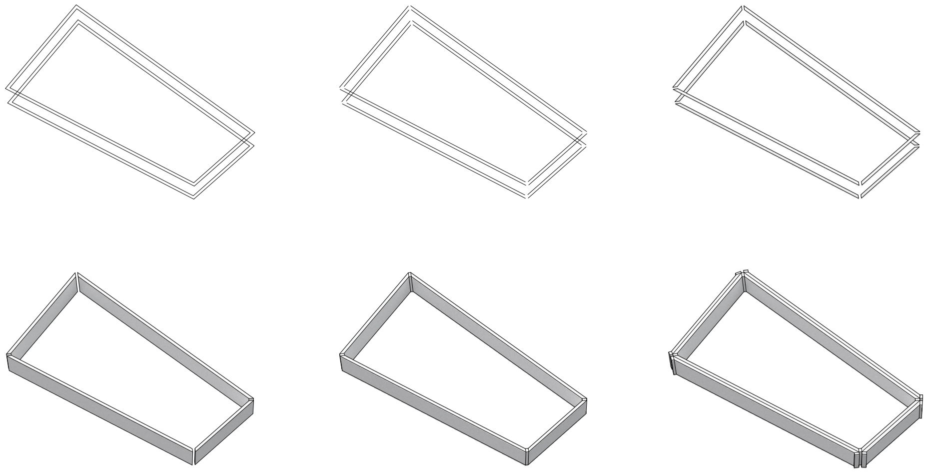


Figure 60. Adding weld to kink-angled beams, from left to right: Polyline edges of the mesh and its planar offset, with the inward offset polylines. Removed edges, by shortening of the lines. Lofting of the shortened lines and their respective shortened offset lines. Beams are constructed from the top and bottom edges by lofting. The welding is created by lofting the edges of the top and bottom faces of the beam elements. A frame of beam elements with the respective neighboring elements and the weld interaction.

Creating a welded connection for a kink angle beam layout requires only a few more design steps than the previous one. From the inwards offset of the mesh lines, shorten both the assumed distance of the weld radius and loft the shortened lines with their respective inward offset. Just like above, you can then loft and then cap the Brep edge of the constructed top and bottom surfaces of the beams to create half of a beam. To construct the welding between the beams, connect the start or end points of the shortened lines and their respective offset, and again, extract the edge lines from those to construct the weld by lofting and capping.

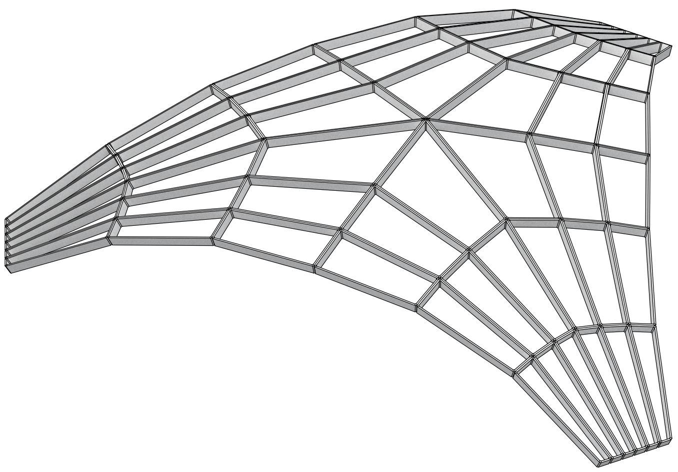


Figure 61. Gridshell from the kink-angled beam with weld procedure.

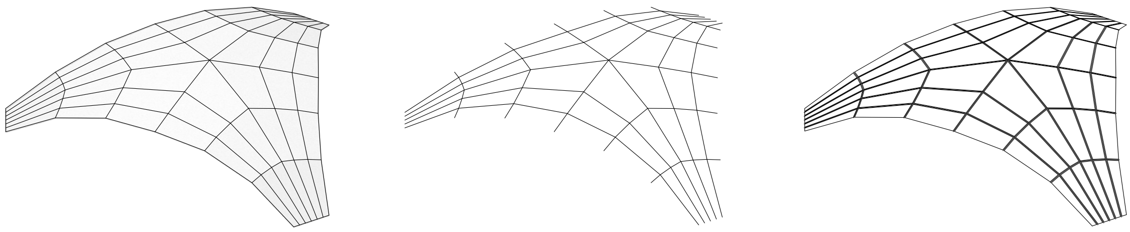


Figure 62. From left to right: joined surfaces created from mesh face edges (to avoid duplicate lines of the edges). Interior edge lines. Interior edge lines and the inwards offset polylines.

Creating non-kink-angled beams with the “simplest” connection for a torsion-free beam layout is a bit more work than the previous examples. The logic used is as follows: Extract the interior Brep edges from the joined faces, extracted from the mesh polylines, and pair each of the lines with the two closest lines from the inward offset. When paired up, these can be lofted together to create the top and bottom of the beam element. Like the previous procedures, the Brep edges of these can then be used to construct the beams.

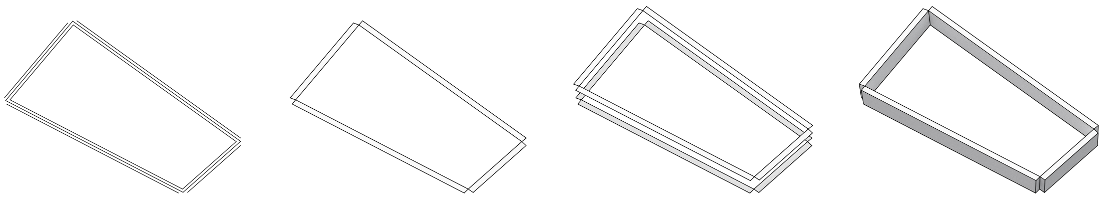


Figure 63. From left to right: paired offset lines (two closest) to each edge of the joined surface faces. Lofting between the paired offset lines. Top and bottom of the beam elements. Solid beams are constructed from the top and bottom edges.

The naked edge elements are created separately as they cannot be constructed by the logic above due to them not having two offset curves opposite of the naked edge lines. Instead, sort the inward offset lines that are closest to the naked edges (as these cannot be extracted from Breps) and loft the top and bottom together, and then simply extrude them outwards in the normal direction of the surface, twice the amount of the inward offset to construct beams with the same width as the interior ones.

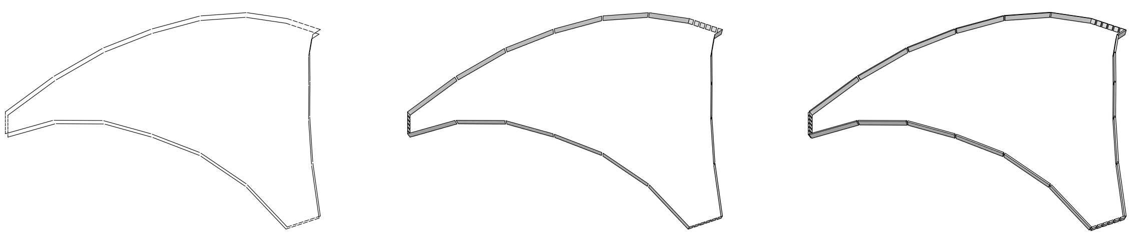


Figure 64. From left to right: sorted closest offset lines to the naked edges. Lofting between the sorted top and bottom lines. Solid beams are created from the normal extrusion of the constructed edge surfaces.

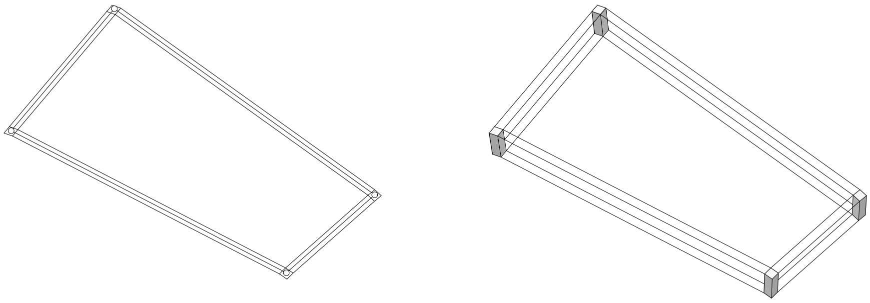


Figure 65. From left to right: mesh edges with highlighted vertices, around which the lines of the short edges of the beam elements are sorted. Lofting between the polylines constructed from the vertices of the short edges of each beam.

From these naked edge elements, extract the lines from the top and bottom part of each, and merge them with the lines used to create the top and bottom elements (the inward offset lines), respectively.

To create the joint or welded node, extract each node point, and make sure there are no duplicates. Sort the lines (exploded polylines) of the inward offsets (which should now include the outer and inner elements, see Figure x), keeping only the closest set of lines to each node. From the lines extract the points that make up them, and create a closed polyline through them, this will ensure that welding is made for corner elements as well as those on the naked edges.

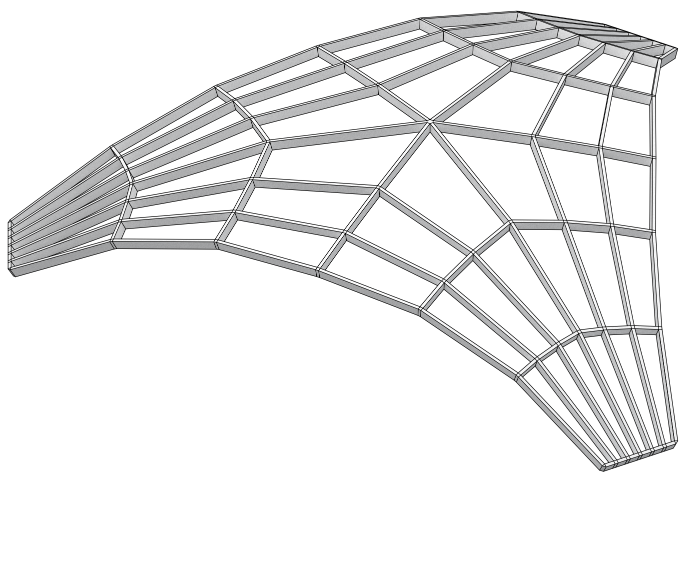


Figure 66. Gridshell from torsion-free beam layout procedure.

Connections

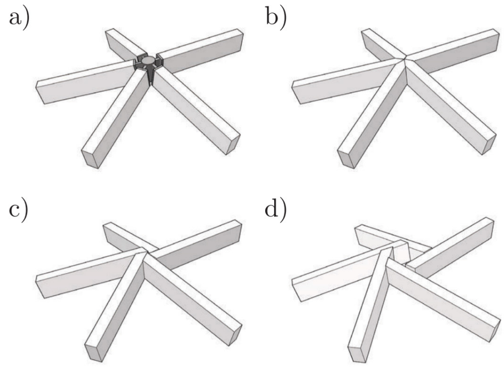


Figure 67. Knot expansion: a) a knot with multiple bars meeting in one point with or b) without an additional joint element, c) an expanded knot with a minimum knot expansion d) a larger knot expansion (Apolinarska, 2018).

For beam or truss elements there are numerous connection types to choose from, as mentioned in the offsetting section, as answers to the geometrical problems of offsetting.

The first and most straightforward solution, a) is to have a connection that resolves the twist in the node. This is suitable for both timber and steel structures and can often be made seamless for the latter. Modeling such a structure is straightforward. From a form found topology or mesh, shorten the mesh edge lines a distance sufficient for the connection, and extrude them in the edge normal direction (the average of opposite face normal). The surfaces can now be extruded in the surface's normal directions to create beam or truss elements. From the end surfaces of the beam elements, a plate can be made for the cross-section and the connector plate can be made simply by non-uniform scaling of the former. The cross-section of the connector plate can then be projected onto the connector (node element) so that it can be lofted. In this example, a cylinder is created at each node in the direction of the average of the neighboring face normal to each node.

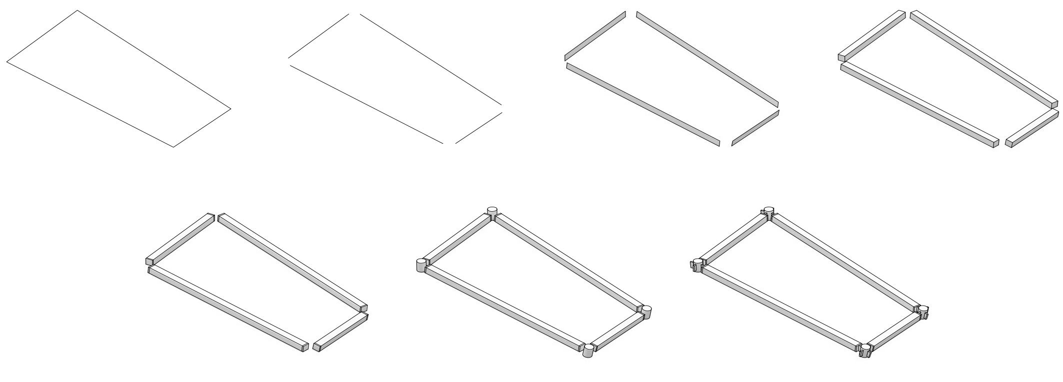


Figure 68. From left to right: polyline of mesh face. Shortened lines from the polyline. Lofting between the top and bottom (from mesh offset) shortened lines. Solid beams from the normal extrusion of the surfaces. End plates are made from the extrusion of the axial faces of the beams. Cylindrical connectors are constructed at each unique vertex. The connector plates are made by directional scaling of the end plates and splitting by the connectors.

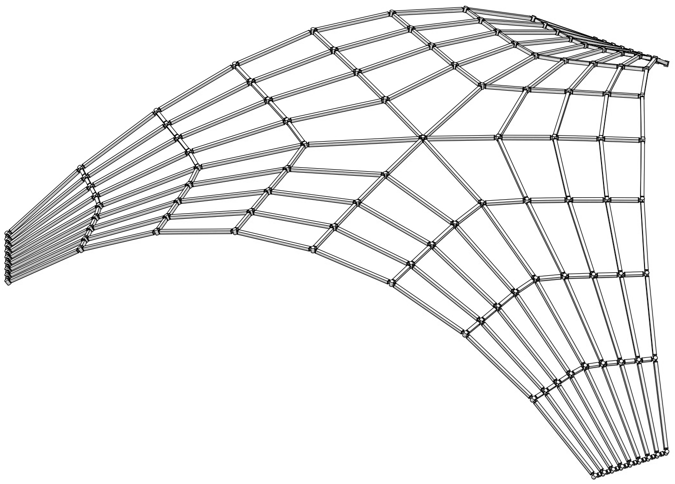


Figure 69. Gridshell from the metal-connection procedure.

Note that the quad-faces of the form found topology or mesh in the example here, are not planar, this is not necessary as the node can resolve the twist that would otherwise occur in the beams or the node. Furthermore, all beam elements are torsion-free, if created in this manner. If the elements are designed to have planar coverings, like glass or boards, face-planarity is easily added to the form-finding constraints. In doing so, consideration will have to be given to the fact that a shared beam between faces without a kink, will result in neither beam being in the plane with the covering panels.

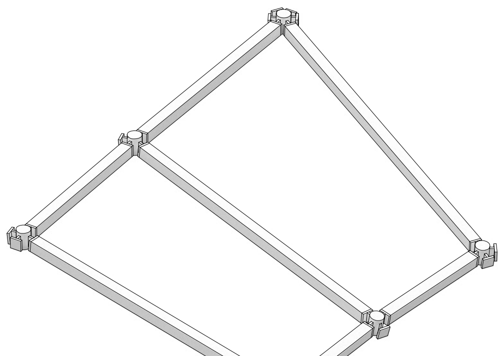


Figure 70. Beam elements with the metal connection

A seamless connection like b) is generally not achievable for complex doubly curved structures, without compromise or optimization. However, hexagons can have such a connection, as only three planes meet in a vertex. It is also possible to have such a connection for quadrilateral conical structures as described in the form-finding section, keep in mind that the beams will need a kink as they are essentially constructed from two face edges.

Reciprocal frames like c) and d) are also known as expanded nodes, or nexorade-structures, and are a type of self-supporting structure made up of interlocking beams. The term “reciprocal” refers to how the beams are arranged and supported, as each beam supports the one next to it, and is in turn supported by the one before it, creating a self-stabilizing system. Generally speaking, reciprocal frames refer to structures of timber beams that fully rest on one another with elements that are sometimes intersecting each other. The reciprocal frame is characterized by the node opening that occurs when the beams rotate around their midpoint. Aspects from this structure can be useful for gridshell design (primarily in timber), as the opening reduces complexity in the node with a beam layout that can follow complex surfaces, such as doubly curved ones.

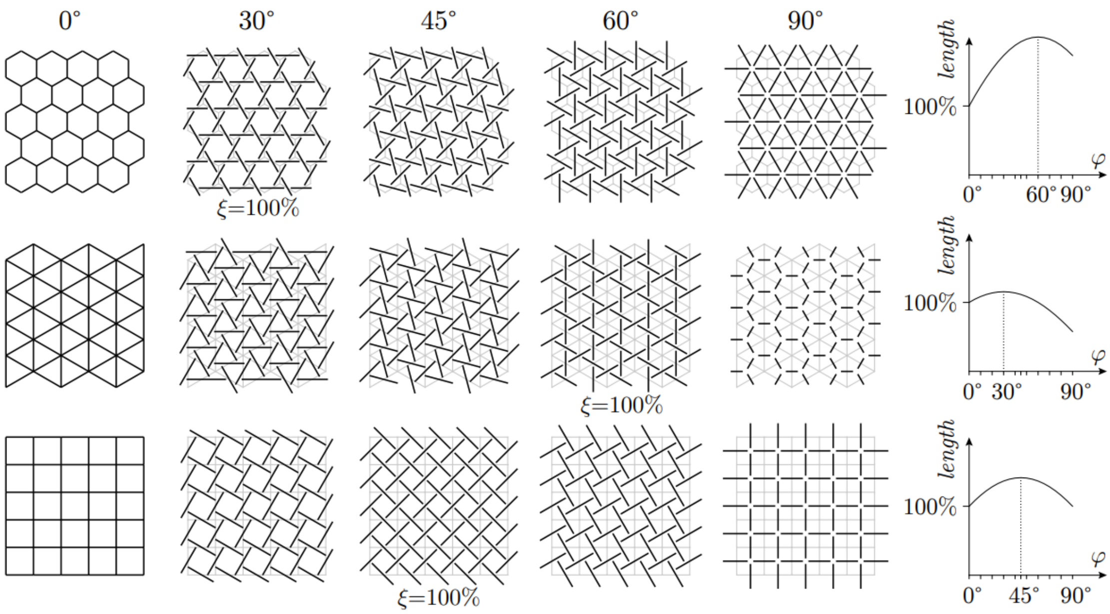


Figure 71. Knot expansion by rotation around each line’s midpoint in regular planar grids, and the resulting change of line lengths with respect to the initial length at 0° (Apolinarska, 2018).

layout that can follow complex surfaces, such as doubly curved ones.
Different knot expansions for triangular, quadrilateral, and hexagonal elements can be seen above and their respective lengthening as a function of the expansion angle of the node. There are many sensitive design parameters to reciprocal frames, which will not be covered in this thesis. A more thorough design investigation of reciprocal frames and optimization is described by Apolinarska (2018) in ‘Complex Timber Structures from Simple Elements - Computational Design of Novel Bar Structures for Robotic Fabrication and Assembly.’ and by Torghabehi (2020) ‘Generative Reciprocity: A Computational Approach for Performance-Based and Fabrication-Aware Design of Reciprocal Systems’.

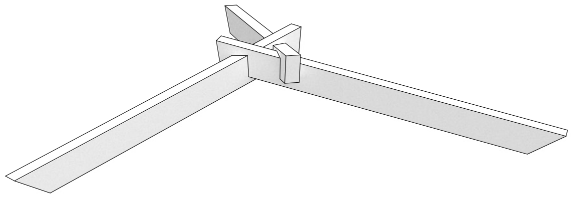


Figure 72. A reciprocal frame of beam elements.

Reciprocal frames as mentioned generally refer to timber elements that rest on one another, as illustrated by Figure x. For this to work, a dome-like form is necessary, but for gridshell design, some type of connection is most likely necessary between elements, especially if the structure curves to become more vertical. Therefore, the elements don't need to overlap or extend past one another but instead rely on a connection between elements. Such connections could be bolts or screws between elements, or one could turn to more creative solutions, for example incorporating a common plate for the beam elements, as seen in Figure X. Keep in mind that designing girdshells with reciprocal frames introduces both bending and shearing and the structural behavior will differ from shell structures, even though they may be based on a funicular form.

Figure 73. A reciprocal frame of beam elements from hexagonal topology, with a shared plate element

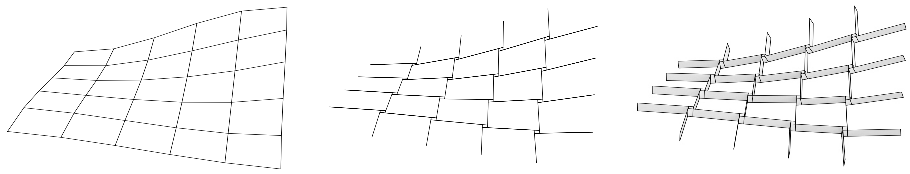


Figure 74. From left to right: quad subdivision of doubly curved surface. Lines rotated 10 degrees around each line's midpoint, around the normal, and extended to the plane of neighboring lines. Extrusion in the normal direction, creating the beam layout.

The design procedure for reciprocal frames is quite straightforward but requires a lot of optimization. Generally speaking, the procedure can look as follow: from the form found mesh or topology, rotate the mesh edges (lines) at their midpoint, around the normal of the edge-line. The rotated lines are then extended to the plane of a neighboring line in cyclical order. From these lines, a 2d beam layout can be made simply by extrusion in the normal direction of the mesh edges. Similarly, 3d beam elements (solids) can be constructed by projecting the cross-section (aligned after the perpendicular frames of the lines) of the element onto the plane of the neighboring (cyclically ordered) elements' closest edge face.

The design procedure for such a reciprocal frame assumes that the elements on the naked edges are supported. In the pavilion example below, the outer-most elements are simply removed, and further optimization is needed to support the naked edge elements, for example, a frame that follows the entirety of the naked edges.

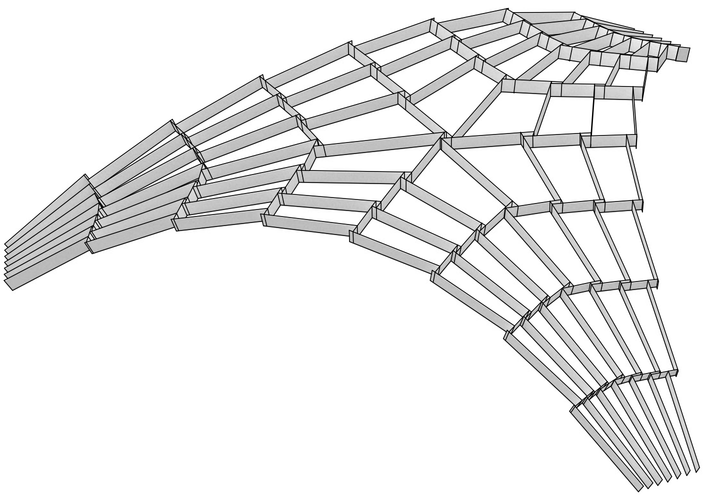


Figure 75. 2d edge-beam layout of a quadrilateral reciprocal frame form-found gridshell topology.

Gridshells and potential applications

Gridshell technology offers a wide range of applications in architecture yet to be attempted or studied in practical scale. While gridshells are commonly used for building roofs and pavilions, they can also be used for a variety of other purposes such as

Architecture and building construction: Gridshells can be used for building roofs, canopies, and other architectural features that require a large span and a light weight structure

Bridges: Gridshells can be used for pedestrian bridges, cable-stayed bridges, and other types of bridges that require a lightweight and strong structure.

Temporary structures: Gridshells can be used for temporary structures such as exhibition pavilions, event venues, and emergency shelters.

Landscaping: Gridshells can be used for creating large-scale landscaping features such as pergolas, trellises, and green roofs.

Sustainable design: Gridshells can be designed and constructed using sustainable materials and methods, making them a viable option for sustainable and environmentally-friendly building and infrastructure projects.

The thesis will initially focus on using a simple pavilion-type structure as an example to illustrate design procedures and analysis. The latter part of the thesis will delve into the less common concepts of application and explore how to implement them effectively.

Gridshell bridge prototype (demonstration)

To showcase the techniques described in this thesis, a design process for a conceptual bridge will be presented. The inspiration for creating a bridge design stems from the promising capabilities of gridshell technology, which has yet to be fully explored in terms of its potential applications. Bridge structures often incorporate funicular elements, such as suspension bridges, tied arch bridges, and arch bridges. Since physics-based form-finding techniques frequently result in a vault or arch-like structures that are commonly used in bridge design, a gridshell bridge was deemed suitable due to the challenges it presents while remaining practical.

1. Flat topology & form-finding study

The skeleton topology (or coarse mesh) was designed parametrically through the logic described in the 'gridshell topology' section. As the bridge is designed without context, it made it suitable to show some variation and flexibility of the design.

Below is a form-finding study through an evolution of topologies, that aim to capture some different pedestrian-bridge scenarios. Note that the topologies are all made with the same configurations, and the UV subdivisions are functions of length to keep elements similar in proportion. Furthermore, the forms are not yet optimized here, they rather aim to create the foundation for the latter form optimization.

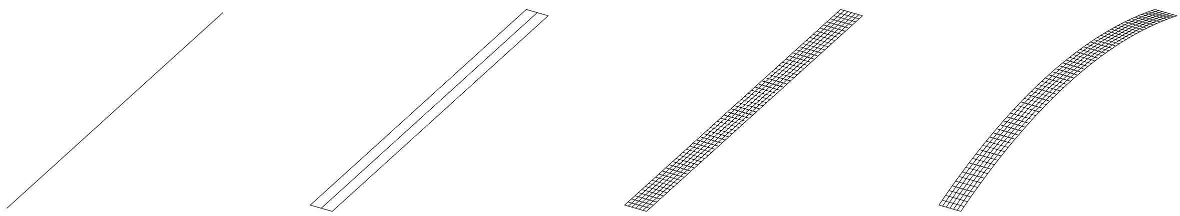


Figure 76. From left to right: line input. Coarse mesh. Flat topology. Form-found topology.

Starting from the simplest input, a single line, the procedure is shown that translates the line to a skeleton mesh and then into a quadrilateral mesh. A quadrilateral mesh was chosen here for the simplicity of the elements, and the ease of translation into triangles. They also behave predictably during form finding and are easy to keep in near similar proportions with lengthwise subdivision of the skeleton mesh. For this procedure, the short naked edges act as anchored supports.

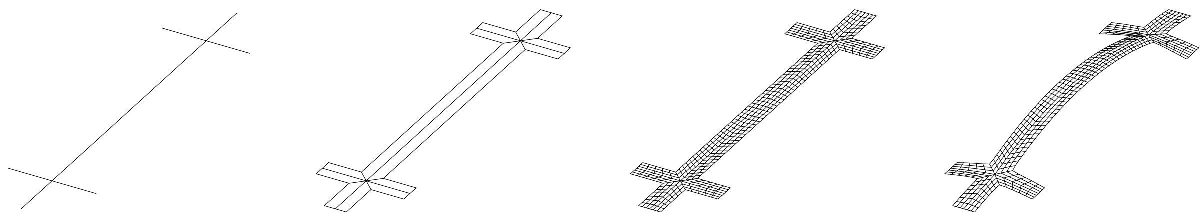


Figure 77. Adding supportive legs.

For more of a shell behavior, rather than an arch behavior, supportive legs were added for bracing and stability to the abutments. From adding these, we can see that the concentrated stiffness near the abutments results in a drastic transition in the slope between the bridge deck and the supports.

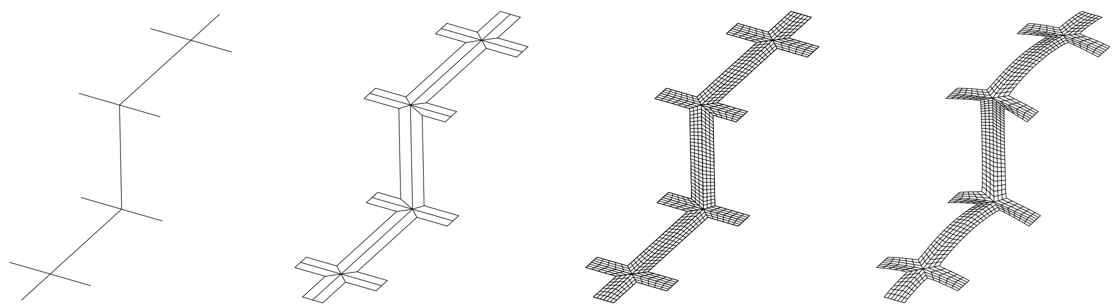


Figure 78. Adding spans.

To study the behavior of multiple spans and how a curved bridge would act, additional spans were included with intermediate leg-arch supports. This allows for observing how the spans always maintain a straight and arch-like shape between the supports.

Bridges typically consist of substructures (providing support to the superstructure), superstructures (the load-bearing element carrying the bridge deck across the span), and the bridge deck. This allows for designing each element based on its contribution to the overall structure. Gridshells can also be designed similarly, but their strength lies in their flexibility, allowing them to serve as both the bridge deck and superstructure. If the gridshell itself is to serve as the bridge deck without a separate structure, the form of the gridshell will need to meet the requirements of both functions.

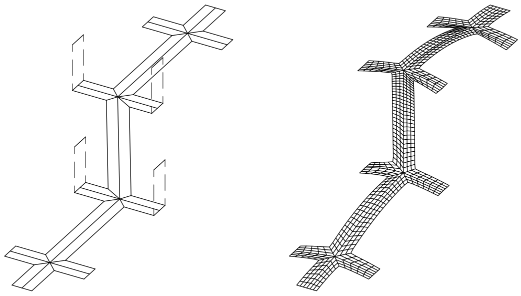


Figure 79. Different heights of supports.

Furthering the form study was done by varying the heights of the supports and investigating the branching of the bridge. Heightening the intermediate supports resulted in a more pleasant slope for the two outer spans, simply by limiting the downward slope. Further optimization of the mid-span could allow for a reasonable slope transition across the multiple spans. Such optimization could, for example, be in the form of taller supportive legs that follow the entire middle span to give it the needed height, and also some needed stiffness to flatten the slope.

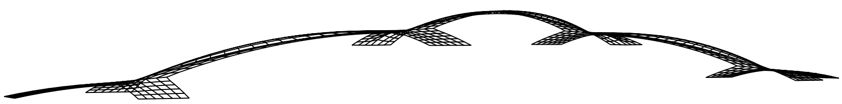


Figure 80. Perspective view of the form- found mesh with heightened supports for the mid-span.

2. Form finding & optimization

To tackle the challenges and requirements of bridge design, the approach is to start with a basic form and iteratively develop it to address these challenges. The flexibility of gridshell technology is a key strength, as the form should be able to respond to as many challenges as possible to be practical.

One of the major challenges faced in the design of a gridshell bridge is the consideration of point loads or varying loads, which are not accounted for in the even load distribution used to derive the form. To address this issue, existing optimized pedestrian bridge structures that are relatively thin have been studied. These bridges typically incorporate some form of pre-tensioning or are aided by cables to improve their performance.

Figure 81 Wasserfallbrücke by Jürg Conzett, one of seven bridges along the Trutg dil Flem, a hike along the Flem River in Switzerland.

One example is the Wasserfallbrücke (waterfall bridge) designed by Jürg Conzett, which is constructed using blocks of stone with self-leveling concrete in between. Across the bridge, a pre-tensioned steel drawcord is laid and anchored at the bridge mounts where two blocks of steel are cast in the concrete. The pre-tensioned drawcord presses the stone blocks together, allowing the bridge to act as if it were heavier and perform better with less material. The drawcords follow the deformation of the bridge and ensure that the force exerted is within limits when the bridge deviates from its arch shape.

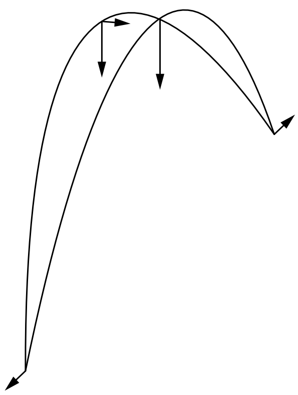


Figure 82. Force components of pulling a straight catenary curve and an angled catenary curve from the same points.

To enhance the strength of the gridshell bridge and increase its capacity to handle varying loads or point loads, the pre-tensioned drawcord concept can be applied to the exposed edges of the structure. However, for this approach to be effective, the cord must either follow a straight line across the bridge deck or a curve that sweeps outwards, as illustrated in Figure X, to ensure that the force components push the bridge together.

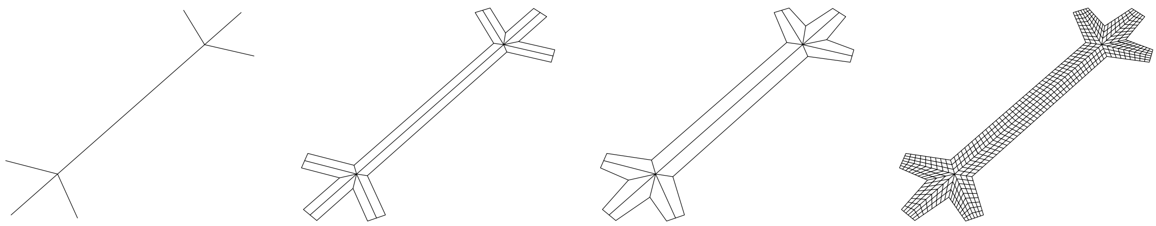


Figure 83. Constructing the flat topology for the bridge

A coarse mesh was generated from a line input like the one previously shown. To achieve compression in two directions from the drawcords, the lines around the valence point (line intersection point) were scaled, resulting in a bridge that is narrower at the start and end points. This coarse mesh was subdivided into quads to receive the flat topology which was subjected to boundary conditions of full anchorage (XYZ) at the start and end points, with the supportive legs anchored to the sloped surface. By constraining the legs only in the X direction (perpendicular to the bridge direction), the optimal form was found, as the supports would otherwise be pulled towards the middle.

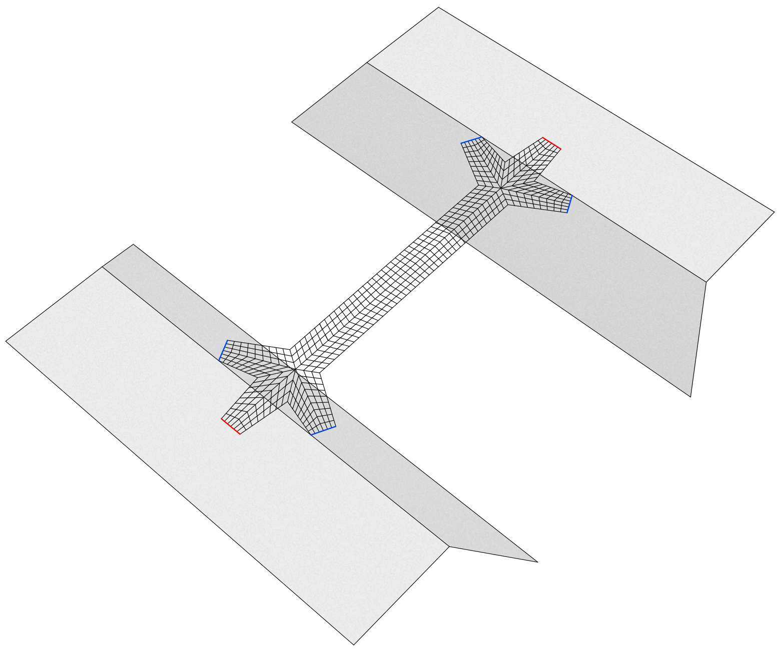


Figure 84. Flat topology and context with red indicating anchoring in XYZ and blue indicating anchoring to the sloped surface and in X direction (perpendicular direction to the flat topology).

The dimensions for this form finding was a span of 32m, and a bridge length of 40 m, with width of 2 m at each end point and gradually increasing to 2.7 m in the mid span.

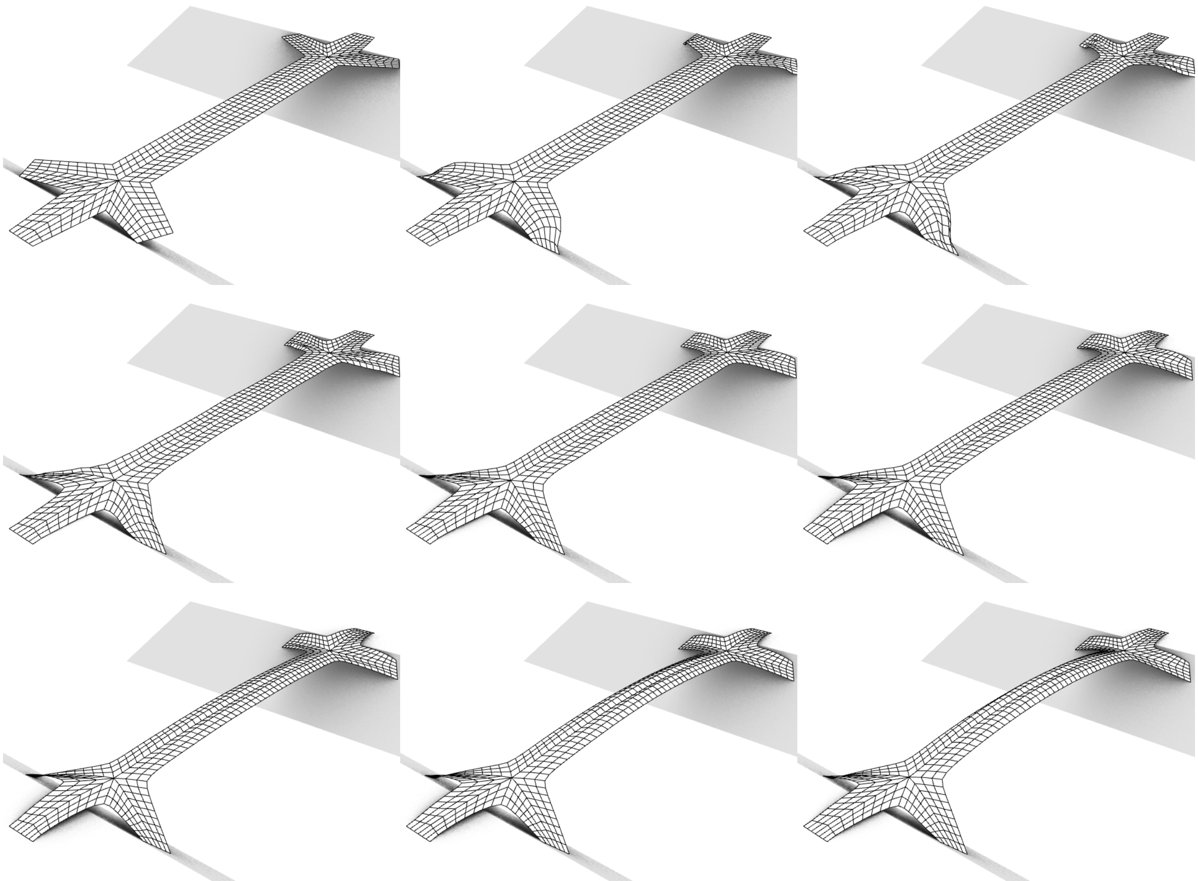


Figure 85. Form-finding of the initial topology.

The initial form derived had only some minor unwanted characteristics, such as the arch-shaped cross-section, which is not desired if the gridshell itself is to act as the bridge deck.

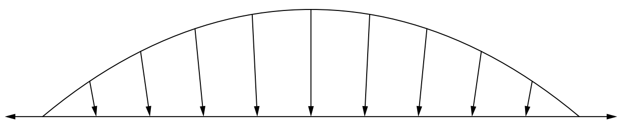


Figure 86. Load from allowing edge lengthening (pre-tensioning)

Applying the drawcord concept was done by allowing the selected edges to stretch or lengthen in addition to the previous boundary conditions. This, in turn, will simulate the forces exerted by the pre-tensioned drawcord onto the form in a distributed pattern in terms of load and direction, by the curvature it follows, rather than an even distribution of load at a vertical angle at every node.

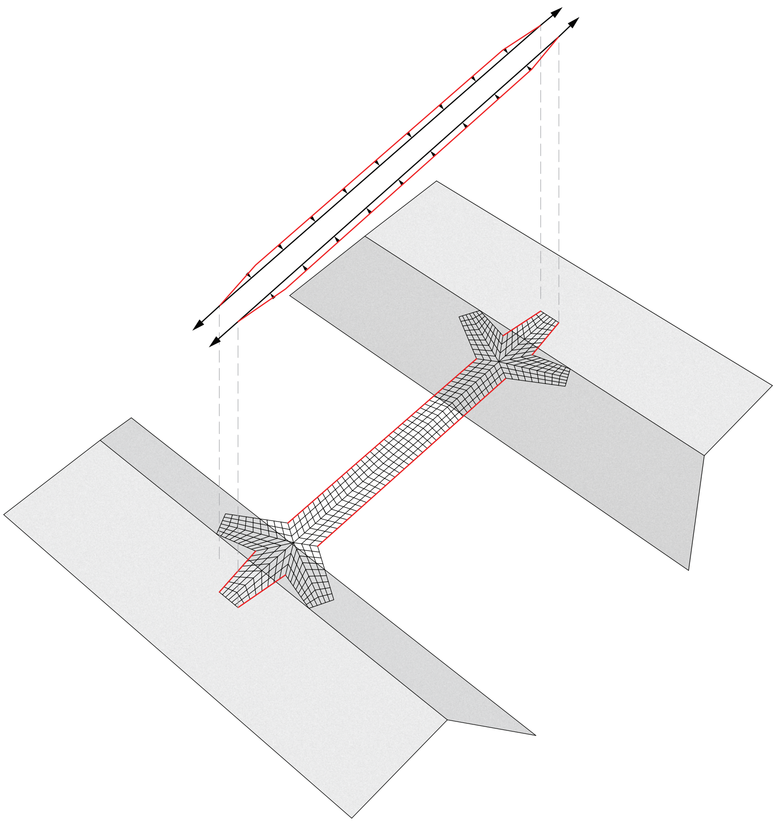


Figure 87. Pre-tensioning was added to the red edges by allowing for the lengthening of those elements by 1.1 times the initial length, in addition to the previous boundary conditions.

The above illustration shows that the drawcords are simulated by not being in contact with the supportive legs, which are angled downwards and towards the sloped surfaces.

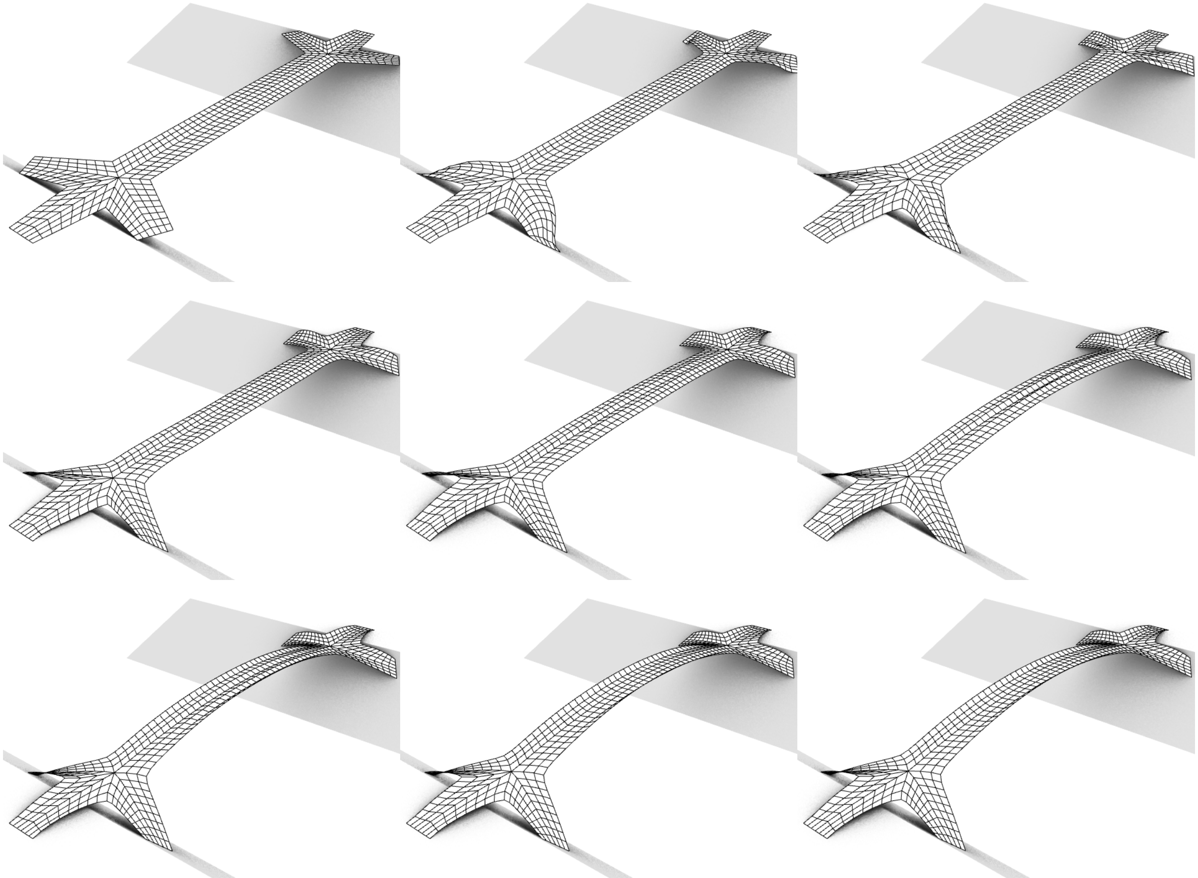


Figure 88. Form-finding with edge lengthening of the selected edges.

The resulting form has more of an overall arch, due to increased loading, with edges raised to counterbalance the added pressure from the drawcords. These raised edges also affect the cross-section, changing it from convex to concave and flattening the middle portion. Similarly, the cross-section can be flattened through mesh smoothing, which essentially forms finding with all naked edges constrained.

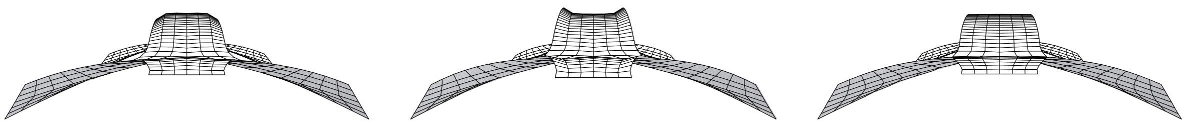


Figure 89. From left to right: initial form-found mesh. With pre-tensioned edges. With mesh smoothing.

3. Tesselation & structure

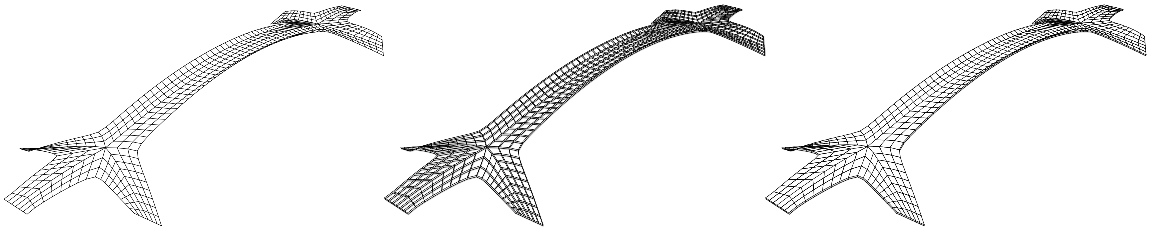


Figure 90. From left to right: form-found mesh topology. Torsion-free beam layout (without constant thickness). The tessela-tion procedure creates elements of constant thickness that meet in a single line.

For a complex form such as the one derived it is not possible to planarize the faces of the mesh without deviating far from the form found. Instead of planarizing the faces directly, the tessellation approach as described earlier was used to derive a structure that is possibly built of stone, like the Wasserfallbrücke. These are all planar elements with a constant height that resolve the initial non-planarity.

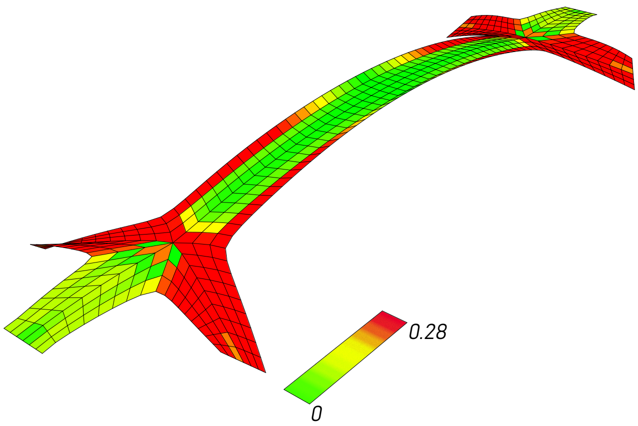


Figure 91. Initial planarity of form-found mesh topology, calculated as shortest diagonals divided by average diagonal length.

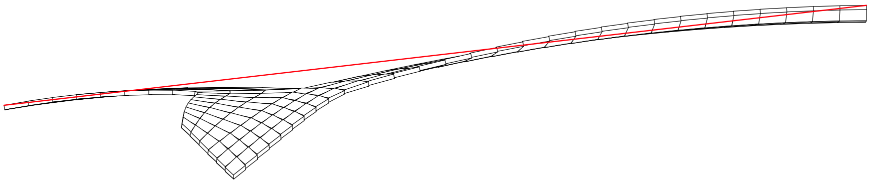


Figure 92. Drawcord interaction with bridge form.

As can be seen above, the drawcord in red will only press against the parts of the bridge that are above the illustrated shortest path between the endpoint and the highest point. The basis for the railing is modeled with this in mind, incorporating the drawcord, and following the form of the bridge.

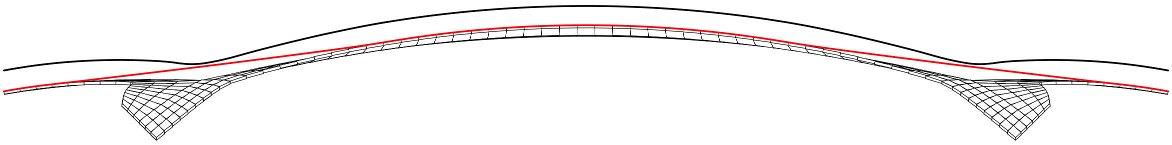


Figure 93. Side view of the bridge with the drawcord in red and the railing outline from the bridge curvature.

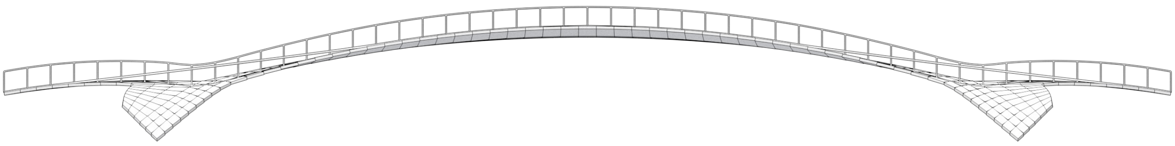


Figure 94. Side view of the bridge with the draw cord integrated into the railing.

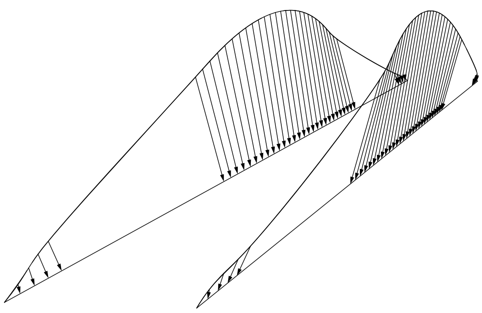


Figure 95. Load vectors from the pre-tensioned cord derived from the geometry.

4. Structural analysis & optimization

For the structural portion, there will not be an in-depth analysis, but rather a simple one to get a general idea of the structural behavior, as well as preliminary sizing of the members. For the analysis karamba3d was used, which is a finite element analysis plugin for Grasshopper.

The analyzed mesh is the one derived from remaking the mesh from the planarized edges. This mesh is not planar, so the quad elements will therefore be translated to triangular shell elements for the FEA analysis.

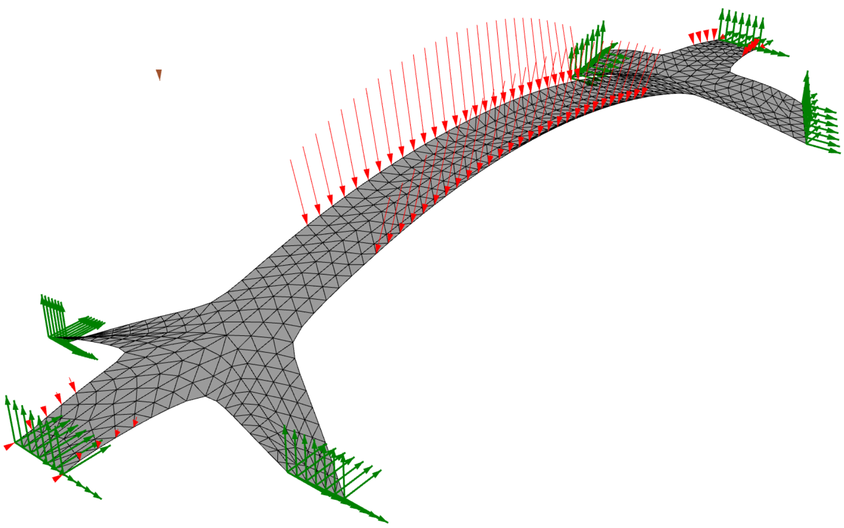


Figure 96. Karamba3d FEA model.

As mentioned earlier the load from the drawcord will not have an even distribution and is simulated as force vectors with the direction and amplitude derived from the relation between the pre-tensioned cord and its non-tensioned state. In addition to the drawcord load, the self-weight of the bridge is considered by the material choice. The model is simply anchored in XYZ at all the supports, with the above-mentioned loading conditions. For the material, high-strength concrete (C100/115) was used as a rough equivalent to the stone material. The mass of the structure comes to 35 tons for elements with a height of 12.5 cm. The load from the cord was initially set to 5x the distance to the straight cord in terms of kilo Newtons. This meant that the total load of the cord onto the bridge deck would be 558 kN, with the amplitudes of the load vectors ranging from 0.08 kN to 11.2 kN, seen in Figure x. The vectors at the mid-span are only 0.58m apart, which makes the pre-tensioning strength large, and any load case of humans significantly small in comparison, which is the premise of the design.

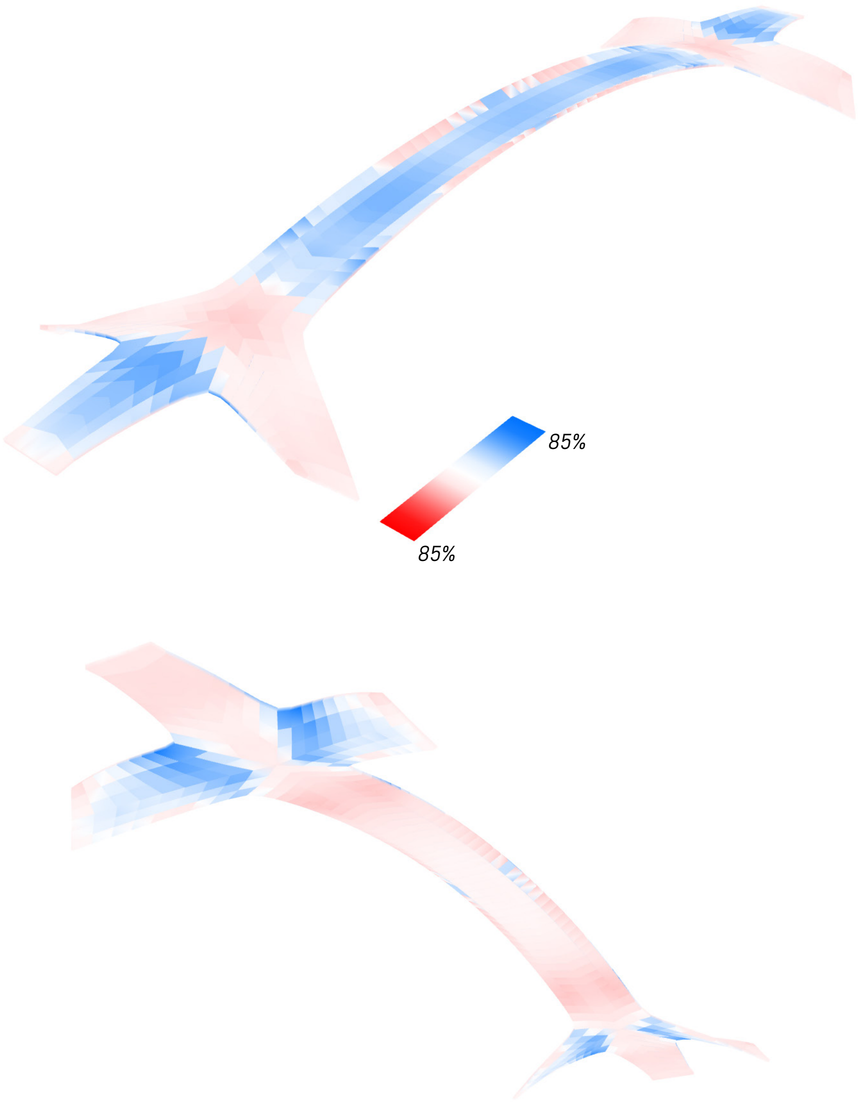


Figure 97. Utilization results with 12.5 cm element height across all elements.

As the form is derived from equilibrium with gravity (funicular form), the load is in theory evenly distributed, and thus all elements should have the same capacity. This is not entirely the case as the analyzed model has a few flaws, such as the material chosen, concrete, which has a weaker tensile strength than its compressive strength. Furthermore, it has to be said that the form analyzed is not the actual form, but rather a proximity. For this model, the maximum displacement was 3.8 cm.

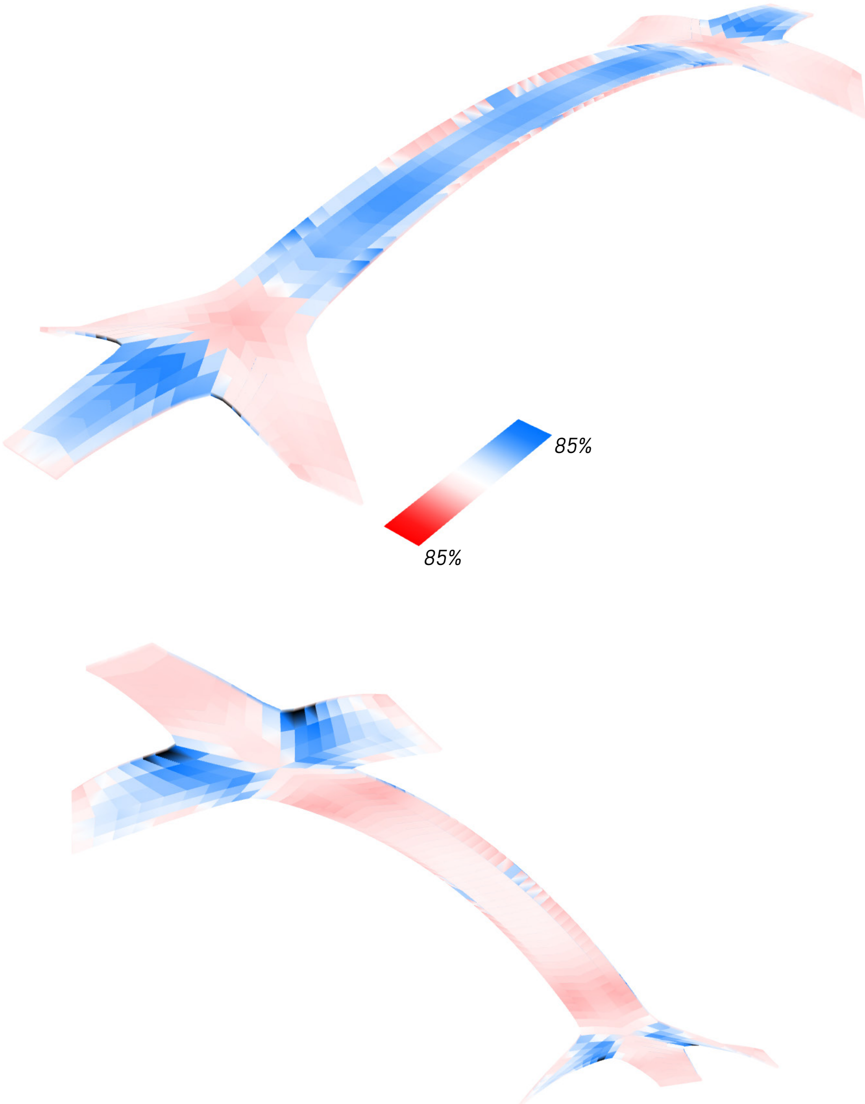


Figure 98. Utilization results with 12.5 cm element height across all elements, and total cord load increased to 782 kN from 558 kN.

To study the behavior under increased load, the load from the cord was further increased to a total of 782 kN, with vector amplitudes increasing to between 0.11 kN and 15.8 kN. The weakest parts as seen above in black, were the elements that reached failure before any other of the elements, noticeably due to the tensile failure of the material.

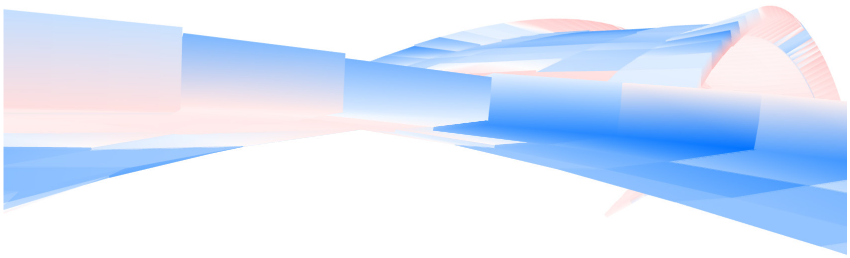


Figure 99. Allowing for varying element heights, between 12.5 cm to 20 cm with 2.5 cm increments.

If we allow the elements to vary in height, according to the utilization needs, a more optimized structure can be found. Above is a close-up of the weakest portion when allowing for incremental sizing of 2.5 cm, resulting in a range of elements between 12.5 cm to 20 cm. This resulted in a structure with a mass of 53 tons with 260 elements that are 12.5 cm, 270 elements that are 15 cm, 14 elements that are 17.5 cm, and the remaining 4 elements that are 20 cm, out of the 558 total elements.

The height of each member is then fed back to the model with preliminary-sized members, scaling the elements in only one direction (as opposed to Karamba3ds scaling in two directions) to retain the smoothness and characteristics of the bridge deck.

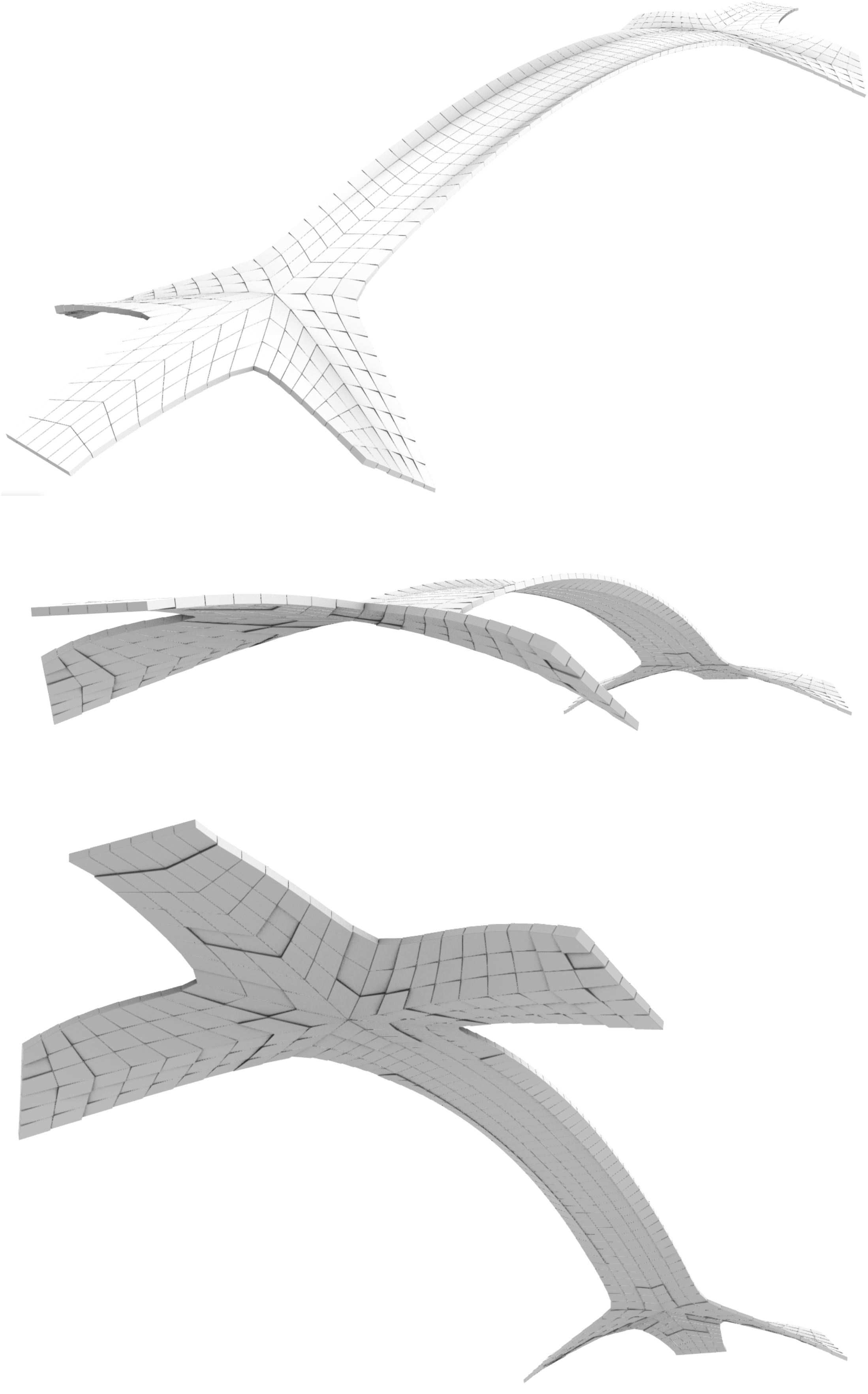


Figure 100. Gridshell bridge geometry.

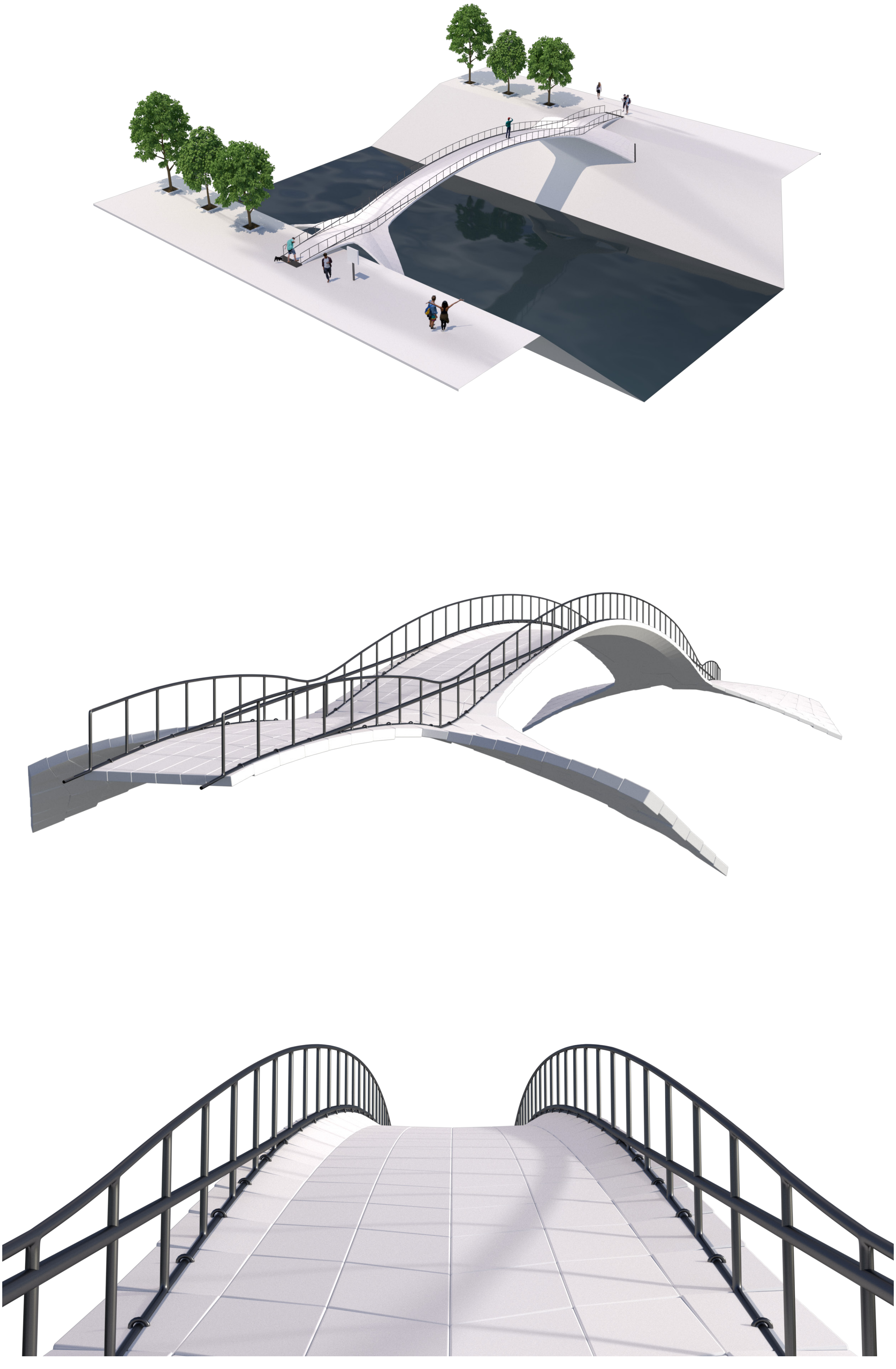


Figure 101. Gridshell bridge renderings.

5. Prototyping

The initial prototype developed in this thesis was a pavilion with a simple form derived from the second tesselation procedure. The objective was to construct the gridshell using male and female connections instead of adhesives.

The waffle-support structure was created by projecting the form-found mesh onto the XY plane and generating a solid from it. The Grasshopper plugin Bowerbird facilitated the creation of the waffle structure from the solid. The supports were manually modeled to fit the geometry.

However, during the construction of the mesh faces, a slight error occurred where the faces did not have consistent outward normals. As a result, some male parts turned out to be female and vice versa. This mismatch prevented the components from being joined, necessitating the use of adhesive to connect them. The presence of black color was a consequence of the flawed closed poly-surfaces.

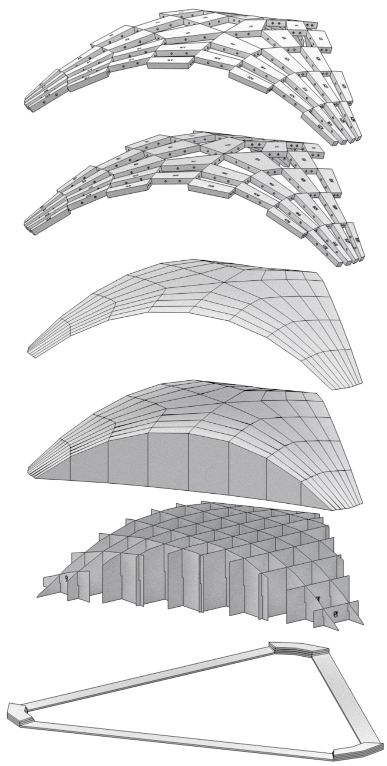


Figure 102. Axonometric eploded view of the physical model

The gypsum material was printed using a ProJet 660 PRO 3D printer. Unfortunately, due to the porosity of the material, the structure was weakened, and the majority of the glue was absorbed, causing the elements to soften. The process of building a physical model to achieve realistic behavior, in line with the simulations is a complex topic that falls beyond the scope and time frame of this thesis.

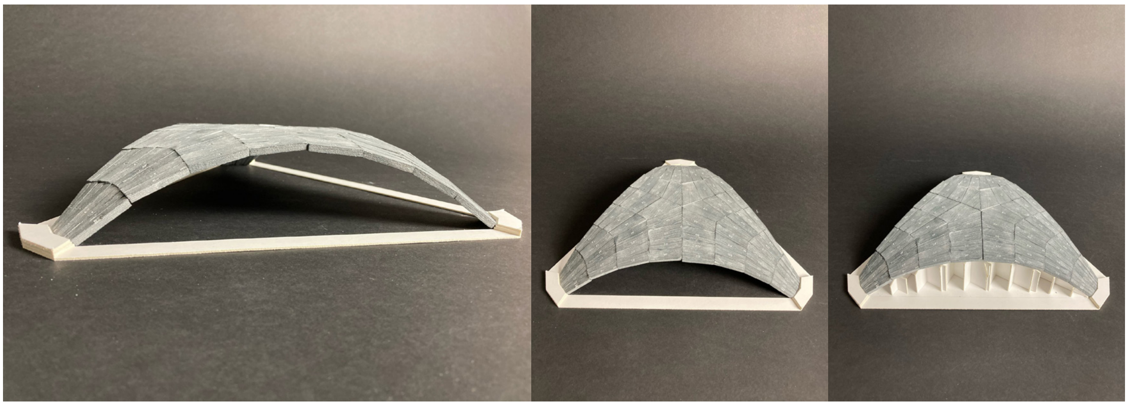
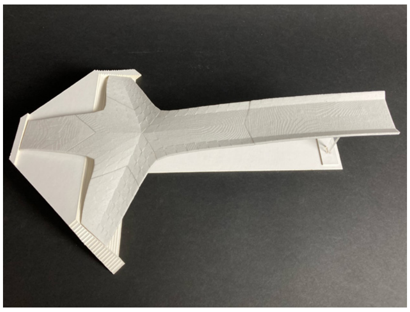
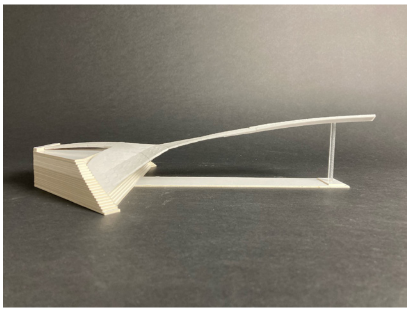


Figure 103. Physical model of the gridshell prototype

For the thesis presentation, a physical model of the gridshell bridge was made by dividing the elements into larger groups, to keep printing costs to a minimum. To overcome to the porosity of the material, the gypsum was impregnated with 3DS Colorbond, strengthening the structure and allowing the groups to be glued together by the coating. The physical model was made in scale 1:50 and was purely for the architectural presentation. It would have been interesting to work further on the theory of the cord and its interaction with the bridge on a more physical level, to add to the plausibility of the potential.





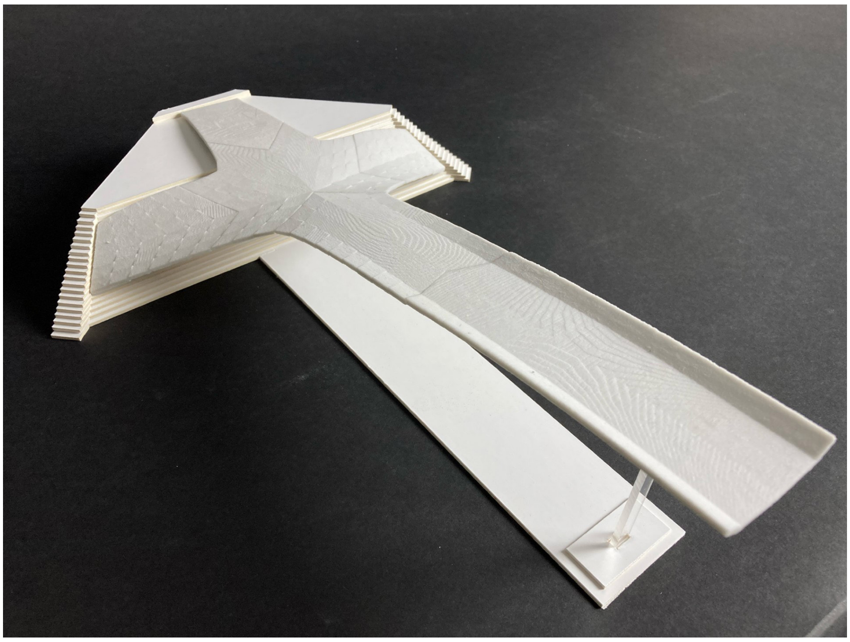


Figure 104. Physical model of the gridshell bridge prototype

References

Sources:

Leung, K.R. (2019). History and theory of Gridshell Architecture. I S. Verderber, T. Cavangh, A. Oak, Thinking While Doing: Explorations in Educational Design/Build. Birkhäuser.

Rigamonti, D. (2022). Gridshell. Designing buildings.
https://www.designingbuildings.co.uk/wiki/Gridshell

Ross, E. & Hambleton, D. (2015). Exact face-offsetting for polygonal meshes, ACADIA 2015: Computational Ecologies, Design in the Anthropocene.
https://www.researchgate.net/publication/283264212_Exact_face-offsetting_for_polygonal_meshes

Liu, Y. Pottman, H. Wallner, J. & Yang, Y. (2006) Geometric Modeling with Conical Meshes and Developable Surfaces, ACM Transactions on Geographics.
https://www.researchgate.net/publication/234796612_Geometric_Modeling_with_Conical_Meshes_and_Developable_Surfaces

Pottman, H. Wallner, J. (2008) The focal geometry of circular and conical meshes, Advances in Computational Mathematics.
https://www.researchgate.net/publication/220391061_The_focal_geometry_of_circular_and_conical_meshes

Apolinarska, A.A. (2018) Complex Timber Structures from Simple Elements Computational Design of Novel Bar Structures for Robotic Fabrication and Assembly. [Doctoral thesis, ETH Zurich].
https://doi.org/10.3929/ethz-b-000266723

Torghabehi, O.O. (2020) Generative Reciprocity: A Computational Approach for Performance-Based and Fabrication-Aware Design of Reciprocal Systems. [Doctoral thesis, University of Michigan].
https://www.researchgate.net/publication/353636199_Generative_Reciprocity_A_Computational_Approach_for_Performance-Based_and_Fabrication-Aware_Design_of_Reciprocal_Systems

Tellier, X. Baverel, O. Douthe, C. Hauswirth, L. (2018) Gridshells without kink angle between beams and cladding panels. Conference: Proceedings of the IASS 2018 Symposium Creativity in Structural DesignAt: Boston, Massachusetts
https://www.researchgate.net/publication/326693023_Gridshells_without_kink_angle_between_beams_and_cladding_panels

Piker, D (February, 2014). Grasshopper3d. https://www.grasshopper3d.com/group/kangaroo/forum/topics/difference-between-stiffness-and-restlenghtfactor

Verstartas, P (September, 2018). Rhinoceros Forums. https://discourse.mcneel.com/t/ngon-solids/70722/5

Piker, D. (May, 2019). Rhioceros Forums. https://discourse.mcneel.com/t/about-pq-mesh-conjugate-curvature-network/83476

Piker, D. (June, 2019). Rhinoceros Forums. https://discourse.mcneel.com/t/ellipsoid-mesh-from-curvature-lines/81480/23?page=2

Piker, D. (February 2021). Rhinoceros Forums. https://discourse.mcneel.com/t/planarizing-quad-panels-on-double-curved-surface/116811/29

Piker, D (February, 2021). Rhinoceros Forums. https://discourse.mcneel.com/t/difference-between-conical-meshes-and-circular-meshes/118429

Piker, D (October, 2021). Rhinoceros Forums. https://discourse.mcneel.com/t/proper-mesh-offset/148952/2

Piker, D (November, 2021). Rhinoceros Forums. https://discourse.mcneel.com/t/how-does-kangaroo-solver-work/92075

Piker, D (September, 2022). Rhinoceros Forums. https://discourse.mcneel.com/t/about-pq-mesh-conjugate-curvature-network/83476/25

Image sources:

Figure 4. Bandura, I. (2012) Amazing gridshell roof at the British Museum (photography)
https://commons.wikimedia.org/wiki/File:Amazing_gridshell_roof_at_the_British_Museum_(7083343621).jpg
CC BY SA 2.0

Figure 5. Unknown author (1891) Vladimir Grigoryevich Shukhov (photography)
https://commons.wikimedia.org/wiki/File:Vladimir_Grigoryevich_Shukhov_1891.jpg

Figure 6. Donskoy (2007) The Structure of the Roof of Upper Trading Rows (Moscow GUM) by Vladimir Shukhov - 1893 (photography)
https://commons.wikimedia.org/wiki/File:Structure_of_the_Roof_of_Upper_Trading_Rows_by_Vladimir_Shukhov_3.JPG
CC BY SA 3.0

Figure 6. Donskoy (2007) The Structure of the Roof of Upper Trading Rows (Moscow GUM) by Vladimir Shukhov - 1893 (photography)
https://commons.wikimedia.org/wiki/File:Structure_of_the_Roof_of_Upper_Trading_Rows_by_Vladimir_Shukhov_7.JPG
CC BY SA 3.0

Figure 7. Unknown author (1897) The double curvature Diagrid Shell by Vladimir Shukhov in Vyksa near Nizhny Novgorod during construction, 1897 (photography)
https://commons.wikimedia.org/wiki/File:Double_curvature_steel_lattice_Shell_by_Shukhov_in_Vyksa_1897_shell.jpg

Figure 8. Forgemind ArchiMedia. (2015) Frei Otto - Montreal Expo67 (photography)
https://openverse.org/image/c719040a-0af2-4b0e-9d1b-a05a9438d680?q=frei%20otto
CC BY SA 2.0

Figure 9. Geller-Grimm, F. (2003) Olympic Stadium (Munich), Germany (photography)
https://commons.wikimedia.org/wiki/File:Olympia_stadion_fg02.jpg
CC BY SA 2.5

Figure 10. Berberich, H. (2012). Multi hall, Herzogenried park, Mannheim (photography)
https://commons.wikimedia.org/wiki/File:Multihalle07.jpg
CC BY SA 3.0

Figure 11. Oosoom (2008). Interior of the wooden gridshell roof of the Savill Building, entrance building to the Savill Garden in Windsor Great Park, Surrey, England. (photography)
https://commons.wikimedia.org/wiki/File:Saville_Building_roof_interior_long.jpg
CC BY SA 3.0

Figure 12. The Wub (2021) A near empty British Museum Great Court, after most of the museum has closed (photography)
https://commons.wikimedia.org/wiki/File:British_Museum_Great_Court,_October_2021.jpg
CC BY SA 4.0

Figure 12. Gary, A. (2017) King's Cross Station, Christmas Day 2017. Closed for the day and deserted. (photography)
https://commons.wikimedia.org/wiki/File:King%27s_Cross_Station,_Christmas_Day_2017_-_concourse_roof.jpg
CC BY SA 4.0

Figure 13. Diliff (2007). The main entrance of L'Oceanografic in Valencia, Spain as viewed from the side across the water. (photography)
https://commons.wikimedia.org/wiki/File:L%27Oceanografic,_Valencia,_Spain_2_-_Jan_07-cropped.jpg
CC BY SA 3.0

Figure 81. Rick Persson (2019) Wasserfallbrücke by Jürg Conzett. (photogrphy)

Abstract

This master thesis investigates the design process of gridshell structures constructed from discretized elements, with a focus on optimizing the design through the exploration of key considerations regarding geometry, structure, and other potential challenges. The research results in a design handbook that provides valuable insight for architects and engineers, that would like to design a gridshell. Additionally, the thesis explores the potential of gridshell technology beyond typical applications with a concluding demonstration of the contents through the development of a bridge prototype that demonstrates the possibilities of gridshell technology.

CHALMERS
UNIVERSITY OF TECHNOLOGY