

Stress Line Driven Timber Gridshell Optimization

Deniz E ARSLAN, Leonardo GEREMIA, Khaled SADEDEN, Po-An HSIEH

Abstract

This paper expands on existing research in the application of principal stresslines to develop free-standing double-curvature structures and more specifically, explores the impact of stressline based derivatives on mass reduction in timber gridshells when compared to other approaches of the physical construction of freeform shapes. The research presents a framework for principal stresslines form finding to create an optimized gridshell. In addition, existing methods are explored for generating principal stress lines its strengths/weaknesses are examined to determine the optimal usage patterns. As an initial proof of concept, a principal stress line based optimization of a gridshell is taken and compared to other timber gridshells approaches, namely small-scale Chebyshev nets and geodesic based designs. Its successfully demonstrated that the use of principal stresslines could be an innovative method for reducing mass in timber gridshells without any negative impact on structural integrity.

Keywords: Principal stresslines, Computer-aided design, Structure optimization, Massing Reduction, Gridshell, Timber Structure Analysis

1.Introduction

Nature is known to create performant geometries and has had a key role in inspiring designers across architectural history. Thus, designers are known to have taken advantage structurally optimized design's potential to create aesthetic architecture. Even though methodologies have varied, the use of connection between structural behavior and architectural geometry inspired innovation. Synthesis through physical modeling, has been utilized to solve spatial and structural issues. Historically leading designers in this field are Antoni Gaudi, and Frei Otto, who are early precursors to open pathways of digitally simulating behaviors of freeform structures from which various studios across the globe developed researches, such as Zaha Hadid, Norman Foster. Of these explorations representing double curvature surfaces as gridshells, common factors are to ease constructability and structural efficiency.

Despite being a relatively recent area of study, the numerical calculation of stress-line-based topology has its origins in Michell's earlier efforts on structural optimization. Michell (1904) developed several well-known optimum truss' analytical derivations though he did not specifically take primary stress trajectories to account, his findings were similar to the principal stress lines for the form he was working with. Whereas civil engineering frequently employs methods linked to stress fields to study load-carrying complex structures, which are related to stress lines, these techniques centered around analyzing structures. Even though there have been a few attempts in architectural field, due to the absence of parameterization and standardization process for producing/ analyzing stress lines remains largely unrealized design community.

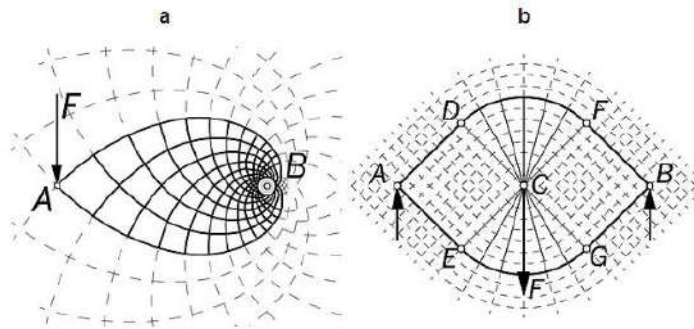


Figure 0. Example works of Michell (1904) [3], a) A single Force (F) in contact with A, normal to AB, b) A single force (F) in contact with C, with C being middle point of AB, with supports on A and B

1. Context

Shells are structural systems that are identified by doubly curved surface, while gridshells could be characterized as a structural typology which are derived from shell archetype. Gridshells being a subcategory could be defined as having shape and strength of a doubly curved shell but constructed as a grid instead of a solid surface (Douthe, 2006 [2]).

Principal stress lines are curve sets that are orthogonal to one another and designate trajectories of internal stresses. By consequence, these lines follow the material strength and naturally create a pattern for optimal topology within the context of a given set of boundary conditions. For the purposes of this paper, the boundary condition of these quadrilateral meshes is assumed to be that of a self-standing structure.

Since stresslines are the geometrical byproduct of a performance analysis of a shape under a certain load condition. A stress-line based gridshell, unlike a geodesic gridshell or a Chebyshev net, has no foreseeable geometric characteristics for each element (beam) before the paths of the stress lines is obtained. Only predictable datum is that at every point in the seeding, there will be a pair of stress line paths intersecting, this implying a valence of 4 at "regular" out-of-stresslines nodes, except on the borders or where singularities are manually or algorithmically added in order to get a closed mesh as a topological disk.

It may be beneficial to understand some terminology in this field to comprehend how principal stress lines are constructed. A principal plane is a plane in which shear stress is zero, and such planes would be perpendicular. Along these planes, the values of normal stresses will either be at their maximum or minimum these are referred as 'principal stresses'. Any element could be decomposed infinitely to discrete elements which the state of stress could be calculated for each point. A system of equivalent principal stresses can be used to represent the complex two-dimensional stress. There are two main methods that are used to calculate these, Graphical method (Mohr's circle) and analytical method using the following formula:

$$\sigma_{\max} = \sigma_1 = \left(\frac{\sigma_x + \sigma_y}{2} \right) + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2} \right)^2 + \tau_{xy}^2}$$

2. Motivation

Our goal is to prove a gridshell engineered with beams oriented/guided along principal stress lines will be able to achieve the same level of utilization with less total material (and therefore, less mass) compared to other systematic approaches for same shape and load cases.

2.1 Methodology

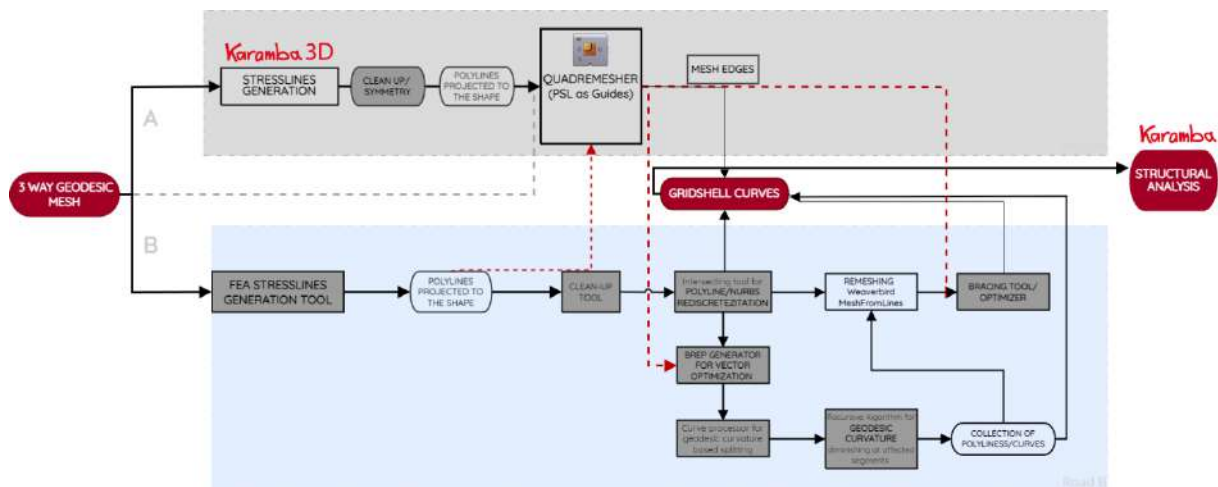


Figure 1: Workflow a) Existing Workflow, Gray b) Proposed Workflow, Light Blue

In figure 1 part 1.b follows a workflow using Karamba3D (Preisinger,2015) to generate principal stresslines. This simplified roadmap was defined to enable this research to reach approximations to a stress-line gridshell via third-party, rhino7, grasshopper algorithms for remeshing [quadRemesh, with a set of stresslines for a given shape as “guide curves”].

Here we must note (Tam and Mueller, 2015) [1]’s criticism of method finding there is not only a lack of documentation but also a lack of discretization of the various stages of stress line interpolation. Thereby users are delivered only the end result, designers being deprived of an option to adjust geometric characteristics.

This realization led us to follow the parallel flow (in figure 1.b) to attain these results in parallel paths there were multiple areas of research needing more comprehension:

- What are the ways to create a quadrilateral meshes using principal stresslines.
- Comprehension of PSL generation and Karamba3D’s vector field/ ways to refine results.
- Understanding the effect of “seeding” structural efficiency

On this path, the proposed workflow (figure 1.b), was initially approached in this research, contemplated the full implementation of the dataflow suggested by Mueller and Tam [1], for stressline generation and processing.

In order to achieve this, in prima facie it was necessary to research to develop several algorithmic tools that could execute the required procedures so that an actual gridshell of stress-lines could be simulated in a structural analysis tool in order to prove or refute the thesis.

Therefore, custom tools were developed as python code to run within Grasshopper’s API for Rhinoceros and were adapted in a way that ensured data inputs would be flexible enough not to depend on applying a node’s algorithm to another custom tool’s output.

In this research’s approach, tools were developed in a manner which respected Mueller and Tam [1], paper purportedly dividing the work among the involved nodes for the data processing flow, and that, depending on the desired construction technique, output could use data either for continuous beam simulation, or for discrete beams simulation as well as inputting data either from other nodes intrinsic to the process established, or from outsources within the grasshopper definition.

Both strategies were developed and implemented simultaneously. Along the process of investigation, both workflows merged in an entwined way:

When hybridizing approaches, grasshopper native components provided input to custom tools which added additional data-structuring and added speed, thus improving less performant areas of the grasshopper ecosystem, eventuality set of stress lines provided. These custom tools also added additional functionality not available in grasshopper natively. For example, custom remeshing functionality was developed, either for FEA generated stress lines, or to improve upon those generated via Karamba3D.

Most tools were fully developed, further explanation could be found under Appendix section and open sources could be reached online. Listed as follows:

- a) FEA stress line generator, b) Cleanup tool, c) Injector tool, d) NewBrep generator,
- e) Curve identifier/ splitter, f) Bracing tool

With further research purported to expand on the encoding of g) Unacceptable binormal rotation Element arches processor. Tested speed proved faster than native grasshopper components and enabled a better control of the variables at the foot of the process.

3. Principal Stress Line Generation and Analysis

3.1 Principal Stress line Generation

Here we outline the process for figure 1.a while conscious of possible inaccuracies and aiming for possible ways to understand the reasoning and resolve them. Having a high-resolution vector field being one of the major requirements to draw the principal stress lines (PSL), here is a summary for variation of first-order approach method (Tam and Mueller, 2015) [1]:

1. Sort the provided edge points and create a closed boundary curve.
2. Take N sample points along the edge curve
- 3a. For each edge point, interpolate the first principal component vector by taking a distance weighted average of the 3 nearest neighbors. 3b. Extend a line segment from the point using the vector. 3c. Repeat the interpolation and extension steps for the new endpoint. Continue until outside the edge curve or until a large angular deviation is encountered. 3d. Apply smoothing algorithm to the newly created lines.
4. Mirror lines if the mesh is symmetrical, around the alluded axis. Repeat for the second principal vector.

What needs to be avoided overlapping, discontinuous lines with unsought intersections. Here we examine the effects of mesh density (subdivision) in the vector field resolution.

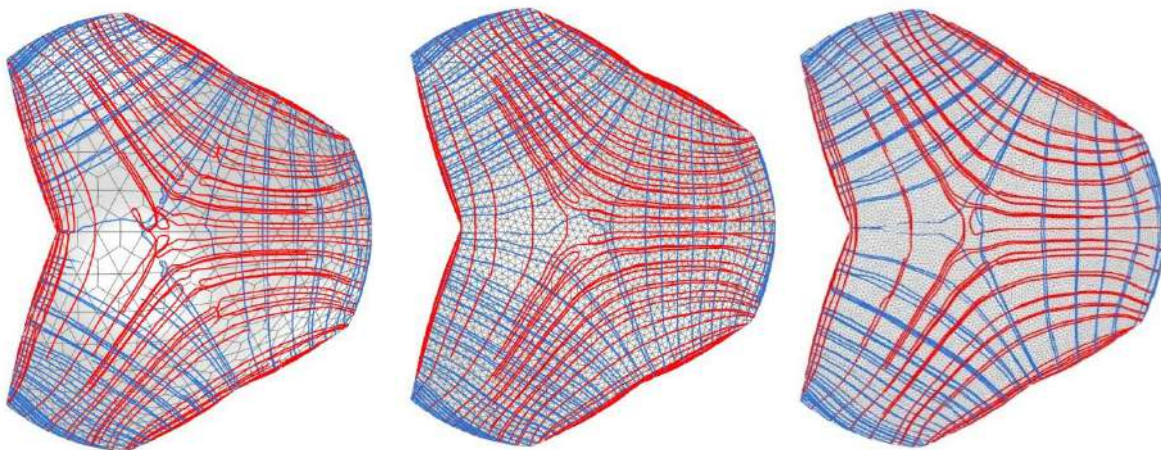


Figure 2: Mesh becoming denser to towards the right side

Density of the mesh (figure 2) does not increase the quality principle stress lines (PSL) that are drawn by using the refining algorithm generated PSLs, but it increases the number of lines generated natively by Karamba3D, but not necessarily the quality of the lines.

Another data point we should consider is the “seeding”. In karamba3d plug-in seeding corresponds to points where orthogonal pairs of PSL will intersect with one another. Below we see the effects of choosing different seeding points to see how they influence the quadremesher generated meshes that is guided by these lines. We also compare the results between native out PSLs and refined lines.

Refinement process is as follows: Remove lines if duplicates, if the end point of a line is equivalent or extremely close to another end point, if a line is subset of a line, if a line is too short (If surface is symmetrical: Mirror all the curves that do not cross symmetry axis. Reconstructs all curves that cross axis to be symmetrical)



Figure 3: Structural Analysis of meshes with different seeding and refinement

Above displayed not only Principal Stress lines before and after refinement, but also seeds that they were generated with and the quadremesher’s output that was made into a gridshell and resulting structural analysis

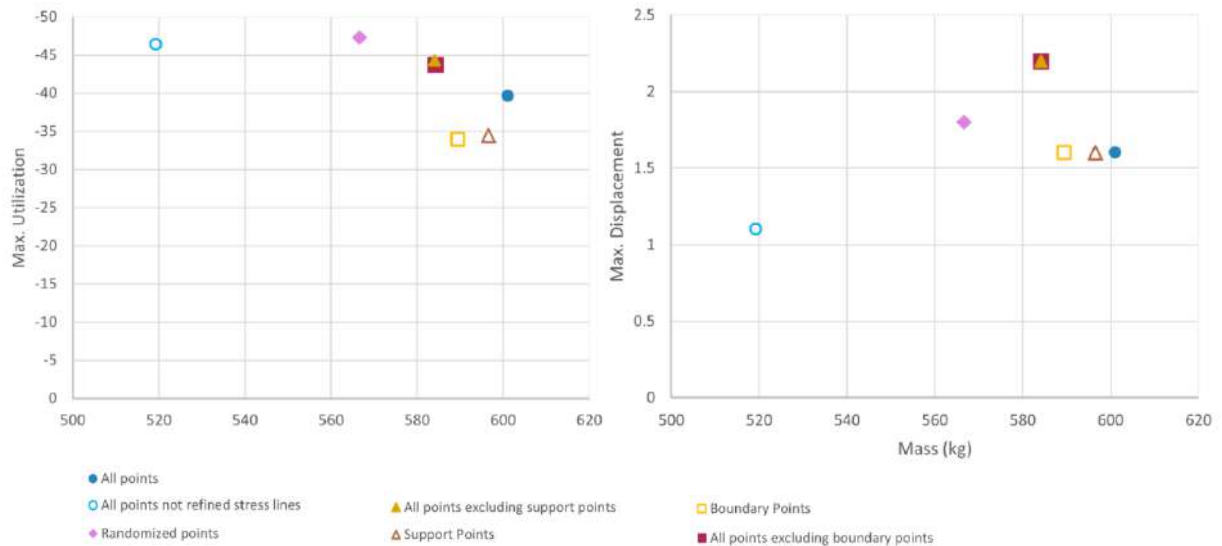


Figure 4: Scatter Chart showing mass against utilization and displacement variables

Accordance with what we see in this graph we notice that rather than the effect of seeding points there is a clear impact of refinements process which does not seem to be positive for structural integrity. One might argue that unrefined lines are up to 8 times denser than latter, and number of lines might have a stronger impact on quadremesher’s guide curves options, this argument does not have sufficient proof in this stage of the research (figure 5).

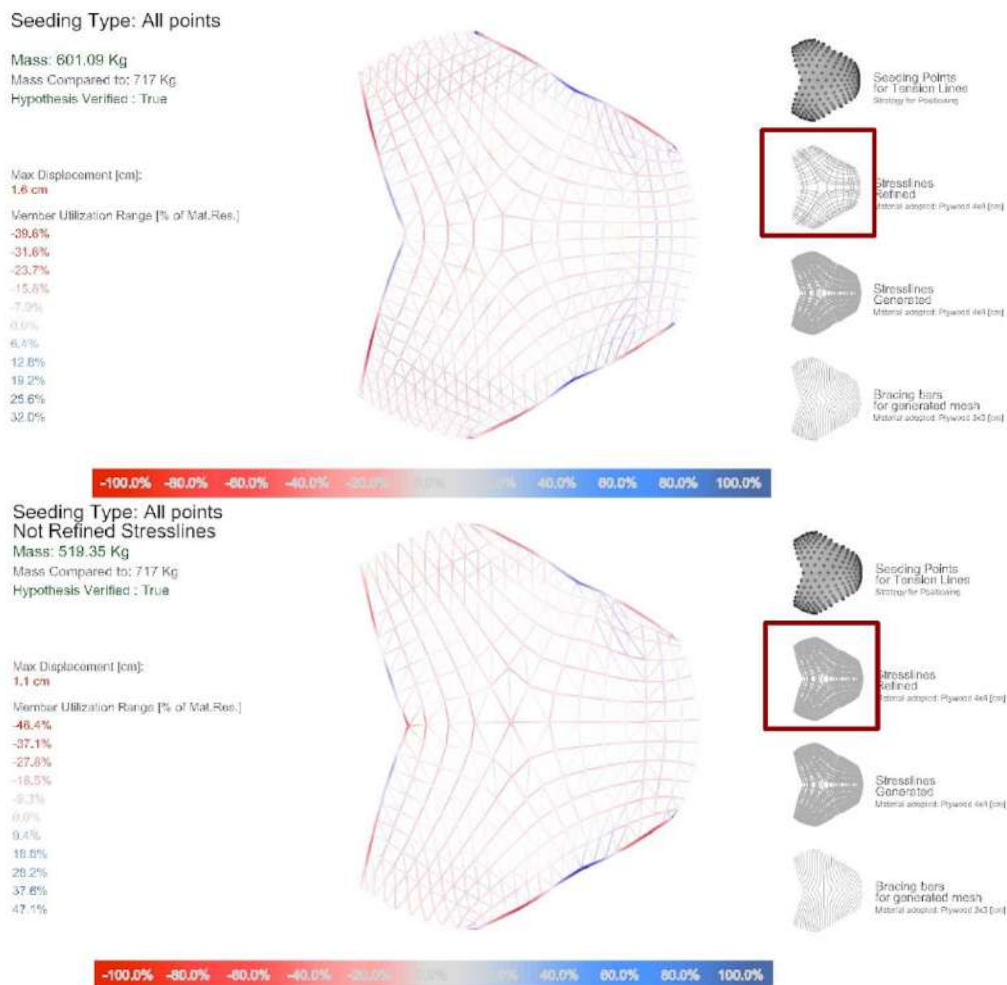


Figure 5: Detail comparison of how generated vs refined act differently

In compliance with the results of path Figure 1.a , next chapter of the research goes on with a gridshell which were generated using Karamba3d plug-in's native unrefined PSL lines seeding randomized.

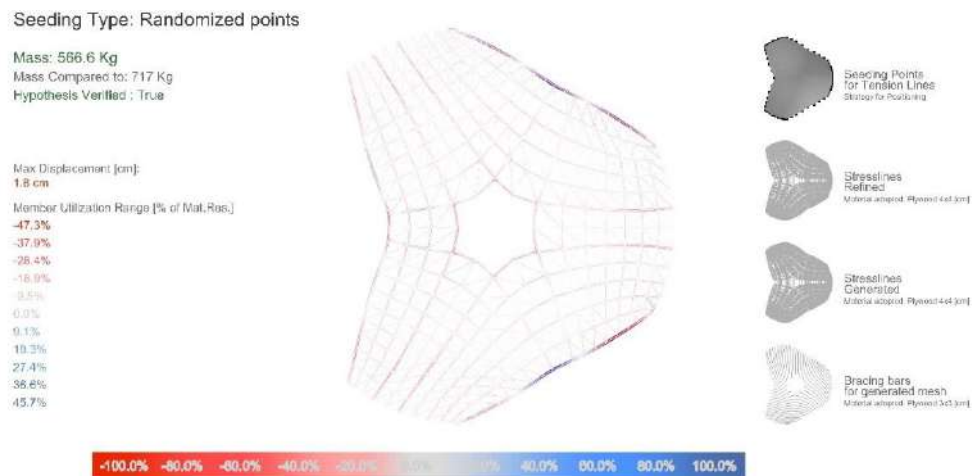


Figure 6: Gridshell chosen to be compared against ones built during course of the semester

4. Structural analysis & Mass comparison:

4.1 Comparing projects:

To evaluate the hypothesis, two projects built during Master of Parametric Design in Architecture at the Universitat Politècnica de Catalunya year 2022 were considered. The projects were compared against a variety of PSL generated gridshells that were derived from each of the original shapes for different loadcases, to assess differences in the geometrical and structural behavior of the PSL set for each of the simulated conditions.



Figure 7. a) Top three images; Chebyshev grid shell made of 1 layer, plywood 5x2cm. Location: Escuela Técnica Superior de Arquitectura del Vallés (ETSAV). b) Bottom three images; 3 ways Geodesic grid shell made of 2 layers, plywood 13x1.5 cm. Location: Alp, Catalonia, Girona

**4.2 Comparison criteria:
3ways Geodesic gridshell vs Stress line gridshell:**

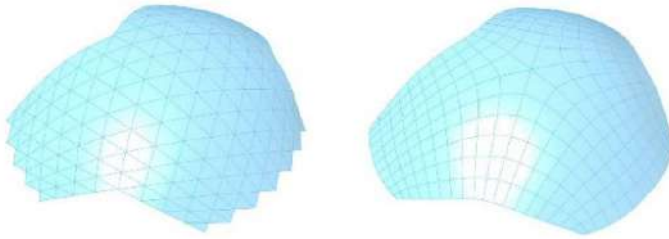


Figure 8. Left, 3 ways Geodesic grid shell mesh. Right, PSL Guided principle stressline mesh

In the mentioned as-built grid shells Karamba3D and Kiwi have been used for the analysis. In this study, Karamba3D results are used for comparison. In order to be able to compare objectively, the fixed comparable components between grids must be identified: shape, load conditions, supports, and utilization. The comparables can then be evaluated for correlations with other variables of interest, such as mass. The shape is discussed in the previous chapters and therefore the remainder of this section will focus on the other fixed compared components.

4.2.1 Loads & supports conditions:

With regards to the load conditions, the worst cases were taken into account. In addition to the self-weight, the maximum snow load (1.9 KN/m², Alps Catalonia region) was applied as a constant load on a significant part of the grid. The wind load and direction were studied and applied from the entrance side, which captures the largest amount of air and is responsible for pushing the grid upward. Alongside the loads, pinned supports are considered. All these loads/support conditions were similar to the 3way and PSL compared gridshells with the caveat that no prestress caused by bending was considered.

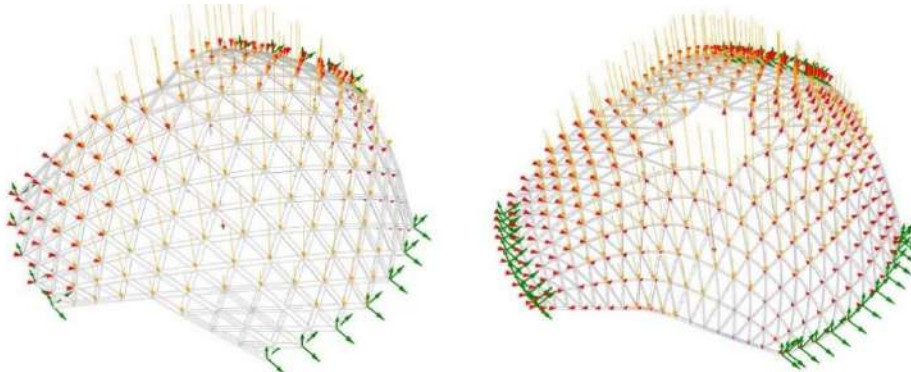


Figure 9. Left, 3 ways Geodesic gridshell. Right, PSL Guided principle stressline gridshell load & supports

4.2.2 Utilization & Displacement:

Utilization and displacement played a crucial role as the main reference variables. For instance, the utilization result from the 3 ways geodesic grid shell karamba analysis are (MIN -47.6%, MAX 37.7%) and it is very important to maintain relatively the same numbers while designing the beams and checking the mass in the PSL gridshell. The results were similar for displacement, which is 3.9 cm for the as built gridshell.

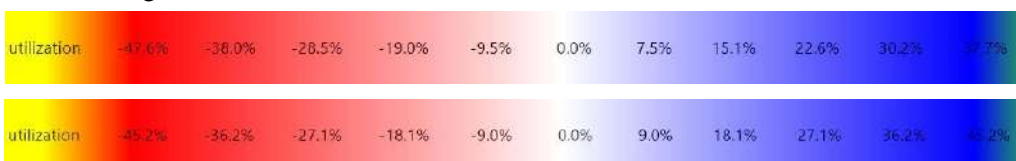


Figure 10. Top, 3 ways Geodesic grid shell. Bottom, PSL Guided principle stressline gridshell. karamba utilization results

4.2.3 Beams Design & Mass Reduction:

Designing the beams was ruled by certain considerations, including verification of geodesic curvature (Kg) and normal curvature (Kn) on one hand and cross sections and materials on the other hand. Curvature analysis studied from PSL grid shell resulted in the presence of a high measurement of geodesic curvature kg (see figure 11) especially at boundaries (openings) and near a singularity. Due to this curvature, it was necessary to design pre-bent glulam beams (6x6 cm).

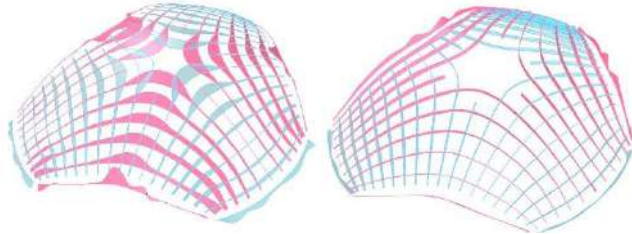
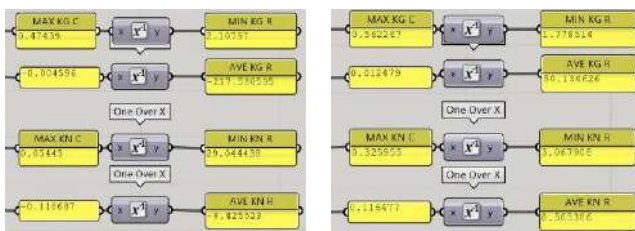


Figure 11. Curvature analysis; left: Kg, right: Kn



Since the PSL grid takes two different directions, the first one holds the load through the compression direction, the plywood beams are designed in cross sections (4x4 cm) larger than the other tension direction (3x3cm).

For both directions, squarish sections are utilized since plywood beams are made of several layers of two directional fabrics, hence they bend following Kg & Kn curvature. Practically, the cross sections were reduced gradually during the design, maintaining the utilization of around 45% close to 3 way geodesic results 47%.

For the bracing, several experiments for two systems have been made (see figure 13): steel cable or timber. As per tests steel cable bracing system added more stiffness and reduced the displacement and overall mass.

However, since the 3 ways geodesic mesh is a triangular system, it is more reasonable to design a diagonal bracing in timber (3x3 cm) which makes the comparison more accurate.

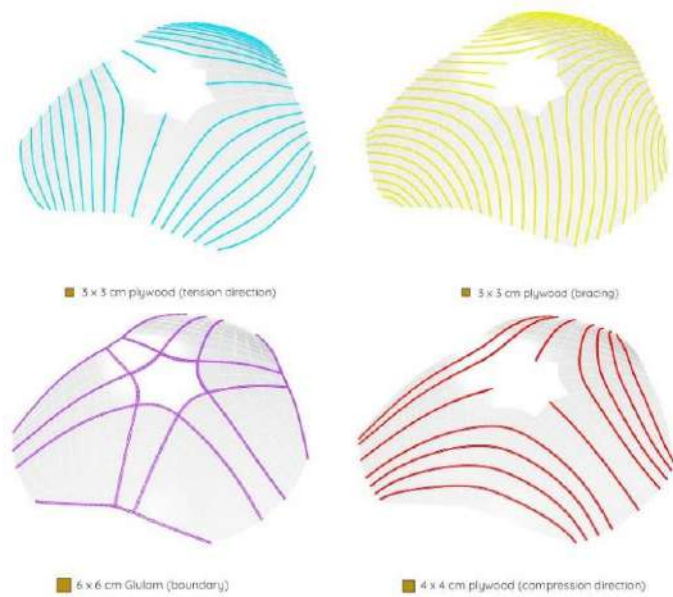


Figure 12 (a).Cross sections

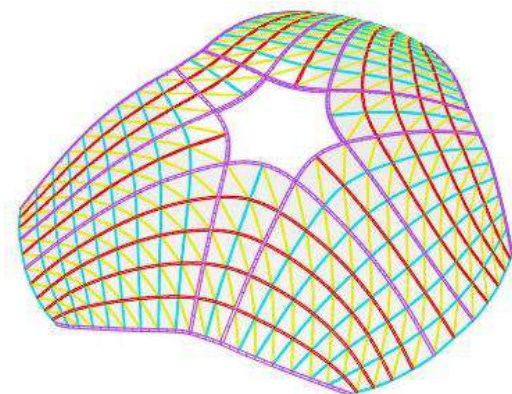


Figure 12 (b).Cross sections

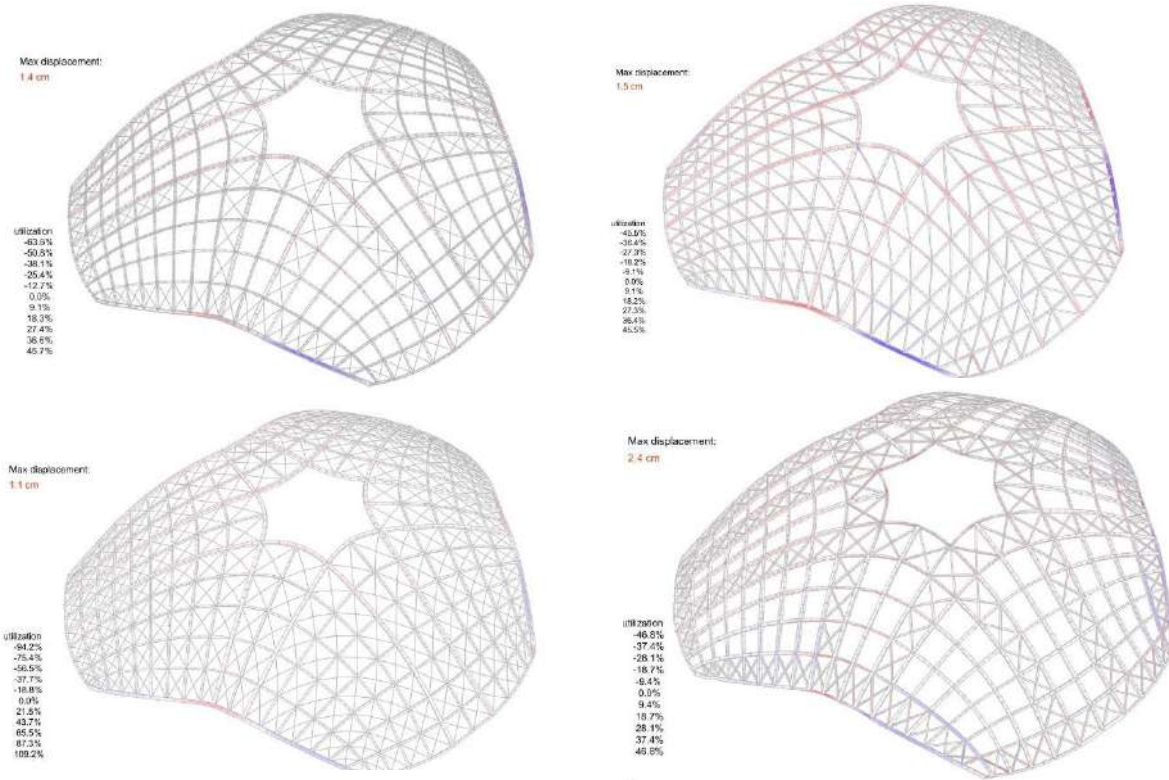


Figure 13. left column: steel cable, right column: timber

<p>Material: Plywood 'S235'</p> <p>E1:922.3[kN/cm²] E2:827.7[kN/cm²] G12:20.7[kN/cm²] nue12:0.3 G31:62[kN/cm²] G32:62[kN/cm²] gamma:6.8[kN/m³] alphaT1:1.2E-5[1/C°] alphaT2:1.2E-5[1/C°] ft1:3.95[kN/cm²] ft2:3.55[kN/cm²] fc1:-2.74[kN/cm²] fc2:-2.46[kN/cm²]</p>	<p>Material: GlulamTimber 'GL36h'</p> <p>E1:1470[kN/cm²] E2:49[kN/cm²] G12:91[kN/cm²] nue12:-1 G31:91[kN/cm²] G32:91[kN/cm²] gamma:5[kN/m³] alphaT1:5.0E-6[1/C°] alphaT2:5.0E-6[1/C°] ft1:2.6[kN/cm²] ft2:0.05[kN/cm²] fc1:3.1[kN/cm²] fc2:0.36[kN/cm²]</p>	<p>Material: Steel 'S235'</p> <p>E:21000[kN/cm²] G12:8076[kN/cm²] G3:8076[kN/cm²] gamma:78.5[kN/m³] alphaT:1.2E-5[1/C°] ft:23.5[kN/cm²] fc:0[kN/cm²] flow-Hypo: Mises; Hypo: Rankine</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 14. material specifications

For the bracing, several experiments for two systems have been made (see figure 13): steel cable or timber. As per tests steel cable bracing system added more stiffness and reduced the displacement and overall mass. However, since the 3ways geodesic mesh is a triangular system, it is more reasonable to design a diagonal bracing in timber (3x3cm) which makes the comparison more accurate.

4.3 Comparison result:

Comparing with 3 ways geodesic grid shell:

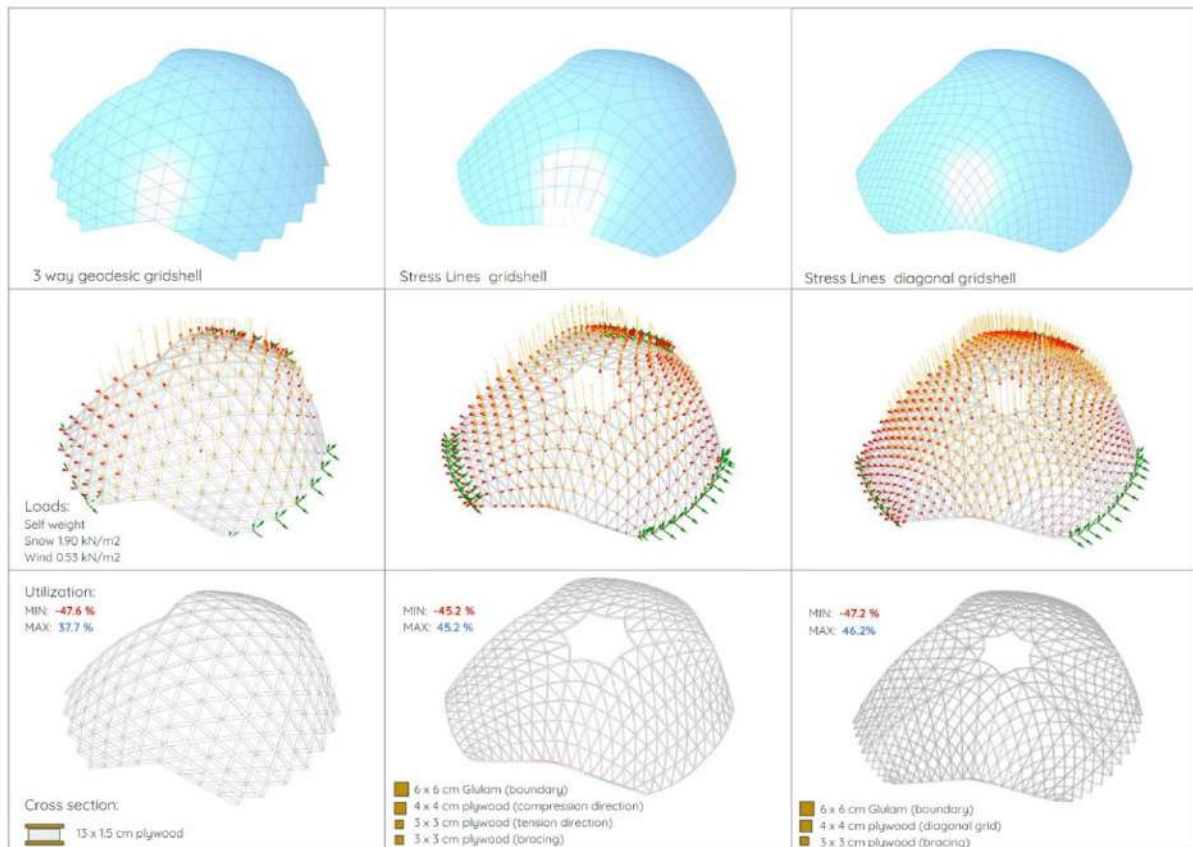


Figure 15. Visual description of three direction geodesic original shape outcomes for meshes, loads and cross-section proposals

By comparing grid shells (figure 15) a significant reduction in the mass for the stress line grid shell is noticed compared to the 3 way geodesic grid shell. To mention, a new version of gridshell is added to compare (diagonal grid shell generated from stress lines) and its mass higher from the two previous grids. As a result; gridshell that generated from stress lines is lower in mass and displacement, thus it is lighter and stiffer (see figure 16).

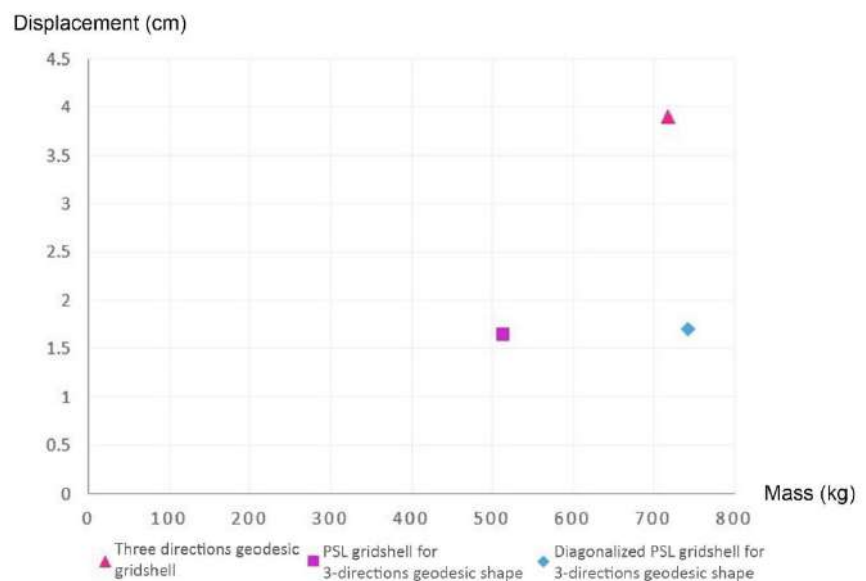


Figure 16. Mass (x axis) vs Displacement (y axis) for three direction geodesic original shape in first original gridshell approach, PSL approach, Diagonalized PSL gridshell approach

Comparing with Chebychev grid shell :

With the same method used to generate and analyze PSL grid shell from 3 way geodesic grid a new PSL grid was generated from Chebychev surface (See figure 11)

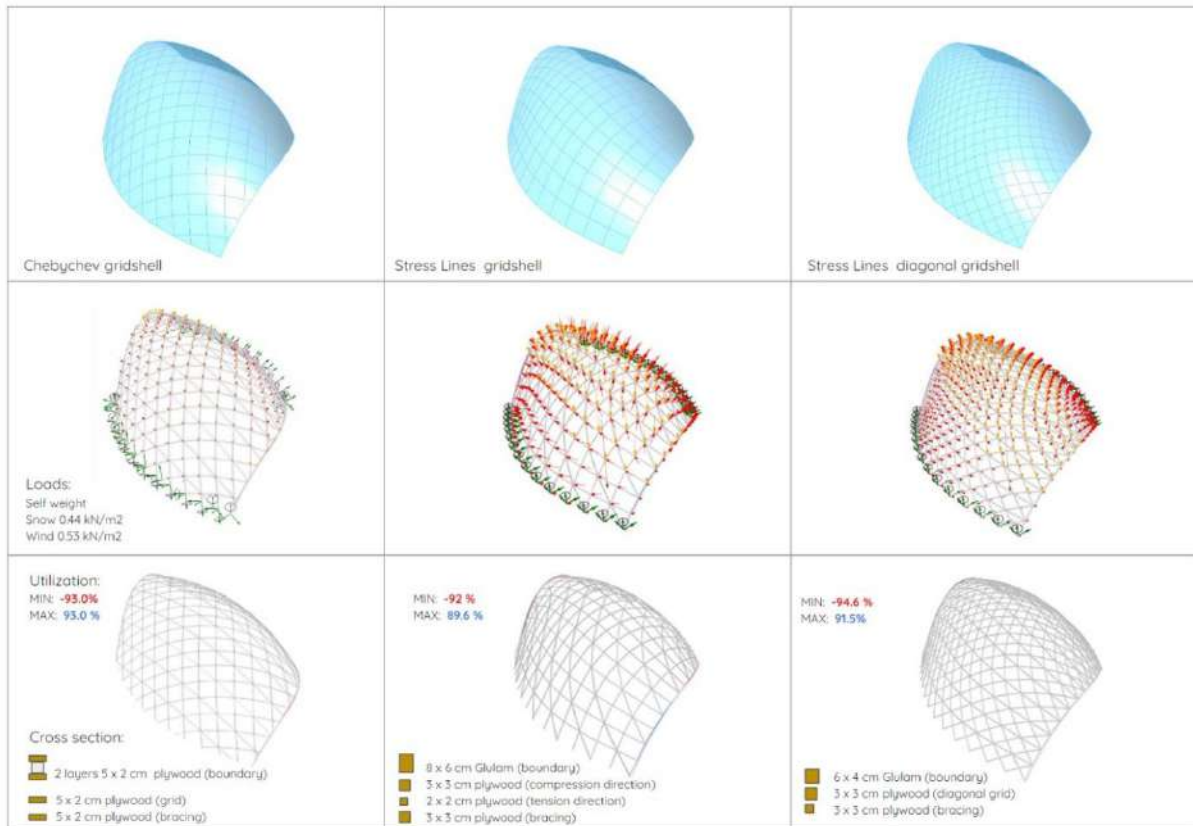


Figure 17. Visual description of chebychev gridshell's original shape outcomes for meshes, loads and cross-section proposals

The new study outcome showed similar results;

At which point the stress line gridshell was lower in mass and displacement compared to Chebychev grid (see figure 18). To summarize, PSL grid shells appeared to be lighter than other grid shell systems.

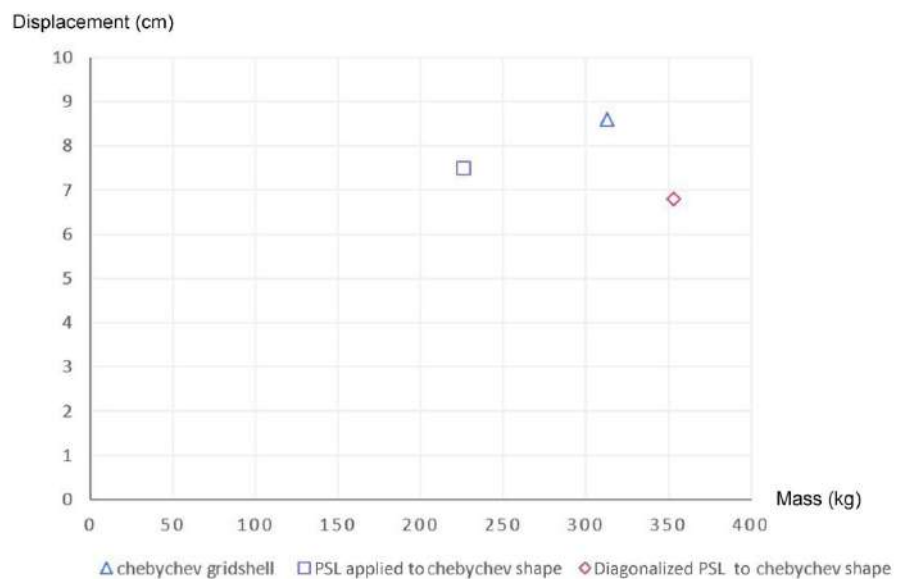


Figure 18. Mass (x axis) vs Displacement (y axis) for chebychev gridshell's original shape in first original gridshell approach, PSL approach, Diagonalized PSL gridshell approach

Below (figure18), are analysis of Principal Stress line gridshells that have been generated using not just gravity but also wind and snow loads. Loads applied are same as the loads that gridshells will undergo on-site in their locations.

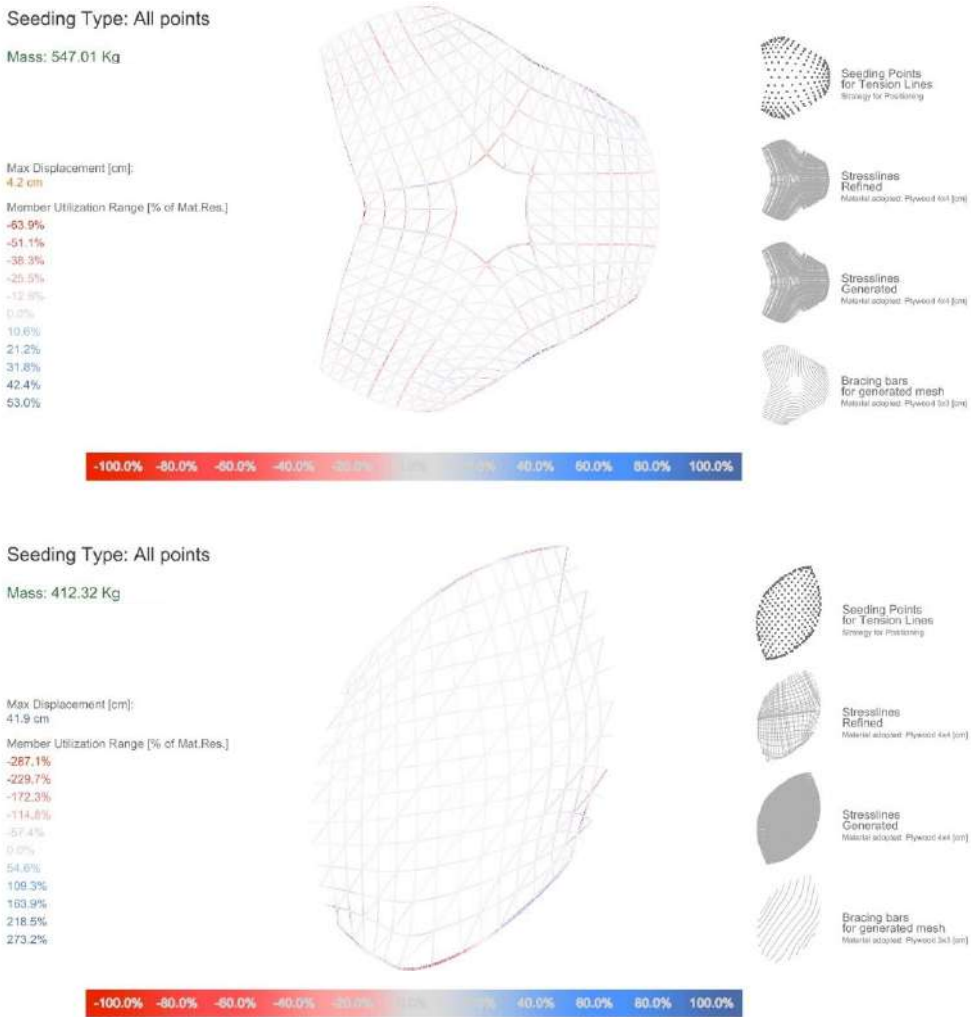


Figure 19: Top, 3way geodesic. Bottom, Chebyshev gridshell PSL Gridshells, generated using self-weight, wind and snow loads

The aim of this assessment was, to identify if the gridshells that were built guided by PSL lines that the forces were already applied to would produce a structurally stronger response than PSL gridshells that were generated only considering gravity and the topology itself. But displacement and utilization were so much inferior previous observations (figure19), including these results in previous analysis would have clouded the data, therefore these analyses were presented separately.

5. Conclusions, Contributions and Further Research

As conclusion of the research, for a collection of load cases applied to both analyzed shapes, the structural behavior of the shape as well as its corresponding collection of stress lines have been explored. It is to be noted that for the several load-cases explored; and due to the intrinsic connection between the geometrical determination of the stress-line paths and its corresponding load-cases for a given shape, any variation in the load-case conditions implied a different set of directions of the stress lines. This limitation impacted the outcome of the research, and it determined that load cases with heavy horizontal loads were discarded in favor of those that were more representative of the concept of rigidity by freestanding form. This resulting sets have been picked for each shape, processed and structurally analyzed as gridshells with topologies driven by those stress lines paths. The hypothesis of this research has been verified on all simulated cases, implying an improvement in resulting mass of the artefact ranging between a 20% and a 50% of material reduction in Kg. when compared to other approaches (three direction geodesic patterning; Chebyshev) for the same shape; thus, inferring that the hypothesis was verified.

Therefore, it can be stated that timber gridshells driven by principal stress lines lead to a noticeable reduction in mass for a similar utilization rate across both the examples and the initial cases; consequently, enabling the statement that, as a general contribution of this work, a lighter gridshell driven with stress lines to be deemed as feasible and convenient in terms of material usage.

As procedural contributions, custom tools have been researched and/or projected in order to enable a greater degree of freedom when manipulating the process of data flow and enlarging the scope of the research, and a workflow has been designed (figure 1) in order to tackle the constructability problems that could arise from the chosen approach.

The completion and improvement of the tools or tool-set developed with the partial scope of producing, processing and fabricating a stress lines driven gridshell has been explored, with some parts still being passable for further development and research within the frame of this paper. Such further research points would imply:

Proposing an algorithm for seeding based on utilization: As has been addressed, the shape of stress lines is a byproduct of the physical behavior of the given shape under certain load conditions. These load cases determining utilization map for the given shape, out of which parameters such as density of stress lines or weighting of them could be achieved in order to improve the general behavior of the consequent gridshell.

Improving the efficiency of the use of information resources in custom tools: Some of the developed custom tools are heavily dependent on brute force approaches, and while profiting on the speed divergence between the python implementation and native Grasshopper components, could still endure code optimization based on the development of an algorithm able to dispatch the input set of stress-lines according to their (equivalent) U and V directions for the given shape, thus reducing the number of operations in a 75%. Additionally, the paradigm of the syntax for every for loop could be revisited in order to reduce, when possible, the number of lines of the code, provided that for every instance of enumeration, all the lines must run consequently.

Creating other custom tool such as Acceptable binormal rotation Element arches processor (g) as a recursive algorithm to lower the geodesic curvature at critical parts of continuous beams: In order to better approximate a constructability approach that lowers the general cost of the gridshell based on simpleness of the behavior of the resulting pieces, would be of great use.

References

- [1] Tam, Kam-Ming Mark & Caitlin T. Mueller. “*Stress Line Generation for Structurally Performative Architectural Design.*” 35th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), Cincinnati, Ohio, Oct. 2015, < <https://hdl.handle.net/1721.1/125063> >
- [2] Douthe, C., Baverel, O. & Caron, J.F: Form-finding of a Grid Shell in Composite Materials. Journal of the International Association for Shell and Spatial Structures, 2006.
- [3] Michell, A. G. M. (1904) “*The limits of economy of material in frame-structures*”, Philosophical Magazine, Vol. 8(47),
- [4] Robin O., “*Topology Finding of Patterns for Structural Design*”, Mar. 2020
- [5] Tsz-Ho, K. & Yongqiang, L., Yong, C., “*A Structural Topology Design Method Based on Principal Stress Line*” Epstein Department of Industrial and Systems Engineering University of Southern California, Los Angeles, CA 90089, Aug. 2016
- [6] Davide, P. & Helmut P., “*Aligning principal stress and curvature directions*” Advances in Architectural Geometry 2018, Gothenburg, Sweden, Sep. 2018
- [7] Tam, Kam-Ming Mark. “*Principal stress line computation for discrete topology design*” Massachusetts Institute of Technology. Department of Civil and Environmental Engineering, Jun. 2015, <<http://hdl.handle.net/1721.1/99630>>
- [8] Junpeng W., & Jun W., & Rüdiger W., “*A Globally Conforming Lattice Structure for 2D Stress Tensor Visualization Article in Computer Graphics Forum*” DOI: 101111/cpt.13991, Jun. 2020
- [9] Cobb, Michael “*Investigating Principal Stress Lines: Optimization of Gridshell Structures*” Delft University of Technology, Oct. 2018

Appendix (Custom Tools)

A.a FEA stress line generator

To improve upon grid shell customization, a custom stress line derivation tool was developed outside of the grasshopper ecosystem. SFEPY was used to compute 3-dimensional stress fields for a grid shell. The stress field was then leveraged to derive displacements, stress tensors, principal stress components, and stress lines. Other technologies leveraged include docker, F3D, and selection of libraries including TETGEN, Numpy, PyVista, and Pydantic.

Summarized workflow is as follows: Tetrahedralize input surface mesh into volume mesh. If surface mesh is not a closed manifold, error. (Tetrahedralization does not work for open manifolds. As a gridshell customized tool input should be open surface mesh converted to a closed manifold of desired thickness in grasshopper via extrusion.). Calculate displacement and stress vectors for newly created volume mesh. Calculate Cauchy stress and strain from stress vectors. Calculates principal component vectors for each node in the volume mesh. [Get the eigen vectors from the Cauchy stress matrices after converting them from Voight notation]

Inputs: Vertices, Faces, Young Modulus, Poisson Ratio, Load Constraints, Fixed Constraints

Outputs: Tetrahedralized Mesh, Displacement Vectors, Cauchy Stress, Cauchy Strain

A.b Clean up tool for stressline generator output

[Link to algorithm flowchart Full Size](#)
[Link to Code](#)

The cleaning tool is an algorithm that takes as inputs the base mesh of the shape, this being the very object that was cast into the FEA tool or Karamba to produce the stress lines; and the set of stress lines as PolylineCurve objects, along a set of Boolean inputs that provide optional and additional functionalities and outputs a functional for remeshing set of stress lines as PolylineCurve objects, all projected to the mesh so that intersections between them are guaranteed to be points; and a new border(s) curve(s) as PolylineCurve(s) objects, with kinks being defined as the succession of start points and endpoints of the output set of stress lines. Stresslines sets generated either via FEA generator for a given seeding pattern (a) or via Karamba seeding, defined as sets of PolylineCurve objects, generally included instances of stress lines that proved idle when incorporated into a remeshing algorithm, for the reasons listed below. Said individual instances, treated within the scope of this research as recurring errors generally present in any approach whatsoever, needed to be identified and culled off the set, so that only functional PolylineCurves remained in the set. Such cases were:

“recurring” stress lines: Polycurves that went from a start point detected in the border curve of the surface/mesh all along its path to a second intersection with the border curve and then back again to the same startpoint. This type of stress line could count >2 intersections with the border(s) curve(s), whether it was only one, or a set of them.

“incomplete” stress lines: Polycurves that went from a start point detected in the border curve of the surface/mesh all along its path to a random point along its path on the mesh. This type of stress lines could count <2 intersections with the border(s) curve(s), whether it was only one, or a set of them.

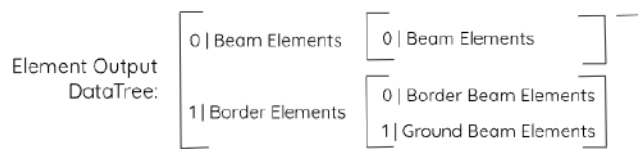
“floating” stress lines: Polycurves that went from a start point a certain point along on the mesh to another point along on the mesh. This type of stress line could count 0 intersections with the border(s) curve(s), whether it was only one, or a set of them.

Among them, if the start point coincides with the endpoint; the stressline is a ring, which isn't an idle element but a functional one. If start point and endpoint are different, the stressline is a "floating" one and therefore, disposable element when it comes to remeshing.

A.c Intersector tool

[Link to algorithm flowchart Full Size](#)
[Link to Code](#)

The intersector tool is an algorithm that takes as inputs the border curve(s) of the mesh of the analysed shape or the border curve(s) output by the cleaning tool as PolylineCurve objects, the set of stress lines to be intersected as PolylineCurve objects and an optional set of PolylineCurve objects the user could add to the set, processes the intersections between them and outputs, for the following data structure (a):



1. Elements as reprocessed PolylineCurve objects.
2. Elements as reprocessed NurbsCurve objects, degree 3.

And subsequently:

3. DataTree(a) of intersection points for Elements as 1 or 2.
4. DataTree(a) of intersection parameters per Element, for Elements as 1 or 2.
5. DataTree(a) of point indexes (b2) for Elements as 1 or 2.
6. DataTree(a) for intersecting Elements indexes (for a flat list indexation of Elements), for Elements as 1 or 2.
7. DataTree(a) for segments between intersections, for Elements as 1 or 2.
8. DataTree(a) for physically flat development of Elements, for Elements as 1 or 2.
9. DataTree(a) for intersection points parameters for physically flat Elements, for s as 1 or 2.

And for a list of non-repeated intersection points (b):

1. List of Topological Vertices
2. List of Topological Vertices Indexes
3. List of connected Vertices per vertex in b1
4. List of Element segments per vertex in b1, as data tree of indexes in b2 containing segment indexes (as flat list of segments in the order they are obtained in a7..

The intersector tool is an algorithm that executes tasks fundamentally possible on GH API with native components, but the main advantage is the script's speed and sorting of the outputs. The need for its development is rooted in an eventual full-custom-tool based approach aimed at remeshing and processing a given set of functional stress lines + border curves for structural analysis.

In this case, the problem for re-discretization of the stresslines arises (Figure A.c.1), for the fact that intersections are flat and on the original mesh doesn't logically imply that the links between intersections are edges of a new mesh, provided that edges must only have 1 segment and therefore all intersections should be processed and the links between them re-drawn.

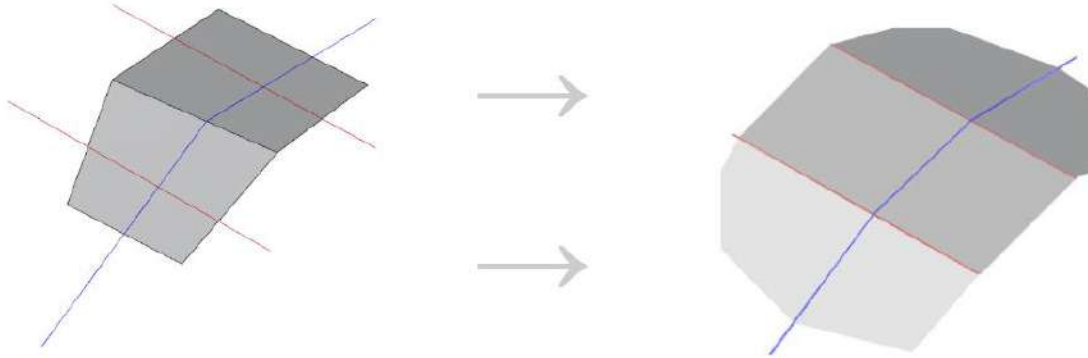


Figure A.c1. This need is absent on the Karamba+QuadRemesher approach, for the input mesh would already be a remeshing that is based on stress lines. However, the custom tool provides for data-structuring and other additional information (a3 to a7, b) that is necessary down the pipeline even for file compatibility, visualization, and as a fast approach to constructive fabrication plans if needed.

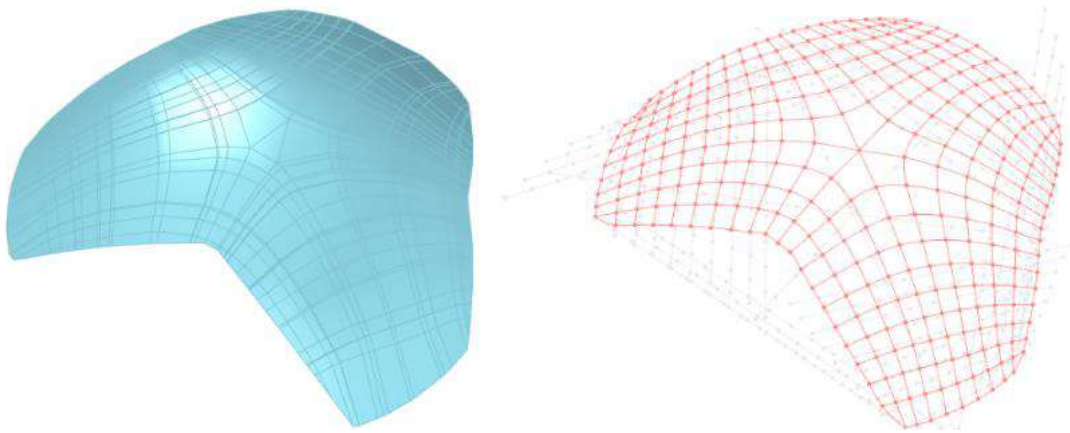


Figure A.c2 (left), Remeshing with WB MeshFromLines component, for the set of segments of the Elements processed by the intersector tool, for Elements as PolylineCurves. (right), Elements as 3rd degree NurbsCurve + physically flat development of said curves.

A.d NewBrep generator

[Link to algorithm flowchart Full Size](#)
[Link to Code](#)

The newBrep generator tool is an algorithm that takes as inputs the flat list of segments of 3rd degree NurbsCurves objects for the Elements of the gridshell as output by the Intersector tool and the list of topological segments per vertex b.4 output of the intersector tool; and outputs a BREP PolySurface object. It was designed with the aim of gaining speed when compared to native GH components, in order to have a 3d representation of the new continuous shape after the re*-discretization implied by the Intersector tool, and its main goal is to provide vectorial information for the construction of Darboux frames at any parameter along the elements, so that geodesic curvature and twisting of the section could be computed and accounted for.

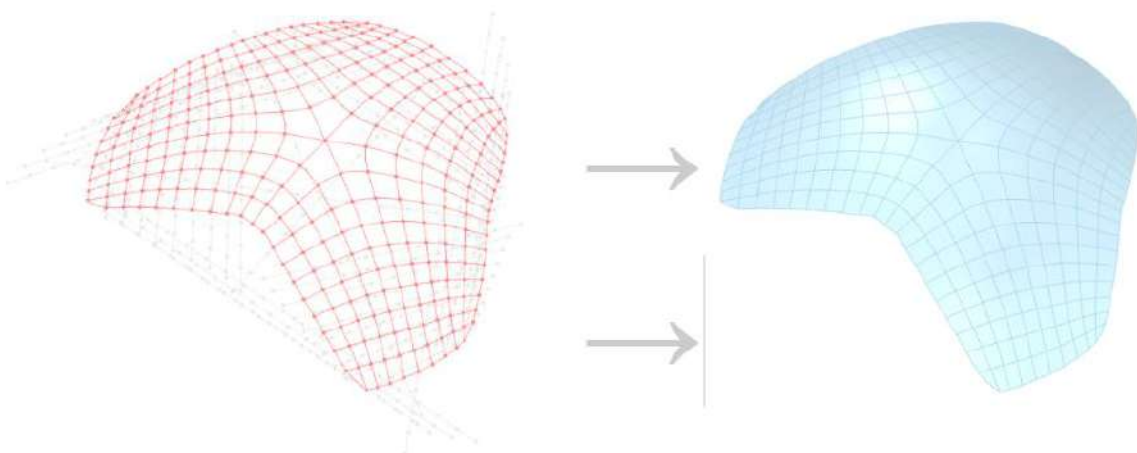


Figure A.d. Implementation of the BREP tool: From 3rd degree NurbsCurve Network to PolysurfaceBREP

A.e Curve identifier/ splitter

[Link to algorithm flowchart Full Size](#)
[Link to Code](#)

The curve identifier and splitter tool is an algorithm that takes as inputs the flat list of Elements, either PolylineCurve or 3rd degree NurbsCurves objects, the Parameter Tree per Element (a4 output of the Intersector tool), the BREP generated by the BREP generator tool, a value of length expressed in meters and a tolerance angle expressed in degrees as float; and outputs two parallel data trees (elements in one list at index n is a null object in the parallel list and vice versa) of either PolylineCurve or 3rd degree NurbsCurve objects:

1. Data tree of unaffected segments, thus being the segments of the input set of curves whose geodesic curvature implies acceptable twistings of the purported section.
2. Data tree of affected segments, thus being the segments of the input set of curves whose geodesic curvature implies unacceptable twisting of the purported section.

Either for a custom-tool-based approach or for a Karamba-stress-lines+Quadremesher approach, this tool facilitates the detection and extraction (for further processing) of that/those pieces (discretized at the gridshell's intersections) of element(s) that need to be processed in order to reach a final gridshell with feasible construction for a given section.

As stated, stress lines aren't the byproduct of an a priori geometrical approach (such as projecting planar curves, or getting planar geodesics if possible, or draping a curve network over) applied to a given shape, but the byproduct of a performative analysis of a given shape under certain load conditions. This implies that there's no previous logical guarantee of certain geometrical properties (such as planarity or twisting) for the set, other than a path that represents the very definition of stress-lines along the analyzed mesh. In this regard, adaptations can either be required to achieve a constructible status, or to facilitate it, lowering its potential costs.

In order to do this, points and binormal vectors (via Darboux frame) are obtained at regular length (input variable) intervals for each element. Subsequently, angles are computed at each of those points, between the binormal at a point and the binormal at its previous along the axis of the element.

That angle is then compared to a tolerance angle (input variable) that accounts for a maximum acceptable binormal torsion per unit of length (input variable), in the logic that for a planar curve along the given shape, all binormals would be parallel, and therefore all angles would be zero; with the caveat that the analyzed shape should be either a topological disk or a perforated disk when looking for geodesics.

For a collection of angles per element, whenever individual or successive unacceptable (larger than max tolerance) angles are found, the algorithm sticks to the previous and next intersection node of the gridshell and asks how many nodes should it go before and after, in the case that the extracted part of the element should be larger on user's request. (Figure A.e). After that, it proceeds to split the Element and derive it to both output trees, either as null object for unaffected curves data tree or as a PolylineCurve/3rd degree NurbsCurve object for the affected curves data tree.

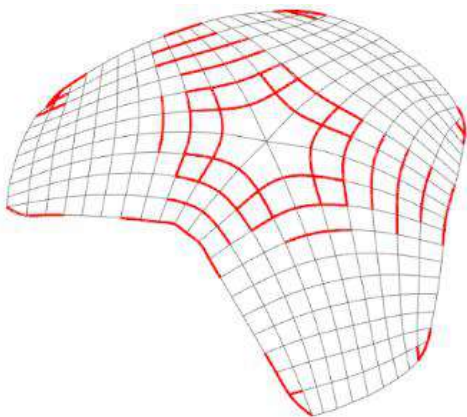


Figure A.e: In red, the unacceptable trunks of a given element, for a maximum tolerance of 3 degrees of binormal rotation per meter of length.

A.f Bracing tool

[Link to algorithm flowchart Full Size](#)
[Link to Code](#)

The bracing tool is an algorithm that takes as inputs the final remeshing of the given shape, either via QuadRemesher or via custom-tool-based remeshing as mesh object and an optional collection of manually drawn curve objects and outputs a data tree of bracing lines per face of the mesh.

Additionally, the algorithm allows the user to opt by an all faces braced case, or for a selective bracing per face case, whose criterium is that those faces in intersecting contact (along their perimeter) with any curve pertaining the set of the drawn curves input as optional would be braced, in the logic that the rigidization that bracing produces isn't necessarily uniform across the analyzed gridshell, and that such requirement can either be the byproduct of an optimization process as well as either that optimization process or any eventual case could either root itself or develop itself into a pattern that adds semiotic meaning to the appearance of the final gridshell.

Should the purported elements of bracing (considering the hypothesis is to lower the general mass of the possible artefact) had been metal cables, both diagonals would have been needed for structural simulation, given the cables incapacity to work withstanding compression. Since for an accurate peer-to-peer comparison similar conditions to that of the analogue cases were needed (being both of the cases braced with wooden elements), only one of both diagonals per face of the mesh was used.

A.g Unacceptable binormal rotation Element arches processor tool:

[Link to algorithm flowchart Full Size](#)

The Unacceptable binormal rotation Element arches processor tool is an algorithm that would take as inputs the BREP object representation of the re-discretized shape (in a custom-tool-based approach, the output of the BREP processor), the data tree of affected segments either as PolylineCurve objects or as 3rd degree NurbsCurve objects, (output 2 of Curve identification and splitter tool), and the same numerical values for length and max. angle tolerance input in the curve identification and splitter tool; and would output a data tree of the same composition (of null and curve objects) that was input, with the processed curves as Bezier-interpolated curve objects now fulfilling the binormal angular requirements previously defined and preserving the original tangent for the parts of the Elements at the start and end points of each piece.

***Singular recursive instance of the general approach before structural analysis:**

After the curves that represent constructive elements were processed in (g), a weaving of the unaffected and processed data trees for Elements would be performed and a joining of the curves within the branches of the trees would need to be ordered in order to reconstruct a list of curve objects representing constructive elements.

Subsequently, this set, and according to the nature of the elements (beam, border beam, ground beam) would be again cast into a second instance of the curve intersector tool, which would provide the physically flat development of each linear element, for fabrication.

