Planar Quads in Architecture Surfaces

Christian Dimitri, UPC BarcelonaTech

September 2018

Planar quads are found everywhere in architectural free-form surfaces. The first chapter will cover the problems behind planar quads for constructability, there benifits, there metrics and there constraints goals for optimization. The second chapter will embed several pre-processing algorithms that generates a candidate mesh for being a planar quads mesh. Therefore, in the third chapter, those candidate meshes will be optimized according to some conditions that qualify them to be PQ meshes. Those conditions are based on scientifique papers. And finally by combining the chapter two and chapter three, iteratively a subdivision method algorithm and a quad planarization will be done in order to have a PQ mesh.

# Constr

*PQ meshes* must show different results from the mere geometry since the planarity of faces should obbey the goals in order to fulfil the basis properties of the *planar quad meshes* [@zadravec2010designing].

## PQ Geometric Properties

The hinge is affected by the high Gaussian curvature on the surface of the *Yas Island Hotel By Zaha Hadid* [@de2011technique]. The difference between *PQ meshes* and *triangule meshes*.

The hinge is affected by the high Gaussian curvature on the surface of the *Yas Island Hotel By Zaha Hadid* [@de2011technique]. The difference between *PQ meshes* and *triangule meshes*.

Since a triangular face in space is always planar, then they are easier to deal with when the curvature is very high. The latter leads the quad panels to bend and the four vertices will want to move from the plane. 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 independant from its neighbouring face’s vertex see @fig:GoemetricPrincipals.

In theory two parallel vectors in space enclosed at each point by two other vectors not necesseraly parallel form a planar face, for more information refer to the author [@glymph2004parametric]. Consider each row of faces is a PQ strip. PQ meshes are composed by vertices with a valence where along each vertex a curve of family A and a curve of family B intersect see [@liu2006geometric]. N-geons can appear with a valence so called singularities.

## Benefits

*Planar quad* meshes are prefered over *triangule meshes* for aesthetical and constructional reasons. The advantages of *planar quads* meshes for construction over other meshes is that:

* *PQ meshes* are less heavy.
* *PQ meshes* are a less material use.
* *PQ meshes* consumes less energy during fabrication.

## Metrics-Measures(Quality)

To have *planar quads*, several measures are mentioned below. For a better quality, the mathematical measures and the conditions are classified by face and by mesh[@fig:Measures and @fig:Measures2}. In addition to that, some conditions will be translated to *custom goals* that will improve the quality of the mesh.

The measurements and conditions applied to the mesh itself are:

Table showing the main measures of PQ meshes see (@gokhale2008practical, @GEOMCHECK and @QuadQualityDefinitions)

Table showing the main measures of PQ meshes see (@gokhale2008practical, @GEOMCHECK and @QuadQualityDefinitions)

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

Table showing the main measures of PQ meshes see (@gokhale2008practical, @GEOMCHECK and @QuadQualityDefinitions)

Table showing the main measures of PQ meshes see (@gokhale2008practical, @GEOMCHECK and @QuadQualityDefinitions)

# Algorithmic Strategies

## Several Pre-Processing Techniques

Several pre-processing techniques will be adapted in order to generate a *PQ mesh* with *planar faces*. The technique will be used depending on the surface type. Translational surfaces is an easy and fast algorithm. However architecture free-form surfaces with high curvature, requires more complex algorithm to generate *PQ meshes*.

### Translational Surfaces

Translational surfaces are limited and easy to generate. The quads generated are 100% the proof that it is generated thru a set of parallel vectors which results a planar face. In addition to that they are homogeneous because adding the same length vector as a constaint 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 principle can appear:

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

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

The family of sectional curves translated over a set of parallel vectors is generated as follow: A random spatial curve called generatrix is translated against another random spatial curve called directrix as seen in [@fig:TranslationPrinciples]. Thus considering that translation by equal length gives homogeneous results of the planar quads.

Geometric principle for translation surfaces and planarity measure fulfilled.

Geometric principle for translation surfaces and planarity measure fulfilled.

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

Elliptical paraboloid

Elliptical paraboloid

In translational surfaces, some geometrical shapes admit boolean and joining operations, for example, the hyperbolic paraboloid is a type of translational surfaces that acknowledge such operations. By translating a parabolic curve over a hyperbolic the result is as seen in [@fig:HyberbolicParaboloid]

Translated hyberbolic paraboloid and joining possibilities

Translated hyberbolic paraboloid and joining possibilities

#### Scale-Translation Surfaces

Centric scale-translation expansion

Centric scale-translation expansion

Scale Translational surfaces are generating by adding a scale parameter to the output curves . After translating the sectional curve on each point equally distant at the directrix curve , the output curves can be scaled uniformly or non-uniformly controlled by the user. The centrical expansion of any curve gives a new curve having parallel edges. The center of expansion can be chosen randomly [@glymph2004parametric]. In this technique the centric expansion has been chosen. The resulting algorithm gives planar quad meshes see [@fig:ExpTrans].

### *Conjugate networks*

Some curve networks are robust and efficient method to extract *PQ meshes* [@liu2006geometric]. Such method admit a huge variety of free-form surfaces. The advantage of designing a *conjugate direction field* is that the user possess freedom while controlling the *PQ mesh* layout [@zadravec2010designing]. Thus, the panels are flat and *discretize the principle curvature lines* see [@liu2006geometric].

Left: High twisting moment. Middle: Stiffening by triangulation. Right: Torsion free alignment.[@zadravec2010designing]

Left: High twisting moment. Middle: Stiffening by triangulation. Right: Torsion free alignment.[@zadravec2010designing]

In addition to that, it can admit free torsion node while aligning the curve networks with the stress and curvature field see [@fig:torsionFree] for more information on statics sensitive layout refer to [@schiftner2010statics].

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

On a smooth surface $S \subset {\Bbb R}{^3},$ the tangent vectors are conjugate if and only if they are treated as two vectors in ${\Bbb R}{^3}$ [@liu2011general]. The CDF is a tool for non-photorealistic rendering in order to visualize the surface topology. Therefore it is usefull for surface remeshing and alignement control. On a triangular face as seen in [@fig:singleCDF] of a triangular mesh ${\Bbb R}{^3}=(V,E,F)$ a CDF is:

A CDF on a Triangular face based on [@liu2011general].

A CDF on a Triangular face based on [@liu2011general].

* Four vectors {}
* Two scalar parameters {}:
  + oriented angle between and
  + oriented angle between and
  + They define the following: and

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

Conjugate curve networks are families of curves : For each unique curves of both family should appear. Since are conjugate then they predefine and get by *integrating the vector field directions conjugate of family* .[@liu2006geometric]

*Examples of Conjugate Curve Networks on Surfaces*:

* Suited for PQ meshes: [@liu2006geometric]
  + *The network of principle curvature lines* see (@fig:conjugateNetworks left).
  + In a translational surface of the form a sectional curve is translated along a another curve generatrix and vice versa see @fig:TranslationPrinciples.
* Not suited for PQ meshes:
  + *Epipolar curves*: The translation of a point along a line and the intersection of the planes threw the points with that surface generate asymptotic curves that are not suited for such meshing see (@fig:conjugateNetworks center).
  + *Isophotic curves are conjugate to the system of the steepest descent curves with respect to the z-axis* see (@fig:conjugateNetworks right).

Various conjugate networks

Various conjugate networks

#### Generating quad-dominant meshes via conjugate direction field

The input mesh should not be random, it is preferable to use a remeshing tool such as mesh machine to control the edges length and the fixed boundaries or others, for more info see [@MeshMachine].

##### Alignment with the curvature [@mueller2018optimized].

Surface tangency extracted on each of the meshes $\Bbb{R}^{3}\_{i}$ by computing the minimum and maximum *principle directions* in red and blue.

Surface tangency extracted on each of the meshes $\Bbb{R}^{3}\_{i}$ by computing the minimum and maximum *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. Givin four different meshes $\Bbb{R}^{3}$, the orthogonality will be introduced for each of the meshes $\Bbb{R}^{3}\_{i}$ by computing the *principle directions* and and storing them in see @fig:PrincipalDirections. This method has been used by [@liu2011general].

##### Interpolating vector field with *N-PolyVector Field* [@mueller2018optimized].

Smoothed vector field using n-polyVector field algorithm

Smoothed vector field using n-polyVector field algorithm

In order to find a smoothed and well aligned vector field on each of the four meshes $\Bbb{R}^{3}$, in this section a plugin called [@Capybara] developed by [@mueller2018optimized] has been used. The algorithm is based on finding the trade-off between neighboring faces so that the parallel transport succeed. It uses the novel method proposed by [@diamanti2014designing] called *N-Poly Vector Field*. While selecting a subset of points [P], the vector field 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 [@liu2011general] where it uses a signed permutation technique in order to find the correct vector’s relation between neighbouring vertices. In [@fig:npolyvectorfield], it is well clear how the smoothed vector field and the parallel transport have been well generated.

##### *Conjugate direction field*

Conjugate field after smoothing previously the vector field .

Conjugate field after smoothing previously the vector field .

After smoothing the vector field in the previous step, a quad mesh can be generated after defining the conjugate networks, [@liu2011general] took further this topic for more information refer to the reading. From the previous step a conjugate vector field with [@Capybara] is computed using an algorithm provided in [@LibDirectional] see [#fig:Conjugatefield].

##### Global parametrization using frame fields

a) The first mesh $\Bbb{R}^{3}\_{1}$, b) The second mesh $\Bbb{R}^{3}\_{2}$. In green the boundary of the cutting path , the isolines, and the frame fields chosen at index .

a) The first mesh $\Bbb{R}^{3}\_{1}$, b) The second mesh $\Bbb{R}^{3}\_{2}$. In green the boundary of the cutting path , the isolines, and the frame fields chosen at index .

If the mesh possess negative curvature and is varying highly, the parametrization has to be done by patches, see @fig:GPmap otherwise the parametrization can be done on a single patch see @fig:GPmap2. The latter depend on the umbilics and the rigids. the algorithm succeeded with all the meshes except for $\Bbb{R}^{3}\_{4}$ which is in continous research.

c) The third mesh $\Bbb{R}^{3}\_{1}$, d) The fourth mesh $\Bbb{R}^{3}\_{1}$ which is still in continous research.

c) The third mesh $\Bbb{R}^{3}\_{1}$, d) The fourth mesh $\Bbb{R}^{3}\_{1}$ which is still in continous research.

This method is based on the *global parametrization with frame fields* [@fig:GPmap] it is generated using a custom component developped by the author of [@Capybara]. It is used to shape the new mesh in a different typology, the latter has to be aligned with some given vetors by interest of the user. The Mixed-integer quadrangulation by [@bommes2009mixed] is one way to do that and another way is using *Anisotropic remeshing to concentrate the elements in the regions with more details*[@mueller2018optimized] published by [@diamanti2014designing] an open source library can do that see [@Libgil] and [@Ebke:2013:QRQ:2508363.2508372].

##### Tracing streamlines

The streamlines are traced on the 2D maps after integrating the Vector field then they are remapped on the 3D meshes. They are generated using a custom component in [@Capybara] that is developed using the order Runge-Kutta for more information refer to [@mueller2018optimized] see [@fig:GPmap and @fig:GPmap2].

##### Extracting the candidate PQ Meshes

The meshes are generated after tracing the streamlines in the previous step. Therefore the results are seen in [@fig:CandidatePQMeshFromCDF]. The resulting meshes are a candidate PQ mesh ready for optimization.

The resulting candidate PQ meshes have a semi regular valence. a) $\Bbb{R}^3\_{1}$ with one singularitie . b) $\Bbb{R}^3\_{2}$ with one singularitie . c) $\Bbb{R}^3\_{3}$ with one singularitie .

The resulting candidate PQ meshes have a semi regular valence. a) $\Bbb{R}^3\_{1}$ with one singularitie . b) $\Bbb{R}^3\_{2}$ with one singularitie . c) $\Bbb{R}^3\_{3}$ with one singularitie .

#### Generating quad-dominant meshes via principle curvature networks

This method is different from the previous one. The network of curves will be generated on each of the four meshes $\Bbb{R}^{3}\_{i}$ using[@Millipede], however the output is not sorted. Although without a special library like [@libigl] and [@Ebke:2013:QRQ:2508363.2508372] to extract automatically a robust-quad mesh [@ebke2013qex] is very hard to achieve. This method is based on the on the mixed integer quadrangulation by [@bommes2009mixed]. Therefore, an algorithm had to be developed in order to extract that candidate PQ mesh using conformal mapping.

##### Computing curvature networks

Curve network computed using [@Millipede] on each of the input meshes $\Bbb{R}^3\_{i}$.

Curve network computed using [@Millipede] on each of the input meshes $\Bbb{R}^3\_{i}$.

The *principle curvature networks* will be generated automatically using the reparamterized component developed in [@Millipede] see [@fig:CurveNetwork].

##### Global parametrization using conformal mapping

Conformal mapping parametrization on a unit plane and curvature color gradiant. remapped and rebuilt on the 2D map.

Conformal mapping parametrization on a unit plane and curvature color gradiant. remapped and rebuilt on the 2D map.

*The curve networks* previously extracted, 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.

##### Extracting the candidate PQ meshes

The resulting candidate PQ meshes have a semi regular valence. a) $\Bbb{R}^3\_{1}$ with four singularities . b) $\Bbb{R}^3\_{2}$ with two singularities . c) $\Bbb{R}^3\_{3}$ with one singularitie and one singularitie .

The resulting candidate PQ meshes have a semi regular valence. a) $\Bbb{R}^3\_{1}$ with four singularities . b) $\Bbb{R}^3\_{2}$ with two singularities . c) $\Bbb{R}^3\_{3}$ with one singularitie and one singularitie .

After mapping the *curve networks* and rebuilding the quad mesh on the unit plane, it is now possible remap the meshes on the original surfaces see @fig:conformalmapping and @fig:CandidatePQmeshes2}.

### Conical meshes

Left: Offset propetry of a conical mesh. Right: *Railway station by B.Schneider* [@liu2006geometric] a conical mesh as glass structure that *discretizes the principle curvature*.

Left: Offset propetry of a conical mesh. Right: *Railway station by B.Schneider* [@liu2006geometric] a conical mesh as glass structure that *discretizes the principle curvature*.

Conical meshes are planar quad meshes which *discretize principle curvature lines*, posses and offset at a constant distance as well as planar connecting elements [@liu2006geometric] see [@fig:ConicalMeshes]. A conical mesh is conical if and only if all of its vertices are conical which means that the four face planes meeting at are tangent to a common sphere [@liu2006geometric] see [@fig:facesConfiguration].

Faces configuration of a conical mesh [@liu2006geometric].

Faces configuration of a conical mesh [@liu2006geometric].

#### The angle criterion of a conical mesh

The sum of the opposite angles on a vertex should always be equals to zero so see [@fig:facesConfiguration]. is a conical vertex if and only if the charaterization of a conical mesh interior angles should respond to this function:

#### The Offset Properties

Triangular meshes are missing the offset property at a constant distance. However conical meshes have this property while generating conical meshes at the offset [@liu2006geometric].

Constant offset of a Conical Mesh see [@pottmann-2008-fg].

Constant offset of a Conical Mesh see [@pottmann-2008-fg].

The fact that the faces of a conical mesh are incident to a common vertex and tangent to a cone with an axis . After offseting the axis remains the same and the faces are still tangent to the cone [@liu2006geometric].

The Languerre transformation [@liu2006geometric] 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.

#### The Normals

*The spherical image* is a fact where the vertice of a PQ mesh built on a unit sphere are converted to the normal vectors of . As the four faces incident to a common vertice tangent to the same cone, the normal vectors on each of the four faces share the same angle with the cone’s axis see @fig:facesConfiguration. Consequently the spherical image of the principle curvature network returns an orthogonal curve network on a sphere [@liu2006geometric].

#### 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 gradiant color that varies in a range between the common meshes see @fig:Anglesbfopt1, @fig:Anglesbfopt2, @fig:normalsbfopt1 and @fig:normalsbfopt2.

##### Angles analysis

Before optimization: The sum of the opposite angles measured in radians for each vertex of the meshes $\Bbb{R}^3\_{i}$ generated via CDF.

Before optimization: The sum of the opposite angles measured in radians for each vertex of the meshes $\Bbb{R}^3\_{i}$ generated via CDF.

Before optimization: The sum of the opposite angles measured in radians for each vertex of the meshes $\Bbb{R}^3\_{j}$ generated via *principle curvature networks*.

Before optimization: The sum of the opposite angles measured in radians for each vertex of the meshes $\Bbb{R}^3\_{j}$ generated via *principle curvature networks*.

##### Normals analysis

Before optimization: The angles difference between the normals and the cones normal are measured in radians for each vertex of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: The angles difference between the normals and the cones normal are measured in radians for each vertex of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

The first mesh $\Bbb{R^3}\_{1}$

Before optimization: The angles difference between the normals and the cones normal are measured in radians for each vertex of the meshes $\Bbb{R}^3\_{j}$ generated via *frame field*.

Before optimization: The angles difference between the normals and the cones normal are measured in radians for each vertex of the meshes $\Bbb{R}^3\_{j}$ generated via *frame field*.

##### Angles optimization

For each vertex on the mesh $\Bbb{R}^3$ minimize the sum of the opposite angles equals to zero using [@Kangaroo3d] solver.

After optimization: The sum of the opposite angles in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

After optimization: The sum of the opposite angles in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

After optimization: The sum of the opposite angles in radians of the meshes $\Bbb{R}^3\_{j}$ generated via *conformal mapping*.

After optimization: The sum of the opposite angles in radians of the meshes $\Bbb{R}^3\_{j}$ generated via *conformal mapping*.

##### Normals optimization

For each vertex on the mesh $\Bbb{R}^3$ minimize the angles difference between the four adjacents faces normals and the cones normal should be equal to zero .

After optimization: The angles difference in radians of the meshes $\Bbb{R}^3\_{i}$ generated via frame field

After optimization: The angles difference in radians of the meshes $\Bbb{R}^3\_{i}$ generated via frame field

After optimization:The angles difference of the meshes $\Bbb{R}^3\_{j}$ generated via frame field.

After optimization:The angles difference of the meshes $\Bbb{R}^3\_{j}$ generated via frame field.

## Optimization (K2)goals

### Planarity

#### Analysis

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

#### Optimization

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

### Diagonals Aspect Ratio

#### Analysis

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

#### Optimization

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

### Element areas

#### Analysis

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

#### Optimization

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via frame field\*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via frame field\*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

### Warping height

#### Analysis

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

#### Optimization

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *frame field*.

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via *conformal mapping*.

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

### Subdivision Strategy principles

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

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

A coarse mesh that approximate the topology of a input surface can be subdivided using catmull clark algorithm [@Weaverbird]. 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:SubdivisionTechnique].

### Curvature and singularities analysis

Singularities with indices -$1\over4$ and $1\over4$ are placed accondring to the curvature.

Singularities with indices -$1\over4$ and $1\over4$ are placed accondring to the curvature.

Givin four different input surfaces, first the curvature is analyzed and the singularities are placed by index see [@fig:Placingsingularities].

### Generating the coarse mesh

The coarse meshes are generated thru isocurves

The coarse meshes are generated thru isocurves

Subsequently to the previous step, a 2D map by patches have been generated. Such method can help out to predict the pre-networking between singularities and to avoid unexpected ones. Therefore it now possible to generate the coarse mesh following the 2D map see [@fig:CoarseMesh].

### 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.

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

Thereafter, the catmull-clark algorithm has be added to the coarse mesh. Therefore, using kangaroo2 [@Kangaroo3d] the coarse mesh has been pulled by constraining the latter’s points on the input mesh. Finally the coarse mesh takes the input shape see [@fig:subresult].

### Optimization

The panels are ready to be fully optimized by constraining the faces under the planarity goal using [@Kangaroo3d] solver, surface fairness and planarity and continuously modified in order to achieve the objectives.

#### Planarity

##### Analysis

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

Before optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

##### Optimization

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

After optimization: Planarity measured in radians of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Diagonals aspect ratio

#### Analysis

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

Before optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Optimization

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

After optimization: Diagonals aspect ratio measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Element areas

#### Analysis

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

Before optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Optimization

After optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

After optimization: Area and the variance measured in sqcm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Warping height

#### Analysis

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

Before optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

#### Optimization

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

After optimization: The height and the variance measured in cm of the meshes $\Bbb{R}^3\_{i}$ generated via subdivision technique.

## Comparison & Synthesis

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{1}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{1}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{2}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{2}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{3}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{3}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{4}$ generated by the three different techniques

Graph showing the difference between the elements areas and the number of panels for each of the first mesh $\Bbb{R}^3\_{4}$ generated by the three different techniques

# Conclusion

PQ meshes are very hard to deal with when the input surface is a free-form. However some algorithms have shown the differences between them and there results. Having a conjugate direction field as a tool to control the mesh layout is very yuseful. Thus generating PQ meshes from curve network is robust as well. The two different methods are almost planar after generation since they are extracted from the principal directions. The conical optimization has proved its robustness 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 optimize it. The boundary condition has been neglected, for further research the latter will be taken in consideration while genearting the PQ meshes.