

Planar Quads in Architecture Free-Form Surfaces

Christian Dimitri, UPC BarcelonaTech

September 2018

Abstract

This paper will cover the preprocessing techniques for planar quad meshes in architecture free-form surfaces. As a first step, we will be covering the problems and objectives behind planar quads for construction, their benefits, their metrics as well as their goals, considering their constraints for a better optimization of the candidate *PQ mesh*. Secondly, we will explain the several preprocessing algorithms that generate a candidate *PQ mesh* ready for optimization and apply them on four different type of surfaces. In addition to that, the output will be optimized according to its properties qualifying it to be *PQ* meshes. The last-mentioned are based on scientific papers, and were applied to concrete architectural projects. Combining chapter two and chapter three iteratively, we will be hitting the last chapter of this paper; generating subdivision method algorithm and a quad planarization in order to have a *PQ* mesh.

1 Introduction

Planar Quad meshes have been nearly ubiquitous in architecture and construction. A large body of data structures and geometry processing algorithms based on them has been developed in the literature and adapted in construction of free-form surfaces. This type of re-meshing has many advantages especially the semi-regular ones, and significant progresses were made in quadrilateral mesh generation and processing during the last years. In this paper, we will study four algorithms behind planar quad meshes and their goals in order to fulfill the objectives. We will apply them on four input surfaces having different curvatures.

2 Construction

In construction, planar quads should always be planar and their distribution on the mesh is preferably equidistant so that their size does not vary a lot. In the first section, the geometric properties of *PQ* meshes are introduced as well as

their benefits over other ones. Therefore, the metrics and measures are split by type and explained graphically and mathematically.

2.1 PQ Geometric Properties

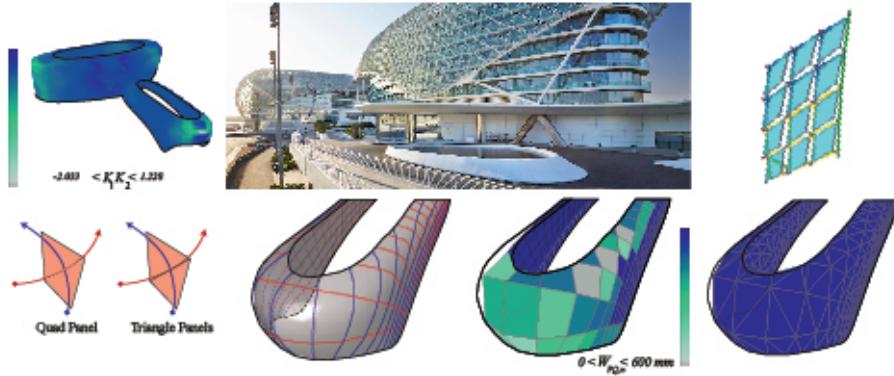


Figure 1: The hinge is affected by the high Gaussian curvature on the surface of the *Yas Island Hotel By Zaha Hadid* (Leon and Shelden 2011). The difference between *PQ meshes* and *triangle meshes*.

A polygon face is planar if and only if its vertices v_n define a plane. A triangle face is always planar, however a quadrangular face can be non-planar since the curvature plays a prominent role against the geometric property of a planar quad. Such constraint is a disadvantage for *PQ meshes* over triangular ones. Thus, if the warping height exceeds a certain limit while measuring it, the four vertices of each of the faces should be independent from its neighboring face's vertex see fig. 1. Knowing that two parallel vectors in space, enclosed at each point by two other vectors that are not necessarily parallel, form a planar face (Glymph et al. 2004). We Consider each row of faces $f_{i,j}$ is a *PQ strip*. The latter is composed by vertices $\mathbf{v}_{i,j}$ with a valence $\pm k/4 (k \in \mathbb{Z})$ where along each vertex a curve of family A and a curve of family B intersect see (Liu et al. 2006). N-geons can appear with a valence $k \neq 4$ so called singularities see fig. 2.

2.2 Benefits

Planar quad meshes may be preferred over *triangular meshes* for construction reasons. In addition, planar quads have the same fabrication and assembling benefits as triangles. The advantages of planar quads meshes for construction over other meshes is that: *PQ meshes* have higher surface to edge ratio than

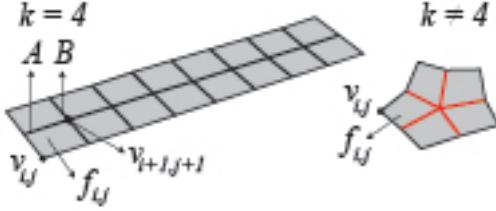


Figure 2: PQ strip and mesh valences and properties

triangles, thus, a lower mullion cost. *PQ meshes* consumes less energy during fabrication.

2.3 Metrics-Figure_3(Quality)

To have *planar quads*, several Figure_3 are mentioned below. For a better quality, the mathematical Figure_3 and the conditions are classified by face and by mesh[fig. 3 and fig. 4]. In addition to that, some conditions are translated to *custom goals* that improve the quality of the mesh.

The measurements and conditions applied to the mesh itself are:

Measures	Technical Drawings	Mathematical Descriptions
<u>Planarity and Convexity</u> A panel is flat and convex if and only if the difference between the sum of the internal angles and 2π is equal to zero		$\delta_{PQ} = \sum ai - 2 = 0$
<u>Warping Height</u> The measures of a quadrilateral element from being planar: Max element corner normal angular deviation from the normal of the mean plane		The warping W_{PQ} at the vertex v is measured by projecting the point on the plane and measuring its distance

Figure 3: Table showing the main Figure_3 of PQ meshes see (Gokhale (2008), guide (2018) and Geometry and Generator (2016))

The measurements and conditions applied to the elements of the mesh are:

Measures	Technical Drawings	Mathematical Descriptions
Element Area The area on each quad is divided by two		$PQ_{area} = \frac{\max(d) \times \min(d)}{2}$
Diagonals Aspect Ratio Maximum distance between diagonals of the quad face divided by the maximum distance of diagonals		$\eta_{PQ} = \frac{\max(d)}{\min(d)}$

Figure 4: Table showing the main Figure_3 of PQ meshes see (Gokhale (2008), guide (2018) and Geometry and Generator (2016))

Graph Pending

Figure 5: Diagram of the algorithmic strategies

3 Algorithmic Strategies

3.1 Several Pre-Processing Techniques

Given four different meshes as inputs with different curvatures, several pre-processing techniques will be adapted in order to generate a *PQ mesh* with *planar faces*. The used techniques will depend on the surface type. Translation surfaces is an easy and fast algorithm to generate specific surfaces. However architecture free-form surfaces with high curvature require more complex algorithms to generate *PQ meshes* see fig. 5.

3.1.1 Translation Surfaces

Translation surfaces are limited and easy to generate. The quads generated are the proof that it is generated through a set of parallel vectors that result in a planar face. In addition to that they are homogeneous because adding the same length vector as a constraint leads to have evenly spaces faces and reduce the variance. If the sectional curves are plane and the vectors are parallel having the same length the result will respond to the design principle of a translation surface. Assuming that one direction of the quad mesh net to be the sectional curve, two design principles can appear:

- The row of longitudinal sectional curves form parallel vectors.
- The row of lateral sectional curves form parallel vectors.

3.1.1.1 Row of sectional curves translated over a set of parallel vectors

The family of sectional curves $p(\mathbf{u})$ translated over a set of parallel vectors is generated as follows: A random spatial curve $p(\mathbf{v})$ called generatrix is translated against another random spatial curve $p(\mathbf{u})$ called directrix as seen in fig. 6. A translation by equal length gives homogeneous results of the planar quads.

Several geometrical shapes have been developed in architecture during the history using translation surfaces. The elliptical paraboloid is the most familiar shape found in architecture. It is generated using the same principle, translating one parabolic curve against another.

In transition surfaces, some geometrical shapes admit boolean and joining operations, for example, the hyperbolic paraboloid is a type of translation surface that acknowledge such operations. By translating a parabolic curve over a hyperbolic the result is as seen in fig. 8

3.1.1.2 Scale-Translation Surfaces

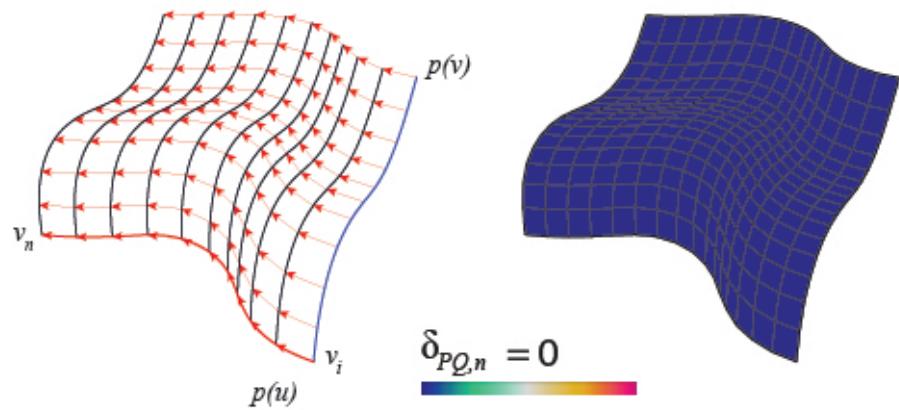


Figure 6: Geometric principle for translation surfaces and planarity measure fulfilled.

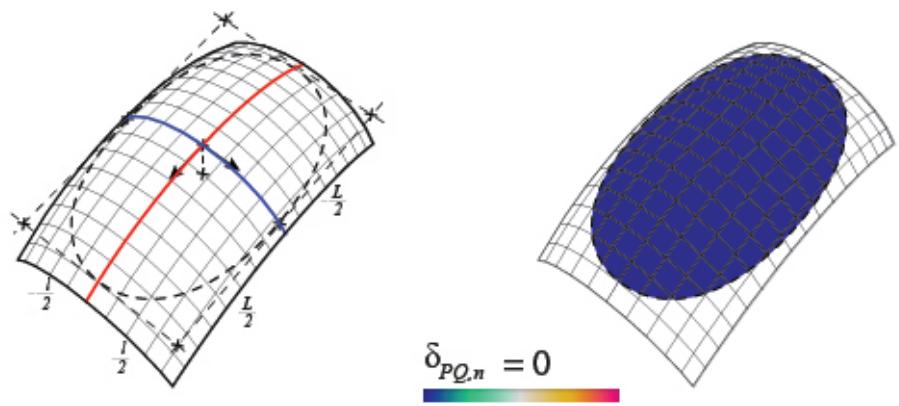


Figure 7: Elliptical paraboloid

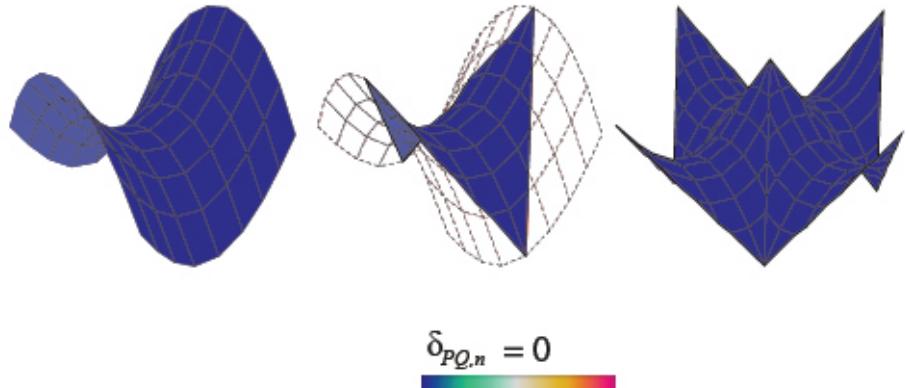


Figure 8: Translated hyperbolic paraboloid and joining possibilities

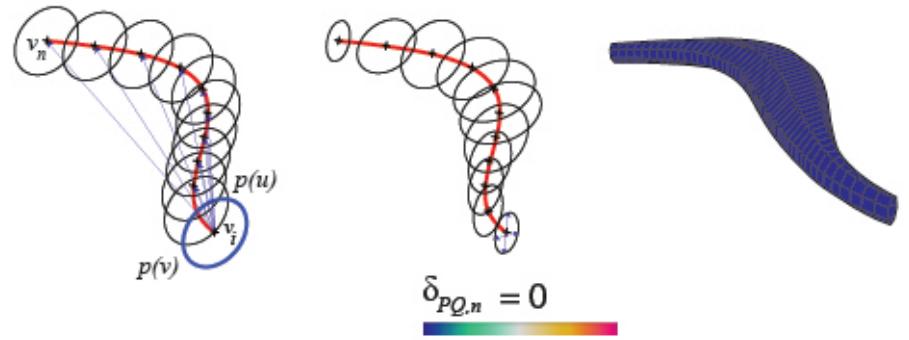


Figure 9: Centric scale-translation expansion

Scale Translation surfaces are generated by adding a scale parameter to the output curves C_n . After translating the sectional curve $p(u)$ on each point v_i equally distant from the directrix curve $p(v)$, the output curves can be scaled uniformly or non-uniformly controlled by the user. The central expansion of any curve gives a new curve having parallel edges. The center of expansion can be chosen randomly (Glymph et al. 2004). In this technique the centric expansion has been chosen. The resulting algorithm give planar quad meshes see fig. 9.

3.1.2 Conjugate networks

Some curve networks are a robust and efficient method to extract *PQ meshes* (Liu et al. 2006). Such method admits a huge variety of free-form surfaces. The advantage of designing a conjugate direction field is that the user possesses total freedom in controlling the PQ mesh layout (Zadravec, Schiftner, and Wallner 2010). Thus, the panels are flat and discretize the principal curvature lines see (Liu et al. 2006).

Left: High twisting moment. Middle: Stiffening by triangulation. Right: Torsion free alignment.(Zadravec, Schiftner, and Wallner 2010)

In addition to that, it can admit free torsion node while aligning the curve networks with the stress and curvature field see fig. ?? for more information on statics sensitive layout (Schiftner and Balzer 2010).

3.1.2.1 The relation between PQ meshes and *conjugate networks*.

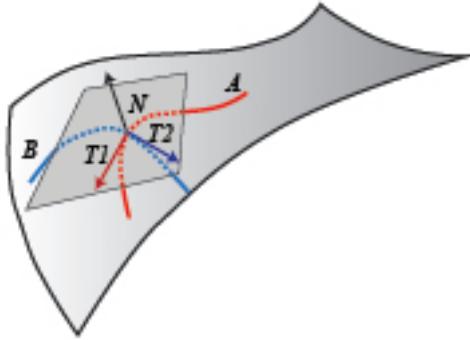


Figure 10: Conjugate curve networks on a surface

As seen in (???, Figure_11), conjugate curve networks are families of curves $A, B \subset \Phi$: For each $p \in \Phi$ unique curves of both family A, B should appear. Since T_1, T_2 are conjugate then they pre-define A and get B by integrating the vector field directions conjugate of family A . (Liu et al. 2006)

Examples of Conjugate Curve Networks on Surfaces:

- Suited for PQ meshes: (Liu et al. 2006)
 - *The network of principle curvature lines* see (fig. 11).
 - In a translation surface of the form $p(u, v)$ $\mathbf{p}(u)$ a sectional curve is translated along another curve generatrix $\mathbf{p}(v)$ and vice versa see fig. 6.
- Less suited for PQ meshes:
 - *Epipolar curves*: The translation of a point p along a line l and the intersection of the planes through the points $p(i)$ with that surface Φ generates asymptotic curves that are not suited for such meshing see (fig. 11).
 - *Isophotic curves are conjugate to the system of the steepest descent curves respecting the z-axis* see (fig. 11).

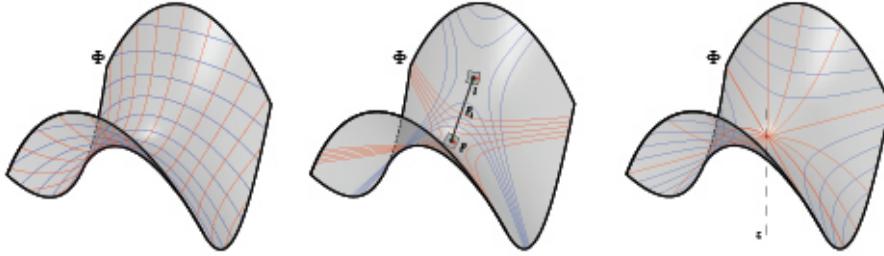


Figure 11: Various conjugate networks

3.1.2.2 What is a CDF on a *triangular mesh*.

On a smooth surface $S \subset \mathbb{R}^3$, the tangent vectors $\mathbf{v}_p, \mathbf{w}_p$ are conjugate if and only if they are treated as two vectors in \mathbb{R}^3 (Liu et al. 2011). The CDF is a tool for non-photorealistic rendering in order to visualize the surface topology. Therefore it is useful for surface re-meshing and alignment control. On a triangular face f_i as seen in fig. 12 of a triangular mesh $\mathbb{R}^3 = (V, E, F)$ a CDF is:

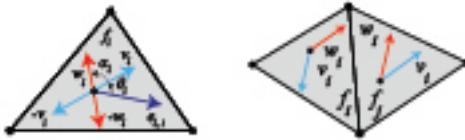


Figure 12: A CDF on a Triangular face based on (Liu et al. 2011).

- Four vectors $\{\mathbf{v}_i, \mathbf{w}_i, -\mathbf{v}_i, -\mathbf{w}_i\}$
- Two scalar parameters $\{\theta_i, \alpha_i\}$:

- θ_i oriented angle between $e_{i,1}$ and \mathbf{v}_i
- α_i oriented angle between \mathbf{v}_i and \mathbf{w}_i
- They define the following: $\mathbf{v}_i = (\cos\theta_i, \sin\theta_i)^T$ and $\mathbf{w}_i = (\cos(\theta_i + \alpha_i), \sin(\theta_i + \alpha_i))^T$

3.1.2.3 Generating quad-dominant meshes via conjugate direction field

When the input is a mesh and not a surface, it is preferable to imply isotropic re-meshing. In this case, the re-meshing tool mesh machine is used (Piker, n.d.). After re-meshing the given input meshes \mathbb{R}^3 , the conjugate direction field $[\mathbf{v}_i, \mathbf{w}_i]$ is particularly generated using a custom plugin called (Greco 2018) developed by (Mueller and Adriaenssens 2018). The candidate *PQ mesh* is generated after applying the global parametrization using frame fields and tracing the streamlines.

3.1.2.3.1 Alignment with the curvature (Mueller and Adriaenssens 2018).

Surface tangency of the mesh having double curvature \mathbb{R}^3 by computing the minimum e_1 and maximum e_2 *principle directions* in red and blue.

The quality of the mesh is always better when the panels are aligned with the curvature or the stress lines. Given four different meshes \mathbb{R}^3 , the orthogonality is introduced for each of the meshes \mathbb{R}_i^3 by computing the *principle directions* e_1 , e_2 and storing them in $[e_1, e_2]$ see fig. ???. This method has been used by (Liu et al. 2011).

3.1.2.3.2 Interpolating vector field with *N-PolyVector Field* (Mueller and Adriaenssens 2018).

In fig. 13, it is clear that the smoothed vector field and the parallel transport have been well generated. In order to find a smooth and aligned vector field $[e_1, e_2]$ on each of the four meshes \mathbb{R}^3 , the algorithm is based on finding the trade-off between neighboring faces f_i so that the parallel transport succeeds see fig. 12 right. It uses the novel method proposed by (Diamanti et al. 2014) called *N-Poly Vector Field*. While selecting a subset of points [P], the vector field $[e_1, e_2]$ is able to be generated smoothly and continuously. It finds the smoothest field by interpolating the two vectors parallelly. This method is different from the one used in (Liu et al. 2011) where a signed permutation method is used in order to find the correct relation between neighboring vectors.

3.1.2.3.3 Conjugate direction field

After smoothing the vector field in the previous step, a quad mesh can be computed after generating the conjugate networks (Liu et al. 2011). From the

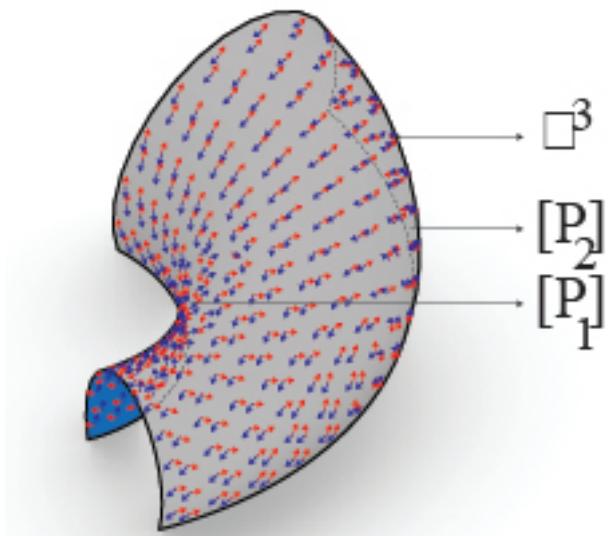


Figure 13: Smoothed vector field using n-polyVector field algorithm

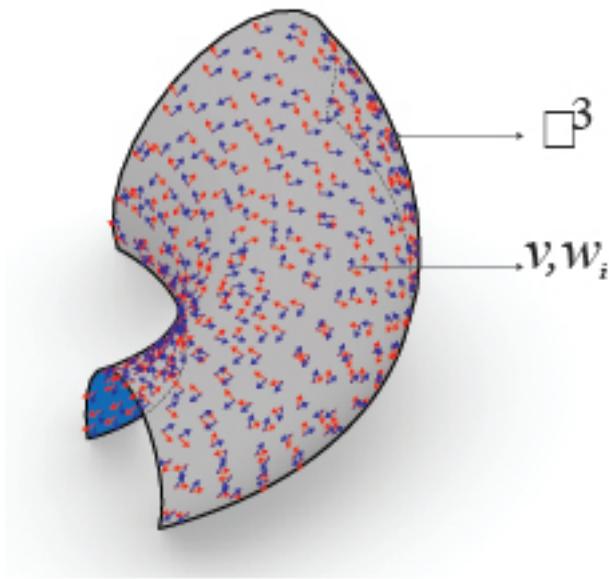


Figure 14: Conjugate field $[v_i, w_i]$ after smoothing previously the vector field $[e_{i,1}, e_{i,2}]$.

previous step a conjugate vector field $[\mathbf{v}_i, \mathbf{w}_i]$ is computed using an algorithm provided in (LibDirectional 2018) see fig. 14.

3.1.2.3.4 Global parametrization using frame fields

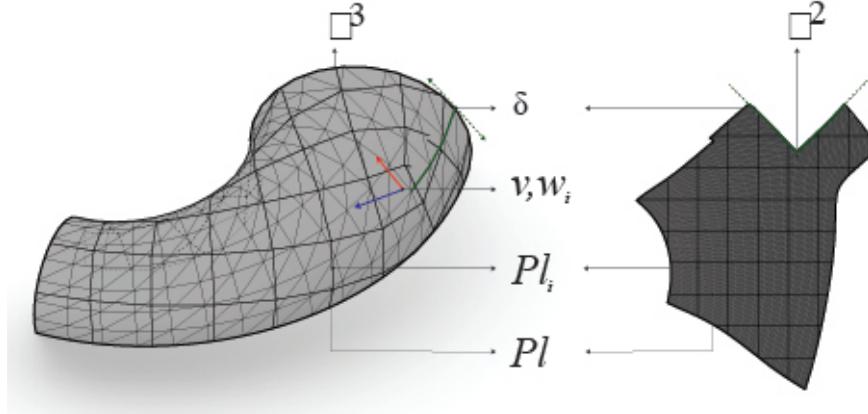


Figure 15: In green the boundary of the cutting path δ , P_{l_i} the isolines, and $\mathbf{v}_i, \mathbf{w}_i$ the frame fields chosen at index i .

If the mesh possess negative curvature, the parametrization has to be done by patches, see fig. 15 otherwise the parametrization can be done on a single patch see fig. ???. The algorithm succeeds on all the meshes except for the last one \mathbb{R}^3_4 where collisions appear. The global parameterization using frame fields fig. 15 is computed at the index i to shape the new mesh in a 2D topology. For each of the given 3D meshes, align the topology with the given vector fields $[e_1, e_2]$ at index i . Therefore, such field can be easily manipulated by the user.

3.1.2.3.5 Tracing streamlines

The streamlines $[P_{l_i}]$ are traced on the 2D maps after integrating the Vector field $[e_1, e_2]$ then they are remapped on the 3D meshes \mathbb{R}^3 . This method is based on the 4th order Runge-Kutta (Mueller and Adriaenssens 2018), see fig. 15.

3.1.2.3.6 Extracting the candidate PQ Meshes

The resulting candidate PQ mesh have a semi regular valence.

Given the conjugate filed $[\mathbf{v}_i, \mathbf{w}_i]$ and the streamlines P_{l_i} , the meshes \mathbb{R}_i^3 are generated by retrieving the faces f_i with the same vertex valence fig. ???. Nevertheless, the resulting meshes are not totally planar and require a further optimization. The mesh with a positive curvature \mathbb{R}_1^3 possess one singularity $k_{PQ} = 3_1$, the mesh with negative curvature \mathbb{R}_2^3 has singularity $k_{PQ} = 5_1$ and

the finally the mesh with double curvature as seen in fig. ?? \mathbb{R}_3^3 possess one singularity $k_{PQ} = 3_1$. To see the other resulting meshes $k_{PQ} = 3_1$ and \mathbb{R}_2^3 see the appendix.

3.1.2.4 Generating quad-dominant meshes via principle curvature networks

This method is different from the previous one. The network of curves $[Pl]$ will be generated on each of the four meshes \mathbb{R}_i^3 using a plugin called (Michalatos 2017), however the output is not sorted. Although, without a special library like (Jacobson, Panozzo, and others 2017) and (Ebke et al. 2013) to automatically extract a robust-quad mesh is very hard to achieve. This method is based on the on the mixed-integer quadrangulation by (Bommes, Zimmer, and Kobbelt 2009). Therefore, in this research an algorithm had to be developed in order to extract that candidate PQ mesh using conformal mapping.

3.1.2.4.1 Computing curvature networks

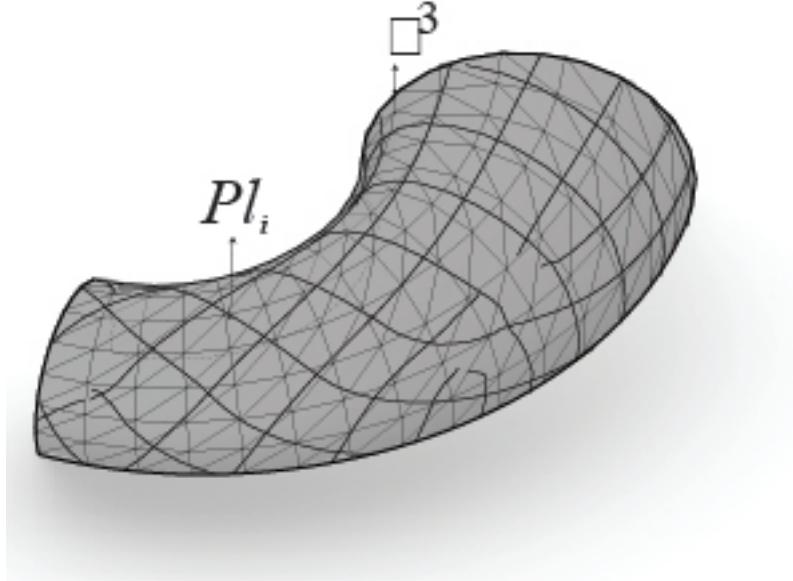


Figure 16: Curve network Pl_i computed using (Michalatos 2017) on each of the input mesh \mathbb{R}_3^3 .

The *principle curvature networks* $[Pl]$ are generated automatically by reparametrizing the input meshes \mathbb{R}_i^3 see fig. 16.

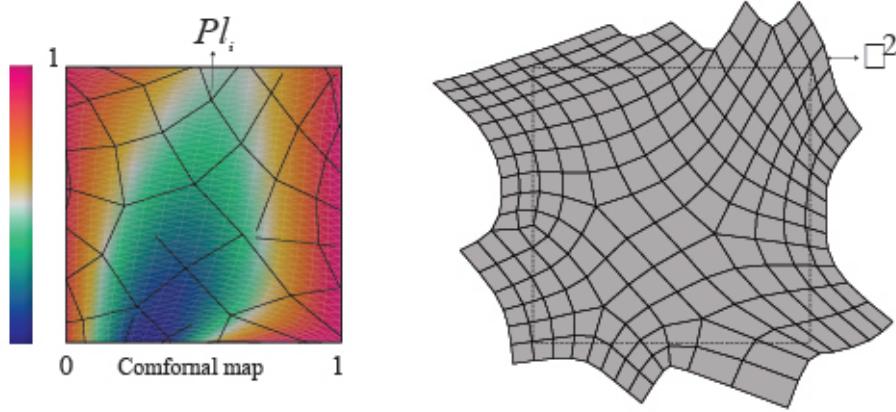


Figure 17: Conformal mapping parametrization on a unit plane and curvature color gradient. Pl_i remapped and rebuilt on the 2D map.

3.1.2.4.2 Global parametrization using conformal mapping

The curve networks $[Pl]$ previously computed, are reparametrized using conformal mapping. Then they are analyzed and rebuilt in order to close naked nodes and form meshes with a semi-regular valence.

3.1.2.4.3 Extracting the candidate PQ meshes

After mapping the *curve networks* and rebuilding the quad mesh on the unit plane, it is now possible to remap the meshes on input geometry. The surface with a double curvature \mathbb{R}_1^3 has four singularities $k_{PQ} = 3_4$, the one with a negative curvature \mathbb{R}_2^3 has two singularities $k_{PQ} = 5_2$, the one with a double curvature \mathbb{R}_3^3 has one singularity $k_{PQ} = 3_1$ and one singularity $k_{PQ} = 5_1$ see fig. 18 and finally the free-form surface \mathbb{R}_4^3 possess three singularities $k_{PQ} = 3_3$ and two singularities $k_{PQ} = 5_2$ see the appendix.

3.1.3 Conical meshes

Conical meshes are planar quad meshes which *discretize principle curvature lines*, possess an offset at a constant distance as well as planar connecting elements (Liu et al. 2006) see fig. 19. A conical mesh is conical if and only if all of its vertices \mathbf{v}_i are conical which means that the four faces meeting at the vertex \mathbf{v} are tangent to a common sphere (Liu et al. 2006) see fig. 21.

3.1.3.1 The angle criterion of a conical mesh

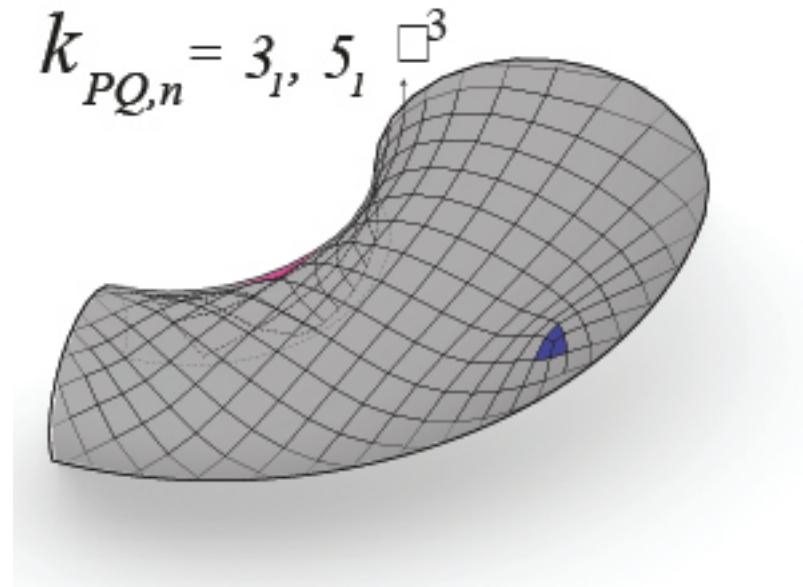


Figure 18: The resulting candidate PQ meshes have a semi regular valence.

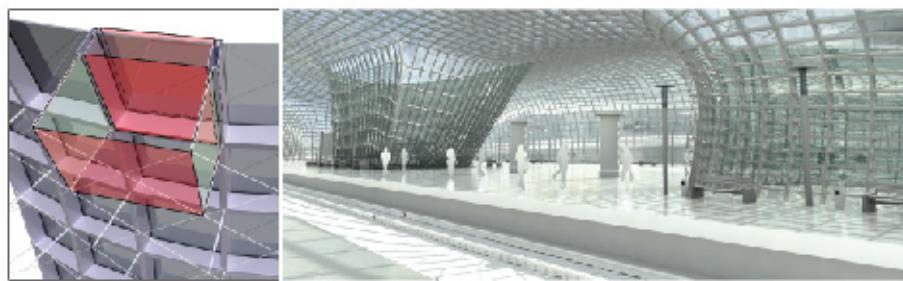


Figure 19: Left: Offset property of a conical mesh. Right: *Railway station by B.Schneider* (Liu et al. 2006) a conical mesh as glass structure that discretizes the principle curvature.

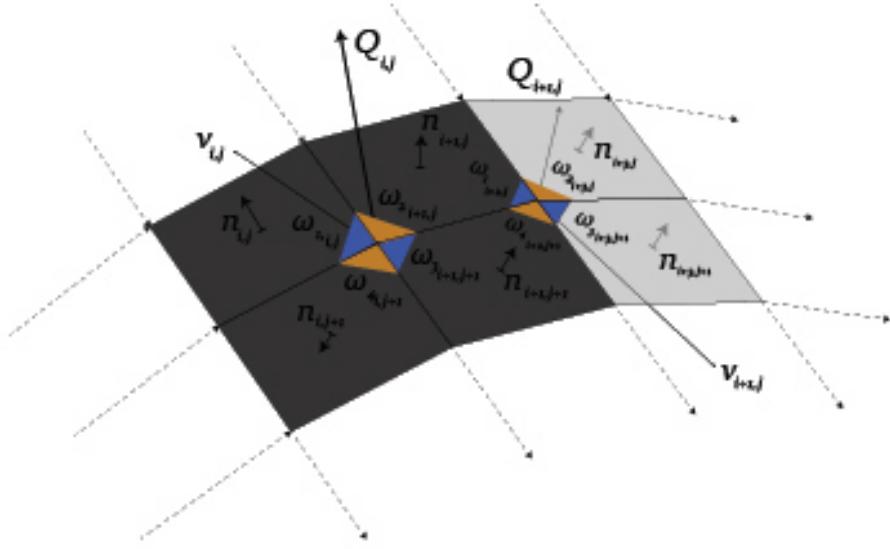


Figure 20: Faces configuration of a conical mesh (Liu et al. 2006).

Assuming that the sum of the opposite angles on a vertex \mathbf{v} should always be equal to zero, see fig. 20, \mathbf{v} is a conical vertex if and only if the characterization of a conical mesh interior angles respond to this function:

$$\omega_1 + \omega_3 = \omega_2 + \omega_4$$

3.1.3.2 The Offset Properties

Triangular meshes are missing the offset property at a constant distance, while conical meshes answer to this property (Liu et al. 2006). The faces of a conical mesh are incident to a common vertex $\mathbf{v}_{i,j}$ and tangent to a cone with an axis $Q_{i,j}$. After offsetting, the axis remains the same and the faces are still tangent to the cone (Liu et al. 2006). The Languerre transformation (Liu et al. 2006) contains one of the instances for offsetting planes by a fixed distance along their normal vector. The Languerre transformation preserves the conical meshes at the offset.

3.1.3.3 The Normals

The *spherical image* is a fact where the vertices \mathbf{v}_{ij} of a PQ mesh built on a unit sphere are converted to the normal vectors of $Q_{i,j}$. As the four faces incident to a common vertex \mathbf{v}_{ij} tangent to the same cone see fig. 20, the normal vectors $n_{i,j}$ on each of the four faces share the same angle with the cone's axis $Q_{i,j}$ see

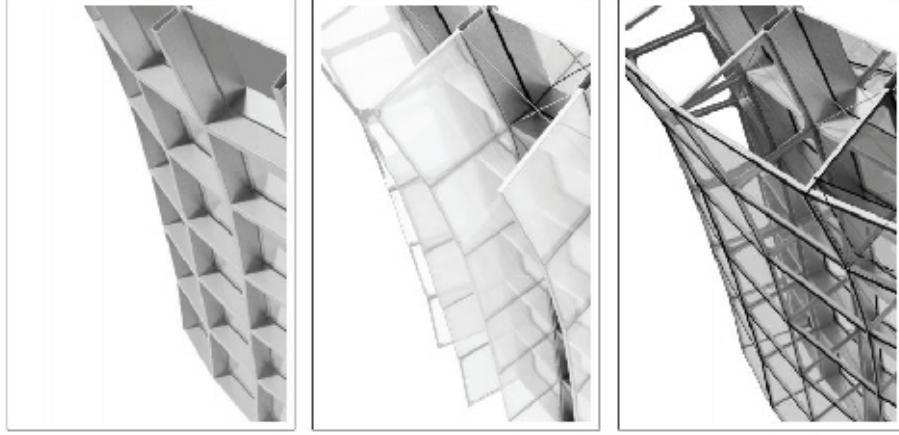


Figure 21: Constant offset of a Conical Mesh see (Pottmann and Wallner 2008).

fig. 21. Consequently, the spherical image of the principle curvature network returns an orthogonal curve network on a sphere (Liu et al. 2006).

3.1.3.4 Conical optimization

PQ meshes generated after computing the principle curve networks are well suited to be optimized using conical meshes conditions. In order to do that, the angles and normals are measured and visualized with a gradient color that varies in a range between the common meshes see fig. 22 and fig. 23.

3.1.3.4.1 Angles and normals analysis

Left: The sum of the opposite angles ω_i measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 . Right: The angles difference θ_i between the normals $n_{i,j}$ and the cones normal $Q_{i,j}$ are measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 .

3.1.3.4.2 Angles and normals optimization

For each vertex $\mathbf{v}_{i,j}$ on the mesh \mathbb{R}^3 minimize the sum of the opposite angles equals to zero $\omega_1 + \omega_3 - \omega_2 - \omega_4 = 0$ using (Piker 2010) solver.

3.2 Optimization (K2)goals

3.2.0.1 Analysis

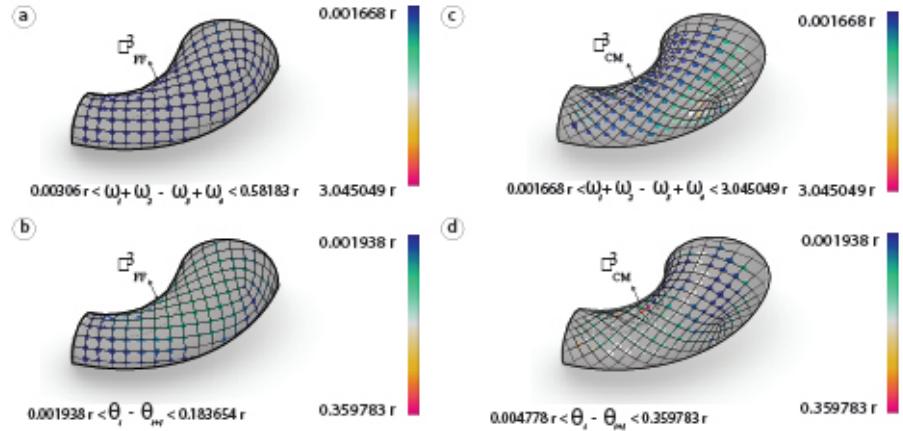


Figure 22: Left: The mesh generated via frame field is measured and shown. Right: The mesh generated via conformal mapping.

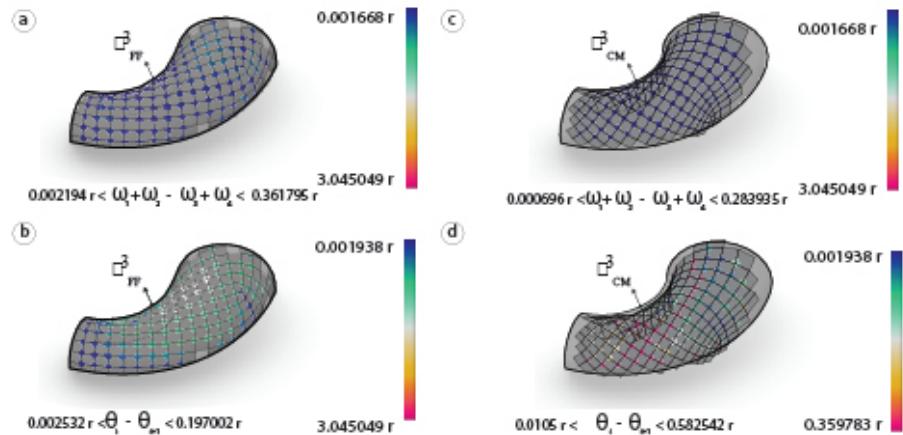


Figure 23: Left: The sum of the opposite angles ω_i measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 . Right: The angles difference θ_i between the normals $n_{i,j}$ and the cones normal $Q_{i,j}$ are measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 .

In order to optimize the panels, the planarity δ_{PQ} is measured in radians, diagonals aspect ratio η_{PQ} and the warping height h are measured in cm, and finally the area η_{PQ} and the variance are measured in cm^2 on the meshes generated via conjugate curve networks as seen in fig. 24.

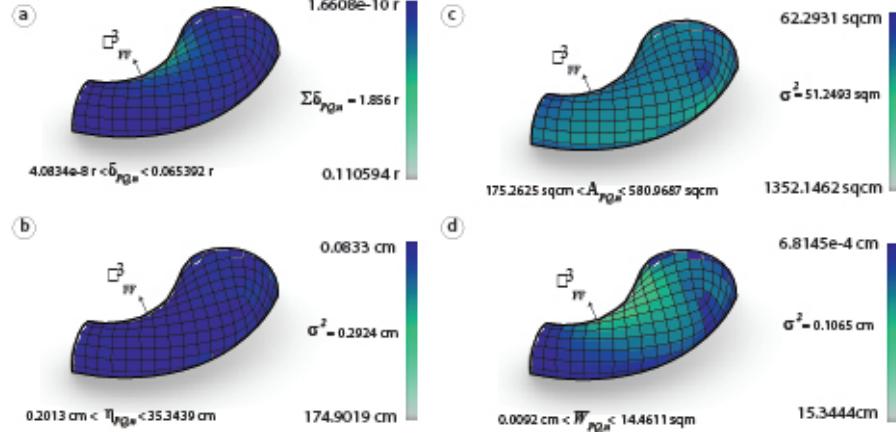


Figure 24: The mesh generated via frame field results are shown and measured above.

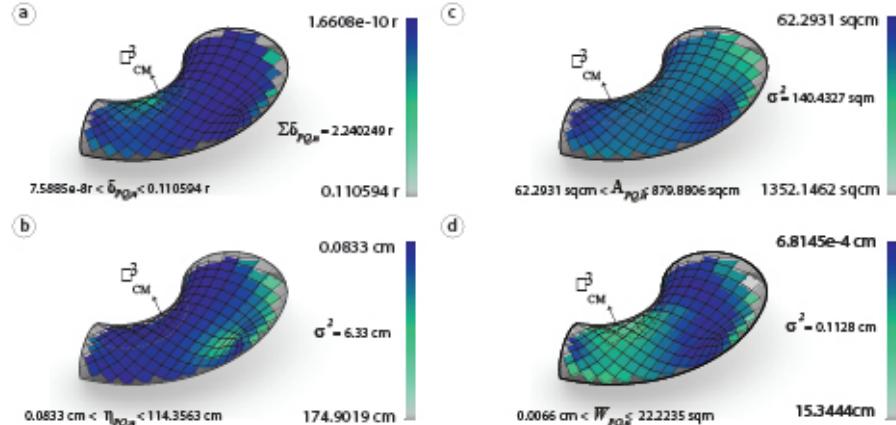


Figure 25: The mesh generated via conformal mapping are shown and measured above.

3.2.0.2 Optimization

After analyzing the panels under their required goals for them to be planar, the elements are optimized using kangaroo2 solver (Piker 2010). The results have been reduced noticeably see in fig. 26.

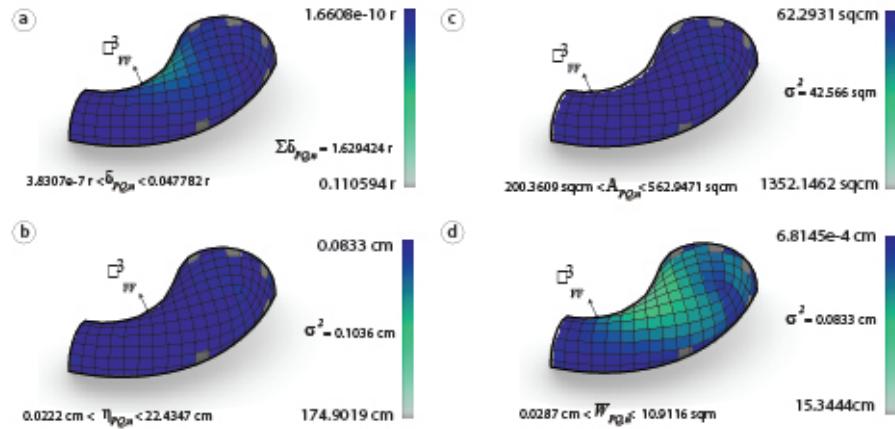


Figure 26: The mesh generated via frame field optimized are shown above.

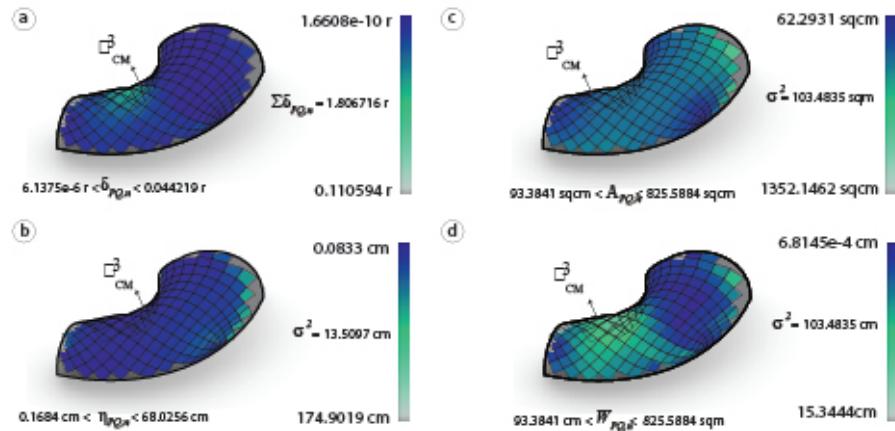


Figure 27: The mesh generated via conformal mapping optimized are shown above.

3.3 Subdivision Strategy (Starting with a Coarse Quad-Dominant mesh)

3.3.1 Principles

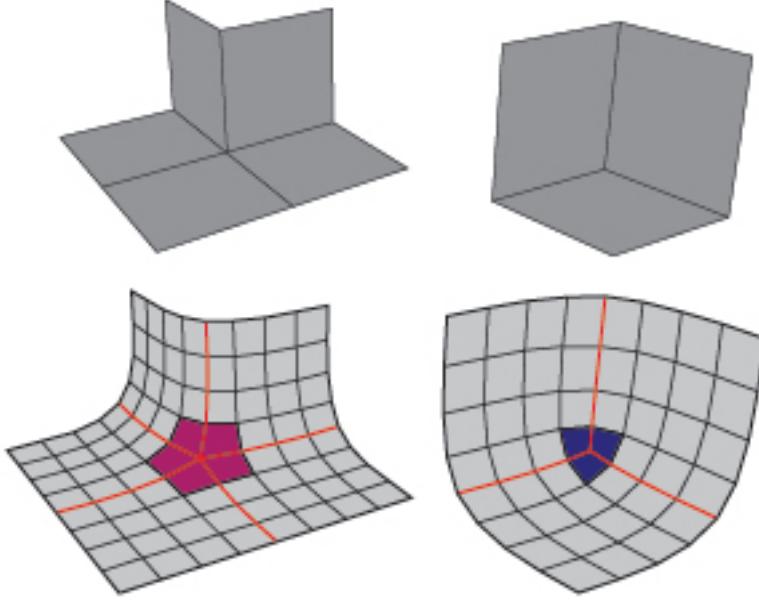


Figure 28: Left: Singularities with negative indices. Right: Singularities with positive indices.

A coarse mesh that approximates the topology of a input surface can be subdivided using the catmull-clark algorithm (Piacentino 2010). For PQ meshes, the valence of the each vertex should be four, vertices with a valence more then four are considered as singularities. After applying the subdivision on the coarse mesh, singularities with negative indices take a negative curvature and singularities with a positive indices take a positive curvature see fig. 28.

3.3.2 Curvature and singularities analysis

On the given input meshes, the curvature K is analyzed and the singularities are placed by index see fig. 29.

3.3.3 Generating the coarse mesh

Subsequently to the previous step, a 2D map by patches is generated. Such a method can help out predicting the pre-networking between singularities and



Figure 29: Singularities with indices $-\frac{1}{4}$ and $\frac{1}{4}$ are placed accordingly to the curvature.

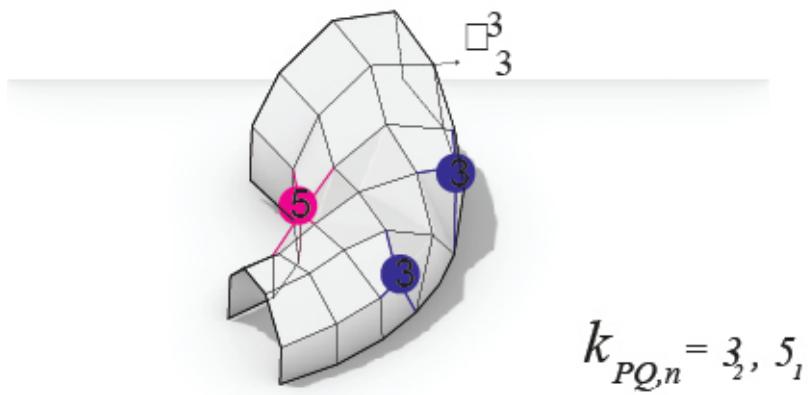


Figure 30: The coarse meshes are generated thru isocurves.

avoiding unexpected ones. Therefore it is now possible to generate the coarse mesh following the 2D map see fig. 30.

3.3.4 Catmull-clark subdivision and pull to mesh

Left: Subdivided mesh using catmull-clark algorithm and singularities in color.
Right: Pulling the subdivided mesh to the input surface.

The catmull-clark algorithm is applied to the coarse meshes. Using kangaroo2 (Piker 2010) the coarse mesh is pulled by constraining the latter's points on the input meshes. The returning outputs are the candidate *PQ meshes* that need iterative optimization fig. ??.

3.3.5 Analysis

After analyzing the panels under their required goals for them to be planar, the elements are optimized using kangaroo2 solver (Piker 2010). The results have been reduced noticeably see in fig. ??.

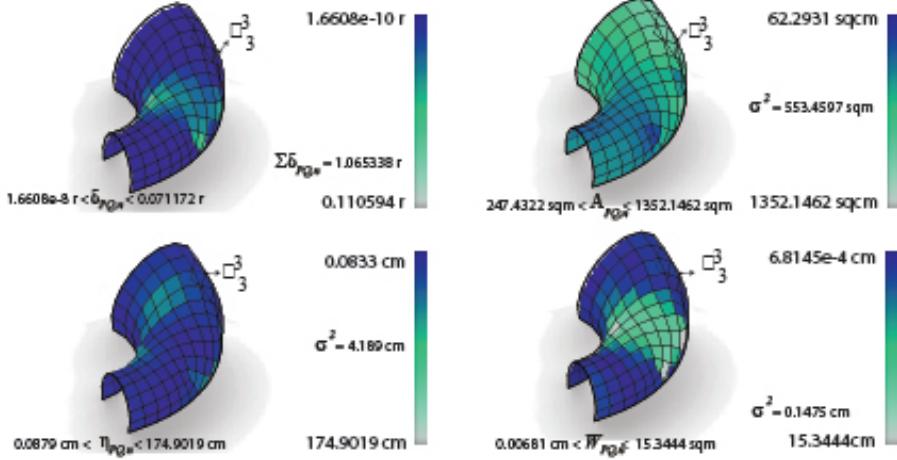


Figure 31: The mesh generated via frame field optimized are shown above.

3.4 Comparison & Synthesis

4 Conclusion

PQ meshes must show different results from mere geometry. The planarity of the faces should obey the goals in order to fulfil the basis of the *planar quad*

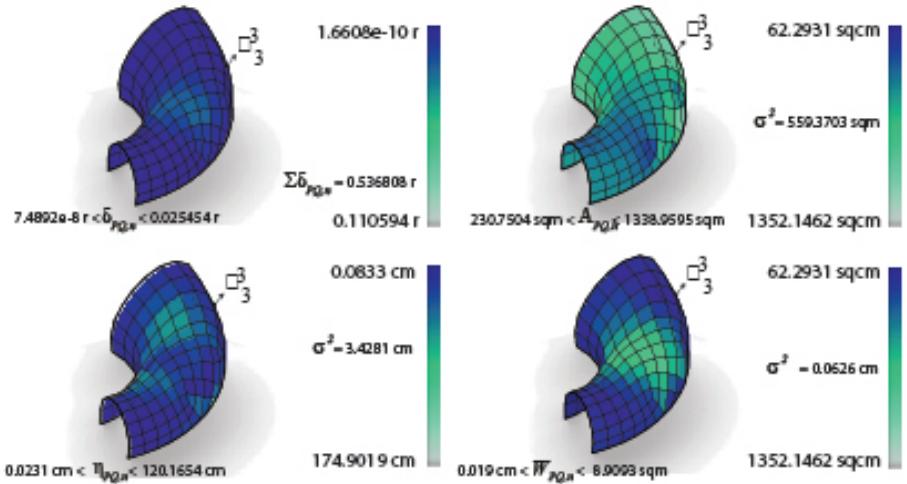


Figure 32: The mesh generated via conformal mapping optimized are shown above.

Graph Pending

Figure 33: Graph showing the MAD of Aspect ratio in cm over the planarity in rad for the four meshes \mathbb{R}_i^3 .

Graph Pending

Figure 34: Graph showing the MAD of Areas in cm^2 over the warping height in cm for the four meshes \mathbb{R}_i^3 .

meshes (Zadravec, Schiftner, and Wallner 2010). They are very hard to deal with when the input surface is a free-form. However some algorithms have shown the differences between them and their results. Having a conjugate direction field as a tool to control the mesh layout is very useful. Thus, generating PQ meshes from curve network is strongly accurate. The two different methods are almost planar after generation since they are extracted from the principal directions. The conical optimization has proven its effectiveness over planar quad meshes. By optimizing and combining the methods, the last one was to generate planar quads by subdividing a coarse mesh and then optimizing it to planar. The boundary condition has been neglected in these methods, however, we recommend the first method due to its smooth results and its manipulation liability.

5 Further work

For further research the boundary will be taken in consideration while generating the PQ meshes. The fourth mesh \mathbb{R}_4^3 that failed in the frame field algorithm has to be developed accordingly to its curvature for it to unwrap while avoiding collisions.

Bommes, David, Henrik Zimmer, and Leif Kobbelt. 2009. “Mixed-Integer Quadrangulation.” In *ACM Transactions on Graphics (ToG)*, 28:77. 3. ACM.

Diamanti, Olga, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. “Designing N-Polyvector Fields with Complex Polynomials.” In *Computer Graphics Forum*, 33:1–11. 5. Wiley Online Library.

- Ebke, Hans-Christian, David Bommes, Marcel Campen, and Leif Kobbelt. 2013. “QEx: Robust Quad Mesh Extraction.” *ACM Transactions on Graphics (TOG)* 32 (6). ACM:168.
- Geometry, CUBIT Sandia, and Mesh Generator. 2016. “Metrics for Quadrilateral Elements.” 2016. https://cubit.sandia.gov/public/15.2/help_manual/WebHelp/mesh_generation/mesh_quality_assessment/quadrilateral_metrics.htm.
- Glymph, James, Dennis Shelden, Cristiano Ceccato, Judith Mussel, and Hans Schober. 2004. “A Parametric Strategy for Free-Form Glass Structures Using Quadrilateral Planar Facets.” *Automation in Construction* 13 (2). Elsevier:187–202.
- Gokhale, Nitin S. 2008. *Practical Finite Element Analysis*. Finite to infinite.
- Greco, Lorenzo. 2018. “Plugin for Advanced Mesh Editing, Remeshing, Quads from Fields, Mesh Analysis.” 2018. <https://www.food4rhino.com/app/capybara>.
- guide, Autodesk Nastran Reference. 2018. “Geometry Check Options.” 2018. <https://knowledge.autodesk.com/support/nastran/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/NSTRN-Reference/files/GUID-69125711-AB89-4A46-8FA9-B7DB2C856A13.htm.html>.
- Jacobson, Alec, Daniele Panozzo, and others. 2017. “libigl: A Simple C++ Geometry Processing Library.”
- Leon, Alexander Peña de, and Dennis Shelden. 2011. “A Technique for the Conditional Detailing of Grid-Shell Structures: Using Cellular Automata’s as Decision Making Engines in Large Parametric Model Assemblies.” In *Computational Design Modelling*, 267–73. Springer.
- LibDirectional. 2018. “Matlab Library for Directional Statistics and Directional Estimation.” 2018. <https://github.com/libDirectional>.
- Liu, Yang, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. 2006. “Geometric Modeling with Conical Meshes and Developable Surfaces.” In *ACM Transactions on Graphics (Tog)*, 25:681–89. 3. ACM.
- Liu, Yang, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. 2011. “General Planar Quadrilateral Mesh Design Using Conjugate Direction Field.” In *ACM Transactions on Graphics (Tog)*, 30:140. 6. ACM.
- Michalatos, Panagiotis. 2017. “Millipede for Structural Analysis and Optimization.” 2017. <http://www.sawapan.eu/>.
- Mueller, Caitlin, and Sigrid Adriaenssens. 2018. “Optimized Quad Gridshell from Stress Field and Curvature Field.”
- Piacentino, Giulio. 2010. “Weaverbird – Topological Mesh Editor.” 2010. <http://www.giuliopiacentino.com/weaverbird/>.

- Piker, Daniel. 2010. “Kangaroo Is an Interactive Physics/Constraint Solver and Grasshopper Plugin for Designers.” 2010. <http://kangaroo3d.com/>.
- . n.d. “MeshMachine for Remeshing.” <https://www.grasshopper3d.com/profiles/blogs/meshmachine-update?id=2985220%3ABlogPost%3A1085830&page=2>.
- Pottmann, Helmut, and Johannes Wallner. 2008. “The Focal Geometry of Circular and Conical Meshes.” *Adv. Comp. Math* 29:249–68.
- Schiftner, Alexander, and Jonathan Balzer. 2010. “Statics-Sensitive Layout of Planar Quadrilateral Meshes.” *Advances in Architectural Geometry 2010*. Springer, 221–36.
- Zadravec, Mirko, Alexander Schiftner, and Johannes Wallner. 2010. “Designing Quad-Dominant Meshes with Planar Faces.” In *Computer Graphics Forum*, 29:1671–9. 5. Wiley Online Library.