The development of computing tools to approximate doubly curved surfaces with quadrilateral planar facets

vorgelegt von
Dipl.-Ing.
Florian Gauss
ORCID: 0000-0002-1695-4357

an der Fakultät VI Planen Bauen Umwelt
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Mike Schlaich
Gutachter: Prof. Dr.-Ing. Klaus Rückert
Gutachter: Prof. Dr. C. K. Williams

Tag der wissenschaftlichen Aussprache: 18.September 2019

Berlin 2020

The development of computing tools to approximate doubly curved surfaces with quadrilateral planar facets

by Florian Gauss

The development of computing tools to approximate doubly curved surfaces with quadrilateral planar facets

by

Dipl.-Ing. Florian Gauss
Diploma in Structural Engineering
University of Stuttgart 1997

submitted to

Faculty VI - Institute of Architecture
Department for Design and Structure
at the Berlin University of Technology
in accordance with examination regulations dated 05.02.2014

in partial fulfilment of the requirements for the degree of a

Doktor-Ingenieur (Dr.-Ing.)

Vorsitzernder	Prof. Dr.-Ing. Mike Schlaich
Hauptberichter:	Prof. Dr.-Ing. Klaus Rückert
Mitberichter:	Prof. Dr. C. K. Williams

Day of viva voce: 18.September 2019

Acknowledgement

Architecture created digitally might not involve the use of actual materials [BeFo97]. However the translation of these architectural visions created by the use of programming, computer modelling and algorithms into actually built objects poses a huge challenge to the work of a structural engineer. For a long time the tessellation of free-form building envelopes in an economic but nevertheless architecturally pleasant way remained challenging and the motivation for this thesis arose in part out of the years of collaboration with architects during my employment with the Advanced Geometry Unit (AGU) at Arup. This thesis would not have been possible without the support of a number of people who accompanied me during years of studies and research with their guidance, advise and not least encouragement:

First and foremost I offer my sincerest thanks to my supervisor, Prof. Dr.-Ing. Klaus Rückert, who for the professional discussions and thought-provoking impulses and for allowing the special frame conditions under which this thesis was realised. I would also like to thank Dr. C.K. Williams for taking over the role of the ‚Mitberichter'.

I owe my deep gratitude to Cecil Balmond, my superior and mentor at Arup's Advanced Geometry Unit, for much valued conversations and advice as well as for giving me the opportunity to work and research in the interdisciplinary environment of the AGU. At AGU I worked with a number of great people whose contribution to the making of this thesis deserves special mention: Paul Jeffries for setting up the framework of the enhanced meshing plug-in, Manja van de Worp for her detailed research on the state of the art of digital fabrication technology and Daniel Hambleton for his research on isothermic surfaces and circle patterns.

I would also like to express my gratitude to the following professionals: Professor Dr. John M. Sullivan for his introduction into the mathematics of polyhedral surfaces and Professor Dr. Ulrich Pinkall who offered advice and insight in the computation and theory of discrete k surfaces. Dr. Chris Williams for his guidance in the theory of principal curvature lines and for the conversation about his work regarding the form finding

process of the British Museum courtyard roof.
Dr. Paul Shepperd for elucidating his research on subdivision surfaces.
Hugh Whitehead and Francis Aish of Foster and Partners‘ specialist modelling group for sharing their knowledge on the parametric modelling of free-form surfaces.
Professor Dr. Pottmann for the very helpful conversation about his research and the recent developments in planar quad meshing.
Dr. Hans Schober for sharing his knowledge about translation surfaces.
Paul Cowell of Simply Rhino UK for providing me with a license of Rhino NURBS modeling programme.
Prof. Dr. Dr. Werner Sobek, supervisor of my diploma thesis ‚Facettierung von Flächen‘, which initiated my studies and research and provided the intellectual foundation for this thesis.

I am indebted to all those who supported me in any respect - especially and at the same time mentioned as representatives for all others - my parents-in –law who unflaggingly gave me back-up with all the day-to-day-tasks and thus provided the freedom for the completion of this thesis.

And finally, I would like to express my very special thanks to my beloved wife Simone for supporting me throughout all my research in various ways but especially for providing the intellectual space which allowed me to realise my dissertation in parallel to my employment.

This work is dedicated to my children Sander and Leandra whose unconditional love kept me going even throughout the toughest task of writing up this thesis.

Schorndorf, October 2017 Florian Gauss

Abstract

Free form architectural design created with the aid of digital media requires a fundamental change of the approach to the entire planning and building process as it might not involve the use of actual materials. However architectural visions created by the use of programming, computer modelling and algorithms have to be translated somehow into real buildings. One of the biggest challenges is the mastering of the produce ability of the building components and foremost - as probably the most significant part of a building - mastering its envelope.
Complex geometry needs to be broken down into elements that not only size-wise but also with respect to their geometrical description can be manufactured and assembled. This process requires the tessellation of surfaces. The triangulation of a surface is a well documented process however from an architectural and economical point of view the planar quadrilateral tessellation of a surface offers several advantages.
Yet the accurate tessellation (every single joint lies exactly in the surface) of an arbitrary surfaces with planar quads remains an unsolved problem.
There are no generally recognised principles of generation to date in contrary to the well researched principle of triangulation. So far the generation of PQ facets is only possible for certain sub-classes of surfaces, which in them self feature simple generating methods.
To address this problem an approximation network is generated which has to follow certain rules: the topology of the network is determined by the properties of the surface which shall be approximated. This means the faces are positioned in such a way so their edges will always follow the principal curvature lines of the surface. Usually the singular points of a PQ mesh surface are characterised by the number of 4 edges per joint. The approximation mesh will be manually sculpted whilst considering the position/ location of those singular points whose number of members differ from the usual 4 edges in a typical joint of a PQ mesh.
The final step of the process is to optimise the approximation network by using dynamic relaxation principles.
A number of case studies carried out on surfaces of different levels of complexity show that the principle of approximation network in combination with dynamic relaxation allows to facet almost any doubly

curved surface designed in a scale typical for the construction industry. There might be a few surfaces whose shape is too complex to apply the tool successfully in the first run. These are usually surfaces which feature areas of high curvature which resulting to small curvature radii.
In this case by making a number of test runs the output surface although at first not satisfactory for panelisation can be used to isolate those surface areas which are causing problems. After applying slight changes to only those parts of the surface the process can be repeated and the output surface should result in a PQ mesh suitable for manufacturing.
Depending on the complexity of the input surface the generated quads are planar or nearly planar within a tolerance that is determined by the user before generating the approximation network. Obviously the tolerance is correlated/linked directly to the envisaged material the quads will be manufactured of.
A method is now available that allows translating almost any doubly curved shape into a network of quadrilateral elements which feature the characteristic of planarity in terms of manufacturing requirements. This might influence the approach of designers to free-form architecture significantly as the realisation of complex geometries so far was only possible for design offices with specialist units who were specially trained in mastering geometry problems.
On the one hand high efficient/powerful CAD and 3D modelling programmes offered tools which allowed the advanced user with a view clicks to create the most complex shapes, objects and surfaces. On the other hand the translation of these virtual buildings into physical objects often necessitated a redesign in sense of simplification and still meant considerable manufacturing costs.

This situation posed a big limitation to the design freedom of architects but if to be made available to the free economy the tool might serve as a link between designer and manufacturer – a kind of translator of exceptional design visions into built reality.

Abstract

Ende des 20.Jahrhunderts entstand durch die stetige Weiterentwicklung von CAD-Software und den zugehörigen 3D-Werkzeugen mit der Freiform-Architektur eine neue architektonische Stilrichtung, die sich äußerst komplexer, häufig ungleichmäßiger konvexer und konkaver Formen bedient (heute auch Blobitecture genannt, nach William Safire, 2002, New York Times Magazine).
Entwürfe der sogenannten Freiform-Architektur, die zunächst lediglich im virtuellen Raum existieren, erfordern jedoch eine grundlegende Umstellung der Herangehensweise für den gesamten Planungs- und Bauprozess:
Visionen im digitalen Raum lassen zunächst häufig Überlegungen bezüglich ihrer Umsetzbarkeit mit reellen Baumaterialien wie Stein, Glas, Stahl oder Holz außer Acht. [BeFo97]. Eine neue Herausforderung besteht somit darin, komplexe Formen, die im Computerzeitalter mittels digitaler Werkzeuge wie Programmierung, B-Spline und NURBS-Modelling sowie der Nutzung von Algorithmen erschaffen wurden, in die gebauteWirklichkeit zu übersetzen. Eine der größten Aufgaben stellt in diesem Zusammenhang die Übersetzung der Großform eines Entwurfs in individuelle Gebäudeteile dar: jeder Entwurf muss, wenn er denn tatsächlich gebaut werden soll, in einzelne Bauteile übersetzt werden, ohne jedoch die ursprüngliche Formensprache des Entwurfs zu stark den Restriktionen der Fertigungstechnik zu unterwerfen. Gleichzeitig sollten die Gebäudeteile aber zum großen Teil vorfertigbar und in hoher Anzahl repetitiv sein, um die Wirtschaftlichkeit und somit Baubaubarkeit eines Gebäudeentwurfs zu gewährleisten. Vor allem die Realisierung der Gebäudehülle als eines der wichtigsten da Gestalt prägenden Elemente eines Bauvorhabens, stellt eine große Herausforderung dar.
Komplexe Geometrien müssen in Einzelelemente unterteilt werden, welche sowohl in Bezug auf ihre Elementgröße als auch hinsichtlich ihrer geometrische Beschreibung mit den in der Industrie verfügbaren Methoden (CAM computer-unterstützte Fertigung und File-To-Factory-Fertigung) hergestellt und anschließend auf der Baustelle montiert werden können. Geometrisch betrachtet erfordert dies die Unterteilung einer im mathematischen Sinn kontinuierlichen Ausgangsfläche durch die Methode der Vernetzung (Tesselierung). Die dreieckige Vernetzung

einer Fläche (Triangulieren) ist eine mathematisch gut dokumentierte Methode, allerdings bietet die Vierecksvernetzung mit ebenen Elementen (Planar Quad oder auch PQ-Vernetzung) aus architektonischer wie auch wirtschaftlicher Sicht eine Anzahl von Vorteilen. Die akkurate Vernetzung (jeder einzelne Knotenpunkt der erzeugten Paneele liegt präzise in der vorgegebenen Fläche) einer beliebigen Ausgangfläche mit ebenen Vierecken stellt allerdings bis dato ein mathematisch ungelöstes Problem dar: im Gegensatz zu den gut erforschten Prinzipien der Triangulierung konnte noch keine allgemeingültige Methode entwickelt werden, mit der eine beliebige Freiformfläche in ein PQ-Netz 'übersetzt' werden kann.
Die Erzeugung von PQ-Netzen ist bisher nur für spezielle Flächenkategorien nachgewiesen, welchen bereits hinsichtlich der Generierung der Ausgangsfläche einfache, gut erforschte mathematische Prinzipien zu Grunde liegen.
Diese Arbeit befasst sich daher mit der Aufgabe, einen allgemeingültigen Lösungsansatz für das Generiereneines PQ-Netzes auf einer beliebigen Freiformflächen zu finden.
Hierfür wird zunächst ein'Topologienetz' auf einer willkürlich erzeugten Ausgangsfläche generiert, das folgende Bedingung erfüllen muss: die Topologie des Netzwerks ist durch dieselben Eigenschaften definiert, welche auch die Ausgangsfläche definieren, die approximiert werden soll. Hierfür wird zunächst eine Analyse der Ausgangsfläche zur Bestimmung ihrer Hauptkrümmungslinien vorgenommen. Im nächsten Schritt können mittels der Hauptkrümmungslinien die singulären Punkte (Nabelpunkte) der Fläche dargestellt werden. Während sich in den Kreuzungspunkten normalerweise jeweils zwei Hauptkrümmungslinien schneiden, ist die Anzahl der sich kreuzenden Linien in den singulären Punkten größer als 2. Diese Flächenbereiche müssen bei der anschließenden Erzeugung des Approximationsnetzes entsprechend berücksichtigt werden. Das Approximationsnetz wird manuell generiert unter Beachtung der für die Facettierung gewünschten Paneelgröße. Abschließend wird das Netz durch mehrmaliges Anwenden einer modifizierten Version der Dynamischen Relaxation soweit optimiert, bis die gewünschte Planarität der Paneele unter Berücksichtigung eines vordefinierten Toleranzwertes erreicht ist.
Eine Reihe von Fallstudien, durchgeführt an Flächen unterschiedlicher Komplexität, beweisen, dass mittels der Methode des 'Annäherungsnetzes' und anschließender Optimierung durch 'Dynamic Relaxation' jede doppelt gekrümmte Fläche mit ebenen Paneelen näherungsweise 'belegt' werden kann.
Vereinzelt kann es vorkommen, dass für Flächen mit besonders komplexer Startgeometrie ein Anwendungsdurchlauf der beschriebenen Methode nicht ausreicht. Hierbei handelt es sich im Allgemeinen um Flächen, die in einzelnen Bereichen sehr stark gekrümmt sind und daher sehr kleine Kurvenradien aufweisen.
In diesem Fall kann mit Hilfe einer Anzahl von Testläufen ein erstes Topologienetz erzeugt werden, das zwar noch nicht die endgültige Lösung darstellt, jedoch als Zwischenstand verwendet werden kann, um diejenigen Flächenbereiche zu ermitteln, die hinsichtlich der gewünschten Planarität Probleme verursachen. Diese Bereiche können dann geometrisch geringfügig angepasst werden, wodurch nach erneuter Anwendung der

‘Dynamic Relaxation’ letztendlich ein Annäherungsnetz mit der gewünschten PQ-Geometrie vorliegt.
Je nach Komplexität der Eingabefläche sind die erzeugten Vierecke planar oder annähernd planar innerhalb eines Toleranzbereichs, der vom Nutzer vor der Anwendung des Prozesses definiert werden muss. Dieser-Toleranzbereich ist durch die technischen Eigenschaften des Materials vorgegeben, aus dem die Paneele hergestellt werden sollen.

Somit steht also ein Verfahren zur Verfügung, mit dessen Hilfe jede beliebige doppelt gekrümmte Fläche mit einem Netzwerk aus planaren-Vierecken angenähert werden kann, wobei der Begriff ‘planar’ nicht in mathematisch absolutem Sinn zu sehen ist, sondern entsprechend der Material bedingten maximal möglichen Abweichung der Paneele.

Die Tatsache, dass für dieses bis dato ungelöste geometrische Problem ein Lösungsansatz entwickelt wurde, könnte die Herangehensweise von Architekten und Entwerfern bezüglich Freiformarchitektur maßgeblich beeinflussen.

Komplexe CAD-und 3D-Modellierungs-Tools ermöglichten dem-versierten Nutzer entsprechender Programmen zwar, mit wenigen “Mausklicks” hochkomplexe Formen zu erzeugen, der Versuch, diese in einen für das Bauwesen typischen Maßstab zu transferieren und dann tatsächlich zu bauen, scheiterte jedoch oft am fehlenden mathematischen Wissen und resultierte nicht selten in einer kompletten Überarbeitung des ursprünglichen Entwurfs.

Entsprechend eingeschränkt war die Gestaltungsfreiheit vielerArchitekten bezüglich freier und somit komplexer Gebäudeformen. Sollte die entwickelte Methode jedoch zukünftig auf dem freien Markt als Planungswerkzeug zur Verfügung stehen, können entwerferische Visionen unabhängig davon, wie komplex die zu Grundeliegende Geometrie sein mag, relativ problemlos in die gebaute Wirklichkeit überführt werden. Es ist ein Bindeglied geschaffen zwischen virtuellem Raum und Realität, zwischen ‘nackter’ Geometrie und den Zwängen der herstellenden Industrie.

Table of contents

Chapter 1 Buildability of freeform architecture and its limitations

1 Buildability of free-form architecture and its limitations

1.1 Introduction

During the non-digital architectural age the architect designed buildings using tools like sketches, drawings and physical models which were all produced by hand. The simple geometrical appearance buildings featured during the 'pre-digital' age clearly reflected the limited design and working methods available in those days when building compositions consisted mainly of a variety of different planar shapes. As William Mitchell [Ko03] describes it, *"Architects drew what they could build and built what they could draw."* If curvy elements formed part of a building's geometry these were usually clearly defined geometrical shapes (Figure 1.1 and 1.2) like spheres, cylinders, cones etc. or segments of them. Basically the architects' creativity was confined to the Euclidean Geometry which until recently formed the basis of all architectural design. According to Schodek at al [SchoBe05] *"...., there is a general relationship among the development of tools for representing design information, the development of technologies for making complexly shaped members, and the specific kinds of architectural designs that were enabled by the relationships that existed at a specific point in time. It is not without interest to note that historically many highly evocative shapes envisioned by designers were simply never built (such as works by Hans Luckhardt, Herman Finsterlin, J.J. Leonidov, and others). Whilst attracting great interest, some designs remained simply utopian and were visualised with sketches and physical models only. Few tools were available to represent these design intentions precisely."*

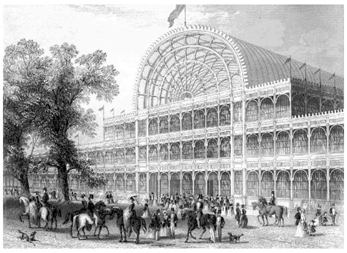


Figure 1.1: Barrel-shaped roof (segment of a cylinder) of the Cristal Palace by Paxton, London, 1851 [Image by Wikipedia]

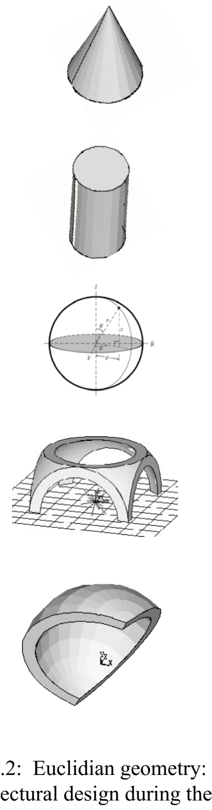


Figure 1.2: Euclidian geometry: basis of architectural design during the 'pre-digital' age [Image by ANSYS Modeling and Meshing Guide]

1.2 The appearance of freeform architecture before the 'Information Age'

The pursuit of going beyond a non-linear design vocabulary existed already in Baroque when the strict orders of the Renaissance era were broken up and concave and convex forms like domes, column groups, gables and architraves with rich ornamental jewellery defined the appearance of a new architectural style. In following epochs the use of curved building parts and components will be found more and more in the design language of architecture. However depending on their complexity and in

lack of experience and knowledge how to build such ambitious designs architects and master builders of these eras had to develop their own strategies to realise their visions - if they could be realised at all:
In this respect the organic and biomorphic forms of Art Nouveau especially in Antoni Gaudi's oeuvre during Modernism, the Catalan variation of Art Nouveau, are a great study object (Figure 1.3) .

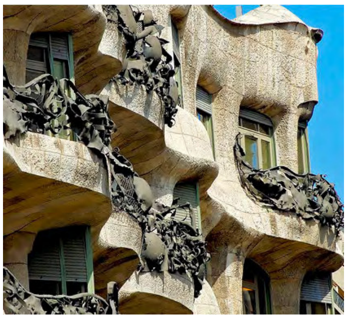


Figure 1.3: Casa Mila by Antoni Gaudi, Barcelona, Spain, 1910 [Image by politforen.net]

Case study: Works of Antoni Gaudi, Catalonia, Spain, 1852-1926
Gaudi was mostly inspired by the free form shapes which he found in nature and his biggest challenge was to find a language to translate them into architecture. He used the principle of ruled surfaces to realise his visions of a new architectural language: hyperbolic paraboloid, hyperboloid, helicoid and cone - just to name a few – formed the geometrical basis of his buildings.

Per definition a surface S is a ruled surface if there is a straight line l through every point of S which lies completely in S. A ruled surface can be generated by moving a straight line along a curve, subsequently the surface consists of a continuous family of straight lines l_x *which are called the rulings or generators of S.*

The fact that ruled surfaces are defined by a family of straight lines allowed realising their structural framework with a latticework of straight elements which was easily to be built even with the technical possibilities of the 19th century.

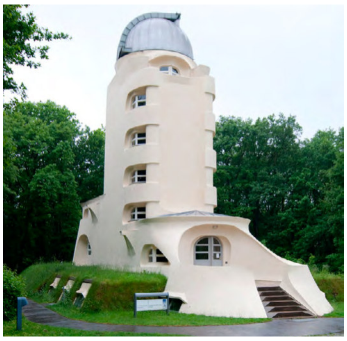


Figure 1.4: Einstein Tower by Erich Mendelsohn, Potsdam, Germany, 1921 [Image by durr-architect]

Case study: Einstein Tower by Erich Mendelsohn, Potsdam, Germany, 1920-1924
Another interesting example is Mendelsohn's Einstein Tower in 1924 (Figure 1.4) whose complex design language was ahead of the times. It had to be realised in a 'coarse' way as the manufacturing techniques to build the doubly-curved shapes weren't available by then: Initially Mendelsohn proposed his astrophysical observatory as a monolithic concrete structure. As this proofed to be impossible to realise the builders had to hark back to construction techniques which didn't do justice to the smooth geometry of the initial design. Eventually the curvy walls were built of brick covered in thick layers of plaster to achieve the envisaged surface continuity.

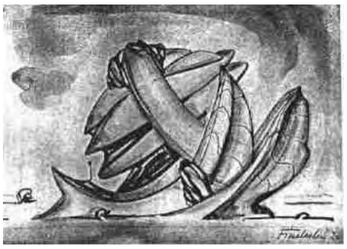


Figure 1.5: House of Glass by Hermann Finsterlin, 1924, never realised [Image by Wikipedia]

Case study: Works of Hermann Finsterlin, visionary architect, Germany, (1887-1973)
The visions of expressionist Hermann Finsterlin show a different aspect of free form architecture that is to say design ideas which were never realised because their complex geometry (Figure 1.5) quite easily wasn't buildable at the time the designer put the sophisticated shapes down on paper.

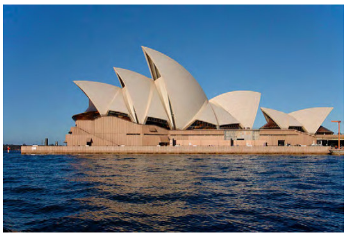


Figure 1.6: Sydney Opera House by Jørn Utzon, Australia, 1957-1973 [Image by Wikipedia]

Case study: Sydney Opera House by Jørn Utzon, Sydney, Australia, 1957-1973
A famous example for rationalising a free form surface to objects of the Euclidean Geometry are the roof shells of the Sydney Opera House by Jørn Utzon (Figure 1.6). The initial design consisted of free form roof

shapes resembling segments of an orange peel. As these proofed unrealisable the scheme "*was simplified through a seven year design development phase* with 12 different shell solutions *each successive form becoming more rationalised and structural elements more repetitive"* [SchoBe05]. Finally they were approximated to surface segments of spheres of different radii. This method allowed describing the underlying surface geometry of each of the different shells based on the geometrical knowledge of the 1950s and 1960s. The next step in the rationalisation process was to find a faceting pattern that resulted in planar repetitive components which could be manufactured with the 'state of the art' techniques of that time. After years of struggle this approach finally made it possible to realise Utzon's sophisticated design from 1953 in the early seventies.

1.3 CAD and CNC to change the design univers

With the introduction of the computer as a drawing and design tool (**C**omputer **A**ided **D**esign - CAD) and the introduction and subsequent development of highly efficient three-dimensional CAD programs the possibilities in terms of visualisable geometry changed dramatically. free form shapes – sometimes referred to as 'blobby' architecture - which until then where only known in the area of product design and here only in the realisation of comparatively small objects, quickly found their way into the design language of the architect.

"It was only within the last few years that the advances in computer-aided design (CAD) and computer-aided manufacturing (CAM) technologies have started to have an impact on building design and construction practices. They opened up new opportunities by allowing production and construction of very complex forms that were until recently very difficult and expensive to design, produce, and assemble using traditional construction technologies....... The consequences will be profound, as the historic relationship between architecture and its means of production is increasingly being challenged by new digitally driven processes of design, fabrication and construction." [SchoBe05].

1.4 Feasibility retards the shape visionaries

Although these new design tools opened up a whole new world in architecture at the same time problems emerged in terms of the construct ability of such ambitious geometrical ideas: Most 3D programs allow the experienced user to create very complex designs of free form shapes with a variety of available 3D tools whilst the sometimes extremely complex mathematical description of these shapes stays hidden for the user somewhere in the 'no man's land' of the computer.

Broadly speaking *"The mathematical development of differential geometry and related general theories of curved surface, provided the needed theoretical basis for understanding complex surfaces, but their formulations remained beyond the reach of design practitioners."* [SchoBe05]

However to realise a computer generated design and to actually build it the question arises how to translate complex digital geometries like a doubly curved surface into a set of information that can be handed over to and used by a manufacturer to produce the required building components.

Digital revolution - Timeline

1963 CAD
Development of the SKETCHPAD system at the Massachusetts Institute of Technology (MIT) by Ivan Sutherland can be seen as the birth of computer aided design

1963 CNC
The design of a trunk lid based on a 2D paper sketch was realised as a 3D clay prototype - for the first time a prototype was produce by means of digital data

Late 1970s PC
Advent of the personal computer

1977 CATIA
Start as an in-house development which was later developed to the first parametric design tool

1987 Introduction of additive manufacturing
C. Deckard developed the idea of producing an object by building it up layer by layer instead of cutting away at a larger chunk of material

1989 WWW
Invention of the World Wide Web

early 1990s
Internet is reaching a critical mass

1992
Introduction of parametric modelling software to the building and construction industry - adaption of CATIA as an architectural design tool ('Barcelon Fish' by F.O. Gehry)

1993 Rhino
Launch of Rhino, a NURBS-based 3D modelling tool

2003 GC
'Generative Components', parametric CAD software developed by Bentley Systems, is first introduced. Already by early 2005 the program is increasingly used in the London architects' scene

2007
Launch of Grashopper, a graphical algorithm editior which operates with Rhino's 3D modeling tools

As Branko Kolarevic states in 'Architecture in the digital age' [Ko03]: *"The information age, ………, is challenging not only how we design buildings, but also how we manufacture and construct them."*

Unfortunately 3D programs have no 'built-in button' to automatically generate the required data out of a 3D model. Therefore there is no standard solution and the way how the construction industry deals with this problem varies widely depending on the geometry itself, the size of the required components and also the materials that are envisaged for the realisation. Each building is a prototype and the challenges to actually build it to this day can only be met by developing a very specific solution both in terms of computational process and production method for each individual project.

There is a good reason why most of the leading architectural companies such as Foster + Partners, Zaha Hadid Architects, Gehry Partners etc. as well as cutting-edge structural engineering companies such as Arup or Buro Happold nowadays all have their own specialist units who deal with complex building geometry and the question how to translate them into reality. Their task is to develop a computational concept to rationalise a three-dimensional model and extract small building components from geometrically complex surfaces or solids. Depending on the concept deemed to be the most suitable the result might be the geometric data for two-dimensional, planar components or three-dimensional curvilinear components. In any case they comprise the numerical input required for digital fabrication.
This is an ongoing research process which will be advanced and refined in parallel to constantly emerging new computer technologies (both in terms of CAD programs and CNC technology), new building materials and improved processing methods.

Information Age - Definition

The period beginning around 1970 and noted for the abundant publication, consumption, and manipulation of information, especially by computers and computer networks.

The American Heritage® Dictionary of the English Language

A period beginning in the last quarter of the 20th century when information became easily accessible through publications and through the manipulation of information by computers and computer networks

WordNet 3.0, Farlex clipart collection. © 2003-2008 Princeton University, Farlex Inc.

A time when large amounts of information are widely available to many people, largely through computer technology (Sociology)

Collins English Dictionary – Complete and Unabridged

1.5 The development of freeform architecture during the 'Information Age'

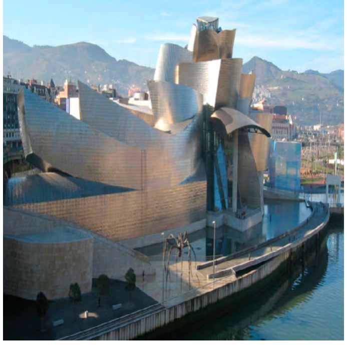


Figure 1.7: Guggenheim Museum by Frank O. Gehry, Bilbao, Spain, 1997
Highly complex envelope realised with the principle of 'paper surfaces' [Image by Wikipedia]

"Many of the forms he (Gehry) is developing now are only possible through the computer...... Bilbao is a perfect example. Prior to the development of the computer applications in the office, they would have been considered something to move away from."

Zaera, Alejandor, Frank Gehry 1991-5, Conversations with Frank O. Gehry, in El Croquis, no. 74-5, 1995

Case study: Guggenheim Museum by Gehry & Partners, Bilbao, Spain, 1997
The principle of 'Paper Surfaces' (also called sheet material surfaces) was developed by Gehry & Partners during a time in architecture when computers although having already found their way into the design and construction process were mainly used for two-dimensional construction drawings and if at all for simple three-dimensional representations of physical models. However, with Gehry & Partners highly ambitious design approach in terms of free form shapes they had to break new ground to be able to realise their projects.
Indeed Gehry & Partners can be seen as the pioneers in working with and actually realising complex free form shapes at a time, when the knowledge about complex geometry and relating computer tools was still very limited.
The principle of 'paper surfaces' or 'sheet material surfaces' was devel-

oped as a compromise between the highly constrained class of planar surfaces and the highly complex class of doubly-curved surface forms.
As the name 'sheet material surfaces' implies the basic idea was to design surfaces or rationalise an existing design to a surface that can be built using common sheet materials and relating economic production methods to realise the cladding.
They can be constructed by flat, flexible sheet materials which to a certain degree are smoothly bent, rolled and twisted to match the form of the physical model (Figure 1.8). This means paper surfaces are based on the mathematical principle of developable surfaces with the difference that the deformation of the cladding panels is accepted within a pre-defined tolerance (subject to the chosen material). One important aspect of paper surfaces is the fact that *"the forms assumed by paper and other sheet materials in scale physical models have important counterparts in full scale construction. These materials may be readily formed by manual methods into curved shapes in space, so long as these forms do not require stretch forming of the material that would produce plastic deformation in the plane of the surface material".* [Sche02]
The idea is to build scale physical models with materials that already in the small scale of model making resemble similar material properties than the sheet material later used in the actual construction process (Figure 1.9). With this approach the model making process acts not only as a design and visualisation tool but also as a tool to rationalise those areas whose curvature is too extreme for being realised with sheet material.
Attempting to realise any doubly-curved shape with paper surfaces might result in wrinkling, creasing or ripping of the sheet of paper which indicates that the designed shape has to be slightly adapted to avoid discontinuities in the smooth surface.

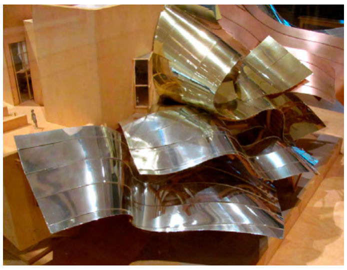


Figure 1.8: Physical presentation model made of sheet materials [Image by rocor via flickr]

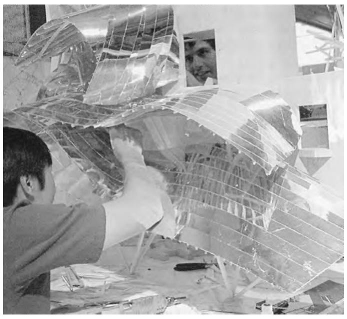


Figure 1.9: Model of 'Paper Surface' Marques de Riscal Winery, Spain, 1998 [Image by Digital Gehry (Li01) page 19]

Case study: The Sage by Foster + Partners, Gateshead, UK, 2003
The complex geometry of the doubly-curved roof of Foster + Partners' music centre in Gateshead was rationalised using the radial geometry of a torus (definition 3.5.3). The surface is composed of different tori patches which smoothly transition from one to another (Figure 1.10). Each patch again is subdivided into individual bands which are cladded with planar quadrilateral panels (Figure 1.11). To guarantee the planarity of each individual face the design of the roof shape was consequently subordinated to the form finding principles for a rationalised surface.

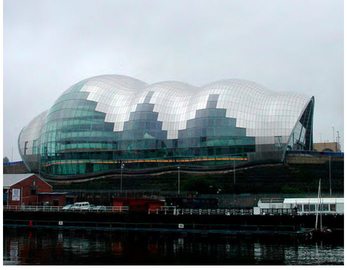


Figure 1.10: The Sage, Gateshead, UK, 2004 by Foster + Partners [Image by Wikipedia]

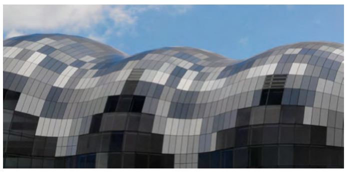


Figure 1.11: Pre-rationalised envelope: The shape of the building envelope was adapted to allow the faceting with planar quadrilateral elements
[Image by Ian Jones via flickr]

1.6 Visionary architecture versus budget

These case studies show that the effort involved in the realisation of free form architecture is immense both in terms of engineering and design.
It seems that complex designs still can only be realised if they have been designed by one of the major architectural companies which have the required knowledge in the field of discrete free form structures and differential geometry as well as solvent clients who can afford the difficulties inherent to complex shapes.
Hence great designs might be doomed to failure if their realisation can't

Frank O. Gehry about the dependency of construction cost and complex building components in 1995:
"Flat pieces cost one dollar, single curvature pieces cost two dollars; double curvature pieces cost ten dollars."

"Value engineering
is essentially a process that uses function cost analysis to reduce cost. Usually done after the design work is complete, it often results in large-scale cuts in program, quality, or complexity."

Digital Gehry: material resistance/digital construction, Bruce Lindsey,, Birkhaeuser 2001 [Li01]

be managed by the people involved in the project.
The aim is to develop tools and methods which allow the designer to live their dreams irrespective of their knowledge in the field of differential geometry and computation.

In recent years the construction industry already found some answers to the task of manufacturing building components with complex geometry. But although the sophisticated manufacturing techniques of the twenty-first century like solid free form fabrication (aka rapid prototyping) allow the realisation of highly complex designs one of the most important questions remain their feasibility in terms of budget, program and collaboration between the different members of the design and construction process. The most unsatisfying situation will always be the redesign of a great idea because the design team wasn't able to deliver a solution that is within budget and still offers an aesthetically pleasing result.

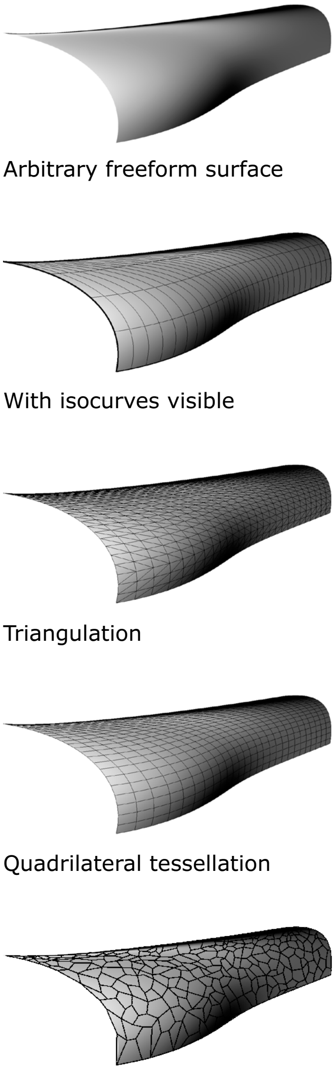
"Geometry alone is not able to provide solutions for the entire process, but a solid geometric understanding is an important step toward a successful realisation of such a project." [Li01]

Generally speaking the difficulty with freeform architecture is the 'breaking down' of a computer generated model into building components which can be manufactured in terms of

- panel size
- complexity in shape
- number of different components and their allocation during assembly

Surface approximation
A particular challenge with free form shapes is the faceting of the envelope which usually represents the biggest part of a building or in more abstract words the subdivision of the building surface. Whilst a computer generated surface is a smooth shape its translation into a geometrical object with relevant information for the construction process is the discretisation of the surface or in other words the approximation of the surface by use of a polygonal mesh.

The underlying geometrical principle of all surface subdivisions is the polygonal tessellation (Figure 1.12) which subdivides a surface using the geometrical description of a polygon (e.g. triangle, quadrangle, pentagon, hexagon, heptagon etc.). This process results in a collection of vertices, edges and faces which define the shape of a polyhedral object. Tessellations of the Euclidean plane always result in planar polygonal subdivisions but it is also possible to tessellate the n-dimensional space which will result in n-dimensional polytopes. These can be either planar or curvilinear depending on the applied method. In theory it seems that the designer has great freedom how to transfer his design idea into buildable components. In reality however this process is limited by considerations like



Arbitrary freeform surface

With isocurves visible

Triangulation

Quadrilateral tessellation

Polygonal tessellation

Figure 1.12: Example of different types of a computer-generated subdivison by means of polygonal tessellation

- How close does the subdivision reflect the initial surface geometry (degree of freedom for shape optimisation)
- Does the generated pattern reflect the design intent
- Difficulties regarding the assembly of a structure which might consist of a myriad of unique components
- Is the desired subdivision on budget or is there a need for further rationalisation (planar versus curvilinear components, non-repetitive building components versus a certain number of identical or at least similar components etc.)

These are all factors with a major influence on the feasibility of a design especially in terms of the budget:

1.7 Number of identical components of surface subdivision

There are systems in place to help with the assembly if a big number of individual building components needs to be allocated precisely as done for Gehry's ‚Experience Music Centre, in Seattle in 2000 or for Foster + Partners' roof of the ‚Great Court Yard, of the British Museum in 2000. Building costs however could have been reduced significantly if the number of individual components had been reduced as done for the roof shells of the Sidney Opera House in 1973.

1.8 Planar versus curvilinear components

With highly sophisticated CNC manufacturing techniques it is not a problem anymore to manufacture doubly-curved cladding elements. However doubly-curved components can only be produced with a limited number of materials hence the choice of cladding material will be restricted. The other aspect is the production process which is far more complex and thus more expensive than for planar components. Consequently the focus of this thesis is directed on a subdivision principle which results in planar components. The most obvious way would be to apply one of the available triangulation principles as these will always result in planar components. However quite often the distinct geometrical appearance of triangulated surfaces doesn't meet the initial design intent of the architect and a 'calmer' geometry is desired which leads to the requirement for a quadrilateral subdivision (Figure 1.13).

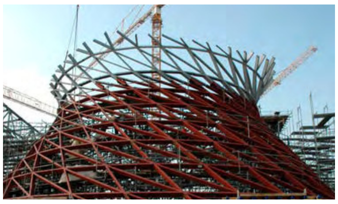


Figure 1.13: Triangulation of the glass roof of the BMW World, Munich, Coop Himmelb(l)au, 2007 [Images by Maira Onofri/ Cepolina]

The planar quadrilateral faceting of a surface is one of the most desirable subdivision principles both from an architectural and an economic point of view. Architecturally it provides a calmer appearance as the edges of the mesh seem to be in some kind of linear order. Economically planarity is less expensive than curved faces.

"A*n object consisting of planar components can be unfolded in the plane* *without distortion and thus fabricated with simple two-dimensional cutting techniques."* [SchoBe05]

Research gap
State of research regarding PQ meshes based on principal curvature lines (Axel Kilian et al.)

Although a number of generation principles for PQ meshes are already known these will all lead to specific types of surfaces.
They all require either pre-rationalisation meaning the form finding process has to be subordinated to the generation principle or post-rationalising for which the initial design idea can only be preserved to a certain degree as the surface has to undergo optimisation processes to make it suitable for quad tessellation. Their mathematical properties will be explained in detail in chapter 03.

To date there is no general method which allows the subdivision of an arbitrary doubly curved surface into a mesh of planar quads.
As stated by Axel Kilian et al.: "*The design of a PQ mesh that satisfies all of the high requirements of aesthetics and that is sufficiently close to a provided input surface is an unsolved research problem. We will be able to compute a solution if the network of principal curvature lines can be used as a basis of a mesh that is then improved via optimisation. However, singularities as well as large variations in cell sizes caused by the flow of principal curvature lines may make this approach unfeasible."*

Purpose of thesis
generation principle to be developed to transform an arbitrary doubly curved surface of any complexity into a mesh of planar quadrilateral faces

The purpose of this thesis is therefore to find a universal generation principle which allows the translation of any arbitrary doubly curved surface into a mesh of planar quadrilateral faces without exerting influence on the design and its formal vocabulary. Ideally the method of solution shall be applicable to every surface irrespective of its complexity and the geometry of the input surface shall be preserved as far as possible.

Fundamentals of PQ mesh generation principles

First the focus directed at the understanding of the mathematical fundamentals of PQ meshes and those surfaces which have been successfully discretised into a PQ mesh. This knowledge is imperative for the further research regarding a universal generation principle.

Surface versus mesh

In mathematics a surface is defined as a continuous smooth sequence of an indefinite number of points. A mesh in contrary is the discretisation of a surface as only the vertices and edges of the mesh are located exactly on the input surface. Areas circumscribed by four adjacent edges remain mathematically undefined (Figure 1.14). Is the mesh defined by straight edges it is the vertices only that form part of the input surface. For a satisfying surface representation an approximation process is required which generates a face density representing the input surface as precisely as possible and at the same time addressing the requirements of the manufacturing industry.

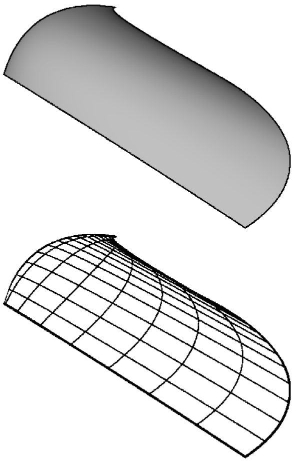


Figure 1.14: Arbitrary doubly curved surface (above) and its translation into a mesh - the discretised version of the same surface (below)

Conjugate network of curves and their importance for PQ meshes

It is mathematically proven that the edges of a PQ mesh always discretise a conjugate network of curves (referred to as CNCs hereafter)

and that theoretically there exists an indefinite number of conjugate networks of curves for every single surface (Chapter 3.4) [BoSu08, PoAs07]. Thus an initial solution to the problem could be to isolate one of these networks from the input surface to use it as basis for the segmentation process. However although the existence of countless CNCs per surface is known so far their mathematical generation principles are widely unexplored.
There are a limeted number of generation principles for CNCs e.g. translation surfaces (Figure 1.15) which are well-researched and documented and although the discretisation of these surfaces delivers a PQ mesh they don't provide a general solution to the problem of PQ faceting as their use would imply the pre-rationalisation of the input surface.
This however will restrict the architects' freedom of design significantly as only doubly curved surfaces based on the generation principles of one of these specific surface types would guarantee a PQ mesh. Hence a more general approach needs to be found.

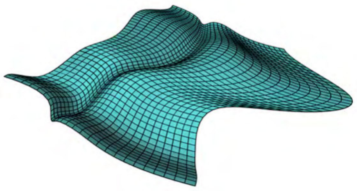


Figure 1.16: Conjugate network of curves on a translation surface

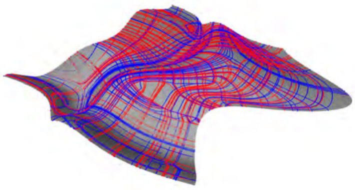


Figure 1.15: Network of principal curvature lines

Network of principal curvature lines

One type of conjugate network of curves is of particular interest though: for each surface there is only one CNC which features the property of orthogonality of the curves in their intersections. This specific network is called the network of principal curvature lines (referred to as LPCs hereafter). As one of the important properties of mesh faces with respect to both the manufacturing process and their appearance (architectural and economical aspects) is the perpendicularity of their edges the LPCs pose an ideal basis for the generation of the output mesh.

- Every surface possesses only one network of principal curvature lines
- It is a generation principle which results in quadrilateral faces which are close to being planar
- the network of LPCs features almost right angles which provides an additional advantage in terms of manufacturing requirements
- Their generation principle is well-research and documented

The LPCs however will provide a mesh suitable as start geometry only for those areas of a surface which feature regular points. In the vicinity of singularities or singular points (Chapter 2.5.3) this approach proofs unfeasible as the flow of the LPCs deflect to unusual configurations and cell sizes which are unsuitable as start geometry (Figures 2.22-2.24).

Crude mesh

An alternative way must be found to generate a continuous and visually appealing PQ mesh which follows the lines of principal curvature in 'regular' surface regions but may diverge in areas where the LPC

flow doesn't provide a suitable start geometry.
By picking the vertices manually - close to the lines of principal curvature but with the freedom to choose their location just close-by instead of exactly on the lines - a generation principle is chosen which takes advantage of the knowledge regarding the LPCs and the geometrical particularities of its relating network but still allows to deal with the difficulties posed by special surface areas. These faces can be sculpted manually and placed in such a way that their edges line up with the LPCs of the target surface. Although the lines of principal curvature are not used directly to generate the crude mesh their geometrical configuration indicates how the PQ approximation mesh is best to be set up.

Mesh subdivision
A subdivision algorithm is applied in order to refine the crude mesh. At the same time the newly generated vertices are averaged to guarantee a smooth and continuous but finer mesh. The subdivision algorithm can be applied iteratively until the desired density of the mesh is achieved. After each iteration cycle the newly generated vertices are 'pulled back' onto the target surface to allow as little deviation from the input surface as possible.

Optimisation
It is obvious that this approach results in faces which are not exactly planar and therefore as a final step an optimisation process shall be developed to change mesh areas with non-planar faces and to force them into planar ones. This process is based on dynamic relaxation principles.

1.9 Main focus and limitations

In the broadest sense the thesis deals with the translation of an arbitrary free form surface into a mesh consisting of vertices, edges and faces. This operation is pre-conditioned by two aspects:

- planarity of the faces
- quadrilateral shape

An additional aspect is the adherence to the original design intent.
There are many different ways how to panelise free form architecture which are still unexplored or whose generation principles can be improved. The reason why this thesis is focusing on a solution for PQ meshes is the obvious advantages in comparison to other panel shapes namely node simplicity, a higher degree of transparency with respect to the reduced number of members per verities and reduced manufacturing costs with respect to planarity of the faces.

Planarity in an absolute versus relative sense

However, regarding planarity the constraint is made that it's not about absolute planarity in a mathematical sense, but planarity in terms of manufacturing requirements. This means the generated faces are close to planar

within a pre-defined tolerance. Depending on the stiffness of the cladding material the tolerance can be chosen more generously for bendy materials like sheet metal respectively less generously for stiff materials like glass. Planarity in a mathematical sense isn't necessary as even for glass panels a certain deformation is acceptable and will result in reduced building cost in comparison to planar facets in an absolute sense.
The quadrilateral shape of the faces is a precondition for reasons of higher structural transparency in comparison to triangulation.
From an economic point of view it could be desirable to go a step further and aim for rectangles instead of quads. Hence a further development of the tool could be the perpendicularity of planar faces.

Generation of rectangles as a step further

Another aspect relevant for the economic viability of a project could be to maximise the amount of repetitive elements for the panelisation. The research in this thesis concentrates on PQ meshes irrespective of the number of different panels as the foremost interest is to provide a general solution for this to date unsolved problem. Still for reasons of production and assembly a high amount of identical elements could be a huge benefit.

Amount of repetitive elements

For the sake of completeness it shall be mentioned that Dr. Yang Liu et al. already developed a planarisation algorithm in 2006 which can be applied to an arbitrary mesh of non-planar faces. However the generation of a suitable starting mesh was left un-discussed and will therefore be address in this thesis.

Dr. Yang Liu et al. planarisation algorithm in 2006 [LiPo06]

1.10 Outline

The analysis and documentation of PQ mesh properties in chapter 2 deliver the mathematical fundamentals which are required to approach the problem of surface segmentation. Elements of planar quad meshes as well as rules and actions relevant for their understanding are explained and illustrated.

Chapter 2
'PQ mesh properties'
The intrinsic properties of PQ meshes are analysed and compiled

In Chapter 3 the mathematical generation principles for surfaces which in the past have already been successfully discretised into a PQ mesh are analysed and discussed in detail: what has been studied so far, how flexible are the well-known methods in terms of surface representation and how can the available knowledge been used for the solution finding process for the specified problem. Basically the 'State of the Art' of surface discretisation in differential geometry is documented.
Notably the special meaning of 'conjugate network of curves' and the 'network of principal curvature lines' for planar quadrilateral faceting are explained as these two principles play a vital part in the solution finding process.

Chapter 3
'PQ mesh surfaces'
Compilation and documentation of PQ mesh surfaces whose mathematical generation principles are known

In Chapter 4 a generation principle is developed which allows the faceting of an arbitrary doubly curved surface with planar quadrilateral faces. The methodology comprises of several steps to get from a crude input

Chapter 4
'PQ mesh approximation'
A generation principle is developed which allows the faceting of an arbitrary doubly-curved surface with planar quadrilateral faces

mesh to an approximation mesh which features the desired planarity in a density suitable for the construction process. Intermediate steps like the refinement of the mesh by use of a subdivision algorithm and the dynamic relaxation process to achieve planarity are explained.

Chapter 5
'Case studies'
The developed tool is applied to surfaces of different levels of complexity and the results are analysed and interpreted

A number of case studies in chapter 5 shall document how successfully the developed tool might perform surface segmentations. The program is applied to and tested with a number of arbitrary surfaces of different levels of complexity to provide a general overview of its potential. The surfaces used for this exercise are especially created for the test run in order to compare similar surface topologies with different degrees of curvature.

However, arbitrarily generated surfaces -although ideal to show certain aspects of a problem- are artificial creations and therefore might differ from the built reality in scale, shape and complexity of curvature. For this reason an additional test run is performed for which the tool shall be applied to surfaces of actually built projects.
For each of the test surfaces the course of the segmentation process is documented with screen shots in a step by step manner to allow a direct comparison of the different surface types.

The results are analysed and interpreted regarding aspects like
- degree of planarity
- mesh density
- adherence to original design intent
- regularity of face pattern

Chapter 6
'Summary and preview'
Scope of research and its achievements, significance for future developments

Chapter 6 shall give an overview of what has been achieved and what could be the future prospects: a preview for possible further developments of the tool and its significance for the construction and manufacturing industry in general as well as for the field of computation and programming in particular.

Chapter 2 Planar quad (PQ) mesh properties

2 Planar Quad (PQ) mesh properties

2.1 Introduction

As the research aim of this thesis is the generation of a planar quadrilateral mesh (called PQ mesh hereafter) from an arbitrary freeform surface In this chapter we will discuss the intrinsic geometric properties of PQ. meshes. For this reason we will break down the mathematical object of a mesh into the components it consists of and start our examination with a single mesh face as the smallest possible unit.
The two relevant properties are the number of vertices which for our particular case is defined as four and the plan
arity of the face. In a mathematical sense the definition of a quadrilateral is trivial and will be disregarded. However planarity of an object consisting of four corner points is essential for the solution finding and therefore we will start with the investigation of the various criteria which guarantee the planarity of a single face.

To get from a single PQ face to the complex structure of a PQ mesh the next step is the PQ strip which as per definition consists of a series of PQ faces linked together in such a way that they form a single row. Mesh strips exist in three generic configurations. As the research aim of this thesis is the generation of a planar cylindrical strips, conical strips and tangential strips of which represent a discrete version of a developable surface [LiPO06].

Similarities and differences of the three PQ strip types will be explained as well as their reference to developable surfaces.

If one links an arbitrary number of PQ strips with each other this will finally result in a PQ mesh. It is mathematically proven that the edges of a PQ mesh always represent the discrete version of a conjugate network of curves with the particularity that all curves of a conjugate network on a surface are in conjugate directions to each other. Hence the properties of conjugate networks of curves and the mathematical principles which in general define curve directions as conjugate are explored.

The lines of principal curvature on a surface are an example for such a conjugate network of curves. However this special network type features

the property that only one network of principal curvature lines exists for any given surface and is therefore explored in more detail.
In addition two more mesh types are investigated and explained because of their particular meaning for PQ meshes: If all faces of a PQ mesh possess a circumcircle such a mesh is called a circular mesh. If faces which have one vertex in common are all tangent to a right circular cone (with the vertex forming the peak of the cone) such a mesh is called a conical PQ mesh. It is important to know that both circular and conical meshes discretise the principal curvature lines of a surface.
Finally mesh transformations and the possibilities and implications offered by them shall be briefly explored in the last section of this chapter.

2.2 PQ Face Planarity Criteria

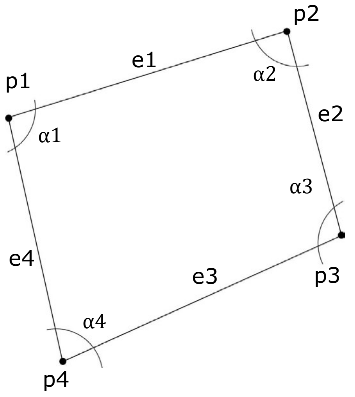


Figure 2.1: Quad Face

Let a quad face (Figure 2.1) be defined by its four corner points p1, p2, p3, p4 edge vectors e1, e2, e3, e4 and corner angels α1, α2, α3, α4.

The planarity of the quadrilateral or quadrangle face can be defined in various ways. The basic condition may be that a quad face is called planar if all corner vertices lie in a plane. If the following conditions are satisfied a mesh face is planar.

2.2.1 Angle sum

The condition of planarity is satisfied if the four corner angles α1 to α4 which are included by the edges e1 to e4 (Figure 2.2) sum up to 2π (360°):

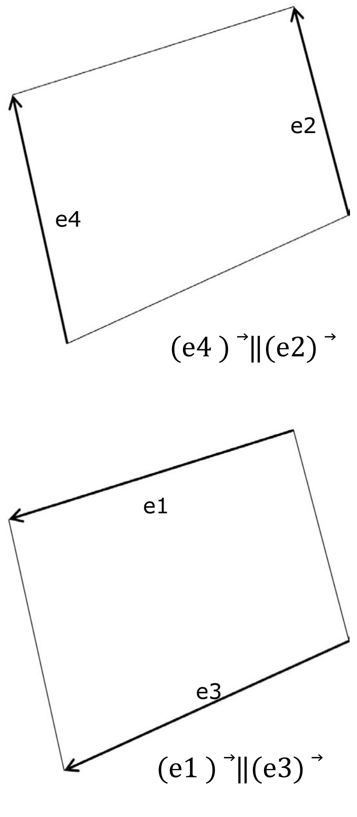
$$\Sigma\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 = 2\pi$$ Equation 2.1

2.2.2 Two Parallel Edges

If we consider the four edge vectors e1 to e4, the opposed edge vectors require to be parallel in space in order to form a PQ face (Figure 2.2). This condition is reciprocal hence we need to check only one pair of edge vectors for parallelism. The conditions of two vectors to be parallel in space is satisfied if their cross product vanishes:

$$e1 \parallel e3 \text{ if } e1 \times e3 = 0$$ Equation 2.2(a)

$$e2 \parallel e4 \text{ if } e2 \times e4 = 0$$ Equation 2.2(b)


e2
e4
$(e4)^{\rightarrow} \| (e2)^{\rightarrow}$
e1
e3
$(e1)^{\rightarrow} \| (e3)^{\rightarrow}$
Figure 2.2: Parallel Edge Vectors

2.2.3 Three Linear Dependant Edge Vectors

If three vectors in space lie in a common plane they are called to be linear dependant. This condition is satisfied if their determinant vanishes. The condition is reciprocal meaning only one of equations 2.3(a) to 2.3(d) needs to be satisfied to prove the planarity of a face.

$$\det(e1, e2, e3) = 0 \qquad \text{Equation 2.3(a)}$$
$$\det(e1, e2, e4) = 0 \qquad \text{Equation 2.3(b)}$$
$$\det(e1, e3, e4) = 0 \qquad \text{Equation 2.3(c)}$$
$$\det(e2, e3, e4) = 0 \qquad \text{Equation 2.3(d)}$$

2.2.4 Vanishing distance of a forth point to a three Point Plane

Three points in space define a plane. Equation 2.4 defines the distance of a point to the defined plane. The reference plane is defined by any three corner vertices of a face. If the fourth vertex of this face has vanishing distance from the reference plane the face is a PQ face. If we check this condition the non-zero event will also give us a value for the non-planarity.

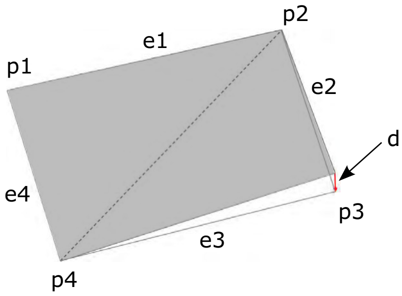


Figure 2.3: Out of Plane Distance

$$d = |v \bullet n| \qquad \text{Equation 2.4}$$

e.g. plane (p1,p2,p4), point (p3)

$v = (p3x - p1x, p3y - p1y, p3z - p1z)$

$$n = \frac{e1 \times e4}{|e1 \times e4|}$$

2.2.5 Four vertices on a circumcircle

A quad face is called to be circular or discrete orthogonal if its four corner vertices all lie on a common circle. The following condition must be fulfilled for a quad face to be circular [WaWa07]:

$$\phi1 + \phi3 - \pi = 0 \text{ and } \phi2 + \phi4 - \pi = 0 \qquad \text{Equation 2.5}$$

$\phi1$ to $\phi4$...interior corner angels of a face

If only one of the five above described conditions is satisfied it is proven that a quadrilateral face is planar.

With the mathematical fundamentals of planarity of a quadrangle defined we will now investigate the unit next in size of a mesh structure.
If a series of planar faces is linked together in a single row the arrangement will form a planar quadrilateral strip. As any PQ mesh can be broken down into a number of different PQ strips their fundamentals are essential and will be investigated in the following section.

2.3 PQ Strip Properties

The arrangement of a series of planar quads in a single row is the discrete version of a developable surface strip [LiPo06]. Therefore the study of PQ meshes and the elements such a mesh consists of is directly linked with the study of developable surfaces and their particularities. In 3.2.2 we will have a closer look at developable surfaces with respect to possible discretisation principles as well as the relationship between developable and ruled surfaces. However the three established types of developable surfaces shall be briefly introduced here as their respective discrete versions are the three main types of PQ strips which PQ meshes of any kind are composed of.
A developable surface S is defined as the envelope of a one-parameter family of planes because each of these planes touches the surface along a straight line. These straight lines are called the rulings (or generators) of S. Depending on the configurations of the rulings three main types of developable surfaces can be isolated:

cylinder surface:	all rulings of S are parallel
conical surface:	all rulings of S pass through a fixed point a namely the apex of the cone
tangent surface:	all rulings of S are tangents of a space curve c

If we now examine the properties of a PQ strip which as mentioned above is the discrete version of a continuous developable surface it becomes obvious that the different PQ strip types can be distinguished in the same way as their respective developable surface types namely by the configuration of their rulings.
In reference to the main types of developable surfaces explained above there are three basic types of PQ strips:

- cylindrical PQ strip
- conical PQ strip
- tangential PQ strip

The only difference to their continuous equivalent is that the space between adjacent rulings of a PQ strip is defined by a plane whereas for developable surfaces single curvature is involved. Consequently this means that the property of single curvature of a developable surface is transformed into a number of planar quads arranged in a single row and joined together in their parallel edges which form the rulings of the PQ strip. Any two consecutive edges of the quads in a PQ strip are parallel in space and therefore co-planar (Chapter 2.2.2). As PQ strips consist of a sequence of planar quads they can be unfolded into the plane without distortion like stretching or compressing. The edges of the PQ strip where the planar quads are joined together represent the rulings of the respective developable surface.
The mathematical definition of the three main types of PQ strips are summarised in the following section.

2.3.1 Cylindrical PQ Strip

The cylindrical PQ strip is defined by a family of parallel lines in space (Figure 2.4). These rulings may be generated by the parallel extrusion of a discrete spatial or planar curve in any direction. The parametric definition of a cylindrical surface is given in Equation 2.6 below:

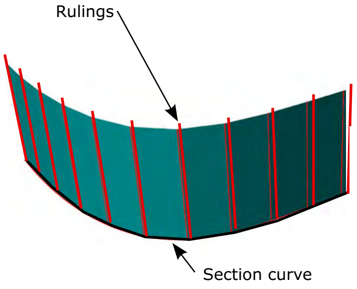


Figure 2.4: Congruent cylindrical PQ strip

$$S[u]=p(u)+z(u) \qquad \text{Equation 2.6}$$

$p(u)$... section curve
$z(u)$... extrusion vector

Two corresponding edges of the PQ strip faces are always part of the rulings which define the PQ strip. The scond pair can be freely chosen as two polygonal lines with their vertices of the rulings. Here can distinguish between *congruent* and *non-congruent* quadrangles:
If the second corresponding set of edges is parallel to the extrusion profile curve the PQ strip faces between too adjacent rulings are called to be congruent. The faces are identical in shape and in size. In absence of the previous condition the quadrangles called to the non-congruent (Figure 2.5).

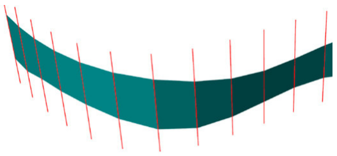


Figure 2.5: Non-congruent cylindrical PQ strip

2.3.2 Conical PQ Strip

If we connect the vertices of a planar or spatial polygonal curve to a single point we obtain a discrete conical surface (Figure 2.6).The conical PQ strip is part of the concical surface. The concical surface is defined in Equation 2.7 below:

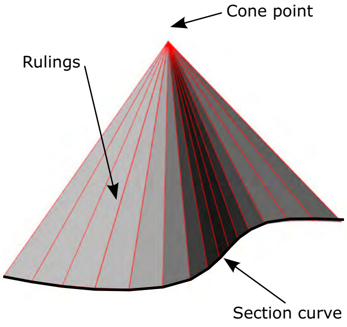


Figure 2.6: Discrete concical surface

$$S[u,v]=p(u)+vz(u) \qquad \text{Equation 2.7}$$

with $z=c_0-\frac{p}{|c_0-p|}$

c_0 ... cone point

The edge defining lines may alternatively generated as the central extrusion of a discrete curve towards a common point.

One corresponding set of PQ face edges lie on the straight generators of the concical surface. Here can distinguish between *similar* and *non-similar* quadrangles:

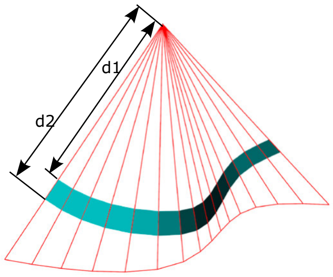


Figure 2.7: Similar face conical PQ strip

If the vertices of the faces between two adjacent generators have a constant distant to the common point the quadrangles are called to be *similar* (Figure 2.7). The faces are identical in shape but not in size. They can be translated and scaled into each other. A similar PQ strip will form a polygonal arc when unrolled. A varying distance will derive in a *non-similar* layout of the PQ strip.

2.3.3 Tangential PQ Strip

The surface defined by a series of tangent curves along a spatial curve is called to be a tangential surface (Figure 2.8). The tangential PQ strip is formed by a part of the he discrete version of the tangent surface. The parametric definition of a tangential surface is given in Equation 2.8:

$$S[u,v]=c(u)+vc'(u) \qquad \text{Equation 2.8}$$

$c(u)$... spatial curve
$c'(u)$... first diveration of spatial curve

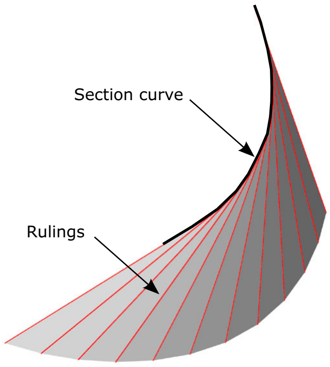


Figure 2.8: Discrete tangential surface

The consecutive rulings are co-planar and their intersections result in a series of points which reciprocally approximate the tangential generating curve. If the distance between the consecutive rulings is chosen to be infinitesimal small the subdivision points will match the generating curve (Figure 2.9).

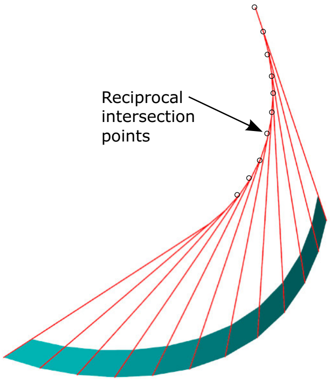


Figure 2.9: Tangential PQ strip

Two opposing edges of each face are always part of two consecutive rulings which define the strip. The condition of *similar* and *non-similar* quadrangles applies in the same way than described for the conical PQ strip.

It is mathematically proven that cylindrical, conical and tangential surfaces are the only existing developable surfaces in 3-dimensional space [PoWa01]. Hence every PQ mesh must be a composition of PQ strips of one or more types of these three configurations.

With the structure of a PQ mesh and its components mathematically defined we can now look at it from a different angle: In the previous section we have stripped the mesh down into its basic elements (mesh face - mesh strip - entire mesh). In the following, we want to examine the mesh structure with respect to its linear components and the resulting curve configurations.

If we consider a PQ mesh with the focal on the configuration of the polygonal curves formed by the consecutive edges it reflects a network with unique properties, namely the discrete version of a conjugate network of curves of the respective surface. The meaning of this special curve network and its relevance regarding a general solution for PQ meshes for an arbitrary freeform surface will be explained in the next section.

2.4 Conjugate network of curves

In chapter 3 we will have a closer look into the different surface types which -when discretised in a certain way- automatically result in a PQ mesh. These are very specific with respect to their appearance and hence unsuitable for a general approach.
The conjugate network of curves of a surface however could potentially offer a first step in the solution finding process as it features two important properties:

- Any surface in 3-dimensional space irrespective of its complexity can be parameterised using conjugate curves.
- If discretised a conjugate network of curves will always result in a PQ mesh, thus featuring faces which are planar quads.

Or in reverse PQ meshes posses the special property that their edges always represent the discrete version of a conjugate network of curves. Such a net is also called Q-net.

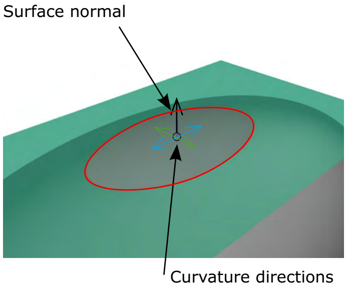


Figure 2.10: Elliptical Indicatrix

Based on this knowledge, in theory one could approach the problem by generating a conjugate network of curves for a surface and then discretise this Q-net to obtain a PQ mesh. However in this context it is important to know that there is not only one but an infinite number of conjugate networks of curves for any given surface [BoSu08, PoAs07] meaning they don't provide a unique solution. Besides, only a very limited number of generation principles are known to date which allow the generation of a Q-net on an arbitrary surface.

As the name implies all curves of a conjugate network are in conjugate direction to each other at any vertex of the net. Hence the question is how to determine such conjugate directions for an arbitrary surface point in order to generate a Q-net. This can be done by means of the Indicatrix of Dupin or the Eulers theorem [Kr59].

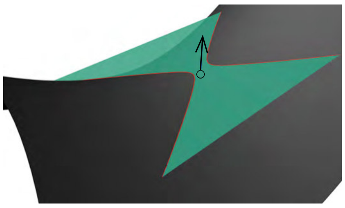


Figure 2.11: Hyperbolic Indicatrix above surface

2.4.1 Indicatrix of Dupin

In the field of differential geometry of curved surfaces the mathematical term ‚Indicatrix' is defined as a planar conic section (resulting in either parabola, ellipse, circle or hyperbola), which describes the local behaviour of curvature of a surface S in a defined point P. It was defined by Pierre Charles François Dupin (1784 - 1873) and is therefore also known as Indicatrix of Dupin
If we consider a sufficiently small environment around a point P on a surface S one can approximate the surface arbitrarily closely by means of a quadric (a surface of 2nd order). If the tangent plane T in point P is moved infinitesimally in the direction of the surface normal vector n and its opposite direction the plane will intersect with the surface and result in one of the three geometrical configurations (elliptic, hyperbolic, parabolic) described in the following chapters.

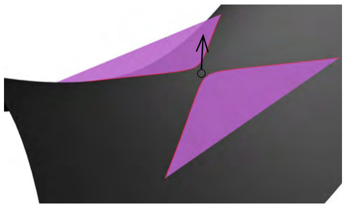


Figure 2.12: Hyperbolic Indicatrix below surface

2.4.2 Elliptic point ($K > 0$, $\kappa_{max,}$ $\kappa_{min} > 0$):

The intersection of the surface with a plane parallel to the tangent plane generates an ellipse (Figure 2.10). The directions of the maximum and minimum radii of this ellipse are corresponding to the maximum (κmax) and minimum (κmin) principle curvature directions. The ellipse can be described as follows:

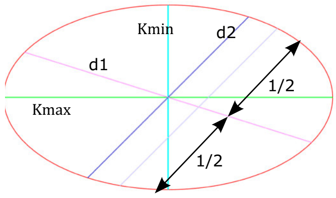


Figure 2.13: Elliptic conjugate diameters

$$\kappa_{max}\, x^2 + \kappa_{min} y^2 = 1,\ R_{max} = \sqrt{}\kappa_{max},\ R_{min} = \sqrt{}\kappa_{min} \qquad \text{Equation 2.9}$$

$\sqrt{}\ R_{max},\ \sqrt{}\ R_{min}$ ... principle semi axis

On a sphere all points are elliptic but with identical values for the maximum and minimum radii of an ellipse and respectively for the principal curvature directions. Hence the Indicatrix for a sphere is always a circle.

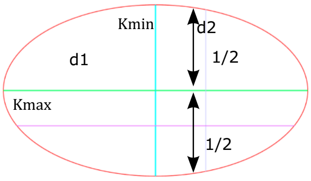


Figure 2.14: Elliptic conjugate diameters

2.4.3 Hyperbolic point ($K < 0$, $\kappa_{max} > 0$, $\kappa_{min} < 0$):

The intersection of the surface with two planes parallel to the tangent plane generates a set of conjugate hyperbolas (Figures 2.11, 2.12). Their main axes are corresponding to the maximum and minimum principle curvature directions. The two asymptotes of the hyperbolas representing the asymptotic directions which comprises of vanishing curvature ($\kappa = 0$). The conjugate hyperbolas are defined as follows:

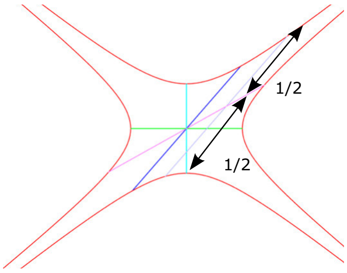


Figure 2.15: Hyperbolic conjugate diameters

$$\kappa_{max}\, x^2 + \kappa_{min}\, y^2 = \pm 1,\ R_{max} = \sqrt{}\kappa_{max},\ R_{min} = \sqrt{}\kappa_{min} \qquad \text{Equation 2.10}$$

$|\sqrt{}\ R_{max}|,|\sqrt{}\ R_{min}|$... principle semi axis

2.4.4 Parabolic point ($K = 0$, $\kappa max > 0$, $\kappa min = 0$):

The intersection of the surface with the plane close to the tangent plane and parallel to it generates two parallel lines. The parallel lines are in direction of the single asymptotic line. With the help of Dupin’s Indicatrix we can now investigate the conjugate directions at any point on the surface.

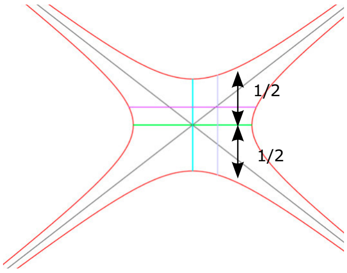


Figure 2.16: Hyperbolic conjugate diameters

2.4.5 Conjugate directions

As explained above the two directions at a point P on a surface S which coincide with the conjugate diameters of *Dupin’s Indicatrix* are called the two conjugate directions.
The two diameters d1, d2 within the Indicatrix are called conjugate if d1 bisects the chords parallel to d2 and vice versa (Figures 2.13, 2.15).

Hence this condition is reciprocal.

It is important to mention that for each point P on a surface S there is not only one pair of conjugate directions but an infinite number of them. This becomes obvious if we have a look at an arbitrary elliptical point on a surface: Once the Indicatrix of Dupin is generated one can choose the first diameter d1 of the ellipse freely. The second diameter results from the condition described in the paragraph above. Consequently one can generate an infinite number of 'conjugate direction pairs' in a single point just by choosing a different 'start' diameter. This method can be applied respectively to hyperbolic and parabolic points.
If we consider the infinite number of conjugate directions there is only one pair which features the particularity of the two directions being perpendicular. These are called the principal diameters and coincide with the principal curvature directions in a point P (Figures 2.14, 2.16) [Kr59].

2.4.6 Conjugate networks

A two directional network of curves (with the direction of one family of curves named A and the other named B) on a surface is called conjugate if at every intersection the tangential directions of A and B are conjugate. From section 2.4.3 follows that the directions A and B are conjugate if the corresponding diameters of Dupin's Indicatrix are conjugate [Kr59].

The tangents at discrete points along curve A in the directions of curves B must form a tangent developable surface with the tangents themselves forming the rulings of the developable surface (see Figure 2.21). This condition highlights the coherence between a conjugate network of curves on a smooth surface S and a PQ mesh as its discrete representative. It also corroborates the relationship between developable surfaces and PQ strips as explained in section 2.3.

As mentioned before there is an infinite number of conjugate networks of curves for any given surface [LiPo06]. Depending on the surface type their generation principles for the Q-nets can be more or less sophisticated. For some surfaces (e.g. those which feature a 'kinematic' generation principle) the Q-net simply consists of the surface generating elements like path curves or rulings. However, surface types with a more complex geometry usually require more complex generation principles for their conjugate networks of curves. Some network examples and the respective surface types they are generated on are listed below [LiPo06]:

> *-**Meridian curves** and **parallel circles** of a revolving surface*
> *-**Generators** of a translational surface*
> *-Network of **principal curvature lines** of any freeform*
> *surface irrespective of its complexity*
> *-Family of c**ontour generators***
> *-Curves of **steepest decent***

Although we have shown in this section that Q-nets might be a potential approach for the generation of a PQ mesh, the resulting network may not be suitable for our purpose, that is it might not serve as a basis for a usable mesh layout.
The reason for this is that the mesh pattern can't be predicted: the existence of an inifinite number of Q-nets for an arbitrary surface means that one would have to generate and compare a certain number of them in order to determine one with a suitable mesh layout.

There is however one conjugate network of curves which is unique for any given surface and whose geometry is determined by the surface shape. The **lines of principal curvature** represent a net which is not only unique for any given surface but also features an orthogonal face pattern which is an additional advantage in terms of the buildability of a structure (as explained in 2.4.5 the principal curvature directions coincide with the only conjugate directions of each surface point which are perpendicular to each other).

In the next section we will investigate principal curvature lines in detail in order to provide the mathematical background for the discrete approximate procedure to compute principal curvature lines on NURBS surfaces which will be proposed in chapter 04.

2.5 Network of principle curvature lines

A doubly curved surface is covered by a network of principle curvature lines simply and without gaps expect of umbilical or naval points (flat points) [STr61] .The lines of principle curvature (LPC) follow the two principal curvatures/directions which measures how the surface bends by different amounts in different directions at a point.

2.5.1 Principle Curvature Directions

The intersection of the normal plane with the surface at a point will generate a planar curve which indicates the normal curvature κ_n for the chosen direction. The normal plane contains the surface normal at the point. If we compute all normal curvature direction at a point we will receive a maximum and a minimum value for every non – umbilical point in the surface (Figure 2.17). These directions are the principal curvature directions which are denoted κmax, κmin or κ1, κ2.
The principle curvature directions are given in the quadratic equation for λ:

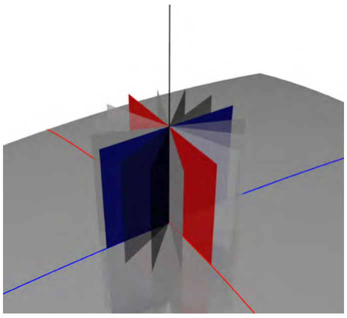


Figure 2.17: Principal Normal curvatures

$$(Fg\text{-}Gf)\,\lambda^2 + (Eg\text{-}Ge)\,\lambda + (Ef + Fe) = 0 \qquad \text{Equation 2.11}$$

E, F, G...first fundamental form coefficients
e, f, g...second fundamental form coefficients

The first and second fundamental from describes the behaviour of curves near point of the surface. For more detailed information please refer to [Str61] chapters 2.2 and 2.5.
The principal curvature direction also be determined with the knowledge of the Gaussian (K) and Mean (H) curvatures [KoMa03, Str61]:

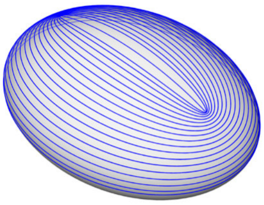


Figure 2.18: Maximum LPC on Ellipsoid

$$K=k_{max}k_{min}=\frac{eg-f^2}{EG-F^2}$$ Equation 2.12

$$H=\frac{1}{2}(k_{max}+k_{min})=\frac{Eg-2fF+eG}{2(EG-F^2)}$$ Equation 2.13

$$k_{max,}\ k_{min}[u,v]=H[u,v]+/-\sqrt{H^2[u,v]-k[u,v]}$$ Equation 2.14

According to Euler (Leonhard Euler, 1707 – 1783) they are always perpendicular to each other.

The multiplicative inverse of the principle normal section curves radii's represents the principle curvatures:

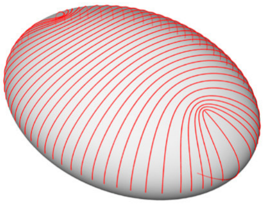


Figure 2.19: Minimum LPC on Ellipsoid

$$K_{max}=\frac{1}{r_{max}},\ K_{min}=\frac{1}{r_{min}}$$ Equation 2.15

With the understanding of the principle curvature directions one can interograte a network of principle curvature lines.

2.5.2 Lines of principle curvature (LPC)

A line of principle curvature is a curve on a surface whose tangents are at every point in the direction of the principal curvatures. There a two sets of lines which are orthogonal to each other which are following respectively the maximum and minimum curvature direction [Str61]. We can obtain a line of principal curvature when solving one of the following differential equations [KoMa03]:

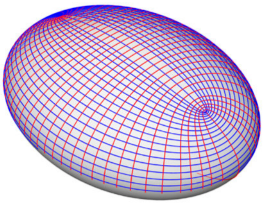


Figure 2.20: LPC on Ellipsoid

$$\frac{du}{ds}=\eta(f+kF),\ \frac{dv}{ds}=\eta(e+kE)$$ Equation 2.16

or

$$\frac{du}{ds}=\mu(g+kG),\frac{dv}{ds}=\mu(f+kF)$$ Equation 2.17

κ... principle curvature
η, μ...normalisation factors of the first fundamental form
E, F, G...first fundamental form coefficients
e, f, g...second fundamental form coefficients

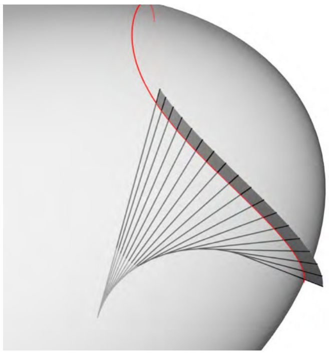


Figure 2.21: Tangential Ruled Surface

The surface normals along a line of principal curvature form a tangential ruled developable surface as displayed in figure 2.21 [Str61].

From previous sections we know that for every surface irrespective of its complexity there is exactly one network of LPCs. Based on Euler we know that these lines are always orthogonal to each other. By means of equations 2.15 and 2.16 we are now in a position to generate the lines of principal curvature for an arbitrary surface.
We also know that the generation of the LPCs require the determination of the maximum and minimum curvature (kmax and kmin) in a point P of a surface S. This is possible for all regular surface points. However, depending on the shape of a surface there might be a number of points which don't show the properties of a regular surface point.
If the local area around a point P of a surface S forms either a plane or a sphere it is not possible to determine the principal curvature directions in P. At every point of a plane all normal curvatures are infinite hence the principal curvature directions are indetermined. At ervery point of a sphere all normal curvatures comprise of the same curvature radius - the radius of the sphere. Hence the principal curvature directions are indetermined. These special point types are called umbilical points of a surface S.

2.5.3 Umbilical points

At an umbilical or naval point all normal curvatures are equal and the principal directions are indeterminate [KoMa03, Str59] as expressed in Equation 2.17:

$$k = k_{max} = k_{min}$$ Equation 2.18

Except for the plane and the sphere the number of umbilical points on a surface is finite. They are located in areas of a surface which are either part of a plane or part of a sphere/either locally spherical or locally planar [Kr59]. At every point of the plane all normal curvatures are infinite hence the principal curvature directions are indetermined. Consequently all points on the sphere and plane are umbilical points.
An umbilical point occurs if the following condition is satisfied [KoMa03]:

$$H^2 - K = 0$$ Equation 2.19

H...Mean curvature
K... Gaussian curvature

Umbilics appear in various forms on a free-form surface such as *isolated points*, *lines* or *regions*. Methods for the numeric extraction of umbilical features on a polynominal surface have been proposed in [CaFa06].

The types of isolated umbilical points can be classified by the number of ridge lines which are a special type of LPCs namely those passing through the umbilic, their index and behaviour close to the umbilical point. The index is the amount of rotation the curvature line experiences when passing through the point.
Isolated umbilical points can be categorised into three basic types depending on the configuration of their ridge lines:

- *Lemon*
- *Star*
- *Lemonstar or monstar*

2.5.4 Lemon

There are is one LPC passing trough the lemon type which changes their sign from maximum to minimum or reverse dependant from which side we approach the point. The LPC passing trough the umbilical point is a straight line in close proximity to it. The ‘lemon’ shape behaviour of the LPC which don’t pass through the umbilical point is displayed in figure 2.22.

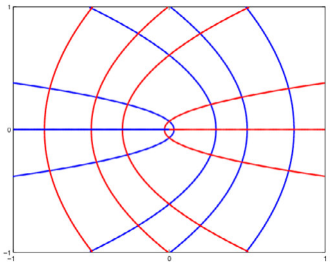


Figure 2.22: Lemon Umbilic [Image by Wikipedia]

2.5.5 Star

There are three LPC passing through the Star type umbilical point, which change their sign at the point. The six lines approaching the umbilical which are next to each other have alternating signs. Three lines approaching the umbilical point cannot be contained in a right angel. The ‘star’ type behaviour of the LPC close the point is displayed in figure 2.23.

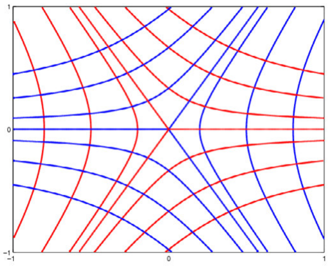


Figure 2.23: Star Umbilic [Image by Wikipedia]

2.5.6 Lemon Star

Also three LPC passing through the lemon star umbilical point which are changing their signs at the point. The three lines approaching the point which are next to each have the same sign and change their sign respectively when leaving the point. The three lines with the same sign can be contained in a right angel. The previous properties distinguish the lemon star from the star umbilical point. The layout of the LPC close to the umbilical point are displayed in figure 2.24.

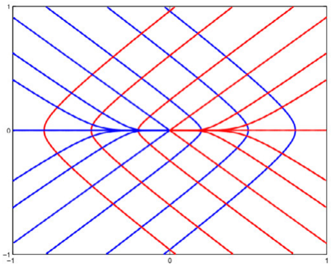


Figure 2.24: Lemon Star Umbilic [Image by Wikipedia]

The knowledge about the characteristics of umbilical surface points and the special configuration the LPCs feature in close proximity to them allows us to isolate those areas of a surface where the LPCs will not offer a mesh pattern suitable for panelisation. It highlights surface areas for which a different approach needs to be found.
As mentioned previously if discretised the lines of principal curvature will result in a PQ mesh with the particularity that this mesh is unique for each surface and offers a definite principle of generation. In addition it

features the property of being per definition a circular mesh. To obtain a comprehensive understanding of the principal curvature lines and the PQ mesh resulting of them we will have a look at the particular properties of circular meshes in section 2.6.

2.5.7 Gauss-Codazzi-Mainardi equations

Alternatively to the method describes in the previous chapter for classical surfaces the Lines of Principle Curvature can be derived using the *Gauss–Codazzi–Mainardi* or *Peterson–Codazzi–Mainardi* equations which consist of a pair of related equations [Str61]:

$$\frac{\partial e}{\partial v}-\frac{\partial f}{\partial u}=e\Gamma_{12}^{1}+f\left(\Gamma_{12}^{2}-\Gamma_{11}^{1}\right)-g\Gamma_{11}^{2}$$ Equation 2.20

$$\frac{\partial f}{\partial v}-\frac{\partial g}{\partial u}=e\Gamma_{22}^{1}+f\left(\Gamma_{22}^{2}-\Gamma_{12}^{1}\right)-g\Gamma_{12}^{2}$$ Equation 2.21

Γ_{jk}^{i} ...Christoffel symbols of second kind
e, f, g...second fundamental form coefficients

Hereby Equation 2.20 relates to the Gauss curvature of a surface to the derivatives of the Gauss map. Hence it is also called Gauss equation. Equation 2.21 is called the Codazzi–Mainardi equation and expresses a structural condition on the second derivatives of the Gauss map and incorporates the mean curvature of the given surface. The equation expresse the remarkable *'Theorema egregium'* which says that the ... *Gaussian curvature of a surface does not change if one bends the surface without stretching it* [WIKI].

Both equations are using the Christoffel symbols which depend exclusively on the coefficients of the first fundamental form: E,F, G. For a detailed definiton of the Christoffel symblos please refer to [Str61] chapter 3-2 (2-7).

If we now set F = 0 and f= 0 we will directly receive the lines of principle curvature as the coordinate lines of the surface.

The principle normal curvatures can be defined as such:

$$k_{max}=\frac{e}{E},\ k_{min}=\frac{g}{G}$$ Equation 2.22

which reduces the Codazzi equations into the following form

$$e_v = \frac{1}{2} E_v \left(\frac{e}{E} + \frac{g}{G}\right), \quad g_u = \frac{1}{2} G_u \left(\frac{e}{E} + \frac{g}{G}\right)$$

Equation 2.23

and can be express in

$$\frac{\partial k_{max}}{\partial v} = \frac{1}{2} \frac{E_v}{E} (k_{min} + k_{max}), \quad \frac{\partial k_{min}}{\partial u} = \frac{1}{2} \frac{G_u}{G} (k_{max} + k_{min})$$

Equation 2.24

If we solve the differential equations for a fixed value of u and v we receive network of principle curvature lines.

2.6 Circular and conical meshes

Circular and conical meshes are special PQ meshes with properties which are interesting for architectural design. Their geometric properties have been studied in [PoWa06], [WaWaLi07] and [BoSu08]. Their key characteristics are summarizes in the following section.

2.6.1 Circular quad configurations

Circular quad faces have been introduced as one of the planarity criteria for quad faces in general in section 2.2.5. A mesh face is called to be circular or discrete orthogonal if all four corner vertices lie on a common circle (Figure 2.25).The following condition must be fulfilled for a mesh to be circular [WaWa07]:

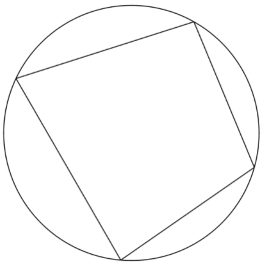


Figure 2.25: Embeded Circular Quad

$$\phi 1 + \phi 3 - \pi = 0 \text{ and } \phi 2 + \phi 4 - \pi = 0$$

Equation 2.25

$\phi 1$ to $\phi 4$...interior corner angels of a face

Depending on the order in which the vertices of a quad are contected with each other two different configurations of a circular quad are possible: as the resulting face can feature edges which intersect or edges which do not intersect one has to distinguish between *embedded* and *non-embedded* circular quads. For an embedded quad its four edges do not intersect (Figure 2.25). In contrast one pair of the opposite edges of the non-embedded circular quad intersect and create an overlapping mesh face (Figure 2.26).

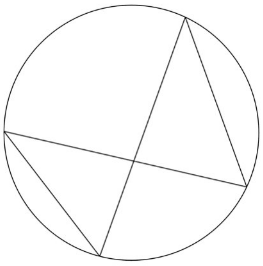


Figure 2.26: Non Embeded Circular Quad

For this reason it is obvious that only embedded quad faces are of interest for the generation of a PQ mesh to obtain a reasonable geometry with respect to the construction industry. Non-embedded quads shall be mentioned only for the sake of completeliness but in the following we will disregard them and concentrate on circular meshes composed of embedded quads and their properties.

2.6.2 Circular mesh

If all faces of a PQ mesh can be circumscribed by a circle meaning if the points which are associated to vertices of elementary quadrilaterals are concircular the mesh is called to be a *circular mesh*. This is of particular interest as a circular PQ mesh represents a discrete version of the principal curvature lines of the surface the mesh is generated on (Figure 2.27). Because of this interrelation with the LPCs circular meshes feature the following properties:

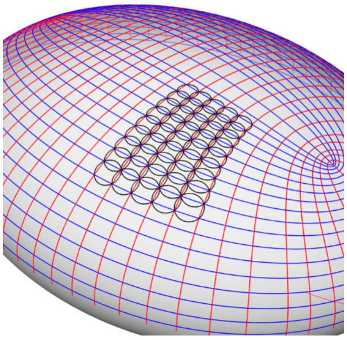


Figure 2.27: Circular Mesh on Ellipsoid

- The edges of each quad are aligned with the principal curvature directions meaning the consecutive edges of the mesh form a polygonal curve which is the discrete version of one principal curvature line of the surface. It is obvious that respectively this property applies to the relationship of every single curvature line of a surface and the aligning consecutive edges.
- Analoguous to the relationsship between continuous LPCs and ruled developable surfaces (explained in section 2.3) the normals at the vertex vectors along a consecutive series of edges form the discrete version of a ruled developable surface.

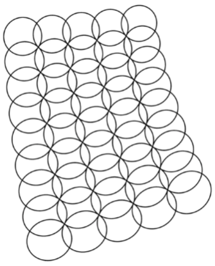


Figure 2.28: Orthogonal Net

The number of edges which meet in an internal mesh vertex indicates if the vertex represents a regular mesh point or an umbilic. For a regular mesh point of a quad mesh the number of edges should usually be four. The rule applies that all vertices different from degree four indicate an umbilical point.

At a vertex V the circumscribed circles of the adjacent faces intersect each other. Hence at a vertex with degree four each circle intersects three neighbouring circles orthogonally and the neighbouring circles touch cyclically (figure 2.28). Based on this fact circular meshes are also called to be discrete orthogonal nets.

A particular property is the fact that any circular mesh can be transformed into a dual mesh on the unit sphere while preserving the planarity criterium of the faces [WaWa07]. Thus it results in a PQ mesh on the sphere. The relationship between PQ meshes on the sphere and their inversions into freeform surfaces have a special relevance for freeform surfaces as they feature an inherent PQ mesh if and only if these surfaces are based on the unit sphere.

The construction of circular meshes on the unit sphere and their dualisation is described in detail in chapter 3.8.

2.6.3 Conformal circular meshes

'In mathematics, conformal geometry is the study of the set of angle-preserving (conformal) transformations on a space ' [WIKI].

A mapping of smooth surfaces is conformal if tangent vectors retain their angles. If all quadrilaterals of a circular mesh are *conformal* the mesh forms a discrete version of an isothermic surface. A quadrilateral/square is called to be *conformal* (Figure 2.29) if its cross ratio is -1 [BHSp05]:

$$aa'/bb'=-1$$ Equation 2.26

a,a'/bb'...opposite edges

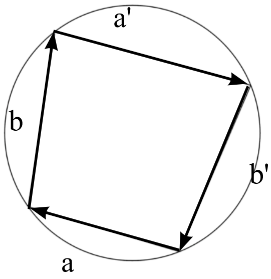


Figure 2.29: Conformal Quad

Per definition an isothermic surface is a surface whose lines of curvature form an isothermic net/system. This again is defined as an orthogonal net of curves on a surface in Euclidean space in which the small quadrangles formed by two pairs of lines from distinct families are, up to infinitesimal quantities of the first order, squares. This means the conformal circular mesh posses the particular property of being a PQ mesh based on the LPCs and featuring only squares as mesh faces.
As explained in section 2.6.1 a circular mesh discretises the lines of principal curvature on a smooth surface. We can therefore transfer the above explained property namely the mesh being a discrete version of an isothermic surface to the smooth surface itself: if the curvature line parmetrisation is conformal the surface from which the mesh is generated is an isothermic surface.
Any isothermic surface can be dualised when applying the Schwarz–Christoffel transformation *(Elwin Bruno Christoffel, 1829-1900 and Hermann Amandus Schwarz 1843-1921)* under preservation of the intrinsic properties of the surface. A detailed definition of this topic will be provided in section 3.8.2. In complex analysis, a Schwarz–Christoffel mapping is a conformal transformation of the upper half-plane onto the interior of a simple polygon. Hence the dualised mesh as the discrete version of a smooth isothermic surface is an isothermic mesh and its faces are conformal, planar and circular.

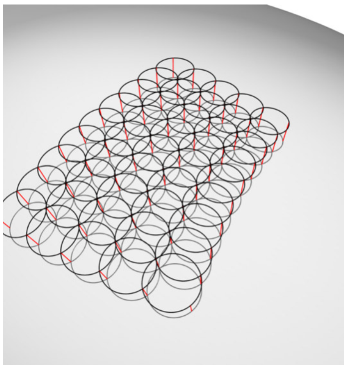


Figure 2.30: Circular Offset Net

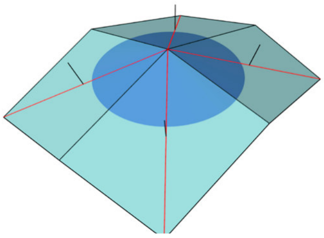


Figure 2.31: Conical Mesh Faces

'The class of isothermic surfaces contains all surfaces of revolution, all quadrics, all constant mean curvature surfaces and ,..., all minimal surfaces.'[BHSp05]

2.6.4 Circular offset meshes

The circular mesh holds constant vertex offset properties. The faces defined by the offset in the direction of the vertex normal by a fixed distance are an offset mesh which preserves the circularity and planarity property (Figure 2.30).

2.6.5 Conical PQ meshes

The faces of a circular PQ mesh are defined by the condition that the faces adjacent to a vertex are tangent to a cone of revolution (Figure 2.31). Also the conical PQ mesh represents a discrete version of the principal curvature lines of the surface the mesh is generated on. The following condition must be fulfilled for a mesh to be conical [LiPo06],[WaWa07]:

$$\omega 1 + \omega 3 = \omega 2 + \omega 4$$

Equation 2.27

$\omega 1$ to $\omega 4$...exterior corner angels around a vertex

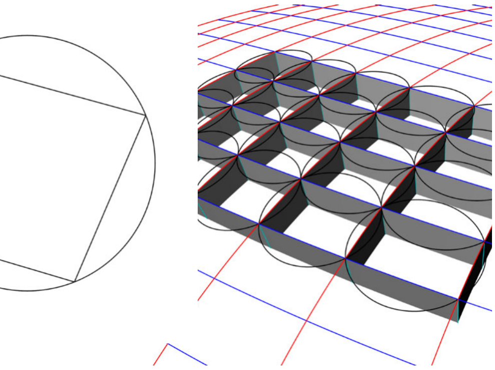


Figure 2.32: Orthogonal Support Structure

2.6.6 Conical offset meshes

The conical mesh holds constant face offset properties. The offset mesh defined by the constant offset in the direction of the face normal preserves the conical and planarity property.

The fact that for both the circular and the conical mesh properties like planarity and angle geometry are preserved when an offset mesh is created makes these mesh types an ideal structure for the cladding industry. A reasonable offset mesh is the main requirement for the build-up of a cladding system for which a support structure is required (represented by the offset mesh) which carries the structure of the actual facets of the envelope (represented by the original mesh).

2.6.7 Orthogonal support structure

Both circular and conical meshes comprise of a orthogonal and planar support structure. The quad faces defined by the connection of the original and offset vertices are also planar and strip wise developable. The supporting face plane is also orthogonal to the initial surface (Figure 2.32).

2.7 PQ mesh transformations

Any affine transformation of the PQ mesh such as scaling with different factors in x, y or z direction will preserve the planarity of the faces. The general definition of spatial transformations are defined in Equations 2.22 (a to c) [PoAs07]:

$$x_1 = ax + by + cz + u$$

Equation 2.28 (a)

$$y_1 = dx + ey + fz + v$$

Equation 2.28 (b)

$$z_1 = gx + hy + iz + w$$

Equation 2.28 (c)

a to i...deformation parameters
u,v,w...translation vector t = (u,v,w)

When applying a affine transformation to a PQ mesh the following properties are preserved:

- *Straightness of edges*
- *Planarity of faces*
- *Paralletity of opposite edges*
- *Length ratio of discrete points on two parallel edges*

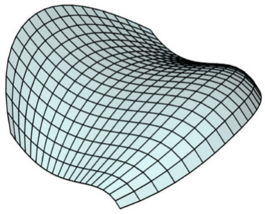


Figure 2.33: Reference PQ Mesh

In general affine transformations are a composition of the four basic types such as *translations*, *scalings*, *shears* and *rotations*.
If we apply only a certain choice out of the deformation parameters a to i, we can obtain the basic affine transformations.
The translation and rotation of PQ meshes are only from trivial interest hence we set the translation vector to zero t = (0,0,0) and do not observe rotation in greater depth.

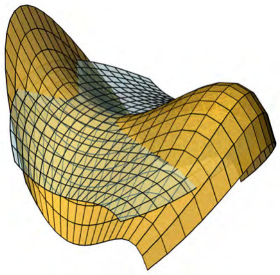


Figure 2.34: Non - Uniform Scaling

2.7.1 PQ mesh scaling

When we only apply the factors a,e to the Equations 2.22 a to c we obtain a scaling of the PQ mesh. For constant values a to c (a= b= c) the scaling is uniform and for non constant values respectively non-uniform (Figure 2.34).

$x_1 = ax$ Equation 2.29 (a)
$y_1 = ey$ Equation 2.29 (b)
$z_1 = iz$ Equation 2.29 (c)

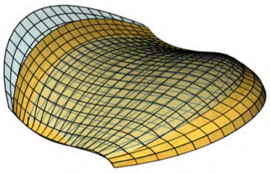


Figure 2.35: Non - Uniform Shear

2.7.2 PQ mesh shear transformation

For a shear transformation parallel to the xy - plane we omit any translation of the mesh vertices z coordinates. The x and y coordinates are translated according to their z values and the constant values c and f. Figure 2.35 shows a non - uniform bi axial shear transformation of the reference surface displayed in figure 2.33.

$x_1 = x + cz$ Equation 2.30 (a)
$y_1 = y + fz$ Equation 2.30 (b)
$z_1 = z$ Equation 2.30 (c)

2.8 Summary

With the knowledge compiled in this chapter we are now in a position to generate a PQ mesh from a freeform surface based on different generation principles. In addition we have a limited number of transformation principles at our disposal to modify the PQ meshes we are able to generate in order to further approximate their geometry towards the input surface.

By contrast in the following chapter surface types are compiled each of whom offer a unique generation principle how to discretise them to a PQ mesh which is well documented in the literature. Although based on conjugate network of curves only some of these meshes is generated by use of the LPCs. A special generation principle exists for each of these surfaces which is explained in detail. Surface types with specific generation principles for PQ meshes offer the advantage that the resulting mesh features absolute planarity of its faces whereas a general generation principle will always result in faces which are only close to planar within a pre-defined tolerance.

The disadvantage however is the constraint in design freedom as each of these surfaces features quite a distinct appearance. This means if the designer is not willing to subordinate his design idea and the accompanying form finding process to one of the shapes offered by these special surface types none of them can provide a solution and a more general approach needs to be found.

In chapter 03 we will have a closer look at these surfaces and their potential for our problem.

Chapter 3 Surfaces with intrinsic PQ mesh representation

3 Surfaces with intrinsic PQ mesh representation

3.1 Introduction

As mentioned in section 2.4 there is a number of subsets of surfaces which feature an inherent PQ mesh representation to their smooth appearance/definition. This means certain elements of the surface like generating curves, meridians or curvature lines (to name just a few of them) might provide a PQ mesh if discretised in a certain way. Such surfaces 'deliver' their respective PQ mesh representation by the way they are mathematically constructed or described.

In contrast we will learn that there are surfaces for which a PQ mesh representation has to be generated specifically and that a generic tool which can be applied to any arbitrary surface is yet to be developed.

Before we look into a possibility of generating PQ meshes for arbitrary surfaces (provided in chapter 04) in chapter 03 we will introduce those surface types for which a parametric representation is known that will lead directly to a PQ mesh. By classifying them into different subsets of surfaces it becomes obvious which characteristics of the respective surface class provides the basis for the discretisation. Finally construction principles for each surface type shall deliver an easy approach to a number of discrete PQ mesh surfaces.

Classical and free form surfaces: First we have to distinguish between surfaces which can be described by a single two dimensional parametric form. These are called classical surfaces hereafter (Figure 3.1). In contrast there is the wide class of surfaces featuring a mathematical description which is a composition of any number of polynominal expressions. These shall be called free form surfaces hereafter (Figure 3.2). A detailed differentiation of these two will be delivered in 3.3.

As mentioned a PQ mesh parameterisation can be inherent in the generating principle of a surface as is the case for surfaces of revolution and translational surfaces (Chapter 2.4.3 and Figure 3.3). It is important to know that surfaces with an intrinsic PQ mesh mainly belong to the class of classical surfaces . Some of them feature the particularity that the curves their generation is based on coincide with a conjugate network of curves.

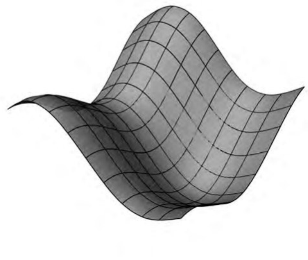


Figure 3.1: Typical example of a classical surface (z = sinx + cos y)

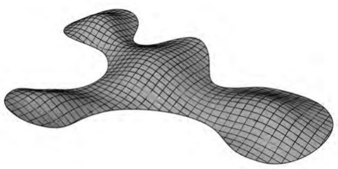


Figure 3.2: Example of a free form surface

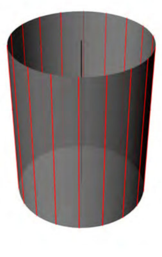


Figure 3.3: Cylinder generated as a revolving surface

Generally a number of different parametric definitions may exist for each individual surface in the R^3 space all of which will lead to an identical smooth surface. An example is the cylinder which can be generated as a revolving surface (straight line segment revolving around an axis) or as a translation surface (circle translated along a straight line segment, Figure 3.4). Each particular parameterisation may lead to parametric curves which coincide with curves of special properties such as asymptotic lines or principal curvature lines. For some surfaces these curves represent a network which can be discretised to a PQ mesh. These shall be identified and their parametrisation as well as the parametrisation of their describing curves shall be explained.

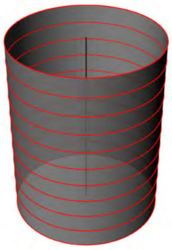


Figure 3.4: Cylinder generated as a translation surface

Any surface in the R^3 space is represented by its two dimensional explicit parametric form [Kr59]:

$$S[u,v] = [x(u,v),y(u,v),z(u,v)] \qquad \text{Equation 3.1}$$

A parametric curve or coordinate curve is formed when setting one of the two variables as constant.

All surface parameterisations which lead to a PQ mesh share the unifying property that they form a conjugate network of curves (Chapter 2.4). In most cases such a conjugate PQ parameterisation also coincides with the principal curvature lines of the surface. As stated in section 2.4 an infinite number of conjugate network of curves exists for each individual surface which when discretised will result respectively in an infinite number of PQ meshes. In theory each of them could be generated by using the knowledge compiled in chapter 02 what would however be a cumbersome exercise.
In contrast to this 'manual generation' in this chapter we want to concentrate on the particular conjugate network of curves inherent to some surface types based on their generation principle or one could say the PQ net which in a way is delivered automatically by generating a surface.

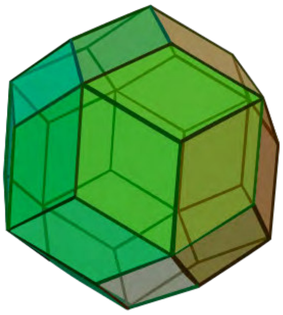


Figure 3.5: Rhombic triacontaherdron tessellation of a sphere [Image by Wikipedia]

In this respect it is fundamental that the only surface class which features an intrinsic PQ mesh parametrisation is that of classical surfaces that is surfaces which can be defined by a single mathematical expression (for a precise definition Chapter 3.3.1).

Depending on the complexity of the surface there are two different ways how to isolate a PQ mesh: For surface types featuring a simple geometric description like revolving surfaces or translation surfaces a uniquely defined PQ mesh can be obtained by using certain generative construction rules (e.g. lamellar or rhombic tessellation of a sphere or a revolving dome as shown in Figure 3.5. and 3.6).
A more sophisticated method for example is required for classical surfaces with constant negative Gaussian curvature (CNGC): in order to obtain a PQ mesh an equal length mesh on a sphere is generated (Figure 3.7)

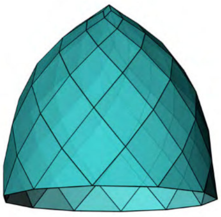


Figure 3.6: Lamellar tessellation of a dome

which is being transformed to receive a CNGC surface of a shape different from that one of a sphere and finally discretised to a PQ mesh (Figure 3.8). Other representatives of the 'more sophisticated approach' to an inherent PQ mesh are discrete minimal surfaces (DMS) based on the sphere and cyclidic nets composed of patches of a cyclide (3.5.2). Obviously these surface types all result in quite distinct output meshes and therefore might have only a very limited applicability. Still it shows a way how to obtain PQ meshes without manually generating them and might offer a solution for particular geometrical tasks.

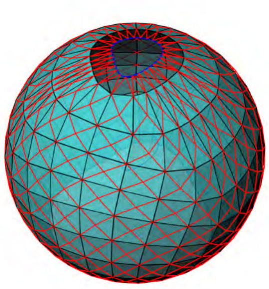


Figure 3.7: Equal length mesh on the sphere

In order to provide an overview of the above described particularities first all surfaces shall be compiled which posses an intrinsic PQ mesh and are already documented and researched in mathematical literature. The next step shall be to investigate and understand the generation principle for each of these surfaces and categorise them accordingly:

- type A: surfaces with simple geometric description thus featuring a PQ mesh based on their generation principle

- type B: PQ mesh representations of complex surface shapes which originate from simple geometric surface discriptions which are transformed into dual surface shapes

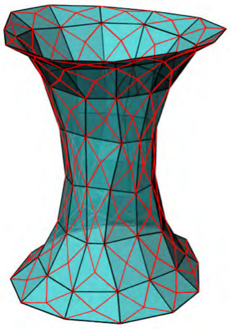
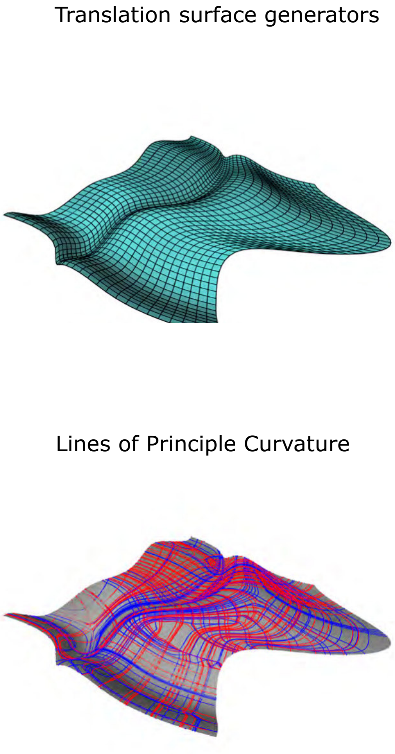


Figure 3.8: Equal length mesh after transformation

For type A we can isolate the curves relevant for each individual surface type (e.g. path curves, profile curves, asymptotes etc.) and show their mathematical description. This method will deliver the parametrisation of one special conjugate network of curves out of the infinite number which are in theory available for any arbitrary surface (Figure 3.9). If discretised this conjugate network of curves will deliver a PQ mesh on the surface. For type B the relevant steps will be described how to obtain a PQ mesh representations of rather complex surface shapes which originate from surfaces of simple geometry like a sphere or a cyclide. The transformation principles to get to the final surface are explained.



Translation surface generators

Lines of Principle Curvature

Figure 3.9: Conjugate network of curves of same surface

Conjugate network of curves as the basis
In chapter 2.3.3 the methodology was described how to generate a conjugate network of curves ‘manually’. If compared now with the few steps required to obtain a PQ-net from a surface whose parametrisation delivers the respective mathematical description it becomes clear that the manual method requires far more time and effort. However the surfaces with a PQ mesh inherent to their parametrisation principle are limited in terms of shape variety and using only those surface types would restrict the design freedom significantly.

We will start our investigation with some mathematical fundamentals and definitions regarding surfaces in general which are essential to understand the analysis of specific surface types as provided in the following.
The difference between single curved and doubly curved surfaces will be

defined and as representatives of these two the relationship and differences between ruled and developable surfaces. Furthermore the class of classical surfaces shall be differentiated from the class of free form surfaces. Finally individual generation principles, the mathematical definition and special properties of surfaces with intrinsic PQ mesh parametrisation will be explained and summarised going from less complex to highly complex surface types with respect to their conjugate parameterisation principles.

3.2 Single curved surfaces

Single curved surfaces feature vanishing Gaussian curvature (K=0) for any given point P of a surface S [Kr59]. From section 2.5.1 we know that the Gaussian curvature is defined as the product of the two principal curvatures, κ1 and κ2. Consequently either k1 or k2 has to equal zero if the Gaussian curvature is to be zero. From section 2.4.4 we know that a point with zero Gaussian curvature is locally parabolic. If this property holds for every single point of a surface S it has to consist of parabolic points only and consequently can be unrolled onto the plane without distortion like stretching or compressing. For this reason such a surface is also called a *developable surface and the planar isometric image is called the development of S.*

If unrolled into the plane one particularity of developable surfaces consists in the preservation of the edge length of their parametric curves as well as the intersection angles of their curves. In mathematics, a mapping that preserves lengths and angles is called an *isometric mapping* [Kr59]. As isometric mappings also preserve Gaussian curvature the converse argument for the fact that a surface is developable is that the surface must have the same Gaussian curvature than the plane namely Gaussian curvature equal to zero.

As mentioned above the Gaussian curvature can only vanish if one or both of the principal curvatures are equal to zero. The only curve however with zero curvature is a straight line which means that in every point P on a single curved surface a straight line passes. For single curved surfaces such a straight line in a point represents one of the two principal curvature lines in P.

In comparison the class of ruled surfaces by definition can be generated by moving a straight line along a path curve thus resulting in surfaces which carry families of straight lines [PoAs07]. At the same time we know that only some ruled surfaces posses the property of being single curved or in other words developable. This means although both surface types carry straight lines these must feature significant differences which characterise the two surface classes.

To distinguish ruled surfaces from the class of developable surfaces in the following paragraph we will investigate properties and particularities of ruled surface in general and ruled developable surfaces in particular.

Special attention will be paid to the configuration of the rulings for both surface types to highlight their commonalities and differences.
For the sake of completeness it shall be mentioned that general ruled surfaces actually belong to the class of doubly curved surfaces. However they are covered in the following section in order to differentiate between them and the class of developable surfaces whose intrinsic PQ mesh will be described subsequently.

3.2.1 Ruled surfaces

Generally speaking a ruled surface is a surface generated by the spatial motion of a straight line called its *generator* or *ruling* along a path curve called the *directrix. H*ence by definition ruled surfaces carry a continuous family of straight lines or in other words through every point of a ruled surface at least one straight line passes which lies on the surface.

Rulings
These straight lines are called rulings and posses the special property that they are asymptotic curves of the surface S.

An asymptotic curve of a surface S is defined as one which is always tangent to an asymptotic direction of S. An asymptotic direction again is defined as one in which the normal curvature is zero. This means for any arbitrary point P on an asymptotic curve c of a surface S the following property holds: if we construct the plane which is defined by the normal of S in P and the tangent of c in P the curve that results from intersecting the plane with S will feature zero curvature in P. An interesting aspect is that asymptotic directions only occur in surface points which feature either negative or vanishing Gaussian curvature [Wikipedia]. From this definition it follows that the Gaussian curvature on a ruled regular surface is everywhere nonpositive and consequently the surface carries only parabolic and/or hperbolic points. [WolframMathWorld]

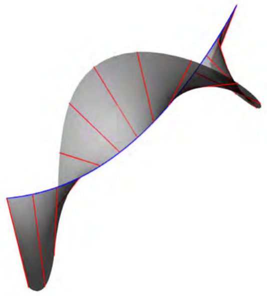


Figure 3.10: Ruled surface I

Parametric representation
The motion of the generator g is determined by the path curve *p(u)* and the direction vector z of g for every value of u as the unit vector *v(u).* A ruled surface may therefore be presented in the following form [Kr59]:

$$S[u,v]=\mathrm{p}(u)+\mathrm{v}\mathrm{z}(u) \qquad \text{Equation 3.2}$$

Ruled surfaces can be generated in two different ways:

I. By moving a straight line (generator) along a single path curve while simultaneously changing the line's direction corresponding to the direction vector of the generator (Figure 3.10).

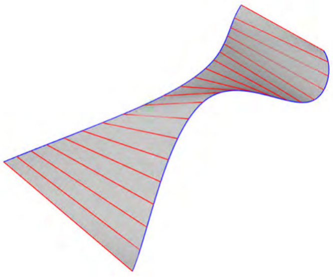


Figure 3.11: Ruled surface II

II. By connecting two sets of corresponding points on two directrix (spatial) curves with a series of straight lines (Figure 3.11).

An interesting aspect of generation principle II is the fact that different parametrisations of the two directrix curves will result in different surfaces: the parametrisation defines the points on the curve and the shape of the surface is determined by connecting two sets of corresponding points of the two directrix curves. Hence the parametrisation of these curves will influence the location of the points and consequently the location of the rulings which ultimatively determine the shape of the surface. A related notion is the parametrisation of a classical surface which is defining its inherent PQ mesh (Chapter 3.1).

If we apply generation principle I a ruled surface such as the *conoid* or the *möbius strip (August Ferdinand Möbius, 1790 - 1886)* will be the result. For the right *conoid* (Figure 3.12) we use a linear path curve with rulings which are all perpendicular to the path curve. The rulings can perform any kind of rotation around the path curve while moving along it. The parametric description of the right conoid is given in equation 3.3 [MW01]:

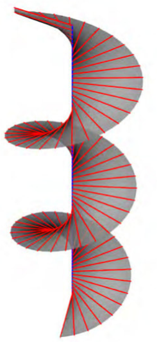


Figure 3.12: Conoid

$$\begin{aligned} x[u,v] &= v \cos \vartheta(u) \\ y[u,v] &= v \sin \vartheta(u) \\ z[u,v] &= h(u) \end{aligned}$$

Equation 3.3

If in contrast we apply a constant rotation (h(u) = 2u, J(u) = u) to the rulings we receive a helicoid (Figure 3.13) which is a special type of the right conoid.

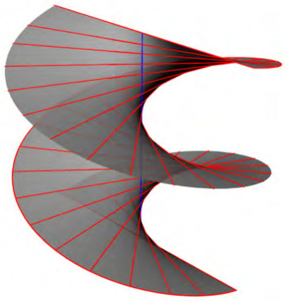


Figure 3.13: Helicoid

The *Möbius strip* (Figure 3.14) is generated by choosing a circle as path curve of the surface. The rulings are continuously rotated perpendicular to the circular path in such a way that the last ruling coincides with the first one. The result is a surface with the special properties of having only one side and one boundary curve and being non-orientable. The parametric definition of the Möbius strip is given in equation 3.4:

$$\begin{aligned} x[u,v] &= r \cos u + v \cos u/2 \cos u \\ y[u,v] &= r \sin u + v \cos u/2 \sin u, \\ z[u,v] &= v \sin u/2 \end{aligned}$$

Equation 3.4

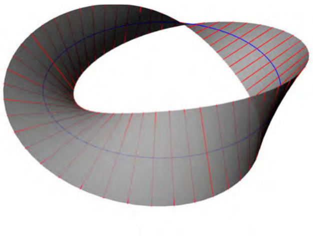


Figure 3.14: Möbius strip

Examples for ruled surfaces generated by using principle II are *hyperbolic paraboloid* (HP) surfaces (Figure 3.15) and the *hyperboloid* (Figure 3.16). The hyperbolic paraboloid and the hyperboloid posses the special property of being doubly ruled surfaces as for any given point P on these two surfaces two straight rulings are passing. The boundary components of a *hyperboloid* are defined by two parallel circles. Parametric definitions for both surfaces are given below:

Hyperbolic parboloid [MW02]

$$x[u,v] = u$$
$$y[u,v] = v$$
$$z[u,v] = u\, v$$

Equation 3.5

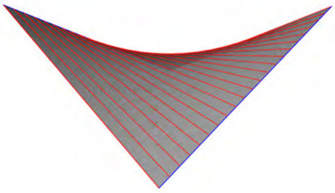


Figure 3.15: Hyperbolic paraboloid

Hyperboloid of one sheet [MW03]

$$x[u,v] = a \sqrt{}\,(1 + u^2) \cos v$$
$$y[u,v] = a \sqrt{}\,(1 + u^2) \sin v$$
$$z[u,v] = c\, u$$

Equation 3.6

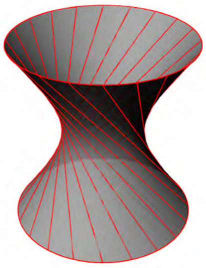


Figure 3.16: Hyperboloid of one sheet

The plane, the tangent developable surface, the cylinder and the cone are ruled surfaces which are at the same time developable. Consequently the rulings of these surface types must feature additional properties to those typical for ruled surfaces. This will be investigated in detail in the following section.

3.2.2 Developable surfaces

The general properties of single curved and thus developable surfaces have been studied in section 3.2. Here we want to investigate the special relationship between ruled surfaces and developable surfaces which are a special type of ruled surfaces.

If a developable surface lies in three-dimensional Euclidean space, and is complete, then it is necessarily ruled. However the converse is not always true [Wikipedia].

We know that a cylinder and a cone are developable surfaces as they are singly curved. We also know that they can be generated by moving a straight line along a path curve meaning they belong to the class of ruled surfaces (Chapter 3.2.1). If we have a look at a hyperboloid we can see that although being a ruled surface it is not singly curved and hence doesn't belong to the class of developable surfaces.

The unifying property of developable surfaces and ruled surfaces is the family of straight lines which both surfaces types are carrying in every arbitrary point. Consequently whatever distinguishes one surface type from the other must be tied to their rulings.
If we have a look at the tangent planes of ruled surfaces the difference becomes obvious: for a ruled surface the tangent plane t in a point P usually touches the surface solely in P and the ruling r through P has no other common points with t than P. Looking at another arbitrary point of r and its associated tangent plane we will see that the tangent plane performs

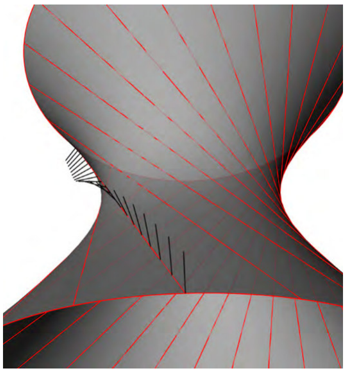


Figure 3.17: Rotating normals of ruling tangent planes

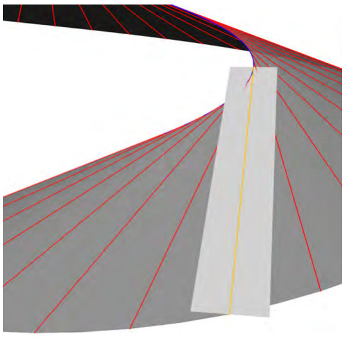


Figure 3.19: Torsal ruling

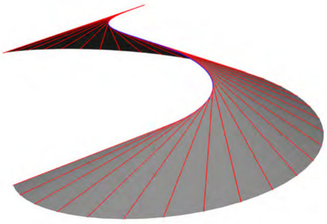


Figure 3.18: Tangent developable surface

a twist (Figure 3.17) meaning it rotates around the ruling [PoAs07]. In comparison there are rulings for which the tangent plane in a point P does touch the ruling not only in P but the ruling is contained in the tangent plane (Figure 3.18). This special type of ruling is called a *torsal ruling*. Those ruled surfaces whose rulings are exclusively torsal are developable surfaces.

A ruling is called torsal if the following condition holds:

‘A developable surface is a ,.., ruled surface with the property that it has the same tangent plane at all points on one an the same generator.’ [Kr59]

All developable surface can be categorised based on their rulings which come in three different configurations:

- all rulings are parallel
- all rulings pass through a fixed point A
- all rulings are tangents of a space curve r

Hence there are four basic types of developable surfaces which can be distinguished by means of their rulings: *cylinders*, *cones, tangent surfaces* of space curves (Figure 3.19) and planes [Kr59]. Any developable surface is a composition of these four basic developable surface types [PoAs07]. Consequently a surface which consists of a number of developable patches which belong to one of the above named types is still developable. However depending on the way these surfaces patches are joined together the surface might feature a smooth appearance or it might feature kinks along the joints.

Another surface configuration might be a mixture of developable and non-developable surface patches. These can be distinguished by means of their ruling types: If a surface features torsal rulings but in a non-exclusive manner only those surface patches are developable for which the torsal rulings are in consecutive order.

3.2.3 Extrusion surfaces

An alternative generation principle for a cone or a cylinder surface (or a plane which is trivial and therefore omitted) is the linear extrusion of a planar or spatial profile curve.
In the field of geometry extrusion describes a dimension increase of an ‘element’ by parallel or central translation in space. A vector determines the direction of the translation. If the translation is to be parallel all points of the curve will follow this direction. If the translation is to be central the vector determines the apex in which all generators of the extrusion surface will meet. The generators of an extrusion surface can be defined as straight line segments which connect each discrete point of the base curve b with the apex A. The base curve can be an open or closed linear

element, either planar or spatial and consisting of straight line segments or being curved. Extruding a line or curve will always result in a developable surface or the discrete version of it.

Cylinder and cone

If a parallel extrusion of a circular base curve is performed a right circular cylinder is generated for which all generators are parallel. The central extrusion of a closed curve generates a generalised cone (open curve results in a conical surface) for which all generators pass through a common point called the apex of the cone. These generation principles apply for all types of conical and cylindrical surfaces.
As briefly mentioned in the introduction to chapter 03 single curved surfaces can be divided into two surface classes which are distinguished by their parametric representation. If a surface can be defined by a single mathematical expression it belongs to the class of classical surfaces. If the description of a surface requires a number of polynomial equations it belongs to the class of freeform surfaces. For single curved surfaces this distinction is irrelevant (and therefore disregarded) as all of them feature an intrinsic PQ mesh without exception. For doubly curved surfaces however this distinction is important as only the class of classical doubly curved surfaces can feature an intrinsic PQ mesh (Chapter 3.3.1 and 3.3.2).

3.2.4 PQ meshes on developable surfaces

In section 3.2.2 we have investigated the special properties of the rulings of a developable surface. A geometrical interpretation of the results could be that for developable ruled surfaces the consecutive rulings are always co-planar hence they lie in the same plane. In contrast the consecutive rulings of a ‘general’ ruled surface are skew lines and therefore they do not intersect and are not parallel.
If consecutive rulings are always co-planar these can quite simply be isolated to describe a planar quad mesh on the surface: if we discretise the profile curve of a developable surface this will result in a polyline with a certain number of vertices V_1 - V_x. We can then isolate the rulings of the developable surface precisely in these vertices. If we now discretise the rulings this will result again in a number of vertices r_1-r_x. This way we obtain a point cloud on the surface. If we finally connect the vertices with straight lines in an order which follows the bounding components of the input surface we will obtain a PQ mesh.

If this principle is applied to a cylindrical surface (surface obtained from a planar base curve and parallel extrusion) the generated parameter curves coincide with the principal curvature lines of the surface. If we choose the same distances between the vertices when discretising the rulings and the circles the generated quads will be congruent. For a cone and a tangent developable surface this condition will lead to quads which are similar (same corner angles and same proportion of edge lengths but scaled in size).

When discretising a surface the distances between the vertices can be chosen to have different length. This will lead to PQ faces which are non-congruent for cylinder surfaces or non-similar in case of cone and tangent developbable surfaces.

3.3 Doubly curved surfaces

In contrast to single curved surfaces a doubly curved surface must feature Gaussian curvature which is different from zero ($K \neq 0$) (Section 3.2). This is a condition which applies locally for a doubly curved surface might feature surface patches with single curvature. As applies for single curved surfaces doubly curved surfaces can be divided into the two classes of classical surfaces and freeform surfaces. For doubly curved surfaces this distinction is essential as only classical doubly curved surfaces can feature an intrinsic PQ mesh parametrisation. Therefore these two surfaces classes are clearly defined in the following.

3.3.1 Classical surfaces

The parametric representation of a classical surface is using elementary linear and/or nonlinear functions which are defining the entire surface in one expression. A good example delivers Scherk's second minimal surface *(Heinrich Ferdinand Scherk, 1785- 1885)* [MW04]. This surface although being described in one expression features quite a complex geometrical shape (Figure 3.20). Still it belongs to the classical surfaces and thus posses an intrinsic PQ mesh. For a detailed description of the generation principle for PQ meshes on minimal surfaces please refer to 3.8 (PQ meshes from circle packings on the sphere). The equation for Scherk's second minimal surface is given below:

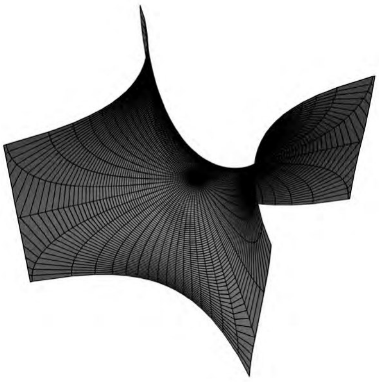


Figure 3.20: Scherk's second minimal surface

$$\begin{aligned} x[r;\theta] &= \ln\left((1+r^2+2r\cos\theta) / (1+r^2-2r\cos\theta)\right) \\ y[r;\theta] &= (1+r^2-2r\sin\theta) / (1+r^2+2r\sin\theta) \\ z[r;\theta] &= 2\tan_{-1}\left((2r^2\sin(2\theta)) / (r^4-1)\right) \end{aligned} \qquad \text{Equation 3.7}$$

with $\theta \in (0, 2\tau)$, $r \in (0,1)$

Amongst other things the theory of classical differential geometry deals with the investigation of curves and surfaces which can be defined by this one parametric expression. From this expression the first and the second fundamental form of a surface can be derived.

By means of the first and second fundamental form the metric properties of a surface like its curvature behaviour (Gaussian curvature, mean curvature and normal curvature) as well as line element and area element can be determined. In general terms one could say they describe the behaviour of curves near a point P on a surface S (Chapter 2.5.1).

An example of the two fundamental surface forms shall be given on the basis of Scherk’s minimal surface:

First fundamental form of Scherk’s minimal surface [MW04]

$$\begin{aligned} E &= 16\,(\,1 + r^2)^2 /\,(\,1 + r^8 - 2\,r^4\,(\cos 2\theta)) \\ F &= 0 \\ G &= 16\,(\,1 + r^2)^2 /\,(\,1 + r^8 - 2\,r^4\,(\cos 2\theta)) \end{aligned}$$

Equation 3.8

Second fundamental from of Scherk’s minimal surface [MW04]

$$\begin{aligned} e &= 8\,(\,1 + r^4)\,\sin 2\theta /\,(\,1 + r^8 - 2\,r^4\,(\cos 4\theta)) \\ g &= 8\,(\,1 + r^4)\cos 2\theta /\,(\,1 + r^{\underline{8}} - 2\,r^4\,(\cos 4\theta)) \\ f &= 8\,(\,1 + r^4)\,\sin 2\theta /\,(\,1 + r^8 - 2\,r^4\,(\cos 4\theta)) \end{aligned}$$

Equation 3.9

3.3.2 Freeform surfaces

In contrast to classical surfaces a freeform surface can’t be described by a single mathematical expression. Depending on the complexity a host of polynomial equations may be required to mathematically define the surface or according to Pottmann et al. *one needs appropriate mathematical algorithms* [PoAs07].

In fact there was no mathematical description of freeform geometry before the 1940s and 50s when CAD and CNC found their way into the world of design. Questions like *‘How to store a surface design digitally’* initiated the development of concepts for digital freeform design. Bezier surfaces, B-spline surface, NURBS surfaces and subdivision surfaces all offer different methods how to design and control freeform surfaces within a 3D modeling package. In general they all have a quadrilateral control mesh (Figure 3.21) which can be used to build and influence a freeform surface. Surface modifications basically change the polynomial functions of higher degree which are describing the actual surface. This mathematical process however stays hidden and only the visual part namely the graphical changes to the surface are visible. Depending on the interpolation technology the control point weight, the knot vector and the degree of the polynomial can be manipulated in order to influence the final shape of the surface via its given control mesh.

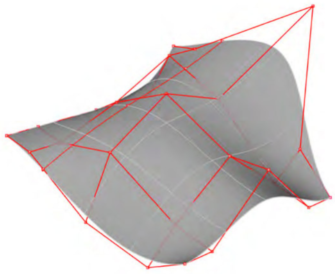


Figure 3.21: Quadrilateral control mesh of a free form surface

The complex mathematical description of freeform surfaces already suggests that there is no intrinsic PQ mesh for any of them. This again underlines the necessity for a general solution.
Still we have to investigate in detail how intrinsic PQ meshes are offered by different types of classical single and doubly curved surfaces as introduced at the beginning of this chapter. An overview will be provided in the following sections starting with less complex surface types going to the highly complex ones.

3.4 Surfaces of revolution

A surface of revolution R (also known as rotational surface or revolving surface) is defined by the rotation of a profile curve about a straight line called the axis A. Any arbitrary planar or spatial curve can be chosen to generate a surface of revolution although a planar curve will result in a better indication of the final surface shape and should therefore be preferred.

For revolving surfaces a planar profile curve always coincides with one of the two families of principal curvature lines meaning one part of the PQ mesh is 'delivered' already by the surface description. The same does not apply for a spatial profile curve meaning for this case isolating a PQ mesh will require more effort.

In order to understand how a revolving surface is defined we will first examine an isolated point P on the profile curve c. It is obvious that due to the rotational generation principle any point on this curve will describe a circle C when rotated about A. The plane which contains C is called the supporting plane S and is orthogonal to axis A. This applies for any arbitrary point on c and thus a revolving surface carries a set of circles which we call parallel circles hereafter. For each of these circles there is a supporting plane orthogonal to A. If we now intersect the revolving surface R with a set of planes M which all contain axis A we obtain a number of section curves which are called meridians curves of R. It is obvious that the meridian curves are planar (section curve from surface and plane) and due to the revolving generation principle are all congruent (Figure 3.22). From the geometrical relationship between axis A and both sets of planes S and M follows that their respective section curves (parallel circles and meridians) are always orthogonal to each other. We have explained earlier that planar section curves of a revolving surface always coincide with one set of the principal curvature lines. Now however we have isolated to sets of planar section curves which in addition to planarity are orthogonal to each other. This implies that the parallel circles and meridians coincide with the lines of principal curvature of the revolving surface. In addition the meridian curves always coincide with the geodesic lines of R (namely curves of shortest path) [Str61].

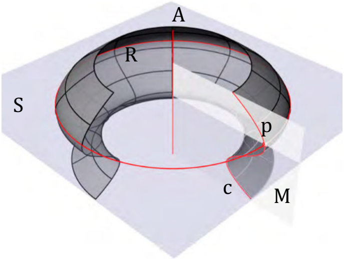


Figure 3.22: Defining elements of a revolving surface

A general surface of revolution can be described in the form:

$$
\begin{aligned}
x[u, v] &= x(v)\cos(u) - y(v)\sin(u)\\
x[u, v] &= x(v)\sin(u) + y(v)\cos(u)\\
x[u, v] &= z(v)
\end{aligned}
$$

note: not for cylinders Equation 3.10

If a planar section curve is chosen as profile curve the parametric representation can be simplified to:

$$x[u, v] = x(v)\cos(u)$$
$$y[u, v] = x(v)\sin(u)$$
$$z[u, v] = z(v)]$$

note: this eq. doesn't apply for cylinders
Meridians for u = const, parallels for v = const Equation 3.11

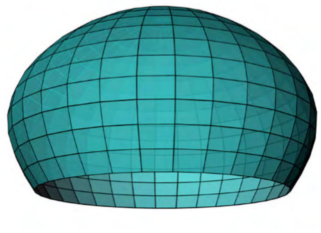


Figure 3.23: Surface of revolution (revolving dome) discretised with a PQ mesh

Figure 3.24: Built example: Lingotto glass dome by Renzo Piano [Image by Antonio Mancinelli

One particular property of surfaces of revolution is the fact that they form a discrete isothermic surface [BoPi94] meaning the lines of curvature of a revolving surface represent an isothermic net which is defined as a mesh of curves which are orthogonal to each other and which up to infinitesimal discrepancies result in square facets. As we know from 2.6.2 isothermic surfaces can be discretised by means of a circular mesh consisting of conformal squares. These are interesting properties inherent to every PQ mesh on a revolving surface which might proof useful for their applicability in the construction industry. A general way how to obtain a PQ mesh on surfaces of revolution shall be described in chapter 3.4.4.

However before concentrating on the generation principle of a PQ mesh there are a few special types of revolving surfaces featuring particularities which we want to highlight in the following sections. Inter alia these include the only revolving surface which is at the same time a minimal surface (catenoid) and those revolving surfaces which allow a geodesic mapping onto the plane (spherical and pseudo spherical surfaces). Besides some of them even offer a further PQ mesh configuration in addition to the 'general' one described in 3.4.4. Both the special surface types and the special PQ mesh generation principles shall be described next.

3.4.1 Surfaces of revolution with constant Gaussian curvature

There is a sub-class of revolving surfaces all of which have constant Gaussian curvature (K=const) and are therefore called constant Gaussian curvature (CGC) surfaces. One of their particularities is that they can be mapped geodesically into the plane (E. Beltrami, 1835-1900).
In general terms a geodesic mapping transforms the geodesic lines of one space into the geodesic lines of another space. A surface which -at least locally- permits a geodesic mapping is called -locally- projectively flat. It is important to know that geodesic mappings only exist in exceptional cases one of them being for surfaces of constant curvature. Thus these surfaces can be called projectively flat [Kr59].
A geodesic mapping preserves geodesic lines. We can distinguish between surfaces of constant positive Gaussian curvature (K>0) which are called *spherical surfaces* and those of constant negative Gaussian curvature (K<0) called *pseudospherical surfaces*.

Both surface types can be represented in the following general form [Kr59]:

$$x[u,v] = \lambda \cos(u/c) \cos(v)$$
$$y[u,v] = \lambda \cos(u/c) \sin(v)$$
$$z[u,v] = \int \sqrt{(1- \lambda^2/c^2 \sin^2(u/c))}\, du$$

Equation 3.12

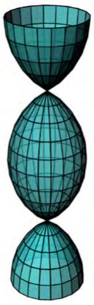


Figure 3.27: Discrete elliptic spherical surface of revolution

Spherical surfaces of revolution (K>0)

There are three different types of spherical surfaces of revolution (K>0) which correspond to the values

$\lambda = c$ for the sphere with radius c
$\lambda > c$ for hyperbolic spherical surfaces and (Figure 3.26)
$\lambda < c$ for elliptic spherical surfaces (Figure 3.25)

Pseudospherical surfaces of revolution (K<0)

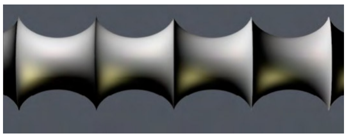


Figure 3.28: Hyperbolic pseudospherical surface of revolution [Image by Geometrie Werkstatt Prof. Dr. Christoph Bohle

Again there are three different types which can be generated by resolving the elliptic integral respectively for the hyperbolic, parabolic and elliptic condition. The parabolic pseudospherical surface can be represented with terms of elementary function:

$$x[u,v]= c \cos(u) \sin(v)$$
$$y[u,v]= c \sin(u) \sin(v)$$
$$z[u,v]= c\ (\cos(v) \log (\tan(v/2))$$

Equation 3.13

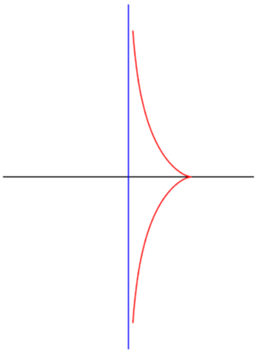


Figure 3.26: Tractrix

Depending on the type of profile curve we receive different types of pseudospherical surfaces.

One example is the revolving profile curve which is called tractrix (Figure 3.27) and which was first considered by C. Perrault in 1670. It can be described with the following function [WP01]:

$$y=\int_z^a \frac{\sqrt{a^2-t^2}}{t} dt = \mp \left(a \ln \frac{a+\sqrt{a^2-x^2}}{x} - \sqrt{a^2-x^2} \right)$$

Equation 3.14

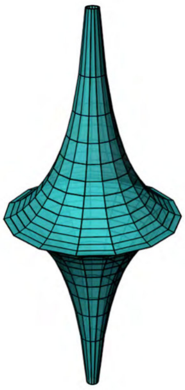


Figure 3.25: Discrete pseudosphere

The particularity of a tractrix t is the property that the length of the segment of the tangent line from any point P on t to the y-axis is constantly equal to 1. The parabolic pseudospherical surface which is generated by rotating a tractrix about its asymptote is called the tractricoid. Other names are tractroid, antisphere, tractrisoid or most commonly known 'pseudosphere'. The pseudosphere has a singular central circle and thins out asymptotically on both ends (Figure 3.28) [Str61]. The name ‘pseudosphere’ derives from the fact that the surface has the same volume as the sphere of same radius however featuring constant negative Gaussian curvature in comparison to the constant positive curvature of the sphere. An interesting aspect of the pseudosphere is that it is always -at least locally- part of a hyperbolic surface structure.

CGC surfaces usually feature an unusual and often complex shape. The special cases of cylinder and cone which feature **vanishing** constant Gaussian curvature (K=0) however belong to the class of single curved surfaces (Chapter 3.2) and are the only two revolving surfaces which are developable.

3.4.2 Surfaces of revolution with vanishing mean curvature (VMC or minimal surfaces)

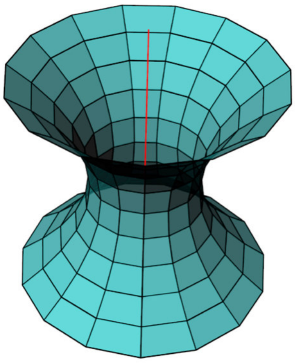


Figure 3.29: Discrete catenoid

Surfaces with vanishing mean curvature (M=0) are called *minimal surfaces*. They may also be characterised as representing the one surface within a closed boundary curve which features the smallest possible surface area (just like a soap film would span a closed wire curve). For this reason they are also known as soap film surfaces. One particularity is that on a minimal surface the asymptotic lines (curves of zero normal curvature) form an orthogonal network of curves which can be mapped *conformally* (angle preserving) onto a sphere. This property also applies to the principal curvature lines of the surface.
The *catenoid* (Figure 3.29) is the only revolving surface which at the same time belongs to the class of minimal surfaces. It can be generated by the rotation of a catenary curve (hanging rope curve) about an axis. The catenoid is represented in the following form:

$$S\,[u,v] = [u\cos(v),\, u\sin(v),\, \mathrm{c}\cosh^{-1}(u/\mathrm{c})] \qquad \text{Equation 3.15}$$

The cantenoid can be mapped isometrically (arc length preserving) onto a right helicoid and will locally coincide with it.
General (non-revolving) minimal surfaces e.g. those generated form a circle packing on the sphere will be discussed in detail in section 3.7.7.

If generalised the definition of minimal surfaces may also cover surfaces with constant mean curvature other than zero. These are investigated in the following.

3.4.3 Surfaces of revolution with constant nonzero mean curvature

In general surfaces which possess constant mean curvature (CMC) other than zero are based on an area minimisation under a volume constraint: Given are a pre-defined volume and one or a number of boundary curves. The minimal surface based on these conditions is the one surface that features the smallest possible surface area for the pre-defined volume within the given boundary curves. Such *CMC surfaces* are closely related to shapes which are used for pneumatic constructions.

The Delaunay surfaces *Nodoid* and *Unduloid (C. E. Delaunay 1816 –*

1872) are the only two surfaces with constant nonzero mean curvature which can be parametrised with a rotational generation principle and thus featuring an inherent PQ mesh.

If a hyperbola is rolled along a fixed line l the movement of the focal point generates a trajectory which forms the profile curve for the nodoid (Figure 3.30). The same applies for an ellipse and the unduloid (Figure 3.31). Generation principles for a PQ mesh for both surfaces are given in 3.4.6.

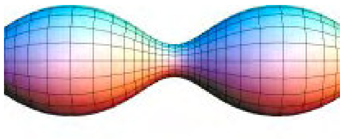


Figure 3.32: Nodoid [Image by 3DXM Consortium]

After having compiled and discussed the parametric definition and the special properties of different types of continuous revolving surfaces we will now investigate the discretisation of revolving surfaces in general and those surfaces which offer an additional PQ mesh configuration in particular.

3.4.4 PQ meshes on general surfaces of revolution

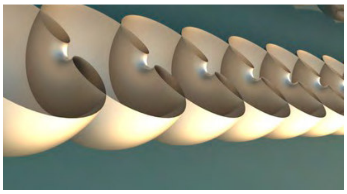


Figure 3.33: Unduloid [Image by GeometrieWerkstatt Prof. Dr. Christoph Bohle

In general the substitution of the planar profile curve and the parallel circles of a revolving surface for a number of polylines and the discretisation of the rotation by a pre-defined angle-degree leads to an orthogonal PQ mesh. If the discretisation of the rotation is uniform we receive facets which are *congruent*. The edges of the PQ facets are formed by the discrete meridians and parallel circles of the revolving surface R which coincide with the principal curvature lines of R. Hence the facets are at the same time circular quads. To obtain the discret version of meridians and parallel circles the u and v values of the parametric representation S [u,v] need to be set as constant. As previously mentioned the discretisation of a surface of revolution always results in a circular mesh of conformal squares thus providing a facet layout of a very distinct appearance. Therefore in the following we want to investigate additional PQ mesh configurations which are however only available for some individual revolving surfaces.

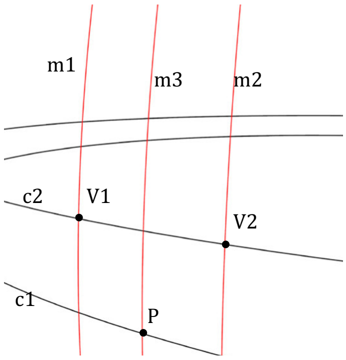


Figure 3.30: Starting circles and parallels

3.4.5 Rhombic PQ meshes for surfaces of revolution with positive Gaussian curvature

A special case are revolving surfaces with positive Gaussian curvature (K>0) as for this surface type we can generate a PQ mesh which leads to a rhombic or lamellar tesselation: As start configuration for the mesh generation we need to isolated two parallel circles with a pre-defined distance and discretise them to polylines with a pre-defined vertex density. We can then construct faces in a diagonal layout (Figure 3.32).
The resulting mesh is uniquely defined for any chosen starting parameters and is covering the entire surface of revolution. The construction steps are described below:

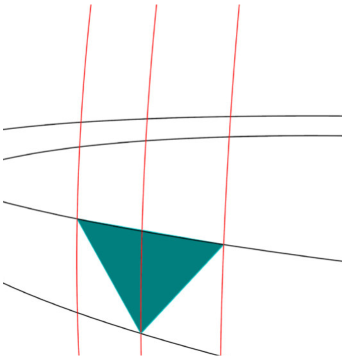


Figure 3.31: Half facet

We choose two parallel circles c_1,c_2 on a surface of revolution with positive Gaussian curvature (Figure 3.32). The chosen distance of the circles will determine the height of the starting triangle which later in the construction process will result in the first mesh facet (a rhombus consisting of two triangles). The even and uniform subdivision of the two parallel circles on the other hand determines the length of the diagonal of the starting facet as two times the vertex distance represents the diagonal length. Each of the vertices V_1, V_2 on the discrete parallel circles determine the location of a corresponding meridian curve m_1, m_2 which runs through one V. The triangular starting facet is defined by an arbitrary point P on the first parallel circle and two vertices V_1 and V_2 on the second parallel circle which are located to the right and to the left of the meridian curve that runs through start point P . These three points define the supporting plane of the initial facet (Figure 3.33. The intersection of the supporting plane with the meridian m_3 through start point P defines the fourth vertex V_4 of the planar starting rhombus (Figure 3.34). This principle will be repeated to generate the first row of rhombs. It becomes obvious that for the definition of points V_x and V_y of each rhombus we use only every other of the vertices which were generated by the subdivision of the parallel circles. To generate the next row of rhombs the procedure will be repeated on the second parallel circle but with a shift by one vertex. This construction principle can be performed until we hit the boundaries of the surface and thus the whole surface will be covered eventually with planar quadrilateral facets in the shape of a rhombus.

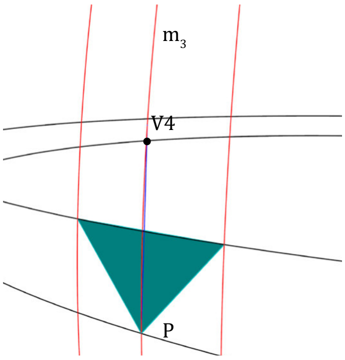


Figure 3.34: Construction of the fourth vertex

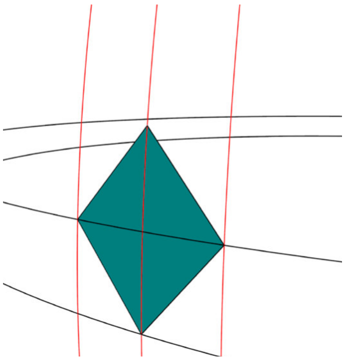


Figure 3.35: Full facet

For this construction principle the facets will feature different sizes depending on the local curvature of the surface as the curvature determines the location of the forth vertex of a rhombus. The facet size however can be influenced by the distance of the parallel circles we start with and their subdivision which defines the distance of the vertices *V_x and V_y*.
Due to the fact that for the construction of the PQ mesh only every other vertex is used we can obtain a second PQ mesh which is congruent to the first one by choosing as the starting point of the second mesh the point precisely to the right or the left of our initial starting point P. This will result in an identical mesh (with respect to facet sizes) but positioned on the revolving surface with a shift by one vertex (or half the diagonal of the rhombus).

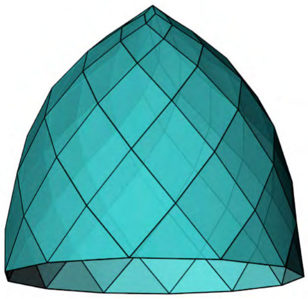


Figure 3.36: Lamellar dome

A well-known built example is the roof structure of Bruno Taut's famous glass pavilion (Figure 3.37) at the 1914 Cologne Werkbund exhibition which was realised using the asymptotic tesselation approach to determine the geometry for the planar glazing sheets.

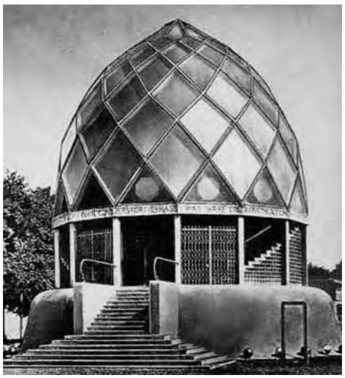


Figure 3.37: Bruno Taut Pavillion [Image by Wikipedia]

3.4.6 PQ meshes for surfaces of revolution with constant mean curvature

A discrete version of the unduloid can be generated using the principle as described in 3.4.4. For the nodoid however there is an additional geometric construction procedure to generate a PQ mesh. This has been proposed by T. Hoffmann in [Ho00]. The required steps are explained below:

First we need to define an ellipse E and choose a starting point Q_0 on this ellipse. Beginning in Q_0 we will subdivide E by performing the following steps: In Q_0 we will define a starting direction/direction vector v_d which is pointing to the inside of E. The vector v_d will hit E in a second point called Q_1 This process can be repeated for a pre-defined number of iterations whilst considering the standard law of reflection (outcoming angle equals incoming angle) and will result in a so called standard billiard (Figure 3.38 featuring a number of vertices on E called Q_0 - Q_x. Due to the 'reflection rule' and depending on the number of iterations the vertices may lie on E in a seemingly non-consecutive order. However they will be named consecutively (Q_0 Q_1 - Q_{x-1}, Q_x) as their order plays a vital part for the following construction steps. Obviously for a larger number of iterations the polygonal lines of the billiard may intersect each other.

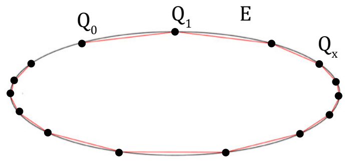


Figure 3.41: Standard billiard based on ellipse E

This procedure results in a number of vertices on E which if connected with one of the focal points F_1 or F_2 of E deliver a series of triangles within E (Figure 3.39). It is important to note that in focal point F there lies a whole series of points F_0 - F_x each of which form one vertex of the newly created triangles. Again one edge of each triangle forms one segment of the boundary polygon which was generated by means of the billiard 'procedure' whereas the other two edges lie within the ellipse.

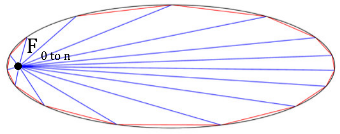


Figure 3.38: Triangles

The boundary polygone can now be unrolled into a straight line l. Those vertices of the triangles which previously coincided with one of the focal points of E will form a new polyline g with vertices F_0 - F_x (Figure 3.40). If we rotate this polyline about its straight axis l, it will result in a discrete version of a nodoid (Figure 3.41). If we finally substitute the smooth rotation of the polyline by a discrete rotation with pre-defined angles we will obtain a PQ mesh. Depending on the number of iterations described for the 'billiard' the resulting discrete nodoid will consist of one, two or even more sphere-like volumes.

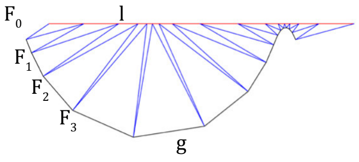


Figure 3.39: Unrolled triangles

3.5 Envelope surfaces

Envelope surfaces in general are defined by the spatial movement of a one parameter family of surfaces. For those envelope surfaces however which feature an inherent PQ mesh the generating surfaces are always one parameter families of spheres S. If the movement of the spheres is discretised (we can stop the moving sphere in certain points) they touch the envelope surface in a circle C_x (Figure 3.42) which is contained in S. This property forms the basis of the PQ mesh parametrisation which will be described later on.

The sub-class of envelope surfaces generated by one parameter families of spheres is called canal surfaces. In the following we will investigate canal surfaces as the only representatives of envelope surfaces with an inherent PQ mesh.

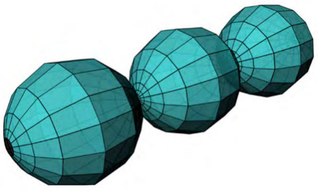


Figure 3.40: Discrete nodoid

3.5.1 Canal surfaces

Canal surfaces are described by the movement of a one parameter family of spheres S_x along a spatial curve c (Figure 3.43). These generating spheres can feature varying diameter while moving along c meaning they are non-congruent.
The spine curve c is defined by the locus of the midpoints of spheres S_x which inscribe the canal surface. The surface itself wraps around these spheres and touches them along the circles C_x each of which is usually one of the small circles of the individual sphere. (The particular case of the surface touching the spheres in the great circles results in a pipe surface). The supporting plane of each circle is at the same time the normal plane n to the spine curve c in the intersection between sphere S_x and c. Canal surfaces exist as open and closed surfaces depending on the spine curve being an open or closed curve (e.g. catenary or circle). An interesting example of a closed canal surface is the *cyclide* or *Dupin cyclide (Pierre Charles François Dupin, 1784 – 1873).* The cyclide (Figure 3.44) is generated by the movement of a number of spheres with varying diameters along a closed circular path curve. Another description of the cyclide is given by the envelope of all spheres touching three given fixed spheres. Each of the fixed spheres is to be touched in an assigned manner, either externally or internally (Figure 3.45). This geometric configuraton is also known as the 3D generalization of the Steiner Chain: The Soddys hexelet. The cyclide can be described in the following form:

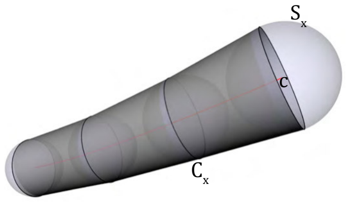


Figure 3.45: Discrete canal surface with generating spheres of varying diameters

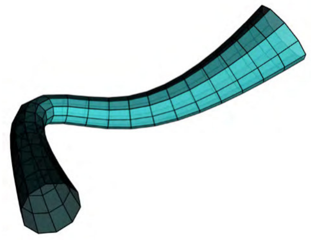


Figure 3.42: Discrete canal surface with open spine curve

$$x[u,v] = (\mu(c - a\cos(u)\cos(v)+b^2\cos(u))/(a- c\cos(u)\cos(v))$$
$$y[u,v] = (b\sin(u) \, x \, (a - \mu\cos(v))/(a - c\cos(u)\cos(v))$$
$$z[u,v] = (b\sin(v) \, x \, (c\cos(u) - \mu)/(a - c\cos(u)\cos(v))$$

Equation 3.16

a, c ,μ...constants with $a^2 > c^2$
$c = \sqrt{(a^2 - c^2)}$
$[u,v] \in [0 ; 2\pi]^2$

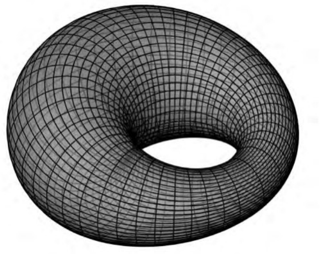


Figure 3.43: Closed canal surface - Dupin cyclide

For this surface type the lines of curvature are all circular arcs or straight lines (which can be defined as arcs with an infinite centre). The sphere which is tangent to the surface at a point and comprises equal radii to the reciprocals of the principal curvature line at this point is called the curvature sphere. For a cyclide the curvature spheres have constant diameter along a principal curvature line and contain the corresponding curvature line as a great circle.

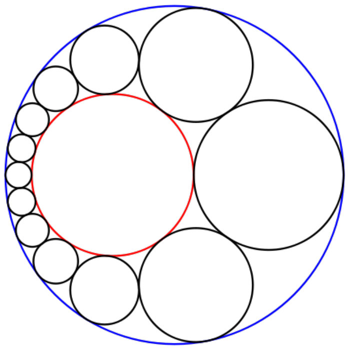


Figure 3.44: Steiner Chain [Image by Wikipedia]

3.5.2 PQ meshes on canal surfaces

A canal surface m may be discretised in an analogous way to the one described for rotational surfaces in 3.4.4. namely by isolating the defining curves of the surface whose intersection will form a PQ mesh: We start with isolating the spine curve s which is defined by the midpoints of the moving spheres that inscribe the canal surface (these are non-congruent spheres with varying diameters). The spine curve will be substitute by a polyline with a pre-defined number of vertices and in each vertex the respective normal plane n will be generated. The intersection of normal planes with the canal surface results in a number of circles on m which are orthogonal to s. We obtain circles which are non-congruent as they result from spheres with varying diameters.

The circles are discretised thus each delivering a number of points. When connected with the points of the adjacent circles by straight line segments this procedure will form the PQ mesh on a canal surface. Depending on the density of the circles and the number of pre-defined vertices on them the PQ mesh might feature a high or low facet density.

There is a further approach to discretise a canal surface namely by means of a series of right circular (truncated) cones which are tangent to a sequence of non-congruent space spheres which have their centres on the spine curve c. Each cone circumscribes two consecutive spheres of different diameters and intersects with the adjacent cone in an ellipse. The result is a chain of truncated cones which form a strip model of the canal surface whose edges are described by the intersection curves (ellipses) of the cones. If the truncated cones are discretised (the ellipse as profile curve of the cone will be substituted by a polyline) we will obtain a PQ mesh with facets that are orthogonal and circular meaning the PQ mesh consists of square planar facets. These properties have been utilised to develop a modelling tool which uses cyclidic patches to design PQ meshes (explained in detail in 3.9
).

If we use non-congruent spheres instead the same procedure will lead to a canal surface and respectively its discrete version.

3.5.3 Pipe surface

The pipe surface is defined as a one parameter family of congruent spheres moving along a space curve c called the spine curve or central curve with the centres of the spheres being located on c. As an alternative generating principle a family of circles with radius r. These circles must be contained within the normal planes of the spine curve with their centre points being located on the spine curve. Pipe surfaces exist as open and closed surfaces depending on the spine curve being an open or closed curve. However, there are special cases of generating curves resulting in special surface types: In the simplest case the pipe surface is a *rotational circular cylinder* with the central curve being a straight line segment. If we choose a circle as the spine curve the resulting pipe surface will be a ring *torus.* Parametric definitions for both objects are given below:

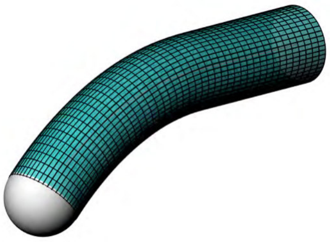


Figure 3.46: Discrete pipe surface

General pipe surface [Kr59]

$$G[x_i,s] = \{x - y(s)\}\,\{x - y(s)\} - r^2 = 0 \qquad \text{Equation 3.17}$$

$y(s)$...spine curve
r...radius of sphere

Torus [MW05]

$$x[u,v] = (c + a\cos(v))\cos(u)$$
$$y[u,v] = (c + a\cos(v))\sin(u)$$
$$z[u,v] = a\sin(v) \qquad \text{Equation 3.18}$$

for $u,v \in [0,2\pi]$,
$c > a$: ring torus, $c = a$: spindle torus, $c < a$: spindle torus

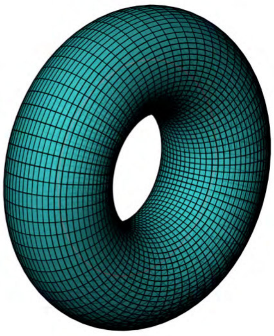


Figure 3.48: Discrete Torus

3.5.4 PQ meshes on the pipe surface

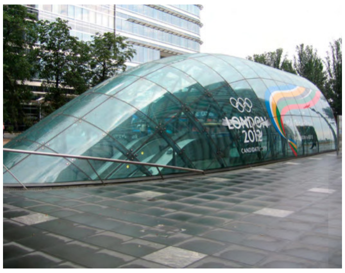


Figure 3.47: Entrance of canary wharf tube station by Foster & Partners [Image by Foster & Partners]

A pipe surface m may be discretised in an analogous way to the one described for the general type of a canal surface namely by isolating the defining curves of the surface whose intersection will form a PQ mesh: First we isolate the spine curve s which is defined by the midpoints of the moving spheres that generated the pipe surface in the first place. This curve will be substitute by a polyline with a pre-defined number of vertices. In each vertex of the spine curve we generate the normal plane n and intersect these planes with the pipe surface m. This way we obtain a number of circles on m which are orthogonal to s.
If we now discretise the circles we obtain a number of points on each circle which represent the vertices of a PQ mesh on the pipe surface. Depending on the density of the circles and the number of pre-defined vertices on them the PQ mesh might feature a high or low facet density.
Another approach to discretise a pipe surface is by means of a series of rotational cylinders which are tangent to a sequence of congruent space spheres whose centres are on the spine curve. Each cylinder circumscribes two consecutive spheres and intersects with the adjacent cylinder in an ellipse. The cylinders form a strip model of the pipe surface whose edges are formed by the intersection ellipses of the cylinders. The cylinders are discretised by substituting the elliptical profile curve by a polyline of a certain density. This will deliver a PQ mesh which features the special properties of being orthogonal and circular.

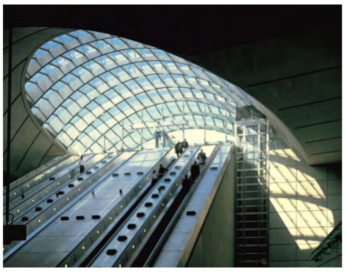


Figure 3.49: Entrance of canary wharf tube station by Foster & Partners [Image by Foster & Partners]

3.6 Spatial motion surfaces

In general spatial motion surfaces are generated by applying a spatial motion to a planar or 3D curved profile curve. Depending on the way the profile curve is moved a number of different subclasses of the spatial motion surface can be created each featuring interesting properties. Some

of the previously described surfaces could as well be classified as spatial motion surfaces (e.g. an envelope surface generated by the spatial motion of a sphere along a path curve) but have been assigned to a surface class according to their most obvious generating feature.
In this section we want to concentrate on spatial motion surfaces which are generated by moving a profile curve along a path curve. Out of these the moulding surface which is a special sweeping surface and the translation surface are of particular interest as these are the only two spatial motion surfaces which feature an inherent PQ mesh and hence will be discussed in detail in the following.

3.6.1 Moulding surface

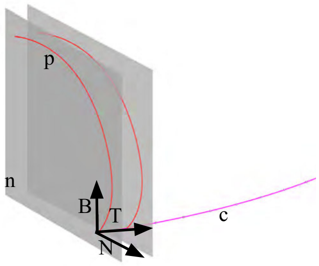


Figure 3.50: Generation of a moulding surface

The moulding surface was first proposed by C. Monge *(1746 - 1818).* It is a special sweeping surface generated by the movement of a profile curve p along a planar path curve c in such a way that p always lies in the local normal plane n of the *Frenet-Serret frame* of c (Figure 3.50). The Frenet–Serret frame of a curve c is defined by the three unit vectors of c which are the tangent, normal, and binormal unit vectors of the curve in a point P, also called T, N, and B.
T is the unit vector tangent to the curve, pointing in the direction of motion. N is the vector orthogonal to T which is contained in the osculating plane of c and B is the cross product of T and N.
Because of the Frenet-Serret frame condition the profile curve when moved along the path curve will always be orthogonal to it. The movement of p defines a moulding surface m which if intersected with planes parallel to the supporting plane of the path curve will result in section curves congruent to the path curve. These curves can be seen as the trajectories of the motion of p.

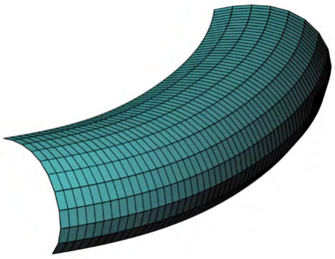


Figure 3.51: Discrete moulding surface

3.6.2 PQ meshes on the moulding surface

A moulding surface can be discretised by applying a combined translation and rotation to a substituted profile curve (Figure 3.51): first both profile curve and planar path curve are substituted by a polyline with a pre-defined number of vertices. The discrete profile curve p_1 may then be translated along the path curve c_1 from vertex to vertex and rotated into the local normal plane of the respective vertex of c_1 thus featuring a rotation during the translation of c. This procedure delivers an array of vertices which, if connected by straight line segments in a consecutive manner, results in a PQ mesh with faces that feature the special porperties of orthogonality and congruency.

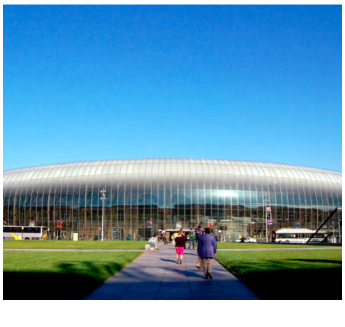


Figure 3.52: Strassbourg station [Image by Seele]

3.6.3 Translation surface

The spatial motion of a profile curve also called *generatrix* along a path curve also called *directrix* will generate the translation surface (Figure

3.53). In geometry a translation is defined as the transformation of an object by a constant offset without rotation or distortion. Any pair of arbitrary spatial or planar curves can be used to perform this operation. As the motion of the generatrix curve is only a translational operation all translated curves isolated at discrete points of the directrix are parallel to the original generatrix. It is a matter of definition which of the two curves will be defined as the profile curve and respectively as the path curve. The resulting surface will be the same. For this reason the parallelism condition is likewise valid for the directrix. This means the translation surface carries two sets of congruent parameter curves: one set based on the generatrix and the other one based on the directrix. The tangents along a generating curve envelop(e) a cylindrical strip which obviously touches the translation surface tangentially. Translation surfaces may be described in the following form:

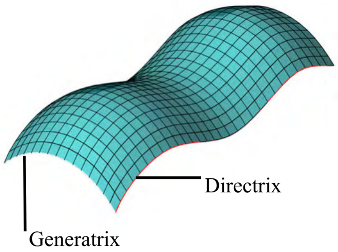


Figure 3.53: General translation surface

$$S\,[u,v] = g(u) + d(v) \qquad \text{Equation 3.19}$$

$g(u)$..directrix curve
$d(v)$..genratrix curve

Although all translation surfaces irrespective of their generating elements posses an intrinsic PQ mesh surfaces generated by planar generatrix and directrix curves will indicate the final shape in a better way than those surfaces generated with spatial curves. If the generating curves are orthogonal to each other this will have an additional positive influence on the surface shape.
If the generating curves belong to the class of *isotropic* curves the translation surface belongs to the class of minimal surfaces [Kr59]. For a definition of isotropic curves please refer to [Kr59]. Scherk's first surface *(Heinrich Ferdinand Scherk, 1785- 1885)* is an example for a minimal surface which can be generated as a translation surface (Figure 3.55). The planar profile curves can be described in the following form:

Figure 3.54: Grid shell hippo house Berlin [Image by Schlaich Bergermann & Partner]

$$\begin{aligned}&\text{Generartix: } x = 0, z = \ln(\cos y)\\ &\text{Directrix: } y = 0, z = \text{-}\ln(\cos x)\end{aligned} \qquad \text{Equation 3.20}$$

There is also a set of rotational surfaces which can be generated as translation surfaces. The *elliptic paraboloid* results from a rotational paraboloid if an independent scaling is applied. It can however be generated as a translation surface by translating two parabolas along one another which must be open to the same side and have parallel axes (Figure 3.56). The elliptic paraboloid parametrised as a translation surface is defined as such:

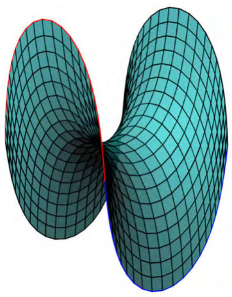
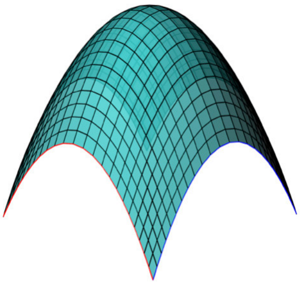


Figure 3.55: Sherk's first minimal surface

$$\begin{aligned}&\text{Generartix: } x = 0, z = a\,y^2\\ &\text{Directrix: } y = 0, z = a\,x^2\end{aligned} \qquad \text{Equation 3.21}$$

Another example for a rotational surface that alternatively may be generated as translation surface is the *hyperbolic paraboloid* (HP surface) (Figure 3.57). In contrast to the elliptic paraboloid the HP surface may be generated by two parabolas which need to be open to opposing sides:



Generartix: $x = 0, z = a\,y^2$
Directrix: $y = 0, z = -a\,x^2$ Equation 3.22

Figure 3.58: Elliptic paraboloid

If we apply a local centric scaling to the generatrix in course of the motion with the centres of scaling being located on a continuous curve we obtain a *scaled translation surface* . The scaling curve and the scaling factors can be chosen freely. However if extreme scaling factors are chosen the shape of the resulting surface is hard to predict and facets may be distorted (Figure 3.58). As discussed in chapter 2.7.1 the scaling transformation preserves planarity of the facets.
The translation can also be performed using a closed curve in combination with a central scaling curve in vertical direction. This will lead to a surface which is similar to a moulding surface. The fundamental difference between these two consists in the orientation of the sections of supporting plane with path curve. For the translation surface the vertical sections are all oriented towards the centre of expansion in comparison to the moulding surface for which the sections are always perpendicular to the path curve.
There is a large variation of surfaces available when using the translation surface approach as H. Schober demonstrated in [Scho02] and [GlSh04].

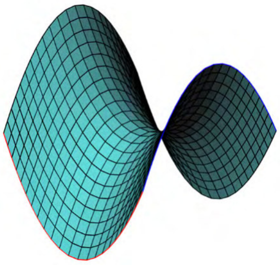


Figure 3.56: Hyperbolic paraboloid

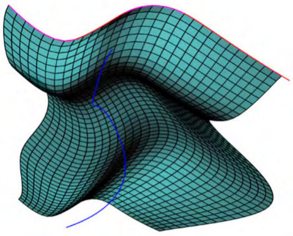


Figure 3.57: Scaled translation surface

3.6.4 PQ meshes on translation surfaces

The PQ mesh of translation surfaces is so to speak built into their generation principle. Both the generatrix and directrix curve may be substituted by polylines with a chord length that can be chosen freely depending on the envisaged mesh density. Instead of the smooth translation as applied for the surface generation the discrete translation of the generating polyline from one vertex to the next of the directing polyline will deliver a mesh of polylines with each segment of a polyline defining an edge of a PQ facet. As each set of polylines are parallel in space the facets are all planar. We have now discussed a number of different surface types. This compilation of generation principles for classical surface and their respective intrinsic PQ mesh parametrisations provides a good overview of PQ meshes ‘readily at hand’ if one is prepared to subordinate the form finding process for a surface to the available geometrical principles. The list below shall summarise all surface types that have been documented as yet:

Developable surfaces
-cylinder
-cone

-tangent surfaces of space curves

Extrusion surfaces

-cylinder

-cone

Surfaces of revolution

-general surfaces of revolution

- Surfaces of revolution with constant Gaussian curvature (spherical and pseudospherical surfaces)
- Surfaces of revolution with vanishing mean curvature (catenoid - minimal surface)
- Surfaces of revolution with constant nonzero mean curvature (nodoid, unduloid)
- Surfaces of revolution with positive Gaussian curvature

Envelope surfaces

- Pipe surfaces (general pipe surface, torus)
- Canal surfaces (general canal surface, cyclide)

Spatial motion surfaces

- Moulding surfaces
- Translation surfaces (general and scaled)

For a general overview of all PQ,mesh surfaces please refer to Appendix A 10.

So far the system how we classified the documented surfaces was according to their generation method (rotational surfaces, translation surfaces and so forth).
In the following we will deviate from this approach to introduce two surfaces which play a special role with respect to their discretisation potential independent from their affiliation with certain generation principles.

The sphere and the cyclide both offer a whole variety of possible PQ mesh parametrisations which all feature certain particularities. Most importantly however the meshes generated on the sphere/them can be tranforrmed by applying a number of transformation principles thus resulting in mesh surfaces with interesting shapes and still featuring the desired PQ tessellation. From the appearance of these surfaces one would not assume a relationship with the sphere or cyclide although they are generated by transformation processes applied on one of those objects.

Therefore the documentation of these two surfaces is handled differently: following a brief definition of the surface itself we will examine the different available discretisation principles and possible transformations of them.

3.7 PQ meshes on the sphere and their duals

In this chapter we will investigate the possible PQ tesselations on a sphere and their dualisations.
A number of different PQ meshes can be generated on the sphere depending on the applied generation principle:
An equal length mesh on the sphere can be inverted to a PQ mesh or transformed to the discrete version of a deformed pseudospherical surface. Circular meshes on the sphere can be transformed to discrete versions of minimal surfaces.

The generating principles of the following PQ mesh types shall be described in detail:

- Diagonal inversion of an EQL mesh
- Discrete K-surface from an EQL mesh
- Discrete minimal surface from circle packings

Before we start looking into these PQ meshes however we want to introduce the unit sphere as a geometrical object and discuss its properties.

3.7.1 The sphere

The sphere is generated by a 180º rotation of a circle about an axis A which must contain the centre of the circle. The sphere may be described in the following form:

$$S\,[u,v] = [\mathrm{r}\cos(v)\cos(u), \mathrm{r}\cos(v)\sin(u), \mathrm{r}\sin(v)] \quad \text{Equation 3.23}$$

<u>Some particularities of the sphere</u>

All points on a sphere feature constant distance r from the centre of the generating circle.
The meridians and parallel circles on the sphere coincide with the *geodesic lines* (curves of shortest path). The sphere has constant positive Gaussian curvature ($\mathrm{K} > 0$) and constant positive mean curvature ($\mathrm{H} > 0$). All points on the sphere are *umbilic points*. Hence the *principal curvature lines* are undefined on the sphere.

In addition to the general PQ mesh parameterisation for rotational surfaces described in 3.4.4 to 3.4.5 (which is applicable to the sphere as a special rotational surface) less obvious PQ mesh layouts can be generated on the sphere. These are of particular interest as there are a number of transformation principles available which change the 'common' shape of a sphere into completely different surfaces whilst keeping the properties of the PQ meshes generated on the sphere. These principles shall be described in the following.

3.7.2 Diagonal 'inversion' of an equal length mesh

An equal length (EQL) mesh or Chebychev net *(Pafnuty Lvovich Chebyshev, 1821-1894)* can be generated on any arbitrary surface and forms a quadrilateral mesh for which the faces all have constant edge length and are parallelograms.

On the sphere S an EQL mesh forms a set of parameter lines which are at the same time *asymptotic lines* (curves of zero normal curvature) of S. If a plane is osculated along an asymptotic line this plane coincides with the surface tangent planes of the sphere at every point of the asymptotic line a. The osculating plane is spanned by the tangent vector T and normal vector N of a point P on a. The cross product of T and N (resulting in B) is hence the normal vector of the sphere in point P.
When using the asymptotic coordinates u,v the EQL mesh can be defined in the following partial differential equation [Pi08]:

$$N \times N_{uv} = 0$$ Equation 3.24

If we substitute the coordinates u,v to:

$$x = u + v$$
$$t = u - v$$ Equation 3.25

Equation 3.24 becomes

$$N \times (N_{xx} - N_{tt}) = 0$$ Equation 3.26

We receive the asymptotic lines of the surface if we set t = const. The EQL mesh is completely determined for any given starting configuration (edge length and direction).
Two approaches to generate a EQL mesh on the sphere have been proposed in [Pi08]:

I. EQL mesh with a fixed point

We choose a point S on the sphere and define a small circle C on the sphere which is parallel to the tangent plane in S (Figure 3.60). The distance from point S to an arbitrary point on the chosen circle will set the edge length of the entire EQL mesh. Then we determine the subdivision of C (number of vertices V_1 - V_x on the circle and their distances) which will determine the layout of the EQL mesh.

When we connect point S with each of the new vertices V_1 - V_x we receive two edges of each facet in the first row of the EQL. With now three corner vertices of one facet defined (one being fixed point S and two

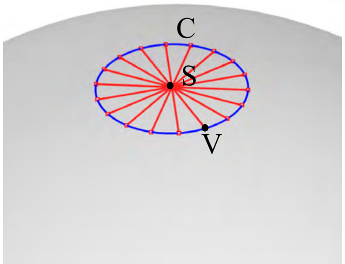


Figure 3.59: Start circle

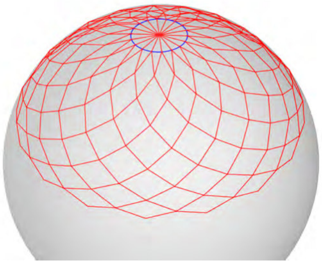


Figure 3.60: EQL mesh with a fixed point

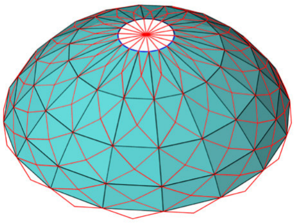


Figure 3.61: PQ faces from a uniform EQL mesh

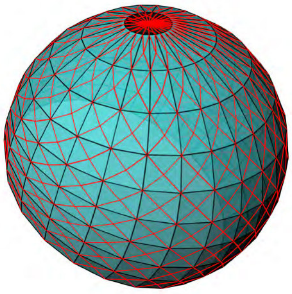


Figure 3.62: PQ mesh non - uniform EQL mesh

consecutive subdivsion vertices on the first parallel circle V_{x-1}, V_x) we can construct the fourth vertex for each facet of the first row which obviously lies on the next (second) parallel circle:
the location of vertices on the second circle is shifted precisely by half the distance of two consecutive vertices of circle 1.
the distance between parallel circle 2 and parallel circle 1 is determined by the edge length of the mesh.
By repeating these two construction steps for a given number of iterations using the constant edge length property the sphere will be covered with facets row by row. If required it can be covered with multiple overlaying meshes.
If a uniform subdivision is chosen all vertices generated within one step lie on a common parallel circle and intersect with a set of meridians (Figures 3.60 + 3.61). A non-uniform subdivision generates a more distorted graph but can still be inverted to a PQ mesh (Figure 3.62). When setting up the EQL one has to consider that if the sum of all edges within one row is less then the circumference of the sphere the EQL mesh will reverse at some point (when the sum of all edges of a row is less than the respective parallel circle) and thus will not cover the whole sphere.
One particularity of this method is that the construction of the mesh can start in second row earliest as the first row quads are collapsed towards the fixed point.

II. EQL mesh from a closed smooth curve

Starting geometry for this construction principle is a smooth closed curve c on the sphere S whose length must be shorter than the circumference of S. In theory c can be located anywhere on S although a location somewhere close to one of the poles results in a clearer EQL. We define a subdivision of c (number of vertices V_1 - V_x and their distances) and the edge length of the EQL. Starting with two consecutive vertices V_{x-1} and V_x on c we can construct a third point of the first facet. If repeated for all vertices on c we will obtain a starting geometry for the EQL mesh consisting of the first row of 'half parallelograms' each consisting of three corner vertices (Figure 3.64). This procedure will be repeated row by row until the complete sphere is covered with parallelograms which form the EQL mesh. Only the small area of the sphere encircled by the starting curve will not be covered by the mesh. This area however can be neglected as the transformation of the EQL mesh will still result in a continuous PQ mesh without any gaps (Figure 3.64).

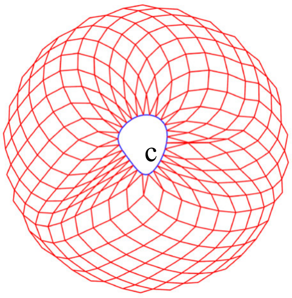


Figure 3.63: EQL mesh with a cone point

Both construction principles result in a mesh with rhombic non-planar facets all featuring the same edge length hence the name equal length mesh. The diagonal 'inversion' of the EQL mesh finally forms a PQ mesh on the sphere. This means we use the vertices defined by the EQL mesh but connect them with straight line segments according to the direction of parallel circles and meridians. This results in a mesh which looses the equal length property but gains the planarity for all facets.
For every EQL mesh on the sphere we can generate two different PQ meshes by using only every other vertex of the EQL mesh for the con-

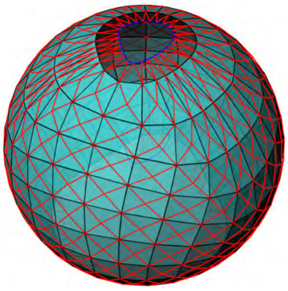


Figure 3.64: PQ mesh with a cone point

struction of the PQ mesh. Hence for PQ mesh A the construction may start in an odd row of the EQL mesh and for PQ mesh B respectively in an even row. Both meshes obviously feature less density than the mesh which uses all vertices and consequently the applicability of this approach depends on the desired facet size and on the chosen edge length and subdivision of curve c.

A PQ mesh face on the sphere can be constructed when applying the following rule:

$$F = [f_{n,m+1}; f_{n+1,m}; f_{n,m-1}; f_{n-1,m}] \qquad \text{Equation 3.27}$$

f... vertices of the EQL mesh on the sphere

For notation of the EQL vertices please refer to figure 3.65. The two methods described above result either in a PQ mesh on the sphere by inverting the EQL mesh or alternatively the EQL mesh can be used as a starting geometry for a transformation process which results in discrete surfaces featuring constant negative Gaussian curvature (K=-1) called K-surfaces.

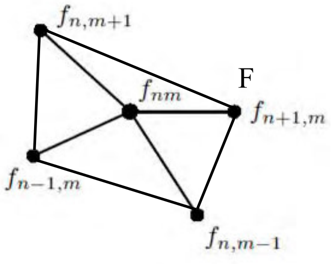


Figure 3.65: Notation of EQL mesh vertices [Image by S. Sechelmann]

3.7.3 Discrete K-surface from an EQL mesh on the sphere

The K-Surface is a *surface* which comprises of constant negative Gaussian curvature (K=-1). In chapter 3.4.1 we have already discussed revolving surfaces with constant negative Gaussian curvature including the pseudosphere. K-surfaces however represent all surfaces with K=-1 meaning also those which cannot be generated by a revolving generation process.
The method proposed by U. Pinkall [Pi08] will allow us to generate a K-surface by means of the Gauss map: a K- surface is the Gauss image of an EQL mesh on the sphere. In other words when we apply the Gauss map to a EQL mesh on the sphere we obtain a discrete K-surface. If the EQL mesh is deformed this will result in an accordantly deformed discrete K-surface. As the Gauss map preserves planarity of the faces the resulting mesh will still be a PQ mesh however featuring a completely different shape than the EQL mesh on the sphere.
Discrete K- surfaces possess the following properties:

I. All vertices surrounding one vertex lie in a plane

This condition guarantees the planarity of a quad face defined by the end vertices of those edges which all meet in the 'centre' vertex. This fact is of particular importance as the PQ mesh of the K-surface is obtained by inverting the K-mesh (discrete K-surface) and using exactly this geometrical constellation for the mesh faces.
The four vertices surrounding a 'centre vertex' also lie on a circumcircle

with a radius equal to the projection of the 'equal length edge' onto the circle plane. This means the K-surface is a circular quadrilateral mesh. As we know from 2.6.2 a mesh forms a discrete version of an isothermic surface if all quadrilaterals of a circular mesh are *conformal* . Hence the K-surface is an *isothermic surface*.

II. All edges have the same length

The second condition allows us to construct the discrete K-surface on the sphere as an EQL mesh (Chebyshev net) which will be dualised by means of the Gauss map into its Gauss image which is the K-Surface. The EQL mesh forms a network of asymptotic curves.
In the next chapter we want to introduce the Gauss map and resulting Gauss image to clearly illustrate the construction of a K-surface by means of an EQL mesh on the unit sphere.

3.7.4 Computation of the Gauss image

In the field of differential geometry, a surface in Euclidean space can be mapped onto the unit sphere by means of the Gauss map which was introduced by Carl F. Gauss in 1825. This means for every point on an surface a corresponding point on the unit sphere exists which can be defined using the function of the Gauss map. Globally this is restricted to surfaces which are orientable whereas locally the Gauss map can be defined for any arbitrary surface.
The Gauss image is the image which is generated on the unit sphere when applying the Gauss map to a surface.
If we invert this method we start with an EQL on the sphere using one of the starting conditions defined in subchapter 3.7.2 (fixed point or closed smooth curve) which delivers an EQL mesh. We can then apply the Gauss map to the EQL mesh to dualise it into its Gauss image namely the respective K-surface.
This K-surface is in itself an equal length mesh which can be easily transformed into a PQ mesh by applying the 'inversion method' as described in section 3.7.2.
This method delivers PQ meshes of very distinct appearance (Figure 3.66). Still there may be a restricted number of problems in the construction business for which this method could provide a solution.
The geometric construction of a discrete K-surface from an EQL mesh on the sphere will be described in the following paragraph.

3.7.5 Reconstruction of a discrete K-surface

We can reconstruct the faces of a discrete K-surface from an EQL mesh by applying the following construction rule (difference equation):

$$f_{n+1,m} = f_{n,m} + [N_{n+1,m} \times N_{n,m}]$$
$$f_{n,m+1} = f_{n,m} + [N_{n,m} \times N_{n,m+1}] \quad \text{Equation 3.28}$$

$f_{n,m}$... corresponding vertices on the sphere and the discrete K-surface
N... normal vector of the plane defined by the vertex and its four neighbours
N_n x N_m... cross product of the two Normal vectors

For notations please refer to figure 3.65

The construction can be described in the following steps:

> *-Choose the start vertex in the second row (or higher) on the EQL mesh on the sphere. (To define the normal vector of a vertex which is required later on we need the adjacent vertices in the following and previous rows hence this wouldn't be possible for row 1)*
> *-Define the start point of the K–surface (which should have a certain distance from the start vertex on the sphere).*
> *-Compute the normal vectors for all vertices of the EQL mesh (as explained in equation 3.28)*
> *-Choose a fixed scale value for the normal vectors or use non-normalised vectors for the construction.*
> *-With the starting values defined we can construct the K-surface vertices successively performing vector additions following the rules given in Equation 3.28.*

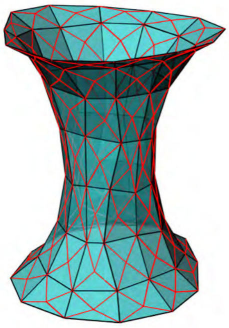


Figure 3.66: K-Surface with a cone point

This method delivers a discrete K-surface from an EQL mesh on the sphere which finally can be inverted into a PQ mesh as described in section 3.7.2.

Aside from the EQL mesh there is another way to generate PQ meshes based on the sphere. This method requires the generation of a circle packing on the sphere which is then being dualised to result in the discrete version of a minimal surface.

The basic principle is similar to the approach described previously: the geometrical object of a sphere is being used to generate a PQ mesh which can then be transformed into the discrete version of a surface whose shape is completely different from that one of a sphere but still featuring the PQ mesh properties.

3.8 Discrete minimal surfaces from circle packings on the sphere

Minimal surfaces are defined as surfaces featuring vanishing mean curvature. The special type of minimal surfaces of revolution and their inherent PQ meshes have already been discussed in section 3.4.2. In this section however we want to introduce a method to generate minimal surfaces with an inherent PQ mesh by transforming a mesh which is initially generated on a sphere.

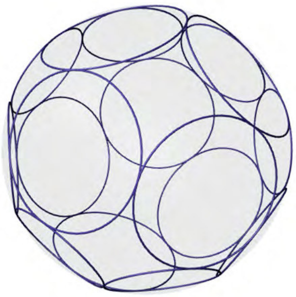


Figure 3.67*: Face circle packing on sphere

In general minimal surfaces belong to the class of free form surfaces and hence do not feature an inherent PQ mesh to their generation principle. However there is a special method how to create discrete minimal surfaces which automatically results in a PQ mesh representation. This method is harvesting the knowledge about a number of well-researched and documented mathematical laws and theorems of differential geometry each of which in itself has no specific use with respect to PQ meshes. By integrating these singular principles into a process which ultimately leads to a PQ tesselation of a minimal surface they play a vital role in the PQ mesh generation process:

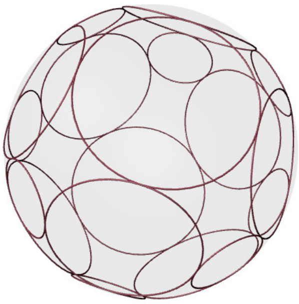


Figure 3.68*: Vertex circle packing on sphere

I In mathematics it is known that a minimal surface (or its discrete version) can always be mapped conformally onto the sphere which means the mapping preserves angles and planarity of faces of the input mesh. In reverse a mesh generated on the sphere can be transformed conformally to a minimal surface again preserving angles and planarity of the initial mesh. Consequently the discrete version of the minimal surface will feature PQ facets if the mesh generated on the sphere is a PQ mesh.

II In mathematics it is also known that there exists a unique circle packing for any arbitrary triangular mesh on the sphere. For every circle packing on the sphere a Koebe polyhedron can be generated which can therefore be seen as the discrete version of a sphere.

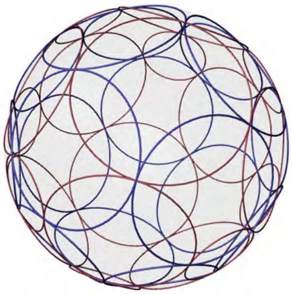


Figure 3.69*: Vertex/Face circle packing on sphere

The Koebe Theorem provides us with the knowledge how to discretise a sphere in a way that the output mesh is suitable for the subsequent transformation into a minimal surface:
If a sphere is discretised by means of triangulation the resulting mesh can be used to generate a circle packing on the sphere. The circle packing represents a Koebe polyhedron which is a discrete version of the sphere. Obviously there are different ways how to discretise a sphere. The Koebe polyhedron however features the special property that it can be transformed by means of the *Schwarz-Christoffel mapping* (Elwin Bruno Christoffel, 1829 - 1900 and Hermann Amandus Schwarz, 1843 - 1921) into a discrete minimal surface.
Our application however does not require any kind of discrete version of a surface but a planar quad tessellation. Consequently we need to find a way how to discretise the sphere so the Koebe polyhedron consists of planar quads. This PQ mesh can then be transformed to the discrete version of a minimal surface which still consists of PQ facets.

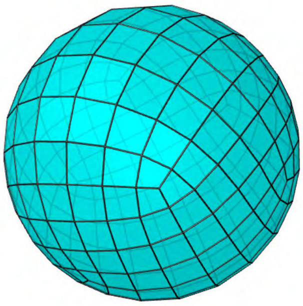


Figure 3.70*: Tessellation of sphere

The generation principle for circle packings on the sphere and the Schwarz-Christoffel mapping will be described in the following section. A method will be discribed how to generate a circle packing of such a type which will result in a PQ mesh on the sphere and consequently in a PQ mesh minimal surface after application of the Schwarz-Christoffel transformation.

3.8.1 The Koebe polyhedron

In general a circle packing is defined as the arrangement of circles within a pre-defined boundary. In our particular case this boundary is the surface of a sphere. The circles can be of equal or varying size but they have to be arranged in a way such that no overlapping occurs and all neighbouring circles are mutually tangent or in other words a circle packing is defined as a configuration of disjoint discs which may touch but do not intersect. According to William P. Thurston [Th97] tessellations of regular polygons correspond to particular circle packings. Hence the packing from a triangulation is different than those based on other tessellations. Circle packings exist in various configurations depending on the envisaged packing density and depending on the mesh they are based on. For our application the packing density is irrelevant however the circle arrangement is important as the special configuration of all neighbouring circles only touching in four points will result in a quad mesh. In general a PQ mesh face can be inscribed with or circumscribe by a circle (Figure 3.71).

Each circle c of the circle packing defines a plane which we call the supporting plane of c. If we intersect these planes the intersection lines will result in a mesh which at the same time describes a special type of polyhedron known as the *Koebe Polyhedron (*Paul Koebe, 1882 - 1945). A Koebe polyhedron is defined as a spherical polyhedron whose edges are all tangent to a sphere S. Hence it can be seen as a discrete version of S. It is obvious that the facets of the Koebe polyhedron are planar as they result from the intersection of a number of planes.
The quad property of the mesh results from the special configuration of the circles which form the circle packing. To understand the computation of such circle packings on the sphere which can be derived into a PQ mesh, we need to start with the Koebe theorem *(Paul Koebe, 1882 – 1945)*:

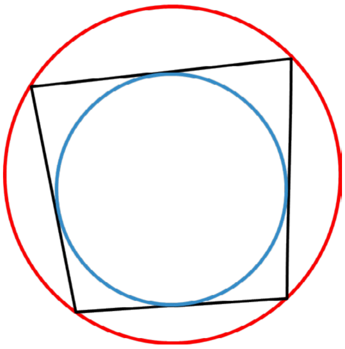


Figure 3.71: Inscribe/Circumscribed Face

The Koebe theorem:
For every triangulation of the sphere there is a packing of circles on the sphere such that circles correspond to vertices and two circles touch if and only if the corresponding vertices are adjacent. This circle pattern is unique up to Möbius transformations of the sphere.

A generalisation of Koebe's theorem has been proposed by A. I. Bobenko and A. Springborn in [BoSp02] which suggests that the faces corresponding to a circle packing on the sphere do not necessarily have to be triangular:

Theorem 1:
For every polytopal cellular decomposition of the sphere, there exists a pattern of circles on the sphere with the following properties. There is a circle corresponding to each face and to each vertex. The vertex circles form a packing with two circles touching if and only if the corresponding vertices are adjacent. Likewise, the face circles form a packing with circles touching if and only if the corresponding

faces are adjacent. For each edge, there is a pair of touching vertex circles and a pair of touching face circles. These pairs touch in the same point, intersecting each other orthogonally. This circle pattern is unique up to Möbius transformations.

In theorem 1 the triangulation condition is softened in order to allow the consideration of quad meshes as basis for a circle packing. Furthermore the existence of two circular meshes is proven, one corresponding to the edges and another one which corresponds to the vertices of a mesh. If these two types of circular packings are superimposed they always intersect orthogonally (Figure 3.72).
The two types of circular meshes are based on the fact that each circle carries a planar face, which is either inscribed with (edge corresponding) or circumscribed by a circle (vertex corresponding).
The Koebe Polyhedron which is generated from a circle packing and its properties are defined in the following Theorem 2 [BoSp02]:

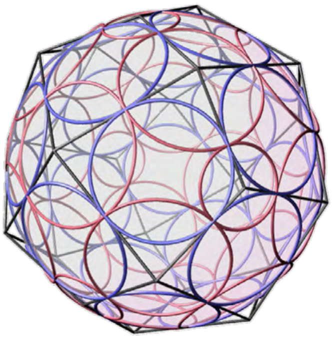


Figure 3.72*: Two types of circular Koebe meshes

Theorem 2:
For every polytopal cellular decomposition of the sphere, there is a combinatorially equivalent polyhedron with edges tangent to a sphere. This polyhedron is unique up to projective transformations which fix the sphere. There is a simultaneous realisation of the dual polyhedron, such that corresponding edges of the dual and the original polyhedron touch the sphere in the same points and intersect orthogonally.

As mentioned in the beginning a circle packing is the arrangement of circles within a defined boundary hence for each surface there might be a number of different circle packing configurations. If we only consider those circle packings for which each circle is touched in four points (meaning each circle has four tangent neighbouring circles) we receive a Koebe polyhedron consisting of PQ facets. Following the condition of the four touching points such a PQ mesh obviously is called to be *circular hence all vertices of each face lie on a common circle* (Chapter 2.6.1). Although the above stated theorems confirm the existence of such circle packings on the sphere their computation requires non-linear approximation. This means there is no 'construction rule' how to arrange the circles on the mesh. The packing configuration has to be determined by means of non-linear optimization which will be outlined in the following:

To reach convergence of a system under optimization we need to start with a *graph* which is combinatorial equivalent to the polytopal cellular decomposition of the sphere. The properties of such a system can be described as follows [Se07]:

-No edge loops are permitted
-No edges share the same vertices (no duplicates)
-At least three edges are meeting in each vertex

* Images generated with software 'Cyclidic Nets' by E. Huhnen-Venedey http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml

If the starting graph satisfies these conditions it is called to be *valid* in the sense that it will lead to a mesh from which the respective Koebe Polyhedron can be constructed.

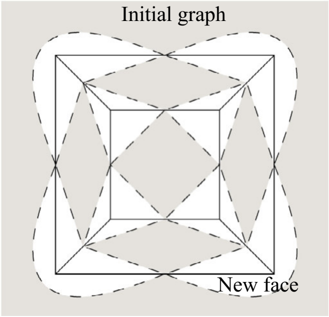


Figure 3.73: Medial combinatorics of the cube [Image by S. Sechelmann]

To receive a circle pattern which reflects the rules given in Theorem 1 we need to apply the *medial combinatorics* to the initial graph which leads to a *medial graph*. Each face of the medial graph corresponds to a face of the initial graph and the same applies for each vertex. The vertices of the medial graph are characterised by the number of four edges meeting in each point. The construction of the medial graph can be described as follows:

> *-Subdivision of the initial graph (the midpoints of each edge will generate a new set of vertices, Figure 3.73)*
> *-By connecting this set of vertices according to the topology of the initial graph a new face is generated*

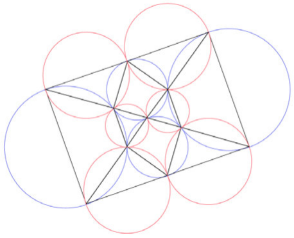


Figure 3.74: 2 D Circle pattern of the cube [Image by S. Sechelmann]

In order to receive a quad mesh we have to restrict the initial graph to consist of quad faces only. These quads can be subdivided to the medial graph whereupon the quad property of the faces is preserved. The subdivision can be refined iteratively until we receive the desired face density. The subdivided graph represents the quad mesh which the circle packing can be generated on (Figure 3.74). Once the packing configuration is generated the radii of the circles need to be approximated in order to fulfil the condition of orthogonal intersections amongst the two sets of circles. Solutions for this process have been proposed by K. Stephens [St05] and A. Springborn / A.I. Bobenko using non linear optimisation and variational principles which can be studied in [BoSp02, Sp03].

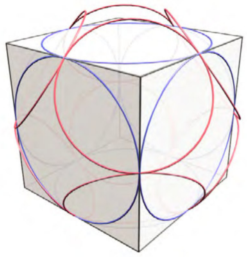


Figure 3.75: Stereopraphic projected circle pattern of cube [Image by S. Sechelmann]

Finally we receive a circle pattern which might be suitable for the Schwarz-Christoffel transformation but which is still in the Euclidean plane and therefore first has to be projected stereographically onto the sphere. As the *stereographic projection* preserves angles and circles, the orthogonal circle pattern will result in an orthogonal circle pattern on the sphere after projection (Figure 3.75).
As explained previously the edges of the Koebe Polyhedron result from the intersection of the supporting planes of the circle packing and form a PQ mesh on the sphere.

The construction process of this circular PQ mesh on the sphere can be summarised in the following steps:

> *-sketching of a planar graph consisting of quad faces with at least 3 edges meeting in every vertex of the graph. This graph has to be valid in a sense as described further above.*
> *-subdivision of the valid graph by connecting the midpoints of all edges to receive a medial graph*
> *-repetition of the subdivision steps to reach the required face density*

-generation of a circle packing based on the subdivided graph
-approximation of the radii of the circles in order to guarantee orthogonal intersections of the two sets of circles
-stereographic projection of the circle pattern onto the sphere
-computation of the Koebe Polyhedron which forms a PQ mesh on the sphere

With the sphere discretised by a PQ mesh based on a Koebe polyhedron finally the *Schwarz Christoffel mapping* can be applied to transform the PQ mesh into a discrete *minimal surface*.

3.8.2 Schwarz-Christoffel mapping of the Koebe polyhedron

As explained in 3.7.4 any arbitrary surface in Euclidean space can be mapped onto the unit sphere by means of the Gauss map resulting in the Gauss image of that surface on the sphere. This process is an involution and therefore a mesh constructed on the unit sphere in reverse may be mapped into a surface in Euclidean space. Depending on the applied transformation method the output surface may feature certain properties: in the particular case of applying the Schwarz-Christoffel mapping to a Koebe polyhedron the output surface namely the Gauss image of the Koebe polyhedron will feature vanishing Gaussian curvature. Hence the Schwarz-Christoffel mapping transforms a mesh on the unit sphere into the discrete version of a minimal surface in Euclidean space.

We know that this transformation process is conformal which proves that a PQ mesh on the sphere results in a PQ mesh representation of a minimal surface.
Minimal surfaces are examples for isothermic surfaces. Isothermic surfaces again are characterised by the fact that if they are discretised by a quad mesh this mesh necessarily consists of conformal squares. The two sets of edges of the conformal squares in consecutive order represent the discrete version of the two families of principal curvature lines of the minimal surface.
A square is called to be conformal (Figure 3.76) if the following condition for the edges is fulfilled:

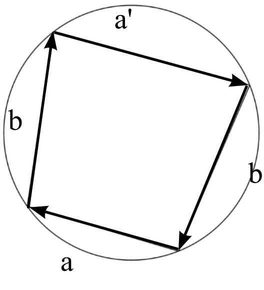


Figure 3.76: Conformal square [Image by S. Sechelmann]

$$aa'/bb' = -1$$ Equation 3.29

a,a',b,b'...edge vectors

Conformal squares feature circumscribed circles or spheres which intersect orthogonally at their points of contact (Figure 3.77). Ordinary vertices have four adjacent edges. In contrast vertices with the degree of three correspond to umbilic points on the surface. We need to know the location of the umbilic points on the surface in order to create the corresponding quad graph for a specific minimal surface as we have to cut the combinatorics along those curvature lines whose intersections represent the umbilic points. Usually the construction of a minimal surface may be performed in patches. Any patch has a number of boundary curves each of which is a curvature line of the surface. If the patches are joined together this will be done along the boundary curvature lines. Consequently adjacent patches have to feature the same boundary curves in order to receive a smooth surface.

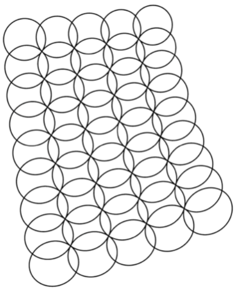


Figure 3.77: Confromal circular mesh

A construction method has been developed by Bobenko, Hoffmann and Springborn [BHSp05]:
We start with a Gauss image on the sphere which in our particular case is the PQ mesh generated from the according circle packing on the sphere. To receive the so-called dual surface from the Gauss image we apply the *Schwarz-Christoffel mapping* also known as *Schwarz-Christoffel transformation or Schwarz-Christoffel dual.*

Two planar quadrilaterals are called to be dual if the following conditions hold [BoSu08]:

Corresponding sides are parallel

(A*B*) || (AB), (B*C*) || (BC), (C*D*) || (CD), (D*A*) || (DA)

Non-corresponding diagonals are parallel:

(A*C*) || (BD), (B*D*) || (AC)

A, B, C, D ... orginal qudrilateral corner vertices
A*, B*, C*, D*.... dual quadrilateral corner vertices

The construction the dual from an original quad by means of the Schwarz-Christoffel mapping can be described in the following steps:

Given is a planar quad face defined by its four corner vertices A to D, the four edges a to d and the diagonal straight connection vectors AC and BD which connect A to C and B to D (Figure 3.78).

-Choose one of the four vertices of the quad (e.g. A) and translate it with a user defined distance to receive the first vertex of the dual quad (A)*
-Choose one of the edge which are adjacent to the starting vertex A (e.g. edge a)
-Reverse the vector of the chosen starting edge (originally the vector is pointing away from the start point towards the end point of the edge)
-Set a scale value for the construction
-Translate the reversed start vector to the start point A of the dual quad and multiply the vector by the scale value. This will define the first edge (a*) of the dual quad*

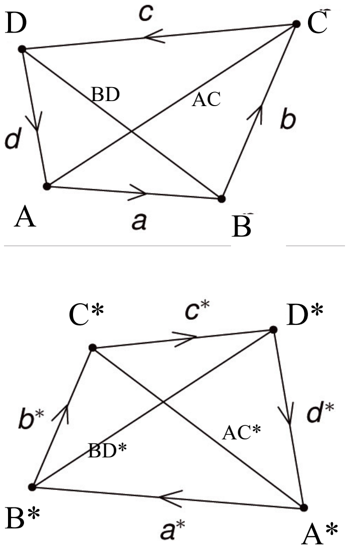


Figure 3.78: Dual quads [Images by S. Sechelmann]

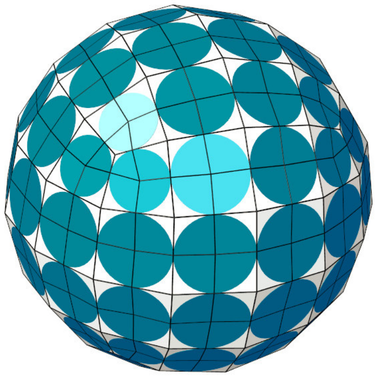


Figure 3.79: Koebe Quad graph [Image by S. Sechelmann]

- *Translate the edge vector b (in original direction) to corner point B* of the dual quad*
- *Translate the diagonal vector BD to A**
- *The third point (C*) of the dual quad will result from/lie at the intersection of edge vector b* and diagonal vector AC**
- *Translate the reversed edge vector c to point C* of the dual quad*
- *Translate the reversed edge vector d to point A* of the dual quad*
- *The last point D* will result from/lies at the intersection of c* and d**

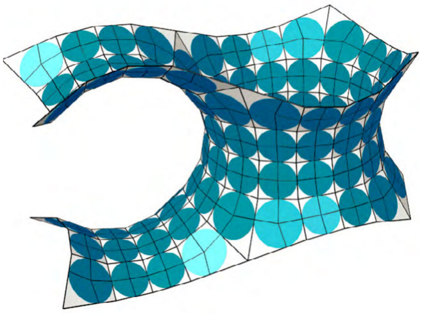


Figure 3.80: Dual Minimal Surface by Christoffel transformation [Image by S. Sechelmann]

Obviously the described construction principle can be applied not only to a single quad but also to a quad mesh. This way we can transform the circle packing created on a sphere to its dual minimal surface. As the circle packing represents a quad mesh its dualisation will result in a planar quad representation of the minimal surface.
The interesting aspect of this way of creating a PQ parametrised surface is that the geometrical starting point was an entirely different surface, namely a sphere.

Java applets where created by S. Sechelmann which produce the Koebe Polyhadra and it's corrsponding Minimal surface. The applet was used for this thesis to produce the images of the Koebe Polyhydra. For further infromation please refer to [Se07] and the following links:

http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml#KoebePolyhedron
http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml#MinimalSurfaces

In the following we want to describe a method to obtain a PQ mesh surface which does not transform the input surface like described previously but uses parts of an existing surface to create a new surface shape out of them.

3.9 Cyclidic patches and cyclidic nets

A *cyclidic patch* is a part of a full *Dupin cyclide* which is cut out along four principal curvature lines that describe the original cyclide: two curves of one parameter family of lines and two from the other family. The intersection of these four curves determine the corner points of the patch. As we know form section 3.5.1 the *Dupin Cyclide* is a special case of a canal surface which holds the unique property that all of its principal curvature lines are formed by circles. The dicretisation of the Dupin Cyclide by using its curvature lines leads to a PQ mesh. With the cyclidic patch being a cut-out of a full cyclide consequently the PQ mesh property applies respectively.

Cyclidic nets are a composition of a number of four-sided cyclidic patches which are joined together along their boundary curves. *Cyclidic patches* - and hence equally *cyclidic nets* which are composed of the patches themselves - can be quite simply discretised by a curvature line parameterisation. The intersection of two curvature lines (one of each parameter family of curves) at the same time always forms an orthogonal coordinate system. This property can be utilised to design PQ mesh surface patches if a number of boundary curves are given. Any cyclidic patch is uniquely defined by its four boundary curves. These curves are circular arcs and their end points are located on a common circle. A method to create *cyclidic patches* by using pre-defined boundary curves has been proposed by A. Bobenko and E. Huhnen Venedey in [BoHu11]. As we are using a curvature line discretisation for the cyclidic patches and nets consequently the generated meshes are always circular and at the same time conical quadrilateral meshes.
In the following chapter we provide a brief summary of the mathematical background of cyclidic patches and cyclidic nets and explain how to obtain a discrete version of them.

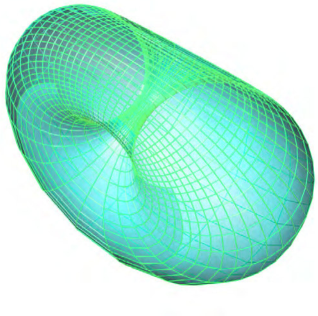


Figure 3.81**: Full Dupin cyclide

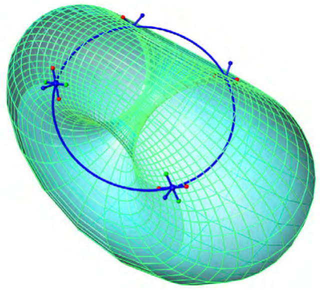


Figure 3.82**: Single cyclidic patch with corner frames

3.9.1 Cyclidic patch

A *cyclic patch* is a surface which is cut out of a full Dupin cyclide (Figure 3.81). The resulting surface patch is bounded by four edge curves (a, a_1, a_2, a_3) all of which form pieces of a curvature line of the original cyclide (Figure 3.82). From chapter 3.5.1 we know that the curvature lines of a cyclide are always circles hence a cut-out of a cyclide consists of circular arcs. The four boundary curves intersect orthogonally at the four corner points (x, x_1, x_2, x_3) of the cyclidic patch and lie on a common circle (Figure 3.83).

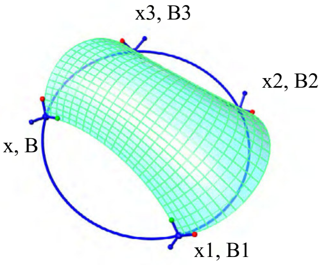


Figure 3.83**: Cyclidic patch with four edge curves and four corner points

Definition and construction principle

A cyclidic patch can be decribed by the four vertex frames (B, B_1, B_2, B_3) which are located at the four corner points of the patch. These frames are orthonormal 3-frames $B = (t_{(1)}, t_{(2)}, n)$ whose vectors $t_{(1)}$ and $t_{(2)}$ are the tangent vectors of the two boundary curves meeting at the corner point which the vertex frame is defined for and vector n the normal vector of the supporting cyclide at that point.
As we know from section 3.5 a cyclide can be defined by a number of spheres moving along a spine curve with each curvature sphere touching the cyclide only along one of its curvature lines which forms a circle. In general curvature spheres are defined as the spheres tangent to a surface with their radii equal to the reciprocals of the principal curvature lines at the point of tangency. Hence the intersection of the sphere with the corresponding surface results in a curvature line which at the same time is the great circle of the sphere.
Each of the four boundary arcs of a cyclidic patch is part of one of the principal curvature lines of the cyclide. The corresponding curvature

* * Images generated with software 'Kobe Polyhydron Editor' by S. Echelmann http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml

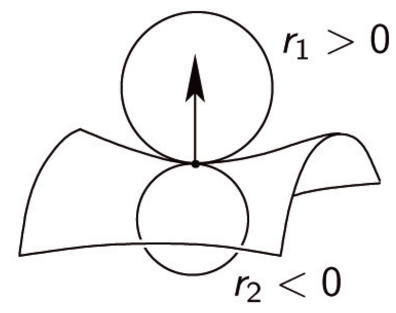


Figure 3.84: Principal curvature spheres touching a surface in its principal curvature directions [Image by E. Huhnen -Venedey]

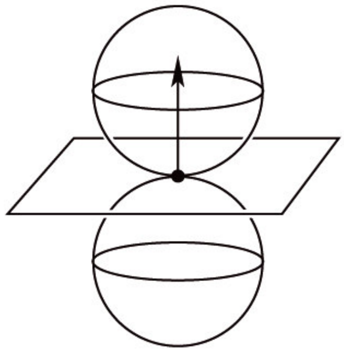


Figure 3.85: Hyperplane between two spheres [Image by E. Huhnen -Venedey]

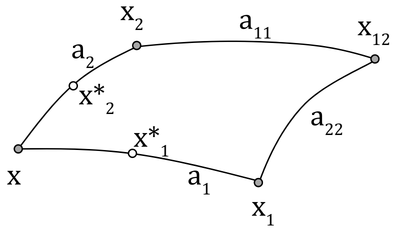


Figure 3.86:Starting patch configuration

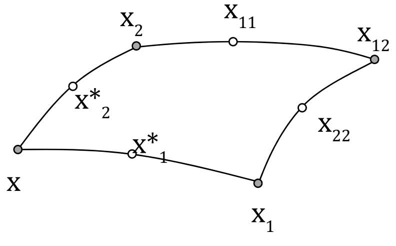


Figure 3.87: Sub arc end points

sphere and respective boundary arc have their centre at the intersection of the normal vectors of the three adjacent vertex frames.

For the construction of a cyclidic patch it is important to know that neighbouring vertex frames (vertex frames of adjacent corner points) comprise of *reflecting angles* relative to their bisecting hyperplane. In Euclidean space a hyperplane separated this space into two half spaces. At the same time it defines a reflection that fixes the hyperplane and interchanges those two half spaces [WIKI].

The bisecting hyperplane of two adjacent corner vertices is determined by the following three conditions [BoHu11]:

i) The bisecting hyperplanes are oriented orthogonal to the supporting plane of the common circle.

ii) All hyperplanes intersect at the centre of the common circle

iii) The hyperplane between two adjacent corner vertices intersects their connecting vector at half distance.

With the four hyperplanes defined and with the knowledge about the condition of the reflecting angles of the vertex frames it is obvious that all corner vertex frames are linear dependent. Hence if we choose the orientation of one vertex frame the missing three are defined uniquely and can be constructed respectively.

Once a cyclidic patch is defined either by cutting it out of a full Dupin cyclide or by constructing it based on four given boundary curves it can be used as the starting geometry to generate a mesh by means of subdivision. The first step of this process will be to generate a sub-patch which by further subdivisions results in a mesh of a certain density depending on the number of subdivision steps.

Generation principle of a sub-patch

Given is a cyclidic patch defined by its corner points x, x_1, x_2 and $x_{ij,}$, the corresponding vertex frames in each of these points and the boundary arcs a_1, $a_{2,}$ a_{11} and a_{22} of the patch (Figure 3.86). Start point for the sub-patch may be point x. If we choose two new points x_1*and x_2* each one lying on one of two adjacent boundary arcs (a_1 and a_2) in point x we can construct a *sub-patch* of the cyclidic patch.
First we can determine the end points x_{11} and x_{22} of the two curvature arcs which start at points x_1* and x_2* as these are located on the opposite edge arcs of a_1 and a_2 at the corresponding subdivision ratio of $x_{1,}$ x_2 in relationship to a_1, a_2(Figure 3.87).

With the knowledge about the hyperplanes and reflecting angles in reference to a vertex frame we can now define the vertex frames in the start and end points of the new arcs. By defining the hyperplane between an existing and a new corner point the 3-frames of the original vertex points can be reflected to obtain the vertex frames in new points x_1*, x_2*, x_{11} and x_{22} (Figure 3.88).

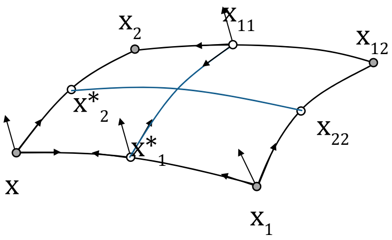


Figure 3.88: Sub patch 3-frames

The start corner point x and the chosen points x_1* and x_2* define the common circle c_1 of the sub-patch. In the same way corner point x_1 which is adjacent to x and points x_1* and x_{22} (start point of one new curvature arc and the end point of the other new curvature arc) define another common circle c_2 of three points (Figure 3.89). These two circles intersect in two points one being x_1 and the other one being the missing fourth corner vertex x_{12} of the sub-patch. The vertex frame in x_{12} can be defined using the hyperplane method.
With all corner points and corresponding vertex frames of the sub-patch defined we can construct the four boundary arcs of the sub-patch connecting corner vertices x, x_1, x_2 and x_{12}. They define a concircular planar quad meaning all corner vertices lie on a common circle. This again proofs that the four points lie in a common plane and thus straight line connections between them describe a planar quadrilateral face.

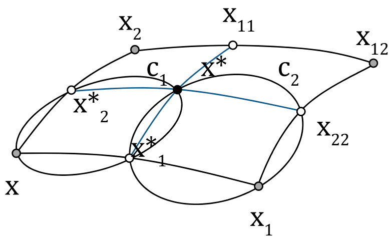


Figure 3.89:Sub patch 3-frames

Obviously we can use this approach to construct as many curvature arcs as required by successively refining the sub-patches of the original cyclidic patch. The result will be a cyclidic patch described by a mesh of PQ facets of a density according to the number of subdivions applied to the original patch.

A related notion to the cyclidic patch with similar properties is the spherical patch which can be constructed based on a sphere instead of a Dupin cyclide.

Spherical patch

The sphere can be seen as a special type of a Dupin cyclide for which the generating family of spheres all have the same diameter and the circular spine curve features vanishing diameter meaning it is described by a point. The patch cut out from a sphere S is called *spherical patch* and similar geometric properties apply as for the cyclidic patch:
Just like for the cyclidic patch the boundary curves of a spherical patch are circular arcs which form part of the one *supporting sphere.* In contrast a cyclidic patch always has four different supporting spheres - one for each boundary curve. The centre of the single supporting sphere S of the spherical patch is located at the intersection of the normal vectors n of the corner vertex frames which at the same time form the radius of S.
The orientation of the four vertex frames are oriented towards one common point. This can be achieved by choosing the four vertices of the patch in a way that they lie on a common square in addition to the condi-

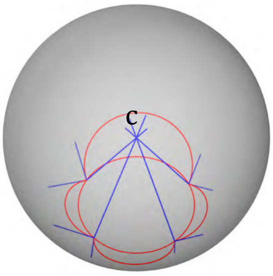


Figure 3.90:Intersecting corner frame normal vectors

tion of being concircular. If we determine the orientation of the vertex frame in the first corner point the other three frames can be constructed by reflection about their corresponding hyperplanes. It is obvious that with all four corner points having the same distance to each other and with the hyperplanes all meeting in the centre point C (Figure 3.90) of the common circle that the reflection of the vertex frames is identical to a rotation about C. Hence their normal vectors will all intersect in a common point namely the centre of the sphere which is the supporting sphere of the spherical patch.

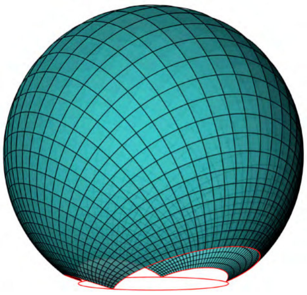


Figure 3.91:Spherical patch

Geometric construction of a PQ mesh based on a cyclidic patch

A discrete cyclidic patch can be constructed uniquely when we choose four concircular corner vertices and determine the vertex frame at one of these vertices. The construction steps which finally lead to a PQ mesh based on a cyclidic patch are outlined in the following:

-Set the diameter of the supporting circle and choose four vertices on the circle (concircular corner vertices)
-Choose the orientation of the vertex frame at one corner vertex (rotation about the three principal axes defined by vectors $t(1)$, $t(2)$ *and* n *of the frame)*
-Construction of the four bisecting hyperplanes
-Successive reflection of the vertex frames in order to receive the three missing corner vertex frames
-Successive intersection of the four normal vectors of the adjacent corner vertices in order to receive the centres of the four boundary arcs
-Construction of the boundary arcs which are defined by centre point and two end points (two adjacent corner points of the patch)
-Choose the number of (even) subdivision steps for each set of opposing edge arcs
-Construction of the curvature arcs at the subdivision points using the sub-patch methodology
- Intersection of all newly created curvature arcs in order to obtain the vertices of the PQ mesh

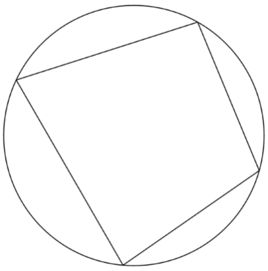


Figure 3.92: Embeded circular quad

When choosing the location of the initial corner vertices we may prefer to consider only configurations which lead to an embedded quad in order to avoid intersecting boundary curves (Figures 3.92 + 3.93). For the sake of completeness it shall be mentioned that in theory a cyclidic patch could be constructed for a configuration with intersecting boundary curves. The resulting mesh however will not be usable for the construction business and is therefore disregarded in this thesis.

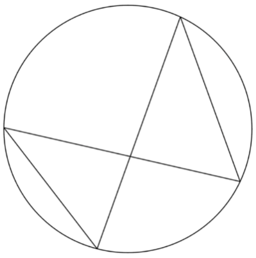


Figure 3.93: Non-embeded circular quad

Geometric construction of a PQ mesh based on a spherical patch

The construction of the spherical patch can be perfomed analougus to the method used for the cyclidic patch as described in 3.9.1.4. The only difference between the construction principles is the configuration of the four normal vectors at the corner points of the patch: In order to receive a spherical patch all four normal vectors of the vertex frames must intersect in a common point. This point defines the centre of the sphere which the spherical patch forms part of. Hence after setting the centre point P we will orient all vertex frame normal vectors towards P. Apart from this additional constraint all other construction steps can be performed as described in chapter 3.9.1.4. The result will be a spherical PQ mesh.
The advantage of the described method is the fact that we can obtain a PQ mesh based on a single cyclidic or spherical patch without having to construct a full cyclide or sphere first. Contrariwise we can start with four arbitrary points with the only constrain that the points have to be con circular to allow the construction of the patch. The disadvantage of this method however is the fact that the entire surface shape is defined by the initial patch and the subdivision into a number of sub-patches only provides a refinement of the mesh in the sense of a higher face density. Hence there is no freedom in surface design with this approach.

Another interesting surface geometry can be obtained by means of a cyclidic net which in general is still based on cyclidic patches but with the difference that not only one patch but a number of different patches form the basis of the net.

3.9.2 Cyclidic nets

Generally speaking a cyclidic net is a composition of a number of different cyclidic patches which are joined together along their boundary curves to result in a new surface. In comparison to the mere subdivision of one cyclidic patch as described in the previous section this approach will allow a greater freedom in shape design. In theory a cyclidic net might consist of patches each one different from the others in terms of size and curvature. The question however is how these can be joined together to form a smooth continuous surface.
Although one method to obtain a number of different cyclidic patches is to cut them out of a full Dupin cyclide these 'singular' patches are not suitable to form a cyclidic net as there is no method available how to determine which patches feature the same boundary curve characertistic in order to join them in a smooth continuous way.

In section 3.9.1 we have explained a method how to obtain a cyclidic patch namely by mathematically generating it based on 4 arbitrary concircular points (vertices (x, x_1, x_2, x_3) and an ortho-normal 3-frame $B = (t_{(1)}, t_{(2)}, n)$ in one of these points. If we use such a patch as the basis of our cyclidic net the question arises how to determine the vertices adjacent to the starting patch and their relating boundary arcs to construct the neighbouring patches.

To understand the mathematical relationship of neighbouring patches in a cyclidic net in detail we first have to introduce the class of principal contact element nets (called PCENs hereafter) as the basis of all cyclidic nets.

Correlation between cyclidic nets, circular nets and principal contact element nets

Per definition in 3.8.1 the four vertices of cyclidic patches always lie on a common circle and per definition in chapter 2.6 circular meshes/nets are maps for which the vertices of each elementary quadrilateral are concircular [BoHu11]. Consequently a cyclidic net which consists of a number of cyclidic patches joined along their boundary curves can be seen as an extension of a circular net with the particularity that each quadruple of circular points represents the vertex set of one cyclidic patch. Based on the knowledge about orthonormal 3-frames sitting in each vertex of a cyclidic patch with adjacent frames being reflections of each other it becomes obvious that there is a 3-parameter family of cyclidic nets for a given circular net.
Consequently a circular net can be seen as the geometrical basis of a cyclidic net: the intersections of the circles the circular net is composed of deliver the vertices for the cyclidic net with each set of four concircular intersections representing the quadruple of one elementary cyclidic patch. The question arises what principal contact element nets are required for if the geometrical configuration of the net is already determined.
Basically the knowledge about two geometrical objects being in principal contact to each other and the net which can be generated based on these contact elements provides the underlying mathematical description to explain the relationship of neighbouring patches in a net and how they can be generated one based on the other.

According to [BoHu11] each principal contact element net contains a circular net as well as a conical net meaning they both belong to the class of principal contact element nets whose defining property is that neighbouring contact elements share a common sphere which coincides with the principal curvature sphere of a surface.

Oriented contact and contact elements

It is important to know that the definition of 'oriented contact' of two geometrical elements is different depending on the two objects which are in contact with each other. With respect to cyclidic nets the relevant topic is oriented contact of two hyperspheres s1 = (c1, r1) and s2 = (c2, r2) which are called to be in oriented contact if they are tangent to each other with coinciding normals in the point of contact (Figure 3.94)

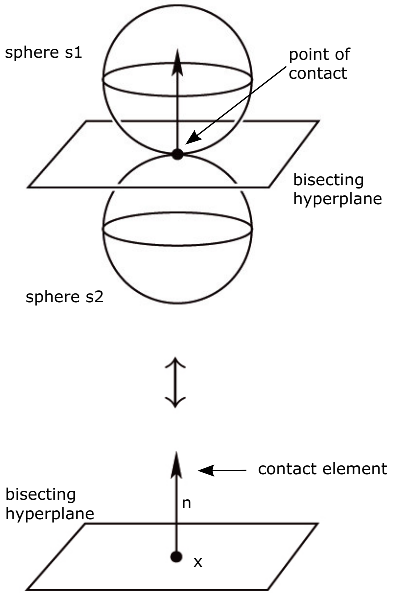


Figure 3.94: Two spheres s1 and s2 in oriented contact (above) and the bisecting hyperplane containing the contact element (below) [Image by E. Huhnen -Venedey]

Definition [BoHu11]

Given are two spheres s1 and s2. These are said to be in oriented contact if there is a unique point of contact X with a unique normal n in X. This normal in turn determines a unique oriented hyperplane p through X, which is in oriented contact with both s1 and s2. Consequently two spheres in oriented contact determine a unique pair (X, n) which gives rise to a 1-parameter family of oriented spheres. This family of spheres is comprised by all spheres in oriented contact with contact point X and the normal n in this point. This constellation of point and vector (also called a pencil because of its characteristic shape) is defined as a contact element and can be identified with the tuple (x, n).

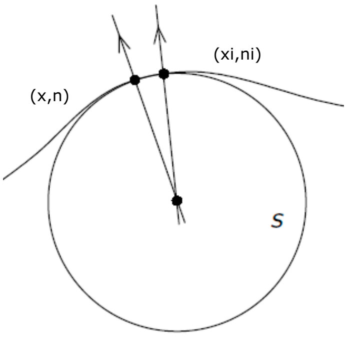


Figure 3.95: Infinitesimal principle contact condition [Image by E. Huhnen -Venedey]

With this knowledge around contact elements we can define a net of principal contact elements as the basis for our cyclidic net.

Principal contact element nets

A map is called a principal contact element net if the following condition applies to all its elements: two neighbouring contact elements (x, n) and (xi, ni) have a sphere s(i) in common with the particularity that this sphere is the principal curvature sphere of the discrete version of the described surface in the point of contact (Figure 3.95).
Principal contact element nets form the basis for cyclidic nets which according to Bobenko and Suris [BoSu07] can be seen as a discretisation of curvature line parametrised surfaces. In this connection curvature lines are characterised by the property that two infinitesimally neighbouring contact elements share exactly one sphere, namely the corresponding principal curvature sphere.

Correlation of neighbouring principal contact elements
For an elementary quadrilateral of a generic principal contact element net with finite contact points, the following holds:

i) The normals of neighbouring contact elements intersect.
ii) The point spheres contained in the contact elements lie on a circle.

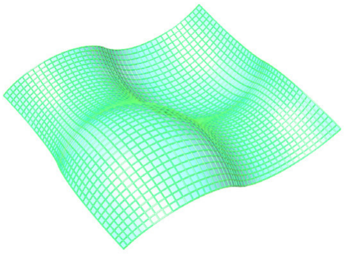


Figure 3.96**: 2D cyclidic net

The above explained mathematical fundamentals about cyclidic nets and principal contact element nets deliver the background how in general cyclidic nets can be composed of a single cyclidic patch. An interesting aspect is that this knowledge can be used to either generate 2D cyclidic nets or 3D cyclidic nets with the difference that 2D nets feature an open surface confined by a number of boundary curve s (Figure 3.96) whereas 3D nets feature a closed surface (Figure 3.97).

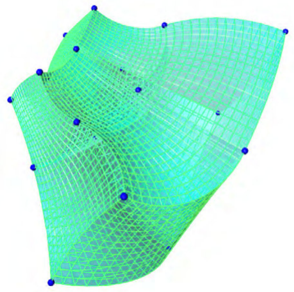


Figure 3.97**: 3D cyclidic net

Geometric construction of a 2D cyclidic net

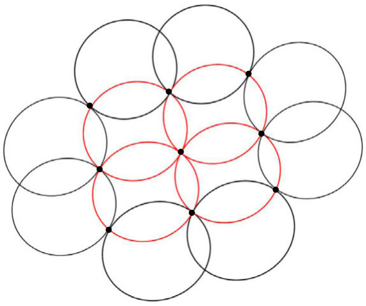


Figure 3.98: No 12 circular net

A 2D cyclidic net can be constructed uniquely for a given circular net and a triply orthogonal frame in one of the vertices of the circular net. The circular net must consist of at least 12 intersecting circles to act as the start geometry of a cyclidic net. The intersections of these circles define the corner vertices of 4 neighbouring cyclidic patches (Figure 3.98). The construction steps how to generate a discrete cyclidic net based on such a given circular net are outlined in the following:

-Choose one net face with corresponding four vertices (x, x_1, x_2, x_3) of the given circular net as starting geometry
-Choose one vertex x of the elementary face as the start point of the cyclidic net
-Choose the orientation of the orthonormal 3-frame $B = (t_{(1)}, t_{(2)}, n)$ in vertex x
-With the start configuration set up we can now construct the first cyclidic patch and all neighbouring patches successively following the construction steps as described in section 3.9.1
For each patch the four corner points are given by the vertices of the underlying circular net.
Once the first cyclidic patch is generated at least two corner frames of the next patch are determined as the adjacent new patch shares two vertices and thus the corresponding 3-frames with the already constructed patch.

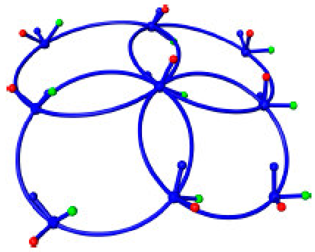


Figure 3.99**: Circles and corner frames 4 patch

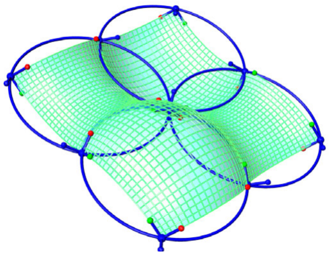


Figure 3.100**: 2D cyclidic 4 patch surface regular

After the generation of the first cyclidic patch one particularity with respect to the construction of 3-frames in neighbouring vertices has to be observed: for all subsequent patches the tangent vector which does not describe the common curvature line needs to be reversed in order to obtain differentiable results.

This process of generating neighbouring patches can be carried out for any size of circular net with the only condition being the minimum required size of four circles (Figure 3.99 + Figure 3.100). As described previously the smoothness of the generated cyclidic net is based on the characteristics of principle contact element nets (PCEN) which any circular net is based on.

For the sake of completeness the construction of 3D cyclidic nets shall be briefly described although their application for the construction business might be rather limited due to their explicit shape.

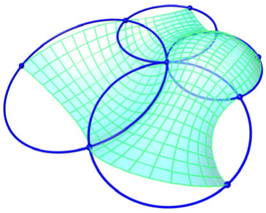


Figure 3.101**: 2D cyclidic 4 patch surface irregular

Geometric construction of a 3D cyclidic net

Just as for 2D cyclidic nets the geometrical basis of 3D cyclidic nets consists of a circular net however the 3-dimensional version of it as shown in Figure 3.97.
For any 3D circular net, there is a 3-parameter family of 3D cyclidic nets with the intersctions of the circular net representing the vertices of the 3D cyclidic net
The result is a closed surface surcumscribed by a number of cyclidic patches whose geometric configuration fullfills the following characteristics:

-All boundary curves represent curvature lines of cyclidic patches
-For each indivicual face the vertices are concircular
-All boundary curves intersecting in a vertex are orthogonal to each other

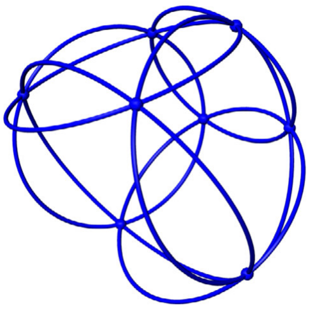


Figure 3.102**: 3D Circular net

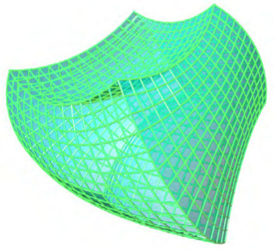


Figure 3.103**: 3D Cyclidic net

A general definition of cyclidic nets in n-dimensinal space is provided by Huhnen [BoHu11]:

A map is called an N-dimensional cyclidic net, if x is a circular net and the frames B;Bi in neighbouring vertices x; xi are related as follows:
The (N - 1)-tuples

$(n^{(1)},........, n^{(i-1),} n^{(i+1)},........., n^{(N)})$

and

$(n_i^{(1)},......... n_i^{(i-1)}, n_i^{(i+1)} ,........, n_i^{(N)})$

are reflections of each other in the perpendicular bisecting hyperplane of the line segment [x, xi], whereas the vector $n_i^{(1)}$ is obtained from $n^{(i)}$ by first reflecting and afterwards changing the orientation (cf. flipping of vectors for the contstruction of a 2D cyclidic net).

With the described method a wide variety of different surfaces can be generated all based on the same circular net. Although the vertices are pre-defined the possibility of choosing the orientation of the vertex frame in the start point allows a certain design freedom. The shape of the underlying circular net defines the 'general structure' of the new surface but the curvature of the individual patches is determined by the orientation of the 3-frame in the start point.

A Java applet by E. Huhnen produces the cyclidic patches, 2D and 3D cyclidid nets and was used to generate the images for this thesis. The applet can be found under the following link:
http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml#CyclidicNets

3.10 Summary

All previously described surfaces and corresponding inherent PQ meshes shall provide an overview of what is known in the field of PQ mesh parametrizations and which surface types could be used for design purposes if the designer is comfortable with a pre-rationalisation of his design. As becomes obvious from the described surfaces this means a certain restriction of the design freedom as only a limited number of surface types are available for the pre-rationalisation approach towards a PQ mesh.
Still the compiled methods deliver a pool of knowledge how certain geometrical elements and objects can be used for the generation of a PQ mesh.
Hence the question is if there is a solution to parametrize any arbitrary freeform surface to obtain a PQ mesh representation of the initial surface by making use of some of the described mathematical theorems and principles but in a new way.
The following chapter 04 describes a method which allows the faceting of an arbitrary doubly curved surface with planar quadrilateral faces. It uses subdivision algorithm and dynamic relaxation principles which will be outlined.

The methodology comprises of several steps to get from a crude input mesh which is generated by hand/manually based on principle curvature lines to an approximation mesh which features the desired planarity of the mesh facets in a density suitable for the construction industry. This requires several intermediate steps like the refinement of the mesh by use of a subdivision algorithm and the dynamic relaxation process of the mesh to achieve planarity of the facets. This process and all required steps will be explained in detail in the following chapter.

Chapter 4 Planar quad (PQ) mesh approximation

4 Planar Quad (PQ) mesh approximation

4.1 Introduction

In this chapter a new method is proposed to approximate a given NURBS surface with a PQ mesh. Powerful universal numerical algorithms exist which can be used to optimise any given starting mesh towards planarity of the faces. But to reach optimal convergence the topology of the starting mesh needs to match the intrinsic curvature behaviour of the target surface. Therefore we focus on the investigation of the curvature proprieties of the target surface which leads us to a high quality starting mesh.

The method consists of four basic steps which are performed fully or semi - automated. An important goal of the development was the implementation of a adequate user interaction to adopt architectural design ambitions. To perform the approximation steps a series of computer programmes have been developed. In the following sub chapters we discuss the underlying theory which is relevant for the particular approximation step.

As discussed in chapter 2.4 the Lines of Principal Curvature form a conjugate network of curves on a surface and is therefore most suitable to derive a PQ mesh from these curves. The first step of the method consist of the computation of a principle curvature line sketch on the given surface. Here the user can choose seed points manually from which the population of the curvature line map starts. The density and the distribution (isotropic or anisotropic) of the LPC map can be influenced via numerical parameters.

Umbilical points (see also chapter 2.5.3) or flat regions play an important role when computing the LPC map. The knowledge of there position on the surface may be considered when choosing the starting points for the LPC map. Hence the detections of such surface features is discussed and also implemented into the software.

With the computed LPC map and knowledge of the umbilical point one can determine the topology of the approximation mesh. We use a so called crude mesh, which can be sculpted manually and populated thereafter using a subdivision algorithm to fix the basic topology. Here the user needs to consider the direction of the curvature lines and place the edges of the initial crude topology mesh along the corresponding directions. The typical vertex has the valence of four expect at/around umbilical points. The detected umbilical points can be classified by the number of LPC terminating in the point (see chapter 2.5.3). For the individual umbilic point types the corresponding crude mesh layout is proposed.

To populate the initial crude topology mesh the Catmull Clark subdivision algorithm is adopted. The subdivision may be performed until the mesh reaches the desired density. To achieve a close fit of the mesh the newly generated vertices are projected onto the target surface.

In the last step the subdivided mesh faces are optimized towards planarity using Dynamic Relaxation principles. Here the initial DR algorithm was modified in order to force the adjacent four mesh vertices of a face into a common plane.

We can summarize the outlined steps as follows:

Step 1: Principle Curvature Line Sketch
Step 2: Topology Mesh
Step 3: Subdivision Mesh
Step 4: Optimization

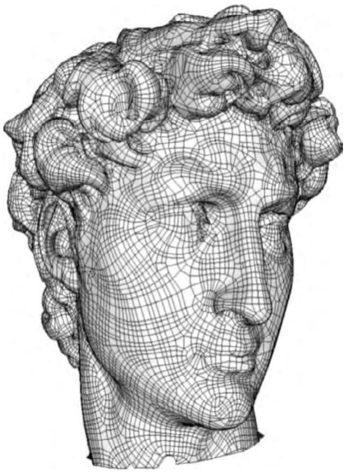


Figure 4.1: Anisotropic Remeshing [Image by M. Marinov, L. Kobblet]

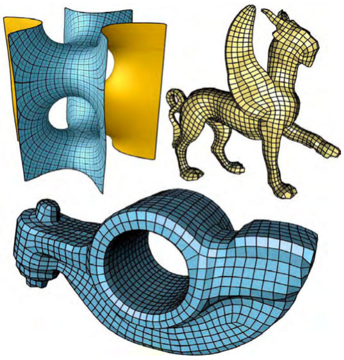


Figure 4.2: Quad Cover [Image by F. Kälberer, M. Nieser, K. Polthier

4.2 Step 1 - Principle Curvature Line Sktech

4.2.1 General

As already identified in chapter 2.4 the principle curvature lines on a surface form a conjugate network of curves. It is understood that the faces of a discrete conjugate net (Q-nets) which approximates an arbitrary surface are planar per definition. Hence a mesh derived from principle curvature lines on a free form surface appears to be a suitable starting mesh for further optimization. In this chapter we will describe a robust method to compute the Lines of Principle Curvature (LPC) on a NURBS surface.

The computation of principle curvature lines and quad meshes generated of them, is from great interest in computer graphics and the mechanical industry. Various methods have been developed in the recent past which employs object matching [KoMa03], anisotropic surface re-meshing [MaKo04] and quad based surface parametrization [KäNi07].

The techniques proposed by the authors mentioned before use sophisticated numerical methods to explicitly integrate curvature lines on manifolds or NURBS surfaces. As we are only interested in a curvature line image which is used as the guidance for a starting mesh, the sophisticated exact integration of the curvature lines is not necessary. A much simpler and processing time efficient method which uses piece wise linear integration of a vector field has been proposed in [DoKi05]. The method adopted allows the user to influence the density and isotropic (uniform) or anisotropic (curvature dependant) behaviour of the curvature line network. The underlying theory and the individual step of the discrete integration method proposed by S. Dong, S. Kircher, and M. Garland is briefly explained in the following chapters.

For completeness it need to be mentioned that a method which builds a PQ mesh directly from a LPC network has been discussed in [Schi07]. The disadvantage of the described method lies in the fact that the overall appearance, variations in mesh density and the location(s) of singular points (umbilics) is determined by the surface itself. To overcome the limitations in architectural design freedom it is proposed to manipulate the kernel radii for the principal curvature direction.

4.2.2 Curvature vector field

The integration of the curvature lines requires the knowledge of the principal curvature directions at any location on the surface. We describe the classical theory of principal curvature directions in chapter 2.5.1. which is based on differentiation of a globally defined surface. Consequently differentiability of the objects studied is a requirement. However, it is not possible to directly carry over these concepts to NURBS or polyhedral surfaces like triangle meshes.
In recent years numerous ways of overcoming this have been investigated. In [CoMo03] ways of generalizing the curvature tensor to triangle meshes stem from discrete differential geometry has been presented which is then used in [AlCo03] for generating the principal curvature direction vector field and tracing principal curvature lines.

In the following we will briefly outline the method for 3D curvature tensor estimation proposed by [AlCo03]:

For a piece wise linear surfaces (E.G. NURBS surfaces) or meshes we need to built a equally piece wise linear curvature tensor field. Therefore the the curvature tensors will be estimated at specific points on the surface or mesh vertices. For a mesh a natural curvature tensor can only be defined on points which lie on a mesh edge [CoMo03]. These tensors T(v) will the be averaged over an arbitrary region B which leads to the following expression [AlCo03]:

$$T_{(v)} = \frac{1}{|B|} \sum \beta_{(e)} \, |e \cap B| \, \overline{e} \, \overline{e}^{t}$$

Equation 4.1

V ... mesh vertex
|B|... Surfaces area around v considered for tensor estimation

β(e)... Angle between adjacent mesh faces, pos: convex, neg: concave (figure 4.3)
|e⋂B|... Lenght of e⋂B between 0 and |e|

e...Unit vector of Edge e

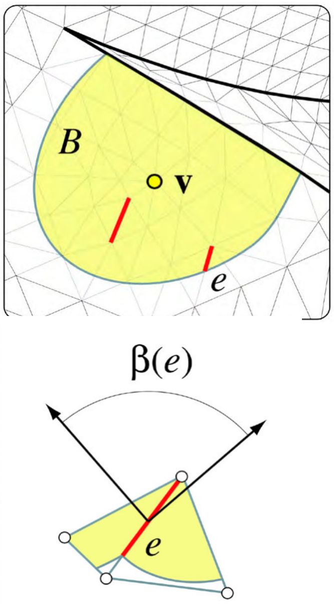


Figure 4.3: Signed Angle between adjacent triangles [Image by P. Alliez, D. Cohen Steiner,O. Devillers, B. Levy, M. Desbrun

The tensor will be evaluate at each point for the defined neighbourhood B which approximates the geodesic disk with the radius r around the point v. A geodesic disc defines a set of point which lie within the geodesic distance of r from the vertex point v. The principal normal vector at v is derived as the Eigenvalue with minimum magnitude from the Eigenvector of T(v). The two remaining Eigenvalues are estimating the two principal curvature direction k_{max} and k_{min}. Please note that the associated directions are switched as such that γ_{min} reflects the maximum curvature and γ_{max} the minimum curvature.

The adopted 3D-CAD software package (Rhinoceros 4.0 ®) allows extracting the principal curvature directions, curvature radii, Gaussian and Mean curvatures at any point of a NURBS surface. The principal curvature tensor field k1, k2 is smooth and orthogonal so that all integral lines may converge at extremal points (umbilical points). An example surface with the principal curvature tensor field is shown in figure 4.4.

It can be observed that Rhino provides the principal curvature directions with sufficient accuracy. This will be in the form of two vectors k1 and k2 which are orthogonal to each other and points away from the reference point p.

The vector direction initially provided by the CAD software will be called to have a *positive* sign. If we reverse the initial vector we receive the vector direction with *negative* sign respectively.

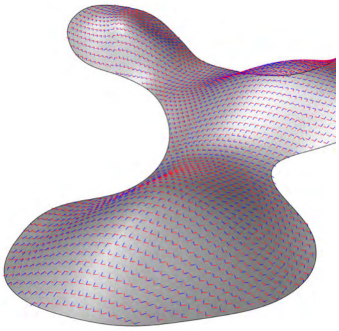


Figure 4.4: Principle curvature vector field

However, a vector field is not uniquely defined in regions close to umbilic areas/points. Hence the distribution of the sign of the initial vector directions which are extracted from Rhino shows inconsistencies. It occurs that points in close proximity to each other show that the initial curvature vector directions pointing in opposite directions. Further the curvature direction of points close to each other may change abrupt from maximum into minimum or vice versa (Figure 4.5).

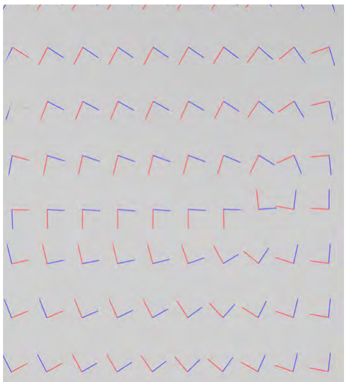


Figure 4.5: Vector field imperfections

These 'noisy' feature of the curvature vector field needs to be considered when we integrate a single curvature line. Several smoothing techniques have been proposed to address this problem. The Laplacian smoothing method proposed by Martin Marinov and Leif Kobbelt [MaKo04] uses the discrete Laplacian operator which originates from the discretization of the heat equation.
In [RaWa06] a Global smoothing approach is described which bases on the minimization of an energy function. The function reduces the angular deviation between adjacent Eigenvectors in a tensor field.

For our application the inconsistencies in the principal curvature field are solve discretely for each LPC independently during their integration. The method will be outlined in the next chapter.

4.2.3 Single curvature line integration

We discretize a curvature line as a piece wise linear curve. The discrete curve is defined by a sequence of nodes n(1 to i) of which the principal curvature vectors in minimum curvature and respectively maximum curvature direction (k1 or k2) are aligned with the tangent vector of the integrated principal curvature line. Each line of curvature either starts from an umbilic point and ends at another one, or has a closed orbit, or can enter and exit from the domain bounds.

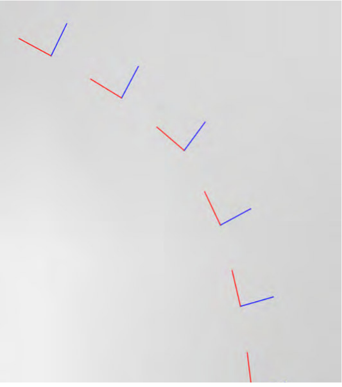


Figure 4.6: LPC vector chain

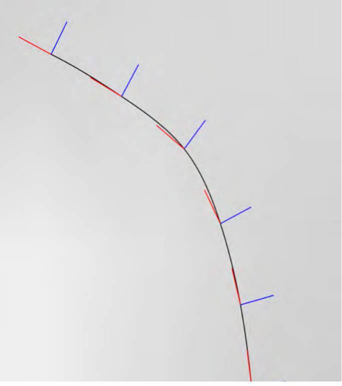


Figure 4.7: LPC NURBS curve

If we adopt the definitions given in [AlCo03] one can trace such a curve in the parameter space (u; v) of the surface by integrating the following ordinary differential equation:

$$\begin{bmatrix} u'(t) \\ v'(t) \end{bmatrix} = \gamma(t)$$

Equation 4.2

γ ... Eigenvector of T(u(t);v(t))

where γ reflects the Eigenvector of the 2D Curvature Tensor T(u(t); v(t)) which is given in Equation 4.3.

$$T = P^t \begin{pmatrix} k_{min} & 0 \\ 0 & k_{max} \end{pmatrix} P$$

Equation 4.3

P^t ... Tensor of principal direction

“We do not need to compute the matrix P in practice. The tensor can be found simply by picking an edge from the 1-ring, projecting it onto the tangent plane, and computing the signed angle α between this projection and the eigenvector of the maximum eigenvalue: the quasi-conformality of our parameterization allows us to now nd the projected eigenvector by starting from the same edge in parameter space, and rotating it by α .” [AlCo03]

As the curvature vectors are oriented orthogonal to each other the symmetric matrix T can be found explicitly.

Our process starts at a given point p on the surface. To find the defining nodes we use a simple search algorithm which successively connects the curvature vectors. For a given start point, a start direction (k1 or k2) and the sampling interval the discrete consecutive nodes can be generated. The closest point to the surface at the tip of the vector in the chosen direction and length defines the next node of the curvature line. The process runs over the surface until it satisfies one of the following stopping criteria:

-Reaches predefined length (steps)
-Hits the defined surface boundary
-Exactly hits an umbilical point

From the given start point we can trace four lines which are in maximum and minimum curvature direction and both positive and negative directions. The so generated nodes can then be joined with straight segments to from a polygonal line or interpolated with a NURBS curve.

As noted in chapter 4.2.2. the curvature tensor field shows inconsistencies in surface regions near umbilic points. We can observe the following cases for a consecutive pair of curvature vectors:

1. Vectors k1/k2 pointing in opposite directions
2. Vector k1/k2 changes direction by roughly 90°

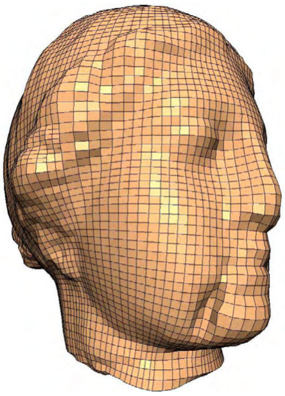


Figure 4.9: Isotropic Mesh [Image by S. Dong, S. Kircher, M. Garland]

Therefore the angle θ between consecutive curvature ki,n ki,n+1 vectors will always be checked and if required correct before the next node is generated. The conditions for the correction of the curvature vectors ki,n can be formulated as follows:

Case 1: θ > (180°±tol) k1/k2 to be reversed

Case 2: θ = (90°±tol) continue with integration in other curvature direction (k1 → k2, k2 →k1)

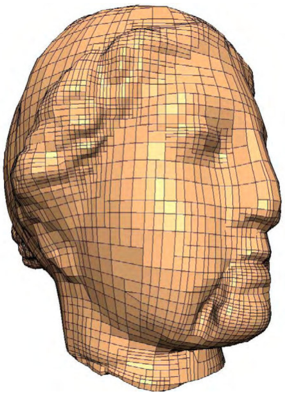


Figure 4.8: Anisotropic Mesh [Image by S. Dong, S. Kircher, M. Garland]

This approach delivered fairly stable LPC integration results which are sufficient to produce a meaningful curvature sketch on a NURBS surface.

4.2.4 Seed point sampling distance

In order to generate a network of curvature lines we need to place further lines next to the initial start curve. A new curve starts at their seed point and it is therefore crucial in which distance and where along the initial curve the new seed point is placed. We can control the seed point location and hence the density of the vector field with the user supplied sampling distance function [DoKi05]:

$$h_1 = h/(1+\alpha \log10 \ (1+\kappa_2)) \qquad \text{Equation 4.4}$$
$$h_2 = h/(1+\alpha \log10 \ (1+\kappa_1))$$

h_1,h_2 ... valued sampling distance along max, min curvature direction
h ... reference sampling distance
α ... curvature sensitivity parameter
κ_1, κ_2 ... local normal curvature in direction

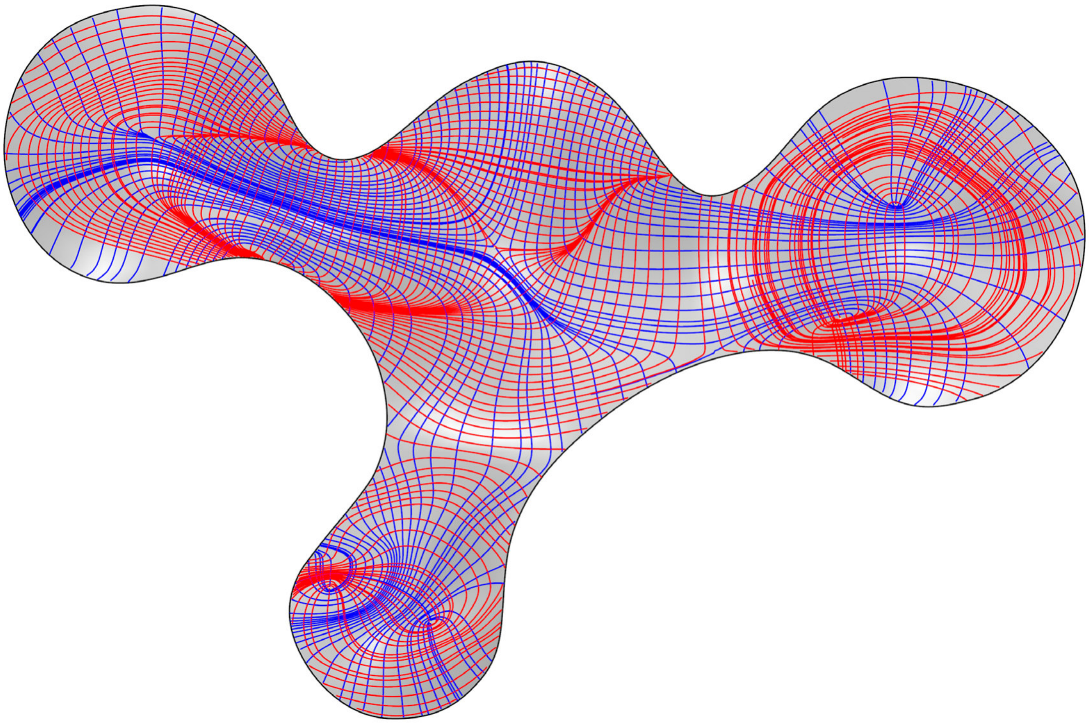


Figure 4.10: LPC Sketch

The density of a curvature network can be categorized as follows:

Isotropic: Uniform spaces, curvature independent
Anisotropic: Uneven spaces, curvature sensitive

With the user defined parameter α one can influence the density of parallel curvature lines corresponding to the local curvature of the surface. For a high α value we generate a dense curvature line arrangement in highly curved areas - an *anisotropic* pattern.

For α = 0 the network becomes curvature independent and evenly spaced (h1=h2=h), therefore an *isotropic* pattern. The authors of [DoKi05] recommend to use a values less the 20 for the parameter α in order to receive meaningful results.

For a given value α at each discretization point along the initial curvature line the h1 or h2 values will be computed. Hence the density of h values will be dependent on the step length of the initial curvature line. At the position with the highest value for h1,h2 the new seed will be placed orthogonal to the initial curve with the distance of h1,h2. This will ensure that the curvature line does not get too dense in highly curved areas.

4.2.5 Extraction of Umbilical points

At umbilical points of a surface the principal curvature vector field shows singular behaviour in proximity. It is therefore important to locate the umbilics before tracing the curvature lines as their location may influence the choice of location for seed points. *An umbilic is a point on a surface where the normal curvatures in all directions are equal and the principal directions are indeterminate. The principal curvature functions are represented in terms of the Gaussian (K) and the mean (H) curvature functions as follows* [KoMa03]:

$$\kappa_1, \kappa_2(u,v) = H(u,v) \pm \sqrt{H^2(u,v) - K(u,v)} \qquad \text{Equation 4.5}$$

κ_1, κ_2 ... local normal curvature in direction 1, 2 (max, min)
H... Mean curvature
K... Gaussian curvature

If we denote $W(u,v) = H^2 - K$ for a point to be an umbilical point the condition $W(u,v) = 0$ must be fulfilled. Hence for an umbilical point the requirement can then be simplified to:

$$H^2 - K = 0$$
or
$$\kappa_1 - \kappa_2 = 0 \qquad \text{Equation 4.6}$$

The strict zero condition of the above formulas will be relaxed to a reasonable tolerance δ_{umb} which is in correspondence with the global tolerance of the CAD software.

$$\kappa_1 - \kappa_2 = \delta_{umb}$$
or
$$H^2 - K = \delta_{umb} \qquad \text{Equation 4.7}$$

δ_{umb}... user defined tolerance value

The reporting of umbilical points, lines and regions has been the topic of various research publications. The authors of [KoMa03] propose the adaptive quad tree decomposition to locate the roots of the Bernstein basis of the polynomial surface description. Another method suggested by [CaFa06] which reports umbilical points as intersections of ridges which are computed whilst solving their implicitisation theorem.

For our application we adopt the quad tree decomposition algorithm as proposed in [KoMa03] in order to locate umbilic points on a NURBS surface. To detect the existence of umbilical points on a confined surface patch [KoMa03] it is requiewd to solve the roots of the Bernstein basis of this patch.

A simpler and less computation time intensive method is used instead. A field of discrete points within a search patch will be checked against Equation 4.7. The process starts with a coarse grid of search points covering the entire surface and a high value for δ_{umb}. The quad tree decomposition algorithm then successively refines the search patches which potentially contain umbilical points whilst increasing the density of search points and the tolerance δ_{umb}.
In order to set the size of the starting patches for the quad tree decomposition we choose the no. of subdivision k_u,k_v for the uv domain of the untrimmed NURBS surface.

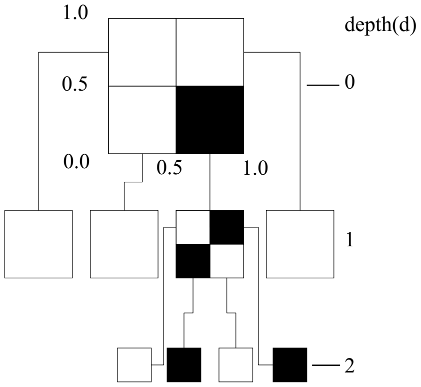


Figure 4.14: Adaptive Quad tree decomposition

Note: Umbilical points which may be detected in patches beyond the trimming boarder of the original surface will be ignored after execution of the algorithm.

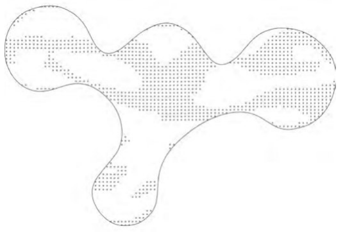


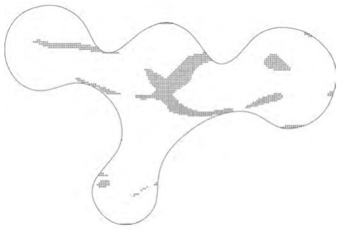
Figure 4.13: Umbilical points Iteration 0

Each rectangular surface domain is then subdivided into four rectangular domains at the mid points of u and v using the de Casteljau algorithm which evaluated with rounded interval arithmetic. The domain for each starting patch at the depth of *d* is described in the following definitions:

$$r2^{-d} \leq u \leq (r+1)2^{-d}$$

$$s2^{-d} \leq v \leq (s+1)2^{-d} \qquad \text{Equation 4.8}$$

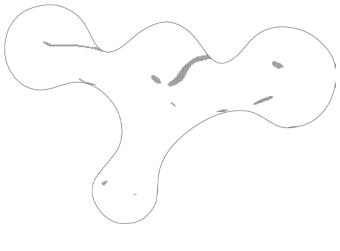
$$r...0 \leq r \leq 4^{d-1}$$
$$s...0 \leq r \leq 4^{d-1}$$



Note: Umbilical points which may be detected in patches beyond the trimming boarder of the original surface will be ignored after execution of the algorithm.

Figure 4.11: Umbilical points Iteration 1

The new generated vertices V_{sub} (v_0 to v_k, u_0 to u_k) define the corner nodes of the new patches Ps and their sub domains (u_1 to u_2, v_1 to v_2). For a given sampling grid subdivision g_s further vertices are computed and checked against the user defined tolerance δ_{umb} for Equation 4.7. If the patch contains vertices which satisfy the given condition it is marked for refinement.



The patches which contain potential umbilic points are subdivided further to receive four new patches. For the subdivided patches new vertices are computed whilst keeping the number of grid subdivision g_s constant. The check against the condition of Equation 4.7 is performed with an increased tolerance δ_{umb}. The sampling point density increases by the factor of two as we keep the grid subdivision constant and the patch edge length halves.

Figure 4.12: Umbilical points Iteration 2

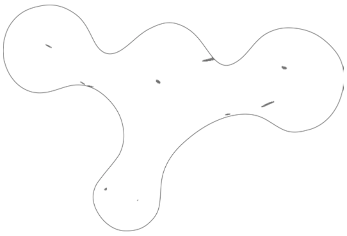


Figure 4.18: Umbilical points Iteration 3

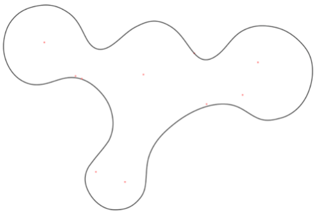


Figure 4.19: Final Umbilical points

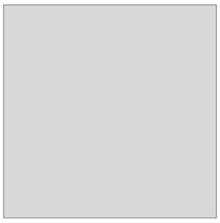


Figure 4.16: Face i= 0, n_f = 1

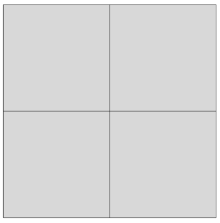


Figure 4.15: Subdivision i = 1 n_f = 4

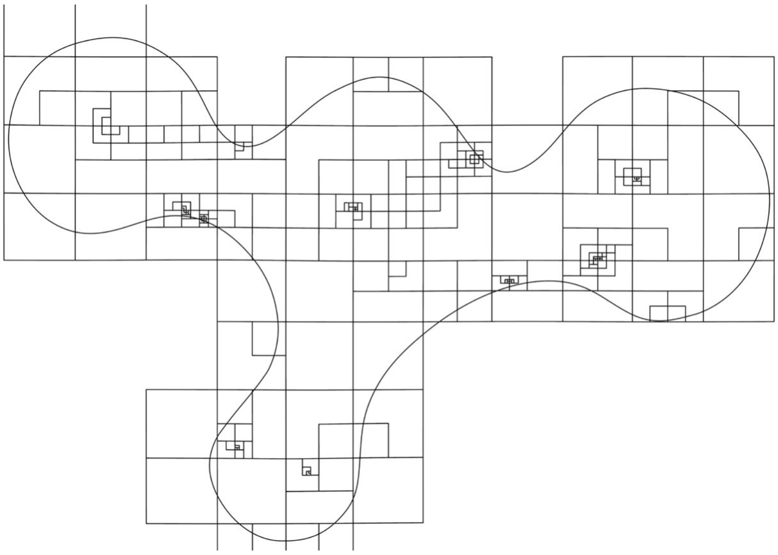


Figure 4.17 Refinement of search patches

This process iteratively refines the regions until the desired final tolerance is reached or a single umbilical vertex is isolated. If the latter is chosen, ridges or flat regions may not converge into a single point. This option is only appropriate in the case it is known that the surface does not contain such features.

To isolate a single umblical vertex a large number of iterations may be required. The user can choose an alternative approach at which approximates the potential umbilic point. At a chosen iteration we compute the convex hull of the remaining candidates in a cluster and set the final umbilic point at their centre if gravity. As this point may not lay on, but very close to the surface, it will be projected back onto it.

4.3 Step 2 - Topology Mesh

In the proposed method we use the LPC network of the target surface as a guidance for the manual sculpting of a crude starting mesh. As discussed in chapter 2.4 the PQ mesh is a discrete version of a conjugate network of curves. The principal curvature lines form such a conjugate network. Hence we desire a approximation mesh which edges are aligned with the lines of principal curvature. With the definition of the crude mesh we also set the topology of the PQ mesh hence we call the crude starting mesh the *topology mesh*.

The topology of the crude mesh needs to be in correspondence to the special features of the target surface. The relevant surface features are the *umbilical points* and the *ridge or valley lines* connecting them. The umbilics can be specified by the number of LPC's which intersecting each other exactly in the umbilic point (refer to chapter 2.5.3). Often the LPC's approaching or leaving an umbilic point follow ridges or valleys on the surface. The information gained from the principal curvature line network and their special feature will be used to sculpt an optimal crude topology mesh. A proposal how to treat areas close to umbilic points and how to connect the umbilical points, lines and regions when sculpting the starting mesh are given in the following chapter.

4.3.1 Mesh density

Before manually sculpting the crude mesh faces the user needs to estimate the crude mesh face sizes and the number of subdivision iterations required. The algorithm (will be outlined in chapter 4.3) allows only for a fixed subdivision sequence as expressed in Equation 4.4:

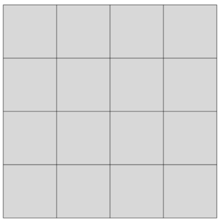


Figure 4.22: Subdivision i = 2, n_f = 16

$$n_f = 2^{2i} n_c \qquad \text{Equation 4.9}$$

n_f ... number of total faces
n_c ... number of crude faces
i ... number of subdivisions

For a given target panel size and number of iterations the rough crude mesh face sizes can be estimated with Equation 4.5

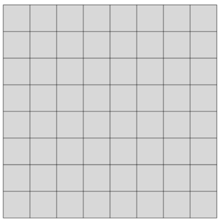


Figure 4.23: Subdivision i = 3 n_f = 64

$$f_{lc} = f_{lt}\, 2^{i} \qquad \text{Equation 4.10}$$

f_{lc} ... face edge length crude
f_{lt} ... face edge length target
i ... number of subdivisions

In order to receive acceptable smooth meshes a minimum of 2 subdivisions should be applied. However the algorithm also allows to choose the number of subdivision and smoothing steps independently. Hence a course mesh can be smoothed afterwards without changing the number of subdivisions.

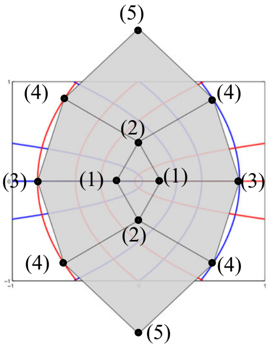


Figure 4.20: Crude Lemon with 2 triangles

4.3.2 Umbilic points

At umblic points we need to allow vertices with a valence other the four. The topology of the crude starting mesh needs to be chosen carefully in order to reflect the singular behaviour of the network of LPC's near umbilics. In chapter 2.5.3 we classified the Umbilical points in the three basic types *lemon*, *star* and *lemon star*. Here we propose the equivalent crude mesh configuration which leads to a subdivided mesh which behaves similar to the LPC's in close proximity to the umbilic point.

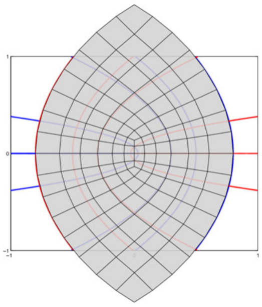


Figure 4.21: Smooth Lemon with 2 triangles

4.3.3 Lemon Type Umbilic

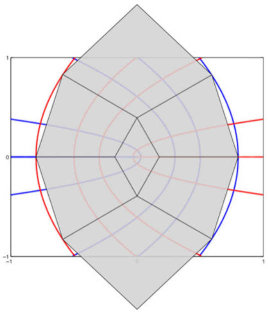


Figure 4.26: Crude Lemon with a diamond

The *lemon* type umbilic occurs when only one LPC is passing the umbilic point. When passing the umbilic point the sign of the line changes from minimum curvature to maximum curvature or respectively. Hence one maximum and one minimum LPC is approaching/leaving the Umbilic Lemon point. The LPC's in close proximity are curling around the umblic point and forming a shape similar to the lemon. For the crude mesh approximation we propose two approaches:

<u>*Crude Lemon with two triangles*</u>

The first approach contains two triangular mesh faces forming the inner lemon with the umbilic point in the centre (Figure 4.22 + 4.23). We locate the two sharing edges of the triangles along the *Ridge LPC* with the umbilic point at their midpoints. We set the length of the Ridge edge and receive two vertices (1) of the initial triangular faces. Then we compute two Lemon LPC's starting at both end vertices of the sharing edge. The third point of each triangel will be placed at the intersection (2) of the previously computed LPC's.

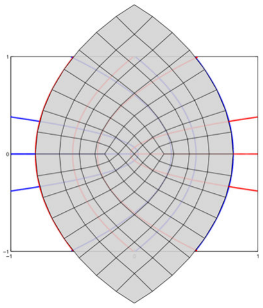


Figure 4.27: Smooth Lemon with a diamond

The second row of the Crude Lemon mesh will now only contain quad mesh faces. There are four faces which share an edge of the initial triangles and two faces sharing only a vertex.

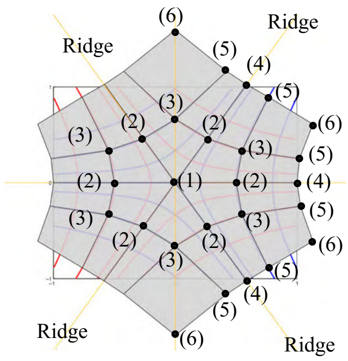


Figure 4.24: Crude Star

For the faces sharing and edge with the initial triangles there are already two vertices (1) given. The third vertex (3) lies on the ridge LPC at a chosen distance. It is appropriate to set the distance equals the length of the shared edge of the triangular faces to receive facets close to a square. To construct the fourth point we use the same approach as for the triangular faces. We compute the second row of lemon LPC starting at the new point (3) and intersect with the first row LPC's to receive point (4). When this procedure is executed for all four faces sharing an edge with the initial triangles the three vertices of the final two faces are already set. The fourth point (5) is then located at the intersection of the second row of lemon umblics.
Additional rows can be constructed using the procedure described before.

<u>*Crude Lemon with a diamond*</u>

The second approach is nearly identical to the first one except that we replace the two inner triangular faces with a diamond shaped quad face. This will avoid too small facets in the centre of the Lemon (Figure 4.24 + 4.25).

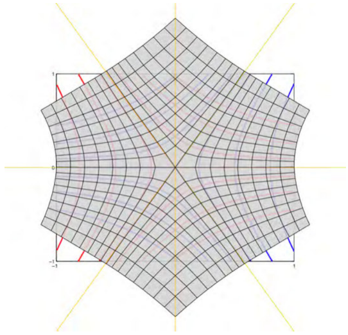


Figure 4.25: Smooth Star

4.3.4 Star Type Umbilic

At the star umbilic there are three LPC's passing the umbilic point which change their sign when crossing the point. The typical angle between the

six Ride LPC's is around 60°. Three ridge LPC's approaching the umbilical point cannot be contained in a right angel. We propose the following layout for the Star umbilic point:
Crude Star

The first row of the crude star contains six mesh faces which all have one vertex located on the umbilic point (1) and two of their edges are aligned with the ridge LPC's (Figure 4.26). Hence we place six vertices (2) on the ridge PLC's in a given distance from the umbilic point. This will define three vertices of the quad faces.

To receive the missing fourth vertex we now compute the LPC's in the orthogonal direction to the ridge LPC's at the new defined vertices (2). We call these LPC's the *tangent LPC's* of the first row of faces. Then we place the final vertices of the first row of quads at each others intersection (3) of these six tangent LPC's. The second row of the crude star consists of 3 x 6 = 18 mesh faces. We again place six vertices (4) of the Ridge LPC's in a chosen distance to the first row. We propose to set this distance as twice the value of the initial distance chosen for the first row faces.

This approach showed good results in matching the LPC pattern when we apply the subdivision algorithm. The Catmull Clark algorithm stretches the edges of the new generated faces in the first row unproportionally around the umbilic point. In order to receive a homogenous distribution of the refined mesh we choose smaller crude mesh faces adjacent to the umbilic point (Figure 4.27).

The neighbouring vertices of the first nodes (5) of the second row faces will be placed at the intersection of the first row tangent LPC's with the LPC's orthogonal to the ridge LPC's - the second row tangent LPC. The missing six vertices are located at the intersections (6) of the second row tangent LPC's.

Additional rows can be added when using the same construction principle as described for the second row. The number of mesh faces will be increased by six additional faces for each additional row.

4.3.5 Lemon Star Type Umbilic

There are three LPC passing through the lemon star or mon star umbilical point which are changing their signs at the point. The three lines approaching the point which are next to each have the same sign and change their sign respectively when leaving the point. The three lines with the same sign can be contained in a right angel. We propose the following crude mesh layout to approximate the LPC in the proximity of the lemon star umbilic point (Figure 4.28).

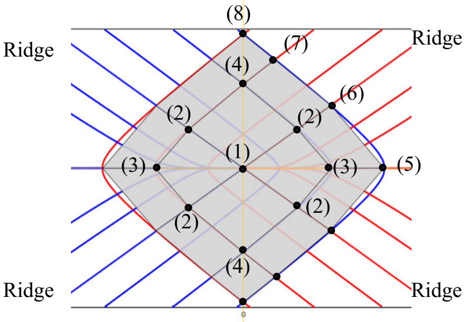


Figure 4.28: Crude Lemon Star

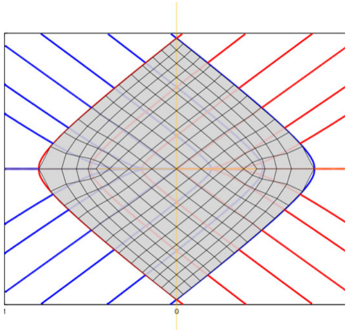


Figure 4.29: Smooth Lemon Star

Crude Lemon Star

The inner row of the *Crude Lemon Star* consist of four mesh faces. All faces have one vertex (1) located at the umbilic point. We locate four mesh vertices (2) at the chosen mesh density distance on the two ridge LPC's with the same sign and which are contained in a right angel. This will define three vertices of the four first row mesh faces. Two of the missing fourth vertices (3) will be placed at each others intersections of the first row tangential LPC's which start at the previously defined vertices. Similar to the *Lemon Type Umbilic point* the tangential LPC's form a lemon shape around the umbilic point hence we have only two intersections of these lines. The last two vertices are located at the intersection (4) of the tangential LPC's with the horizontal ridge LPC's. Here we have to accept that at the inner row none of the mesh edges is aligned with the horizontal ridge LPC. The second row consists of ten crude mesh faces. We start with the faces which edges are aligned with the horizontal ridge LPC. Two new vertices (5) are placed on the ridge LPC in the typical mesh distance from the vertices which are already located on the ridge. We repeat this approach with the four missing ridge LPC's and receive four mesh faces (6). These vertices also define the second row tangential LPC's. Please note that in case of the *Curde Lemon Star* the second row vertices located on the horizontal ridge LPC's are not aligned on the outer tangential LPC's.When we now intersect the first row tangential LPC's with the second row tangential LPC's we receive four new vertices (7) which define also four new faces. The intersection of the second row tangential LPC's with each other define the last two vertices (8) and hence that last two faces.

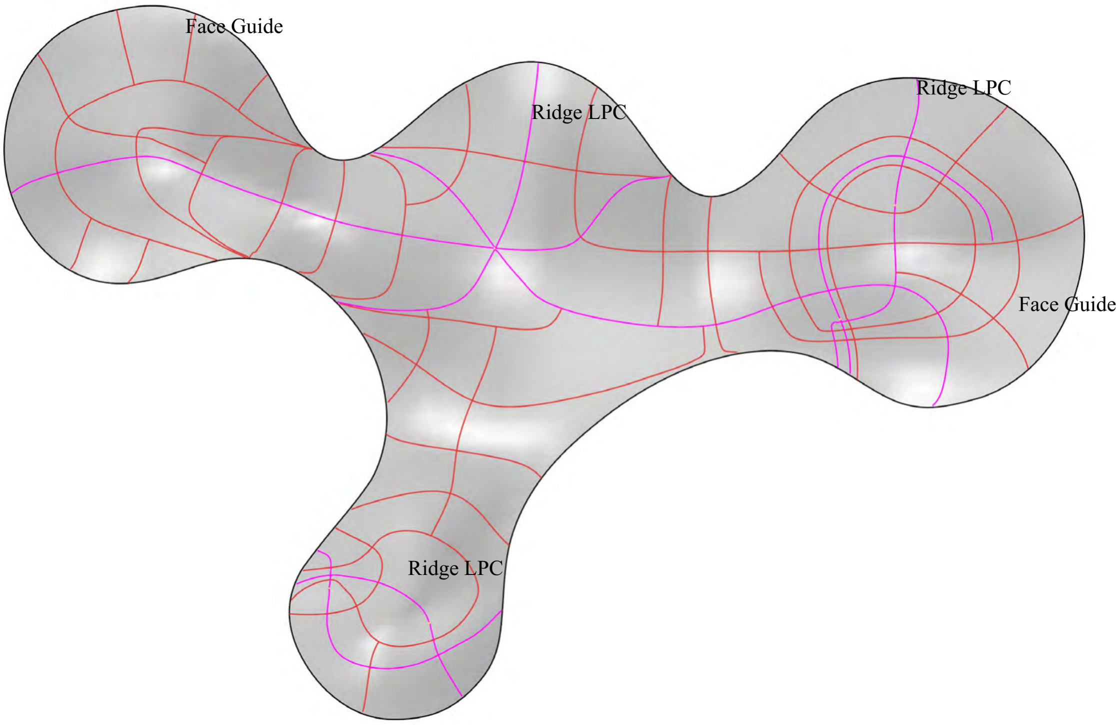


Figure 4.30: Guide LPC's

4.3.6 Sculpting the Topology Mesh

The topology mesh needs to be manually sculpted by the user. We can outline the required steps as follows:

-Determination of target crude mesh face size
-Classification of umbilical points
-Visual extraction of ridge LPC's and face guides
-Selection of face edge LPC's
-Sculpting of crude faces around umbilical points
-Sculpting of crude faces in between umbilical points

Target size of crude mesh faces and number of subdivisions can be determined according chapter 4.3.1. The surface feature (LPC'S, Umbilic points) which have been extracted as described in chapter 4.2 function here as guides for the sculpting. The detected umbilical points can be classified into the three basic types *lemon*, *star* and *lemonstar* as describe in the previous chapter. Here the number of LPC's which enter leave and umbilic points is the most distinct feature for their classification. These LPC's are also called *ridges* and need to be considered mandatory as a sculpting guide. Additional LPC's , in full length or parts of them, may be choose in the distance of the target crude mesh face. We call these the face guides (Figure 4.30).
First the mesh crude faces around the umbilical points need to placed following the layout describe in the previous chapter. The intersections

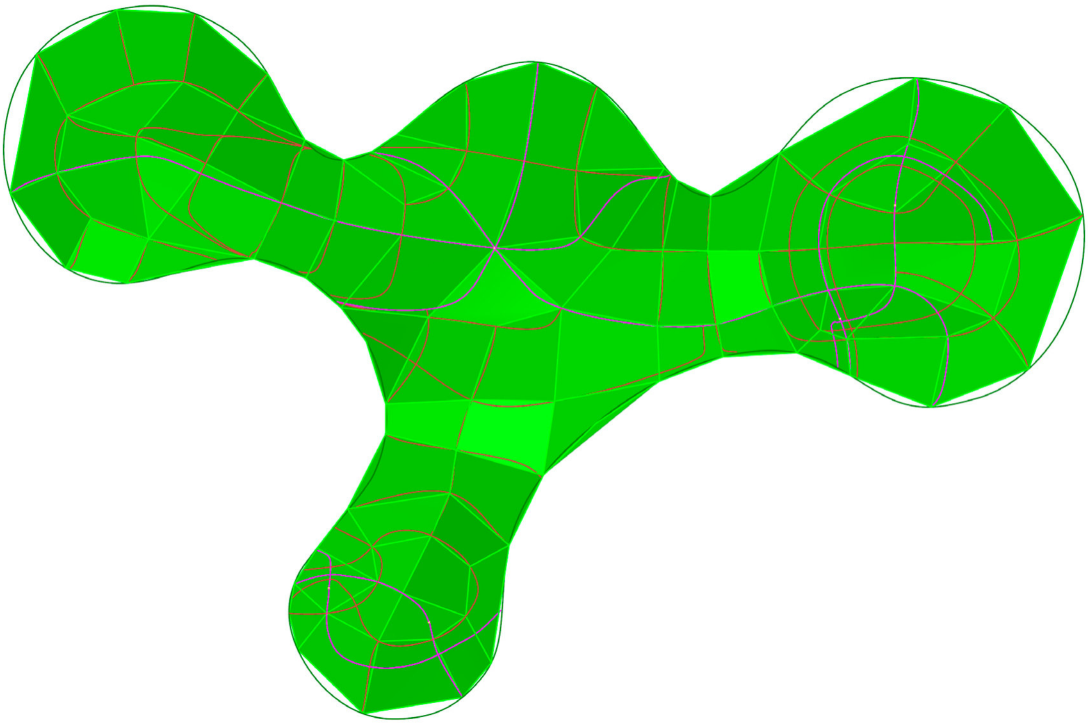


Figure 4.31: Crude Mesh

of the ridge and face guides propose the location for the vertices of the remaining crude faces (Figure 4.31).
It is obvious the quality of the guide network is dependant of the curvature behaviour of the target surface. In regions where the LPC's are evenly spaced and show consistent curvature the curves can be taken almost literally as face edges. In regions where the LPC's densify or change directions sharply is need to be accepted that the mesh faces may cross LPC's rather then be aligned with them.

4.4 Step 3 - Subdivision Mesh

To populate the manually sculpted *topology mesh* we apply a subdivision algorithm. The so called subdivision surfaces are generated by the sequential subdivision and averaging of a coarse input mesh. The procedure can be repeated infinitely and produce a B-Spline surface in their limit. The topology of the subdivision surfaces converges after the first iteration dependant on the scheme applied and is preserved in the following refinement steps. Here we constrain the subdivision to the target surface. This can be either performed strictly on the surface or as a user defined attraction which may smooths uneven target surfaces.

4.4.1 The Catmull Clark Algorithm

Subdivision algorithms have been proposed first in 1978 by Daniel Doo / Malcom Sabin [DoSa78] and Edwin Catmull / James Clark [CaCa78] simultaneously. Both schemes produce quad dominated meshes, although the Doo-Sabin scheme also generates irregular vertices other then with the valence of four. The Catmull-Clark approach results in a quad only mesh after the first refinement step and is therefore more suitable for the proposed application.
Catmull-Clark's algorithm computes a face point for each face, followed by an edge point for each edge and then a vertex point for each vertex. With the so generated vertices a new finer and averaged mesh can be constructed.
In reference to the notation of a standard bicubic B-spline patch as displayed in Figure 4.32 we recall the computation of the new faces:

1. Computing Face points

The new face point q_{11} needs to be computed first. Face points are located in the centre of gravity of each face and reflect the average of all adjacent vertices of the face:

$$q_{11}=\frac{p_{11}+p_{12}+p_{21}+p_{22}}{4}$$

Equation 4.11

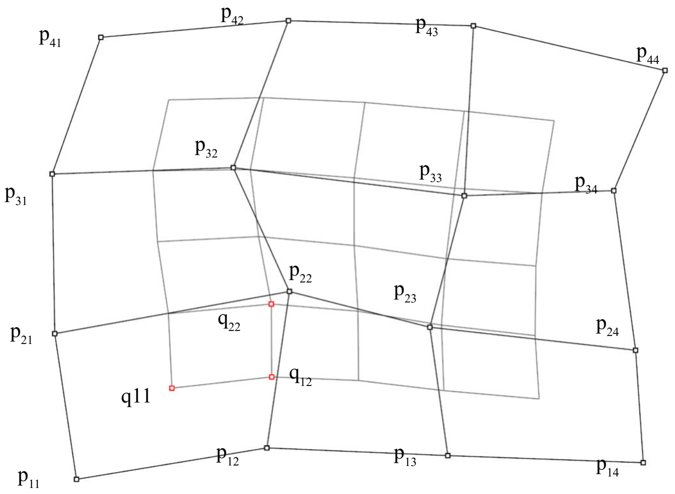


Figure 4.32: B-Spline patch

2. Computing Edge points

Next the edge point q_{12} are computed. The edge points are defined as the average of the two endpoints of that edge and the two face points of the edge's adjacent faces:

$$q_{12}=\frac{\frac{C+D}{2}+\frac{p_{12}+p_{22}}{2}}{2} \qquad \text{Equation 4.12}$$

C... adjacent face point q_{11} as in Equation 4.11
D... adjacent face point $q_{13} = (p_{12}+p_{13}+p_{22}+p_{23})/4$

3. Computing Vertex points

The computation of the new vertex point q_{22} is defined by:

$$q_{22}=\frac{Q}{4}+\frac{R}{2}+\frac{p_{22}}{4} \qquad \text{Equation 4.13}$$

The two intermediate points Q and R are given by:

$$Q=\frac{p_{11}+p_{12}+p_{21}+p_{22}}{4}$$
$$R=\frac{\frac{C+D}{2}+\frac{p_{12}+p_{22}}{2}}{4} \qquad \text{Equation 4.14}$$

Q...the average of all new adjacent face points of the current vertex v
R... the average of all mid - points of the edges adjacent to current vertex v

The equations 4.11 to 4.14 can be generalized into a set of geometric rules which can be applied to arbitrary topologies:

-New face points - the average of all the old points defining the face

-New edges points - the average of midpoints of the old edge with the average of the new face points sharing the edge

-New vertex points - the average $Q/n + 2R/n + S(n-3)/n$

Q, R... as defined in equation 4.9
S ... old vertex points

4. Building New faces

The edges of the new faces are then defined as follows:

-The connection of each new face point to its adjacent edge points

-The connection of each new vertex point to the edge points of all old edges incident on the old vertex point

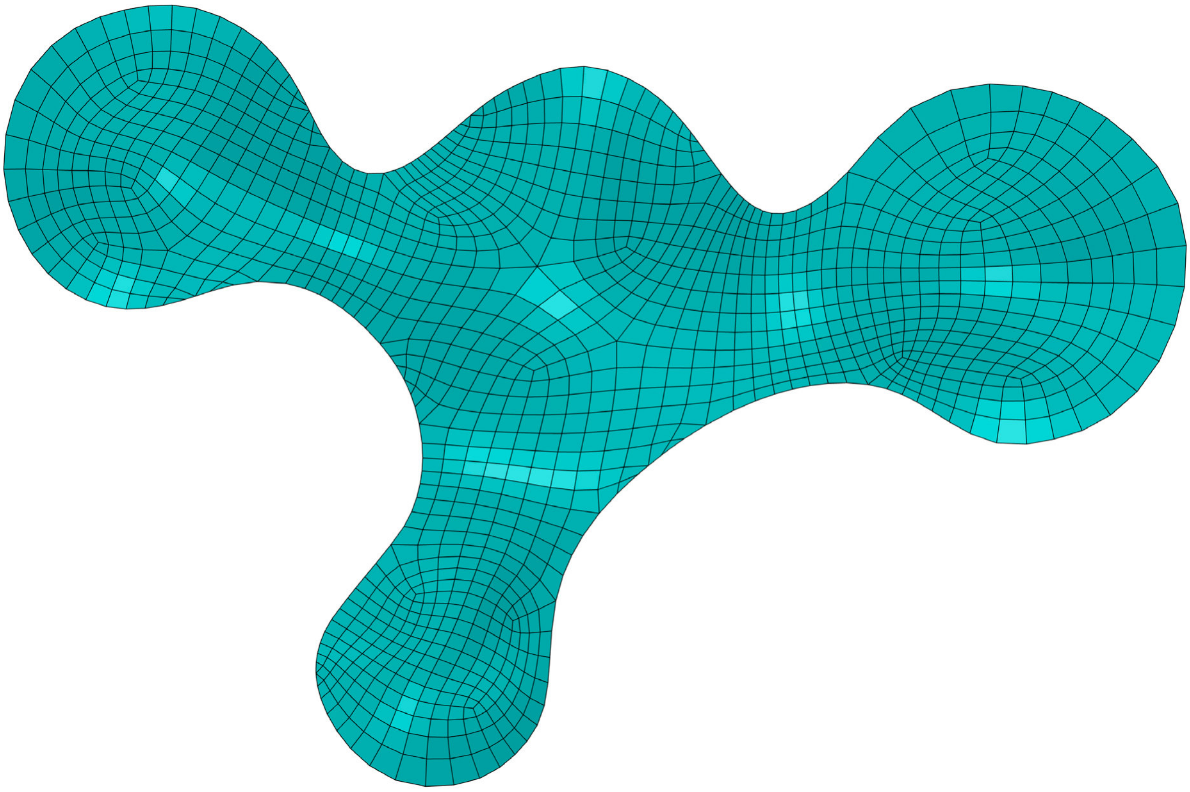


Figure 4.33: Subdivided Mesh with surface and perimeter curve constraint - 2 iterations

4.4.2 Constraint Subdivision

In order to gain further user control over the refined mesh generated by the algorithm we introduce the following constraints:

vertex constraint - individual vertices can be fixed in place

curve constraint - vertices within a user defined distance to a curve will be projected orthogonal onto this curve

surface constraint - vertices within a user defined distance to a reference surface will be projected onto this surface

The constraint mechanism will be performed after each subdivision iteration.

4.5 Step 4 - Optimization

The dynamic relaxation algorithm is suitable to solve approximation problems of discrete systems. The method is based on the discretization of lumped mass system which oscillates about the equilibrium position under the influence of internal and external forces. The relationship between the individual nodes is defined in terms of stiffness similar to finite element methods. The iterative process updates the geometry after each iteration according to the acting forces under consideration of pseudo-dynamic damping. The pulling forces may be formulated specifically for the particular optimization target. We only need to specify a new set of Internal Forces which justifies the desired condition and execute the iteration under the given damping scheme. For the triangulated grid shell structure of the courtyard roof at the British Museum the relaxation algorithm has been manipulated in order to achieve nearly regular faces. The method proposed by Chris K. Williams in [Wi01] starts with a mesh generated in the plane by the simple subdivision of segments with straight boarders. The so generated start mesh is then projected and fixed to the design surface. The algorithm moves the vertices until all edge lengths are within a specified boundary in order to generate a smooth transitions between of the initial triangulated patches.

4.5.1 Dynamic Relaxation (DR)

The dynamic relaxation method is widely used for form finding and nonlinear structural analyses of fabric or cable structures. The method uses the dynamic equation of a damped system with or without externally applied loads to calculate the static behaviour of structures. To reduce the computation storage requirements the DR method is formulated as direct vector method which only considers geometric nonlinearity.
The motion of mesh nodes representing a 2D structure is traced over time until the sum of residual forces in the nodes reach the convergence

criteria. This indicates the state of equilibrium of the structural system. The first DR algorithm for structures was proposed in 1965 by A. S. Day [Da65] and J.R.H. Otter [Ot65]. The appropriate choice of damping influences how rapid a system converges and has been studied by Papadrakakis in [Pa81].

4.5.2 Dynamic Relaxation Theory

The DR algorithm targets to obtain the equilibrium configuration for a damped solution of the dynamic equations of motion. The equation of a viscously damped motion in the centred difference form can be formulated according M.R.Barnes [Ba]:

$$R_{[j,t]} = M_{[j]/\Delta t} (V_{[j,t+\Delta t/2]} - V_{[j,t-\Delta t/2]}) + C_{[j]/2} (V_{[j,t+\Delta t/2]} - V_{[j,t-\Delta t/2]})$$

Equation 4.15

R...Vector of residual forces for the jth degree of freedom at time t
M...Mass
V...Gradient vector
C...Correction vector
j...degree of freedom
t...time

The substitution of C, A,B and rearrangement delivers:

$$V_{[j,t+\Delta t/2]} = A\, V_{[j,t-\Delta t/2]} + B_{[j]}\, R_{[j,t]}/\Delta t$$ Equation 4.16

with
$C_{[j]} = M_{[j]}\, C`/\Delta t$
$A = (1- C`/2)/(1+C`/2)$
$B_{[j]} = \Delta t^2/2M_{[j]}\, (A+1)$

The total nodal deflection $\delta_{[j]}$ at the time $t+\Delta t$ is given in Equation 4.17:

$$\delta_{[j,t+\Delta t]} = \delta_{[j,t]} + A\, (\delta_{[j,t]} - \delta_{[j,t-\Delta t]}) + B\, R_{[j,t]}$$ Equation 4.17

The above Equation is written at each node of the structure for the principal stiffness directions and all mass components M_j are adjusted proportional to the direct stiffness K_{ji}. It may be shown that the critical time interval is:

$$\Delta t_{crit} = \sqrt{(2M_j/K_{ji})}$$ Equation 4.18

K_{ji}...Stiffness matrix

The optimum convergence for the Dynamic Relaxation method corresponds to the critically damping of the fundamental mode and is given in Equation 4.19:

$$\lambda_{opt} = \sqrt{A} \qquad \text{Equation 4.19}$$

When using the relation of the critical time interval expressed in Equation 4.17 for Equation 4.15 we receive:

$$\delta_{[i+1]} = \delta_{[i]} + A\,(\delta_{[i]} - \delta_{[i-1]}) + (A+1)\,\{S\}\,R_{[i]} \qquad \text{Equation 4.20}$$

{S}...Stiffness matrix

The critical damping C' needs to be evaluated newly after each iteration run. This can be performed using an automated adjustment procedure. The convergence rate of the incremental deflection vectors can be expressed in the following function:

$$\lambda = \Delta\delta_{[i]}/\Delta\delta_{[i-1]} = (\delta_{[i]} - \delta_{[i-1]})/(\delta_{[i-1]} - \delta_{[i-2]}) \qquad \text{Equation 4.21}$$

If we use Equation 4.16 we receive:

$$A_{[i]} = ||\Delta\delta_{[i]}||^2 \,/\, ||\Delta\delta_{[i-1]}||^2 \qquad \text{Equation 4.22}$$

If we want to transform the above Equation to a common global coordinate system {S}, to which the deflections and residuals can be referred to, we amend Equation 4.17 to:

$$\{S'\} = \{T\}\,\mathrm{T}\,\{S\}\,\{T\} \qquad \text{Equation 4.23}$$

{T} ...Block diagonal matrix of nodal transformations

4.5.3 The Dynamic Relaxation Algorithm

In this chapter we explicate the individual steps of the dynamic relaxation algorithm derived from the theoretical basis outlined in the previous chapter. The DR method has been implemented into a 3D-CAD environment (Rhinoceros 4.0 ®) as a *'PlugIn'* application. The programming has been performed using the VB.Net language. The algorithm consists of the iterative computation of a series of vector operations applied to the vertices of a given mesh. As the first step of the process the initial mesh needs to be investigated in order to determine the internal adjacencies of its *faces* (ϕ), *edges* (ε) and *vertices* (ϖ). For a quad mesh we can define the following relations:

A face has 4 adjacent vertices and 4 adjacent edges: $\phi\{\varpi_1 ... \varpi_4\},\{\varepsilon_1 ... \varepsilon_4\}$
An edge has 2 adjacent faces and 2 adjacent nodes: $\varepsilon\{\phi_1,\phi_2\},\{v_1,v_2\}$
A vertex has the minimum of 2 adjacent edges: $\varpi\{\varepsilon_1 ... \varepsilon_2 ... \varepsilon_n\}$

ϕ,ϕ_i ...current face, adjacent face
$\varepsilon,\varepsilon_n$...current edge, adjacent edge
ν,ν_n...current node, adjacent node

Definition 4.1

The above defined adjacencies will be computed and written to the starting mesh in order to extract the relevant information during the iterations The following operations need to executed at each iterative step. For all mesh vertices the weighted nodal residual (r') needs to be evaluated first and is given as:

$$r'_{[i]} = (f_{[i]} + acc_{[i-1]}\, M + vel_{[i-1]}\, C) - f_{[i-1]}$$

Equation 4.24

r'...Weighted residual vector
f...Internal force vector
acc...Acceleration vector
M...Fictitious mass
vel...Velocity vector
C...Fictitious damping
i...Iteration

The current internal force vector ($f_{[i]}$) of a node point (p) is equals the sum of all force vectors acting on the current node. The vectors acting of the node are given by the geometry of the adjacent edges/vertices expressed as vectors pointing towards the node.

$$f_{[i]} = \Sigma\,(e_1 ... e_n)$$

Equation 4.25

with $e_n = \varpi - \varpi_k$

The resulting vector ($f_{[i]}$) reflects the *out of equilibrium* value for the particular position which needs to be minimised. In case of a node with valence four the equilibrium state is reached if the crossing pairs are pointing in opposite directing and have the same resulting value which reflects the tension only condition for form found structures. Additional external acting force may be also added to the resulting vector at this stage.

The term "acc * M + vel * C" is a modifier to the central difference integrator in order influence the convergence behaviour of the algorithm, consisting of weighted results of the previous run. For the first iteration we set the Acceleration (acc), Velocity (vel) and Forces (f) equals to the zero vector. The parameters M and C are constants reflecting the *fictitious mass* and the *fictitious damping* of the system.

The current value for the Velocity ($vel_{[i]}$) is determined in Equation 4.23. We use the results of r' and vel from the previous run and some constants F1,F2:

$$vel_{[i]} = (vel_{[i-1]}(F1 / F2)) + (r'_{[i]}/ F2) \qquad \text{Equation 4.26}$$

The parameters F1 and F2 are controlling the damping and are defined below:

$$F1 = M / \Delta t - C / 2 \qquad \text{Equation 4.27}$$
$$F2 = M / \Delta t + C / 2 \qquad \text{Equation 4.28}$$

Δt...Timestep

The constant time step (Δt) can be chosen by the user in order to influence the speed of convergence of the system. We may alter the value for time step during the relaxation process.
The value for Acc is set in the *current* iteration to be used in the *next* iteration and is defined as the difference between the current Velocity ($Vel_{[i]}$) and the previous Velocity ($Vel_{[i-1]}$):

$$Acc = Vel_{[i]} - Vel_{[i-1]} \qquad \text{Equation 4.29}$$

Finally we use the new computed Velocity (Vel[i]) to update the node position:

$$p_{[i]} = p_{[i-1]} + (Vel * \Delta t) \qquad \text{Equation 4.30}$$

p...node point

To evaluate the final nodal residual vector (r) of the iteration we need to consider the user defined control or boundary conditions. If a mesh node is within a specified tolerance of a control object it 'snaps' to it. The control objects may be defined as points (fixed), curves (sliding) or surfaces (orthogonal projection). The option of a user defined attraction rather then the strict snap to control object is available. If attraction is chosen the direct connecting line between node point and constraint object will behave like a virtual spring. The farther away the node point from the constraint object the larger the pulling forces. We choose the following linear fall off function:

$$p_{[i]} = p_{[i-1]} + (p_{control} - p_{[i-1]}) \Omega \qquad \text{Equation 4.31}$$

for $|p_{control} - p_{[i-1]}| <$ snap tolerance
Ω...weight ($0 \geq \Omega \geq 1$)

After the boundary conditions are considered in the update of the node positions the nodal residual vector (r) can be computed:

$$r = p_{[i]} - p_{[i-1]} \qquad \text{Equation 4.32}$$

The global residual value (R) can be computed when adding all absolute values of the nodal residual vectors:

$$R = \Sigma\ |r_{[i]}| \qquad \text{Equation 4.33}$$

The procedure gets repeated for predefined number of iterations or until the target residual is reached.

4.5.4 Planar Relaxation

As mentioned in chapter 4.5.1 the DR algorithm has been previously used to optimise structures towards other targets then to achieve tension force equilibrium in all vertices. We only need to define a new set of pulling forces which are acting towards the desired condition of the mesh. In our case we desire that all adjacent faces to a node become planar. To achieve this target we need to redefine the internal forces with new pulling forces acting towards planarity of all mesh faces. Let us introduce the new pulling forces (f_p) :

$$fp_{[i]} = \Sigma\ (\nu_{p1} \ldots \nu_{pj}) \qquad \text{Equation 4.34}$$

ν_{pj}...out of planarity vector adjacent face
j...number of adjacent faces

For every vertex we compute the out of plane vectors (n_p) of the adjacent faces. The plane of the adjacent face is defined by the vertices of this face, other then the current one. The out of planarity condition can be expressed with the distance (d) to three point plane criteria as described in chapter 2.2.4. In order to compute the out of planarity vector (ν_p) we need to find the closest point of the current node on the adjacent face plane. This point is called the root point and can be defined as follows:

$$p_r = p + t_0 * n \qquad \text{Equation 4.35}$$

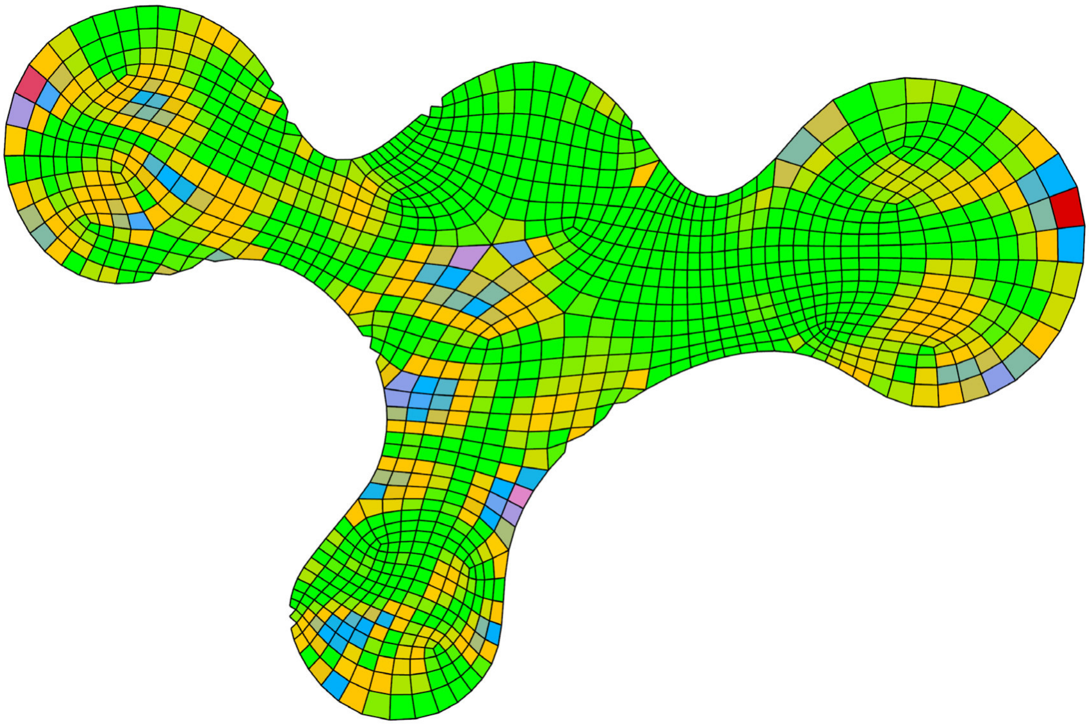


Figure 4.34: Planrity graph for subdivided mesh - max non-planar: 121mm

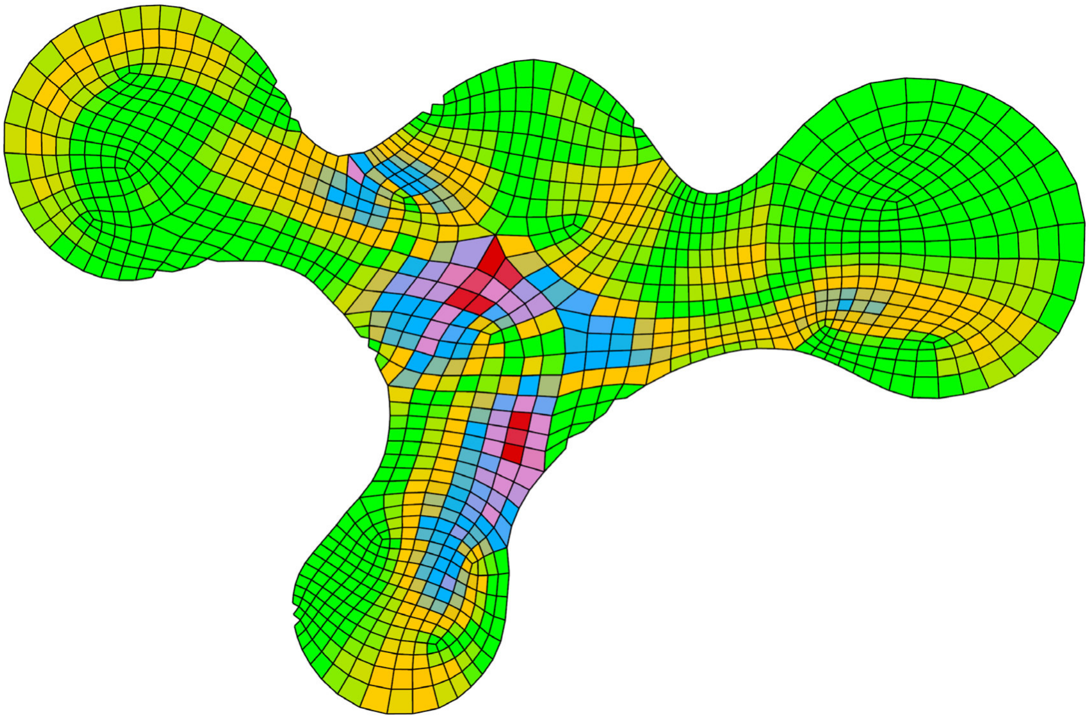


Figure 4.35: Planarity graph for optimized mesh after 1632 iterations - max non-planar: 9mm

with $t_0 = d - n * p/n^2$

for d refer to Equation 2.4
n...normal vector of adjacent face plane

Finally we replace the internal forces (f) in Equation 4.24 with the new defined planar pulling forces (f_P) and receive:

$$r'_{[i]} = (fp_{[i]} + acc_{[i-1]} M + vel_{[i-1]} C) - fp_{[i-1]} \qquad \text{Equation 4.36}$$

4.6 Summary

With the described 4 step method any surface geometry can be approximated with a PQ mesh. The outlined procedure to generate the Lines of Principles Curvature (LPC's) and the umbilical points produces results which are accurate enough to understand the curvature behaviour of the target surface. It is understood that the generated lines may not be accurate in a strict mathematical sense. As these lines are mainly used as the guide for the sculpting of the topology mesh this fact may be negligible. With the availability of the LPC's sketch and approximate location of the umbilical points the crude topology mesh can be sculpted which follows the curvature of the surface. Specify mesh layouts in close proximity of the 3 different umbilical point types (lemon, star, lemonstar) are proposed. The applied subdivision algorithm (Catmull - Clark) generates a smooth mesh on the target surface in the desired density. Finally the proposed numerical optimization of the still imperfect mesh uses a modified version of the Dynamic Relaxation (DR) Algorithm. The procedure works computational efficiently and shows stable convergency. In the next chapter the case studies are executed to verify the proposed method. The results of a trial optimization is displayed in Figure 4.33 before optimization and on Figure 4.43 after optimization. In the next chapter we will test the functionality of the proposed procedure in greater depth.

Chapter 5 PQ mesh approximation case studies

5 PQ mesh approximation case studies

5.1 Introduction

In this chapter the proposed PQ mesh approximation method is tested. First we introduce mesh quality criteria in order to measure the improvement or degrading of the mesh during the optimization. The *relative planarity* (rp_n) set the absolute planarity in proportion to the size of the mesh face. This will consider the fact that larger glass panes allow larger elastic deformations than smaller ones. An important visual quality criteria is the smoothness or continuity of a mesh. Therefore we introduce the *face continuity* (fc_i)and the *edge continuity* (ec_n).

In order to evaluate basic functionality of the Planar Relaxation (PR) algorithm a series of simple meshes will be optimised toward planarity. Each individual optimization will aim to demonstrate the performance of the algorithm under a different aspect such as to handle large deformations, optimise closed objects or approximate pre-defined solutions.

Finally we will investigate different meshing approaches for one theoretical surface (Ellipsoid) and two already realized glass steel roofs: British Museum Courtyard roof and Westfield Shopping Centre both located in London/England. For the ellipsoid surface we mainly evaluate the two proposed approaches for the meshing of lemon type umbilics.
The realized glass structures both contain only triangular glazing panes. Here we test different alternative meshing solutions containing only quad faces.
The first solution comprises the of quads which are defined by two joined triangles of the original mesh. We then further refine the topology mesh considering the alignment of the faces edges with the principal curvature lines of the original roof surface. For some solution we introduce constraint conditions along their perimeter. During optimization process the planarity-, continuity properties and and the distance of individual vertices travelled from the starting geometry is monitored and summarized in a table.

5.2 Mesh Quality Criteria

The quality of the optimization procedure can be measure directly by the achieved planarity of the mesh faces. Here we can distinguish between *absolute planarity* (ap) and *relative planarity* (rp) of a face. For the defi-

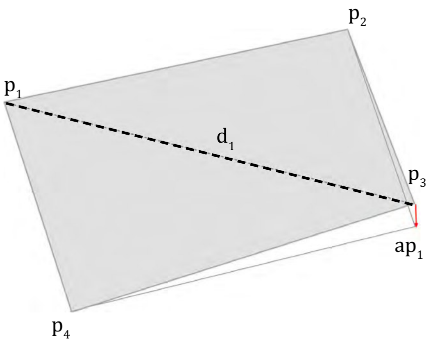


Figure 5.1: Relative Planarity 1

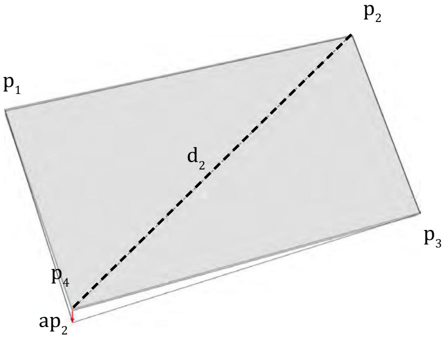


Figure 5.2: Relative Planarity 2

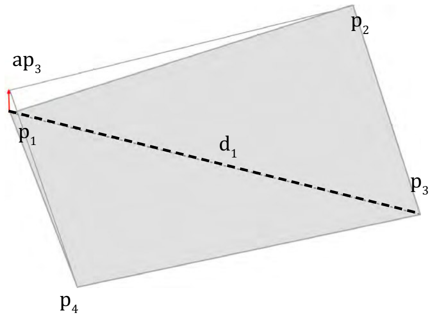


Figure 5.3: Relative Planarity 3

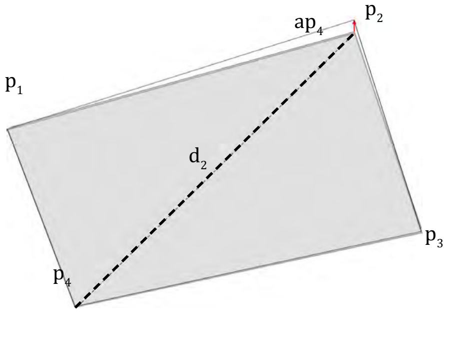


Figure 5.4: Relative Planarity 4

nition of absolute planarity please refer to chapter 2.2.4. In the following section we introduce the relative planarity of a face.

In general a smooth and continuous mesh surface will be most desirable the for architectural purposes. As an interpretational quality criteria we introduce the face continuity and edge continuity at a mesh vertex. Their definitions will be outlined in chapters 5.2.2 and 5.2.3.

5.2.1 Relative Planarity

The *relative planarity* (rp_n) represents the relation between the absolute planarity and the diagonal size of the mesh face. Here we need to evaluate four relationships in order to receive the most onerous value:

$$rp_{1,n} = d_1/ap_{1,n}$$ Equation 5.1

$$rp_{2,n}= d_2/ap_{2,n}$$ Equation 5.2

$$rp_{3,n} = d_1/ap_{3,n}$$ Equation 5.3

$$rp_{4,n} = d_2/ap_{4,n}$$ Equation 5.4

d_1 ... distance between p_1 and p_3
d_2... distance between p_2 and p_4
$ap_{1...4,n}$ ... absolute planarity

The larger value of the two relationships will be set as the relative planarity value of the face.
For the assessment of real glass panes the relative planarity value is more meaningful the absolute planarity as larger sheets can accept larger elastic deformations then smaller ones.

5.2.2 Face Continuity

In order to measure the smoothness of the mesh before and after the optimization we introduce the *face continuity* (fc_i) property. The face continuity measure the angles between faces in a mesh which share an edge. We need to evaluate the normal vectors of all adjacent faces at a vertex. In the most case the faces are not planar. Therefore we need to evaluate the normal vectors of the two triangular faces, which are building the quad face and set the average of the two triangular face normal vectors as the quad face normal vector. From the resulting set of normal vector angles we choose the largest and set it to the vertex's face continuity angle fc.

$$fc_n = \max \measuredangle\{n_{n,1\text{ to }v-1}, n_{n,2\text{ to }v}\}$$ Equation 5.3

$n_{n,v}$ ... normal vectors of adjacent faces at vertex n
v ... valence of vertex

The face continuity will not give any information about the smoothness of the edges crossing the current vertex. The adjacent faces may comprises of a small face continuity angle but have discontinues faces edges. Therefore we introduce the edge continuity property which is outlined in the next chapter.

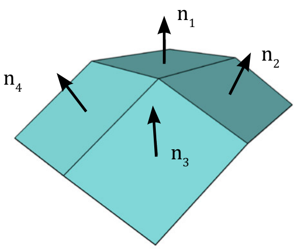


Figure 5.8: Face Continuity fc

5.2.3 Edge Continuity

For the evaluate the *edge continuity* (ec_n) we measure the angles between all adjacent edge vectors $\{e_1,e_2,e_3,e_4\}$ pointing outwards of the current vertex. The state of ideal continuity may be reached when all edge vectors are equally spaced. Hence the edge continuity criteria measures the deviation from the ideal angel (iec_i). Dependant on the number of faces adjacent to the current vertex the ideal angel derive to a portion of 360° (2π) for vertices other then edge or corner vertices. For edge vertices the ideal condition set to be a portion of 180°(π) and for corner vertices to 90° (1/2 π). It is obvious that this state can only be reached when all adjacent edge vectors share a common plane.

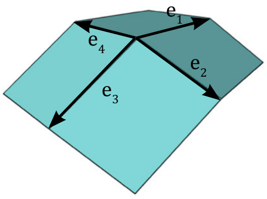


Figure 5.5: Edge Continuity

$$ec_n = \max |\sphericalangle\{e_{n,1 \text{ to } v-1}, e_{i,2 \text{ to } v}\}-iec_i| \qquad \text{Equation 5.4}$$

v ... valence of vertex
iec_n ... 360°/ v   inner mesh vertex
iec_n ... 180°/ v-1   edge vertex
iec_n ... 90°/v-1   corner vertex

The proposed continuity values can only provide an indication of an improvement or decrease relative to the starting mesh.
During the optimization of the cases studies documented in next chapters we will measure the previous defined values after each iteration step and store them in a graph.

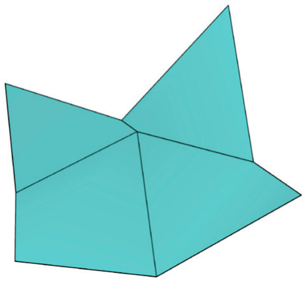


Figure 5.7: 2 x 2 mesh non-planar ($rp_{max,0}$ = 1.09)

5.3 Basic functionality tests

The first series of optimization test will be executed a series of simple geometry meshes in order to evaluate the general functionality of the Planar Relaxation (PR) algorithm. Each individual test aims to demonstrate the performance of the algorithm focussing under a single aspect.

-Large deformations
-Highly deformed meshes
-Closed Objects
-Approximated pre-defined solutions
-Constraint Meshes

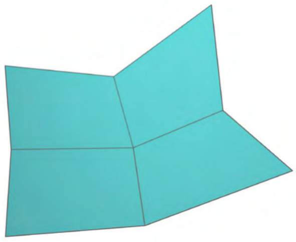


Figure 5.6: 2 x 2 mesh planar (16 iteration)

The individual test meshes and optimization results are briefly outlined in the following chapters.

5.3.1 2 x 2 mesh

The first test demonstrates the initial functionality of the PR Algorithm. We apply the optimization to a unconstraint mesh with 4 faces which comprise of the average edge length of 700 mm. The algorithm start with out of planar values larger then 100% and finds a rp$_{max,16}$ = 0.0001 planar solution within 16 iterations. The maximum displacement occurs in the central vertex and results to 340 mm. Images of the results are displayed in figures 5.7 to 5.9.

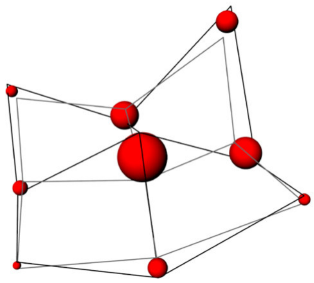


Figure 5.15: 2 x 2 mesh displacements (340mm)

5.3.2 5 x 5 mesh highly deformed

To evaluate the PR Algorithms capability the perform large movements we execute a highly deformed 5 x 5 face. The maxim relative planarity counts to rp$_{max,0}$=2.19. We leave the edges unconstraint allow free movement of the vertices. The algorithm find as planar solution after 65 iterations.

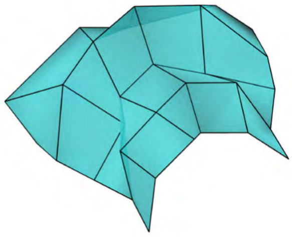


Figure 5.12: 5 x 5 mesh non-planar (rp$_{max,0}$=2.19)

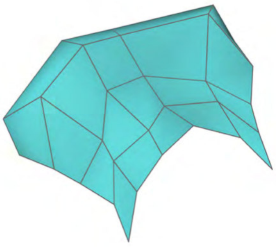


Figure 5.13: 5 x 5 mesh planar (65 iterations)

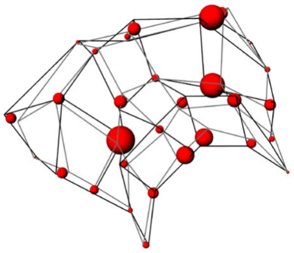


Figure 5.14: 5 x 5 mesh displacements (340mm)

5.3.3 Deformed Cube

The next test mesh is closed object: We perform the optimization for a twisted cube mesh consisting of 6 faces. The optimization is performed without any constraint. The algorithm untwist the cube until all faces are planar. Only 11 iterations are required to reach rp$_{max,11}$ = 0.0001. The faces edges which had varying length before the optimization have similar length thereafter.

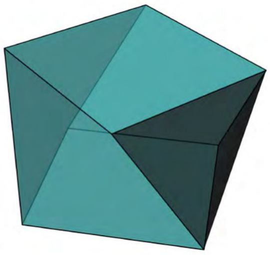


Figure 5.9: Deformed cube non- planar (rp$_{max,0}$=0.68)

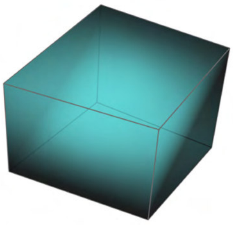


Figure 5.10: Planar cube (11 iterations)

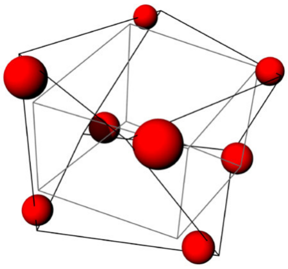


Figure 5.11: Mesh displacements (410mm)

5.3.4 Approximation of pre-defined solution

The next test is intended to evaluate the algorithms capability to approximate a already pre-defined planar mesh layout. As test surface we use the Elliptic Paraboloid. The related PQ mesh (Figure 5.16) is generated using the translation surface principle as discussed in chapter 3.7.3. The mesh comprises of 529 faces with the average edge length of 2500mm.
We then deform 4 mesh vertices in the top of the domed shape (Figure 5.17). The PR Algorithm is the executed considering fixed boundary conditions along all four the perimeter curves.

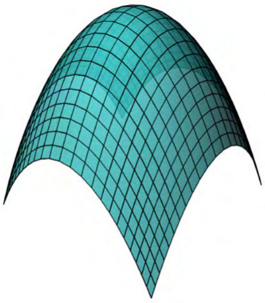


Figure 5.16: Elliptic Paraboloid mesh

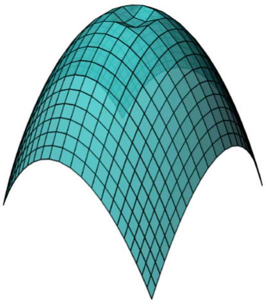


Figure 5.17: EP mesh deformed

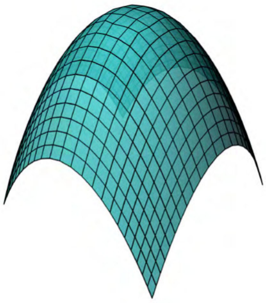


Figure 5.18: EP mesh planar (647 iterations)

The planar solution was found after 647 iterations. However, the original shape generated by translation could not be reproduced exactly with the PR Algorithm. In plan view it is visible that the initially introduced deformation of the faces in the apex of the dome is still visually noticeable. Only the vertices in close proximity to the imperfection where moved by the PR Algorithm.
The apex vertices of the planarized remains 590mm lower then the at the original mesh (figure 5.20).

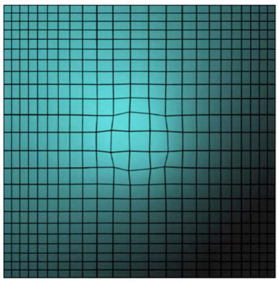


Figure 5.19: EP mesh planar top view

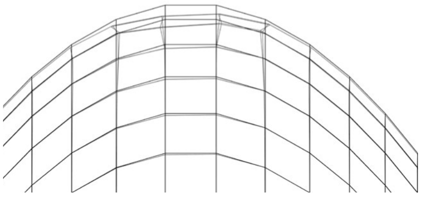


Figure 5.20: EP mesh planar side view

5.3.5 Constraint mesh 22 x 28 faces

As the last generic performance test we apply the of the PR algorithm to a mesh with constraint boundary conditions. We execute a 22 x 28 mesh which is draped over an ellipsoid. Three of the four edges are considered to be fixed. These edge constraint lines are lying in a common plane.

The relative maximum non-planarity of the faces counts to $rp_{max,0}$= 0.27. The non-planar faces occurs along the ridge where the deforming body punches into the previously un-deformed mesh (Figure 5.21).
The algorithm produces mesh faces with maximum non-planarity values of $rp_{max,13500}$ = 0.01 after 13500 iterations. The large number of iterations required is a result of the fully constraint corner faces. Three of their four vertices are lying on the edge constraint lines and are allow to move only along these. Hence the face can only become planar if the forth vertex lies in the plane defined by the three constraint lines. Therefore the entire mesh can only become 100% planar as long the all vertices coincide with the constraint plane. In other words a notional curvature of the mesh surfaces is only possible within the limits of the planarity tolerance given.

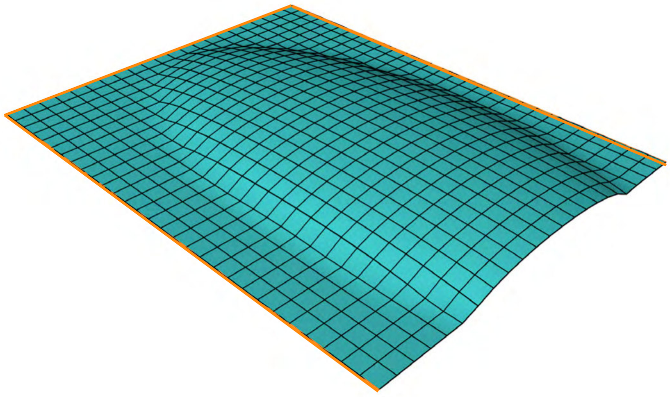


Figure 5.23: Constraint Mesh

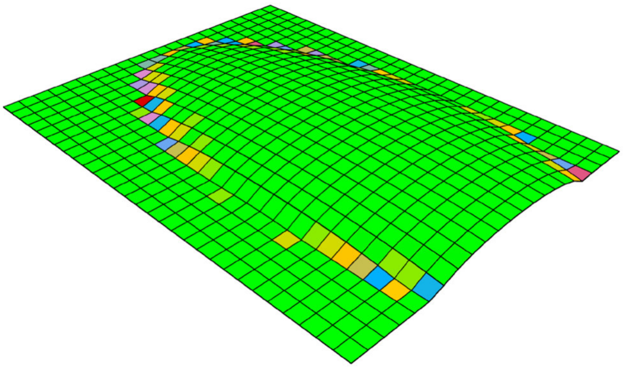


Figure 5.24: Constraint Mesh mesh - non-planar ($rp_{max,0}$ = 0.27)

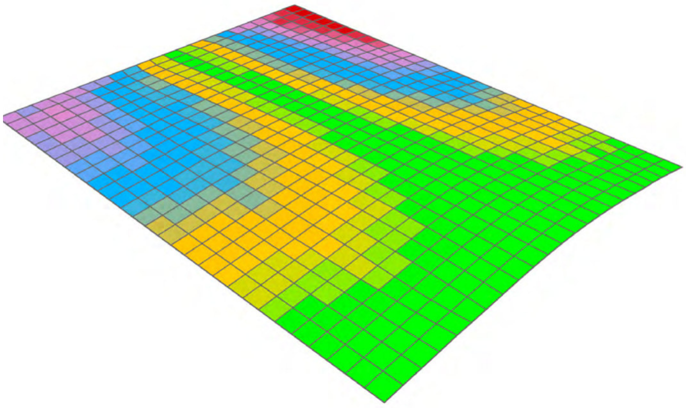


Figure 5.21: Constraint Mesh planar (13500 iterations)

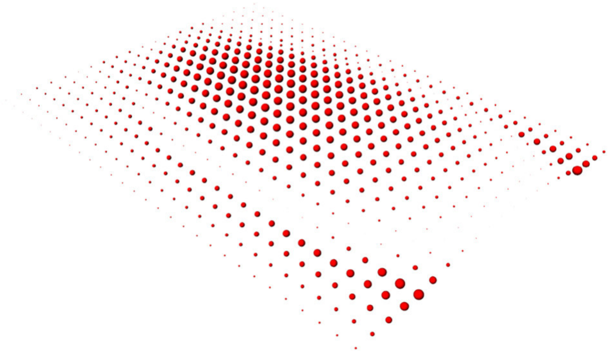


Figure 5.22: Constraint Mesh displacements (1108mm)

In the following chapters the approximation results of theoretical surfaces and realized glass roofs are documented. We follow the individual steps as described in chapter 4. On the basis of the principle curvature line sketch (chapter 4.2, Step 1) different variations of topology meshes are sculpted in order to evaluate their performance during the optimization process. Here we monitor the planarity and continuity properties as well as speed of convergence and the distance of individual vertices travelled from the starting geometry during the optimization.

5.4 Case Studies - Ellipsoid

As the first case study investigate on halve of an ellipsoid surface. The originally closed surface comprises of a four fold symmetry. The principles curvature line graph and the location of umbilical points is well understood and documented.
We can find two umbilical points on the surface. Both umbilical points can be classified as lemon type. In chapter 4.3.3 we proposed two crude mesh approximations: crude lemon with two triangles (mesh 01, figure 5.24) or with a diamond (mesh 02, Figure 5.27). We will investigate the two proposals for this study.

A selection of LPC's will be extracted in order to function as guides for sculpting the crude topology mesh (Figure 5.26 + 5.29).

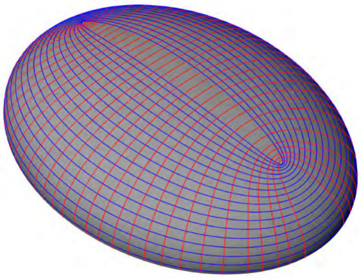


Figure 5.25: Ellipsoid LPC sketch

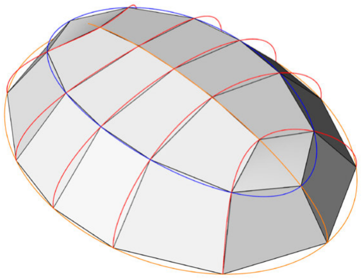


Figure 5.26: Crude Mesh 01- with Diamond

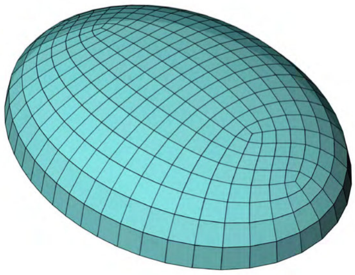


Figure 5.27: Smooth Mesh 01

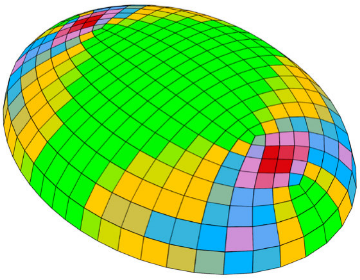


Figure 5.28: Mesh 01 - planarity rp_i= 0.034

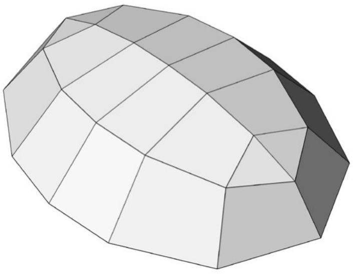


Figure 5.29: Crude Mesh 02 - with two triangels

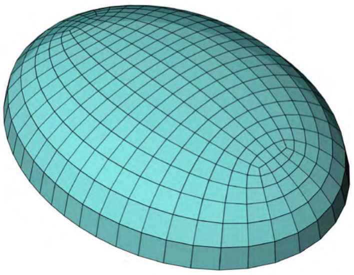


Figure 5.30: Smooth Mesh 02

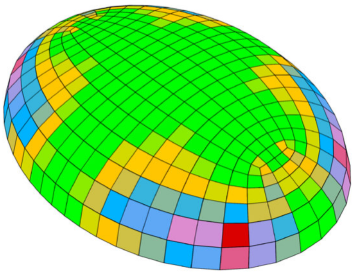


Figure 5.31: Mesh 02 - planarity rp_n=0.0015

For both meshes 2 iterations of Catmull Clarke subdivisions are performed and the mesh is snapped to the ellipsoid surface.
For mesh 01 the faces in proximity of the umbilical points show the largest non-planarity values of rp_i = 0.034 (Figure 5.26).
The maximum non - planarity values for mesh 02 ($rp_{i,max}$=0.015) occur along the perimeter as we can see in Figure 5.29.
Hence the crude topology mesh with two triangles delivers better approximation results around the lemon umbilical point for the tested surface geometry. As the ellipsoid surface is showing a fairly ordered principle curvature graph the generated lines can be almost literally taken to sculpt the crude topology mesh.

We execute the PR algorithm for both meshes until we reach a relative planarity value of $rp_{max,i}$=0.001. As a consequence of the quality of the starting geometry mesh 01 required more iterations (1013) then mesh 02 (545) to reach the given target. Analogous the deviations caused by the optimization are for mesh 01 2.7 times larger then for mesh 02.

The explicit results of the optimization runs are documented in Appendix A1 +A2.

5.5 British Museum Great Court Roof

The roof over the Court of the British Museum covers the area between the central Reading Room and the adjacent buildings.
It was designed by Foster and Partners (architects) and Buro Happold (engineers) and was fabricated and erected by Waagner Biro.

The structure of the roof is formed by a triangular steel grid of box sections which are welded to the node pieces. The triangular facets are glazed by 3312 flat double glazing units with the overall area of approx. 6,750 m².

In order to achieve nearly regular faces a manipulated form of the Dynamic relaxation method was proposed by Chris K. Williams in [Wi01]. It starts with a mesh generated in the plane by the simple subdivision of segments with straight boarders. The so generated start mesh is then projected and fixed to the design surface. The algorithm moves the vertices until all edge lengths are within a specified boundary to generate a smooth transitions between of the initial triangulated patches.

In this chapter we test several meshing options for the roof glazing which uses flat quadrangular panels. A planar quad tessellation of this project was proposed in [ZsSch10] by M. Zadravec, A. Schiftner, J. Wallner. The authors use a field of conjugate directions which are transverse everywhere, the so called TCD field, in order to define the layout of the quad panels.

Figure 5.32: British Museum Great Court Roof [Image by Foster & Partners]

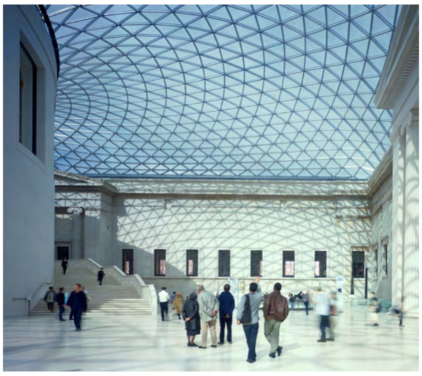


Figure 5.33: British Museum Great Court Roof [Image by Foster & Partners]

5.5.1 BM_Mesh 01: Rhombic Tessellation

As the first planar meshing test we evaluate the original triangular mesh layout (Figure 5.34) as a quad version. For that we join two of the original triangular mesh faces into one quad face. Two adjacent triangular faces are joined along their edges which are in radial direction to the centre of the reading room building. The overall number of faces reduces from previous 3312 faces to now 1687 faces. Due to the diagonal layout we need to accept 66 triangular faces mainly along the central opening. During the optimization these faces will be ignored.
The most non - planar facets are located in the four corners of the mesh. This will become obvious when we compare the layout of the edges with the Principle Curvature Lines Graph (Figure 5.37). The curvature lines and the mesh edges show little alignment in these areas.

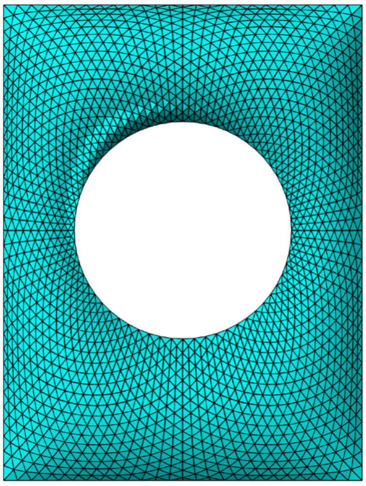


Figure 5.34: British Museum Triangulated Tessellation

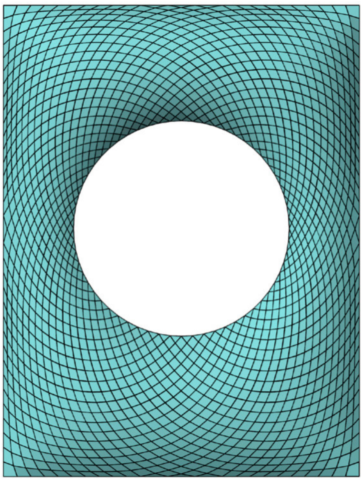


Figure 5.35: BM_Mesh 01 Rhombic Tessellation

It is expected that the mesh layout will not show good convergence during the planar optimization process. Hence we abandon any constraints in order to receive a solution within the target planarity value. The optimization target will be set to 1% ($rp_{max,i}$ = 0.001) which is sufficient when using double insulating glazing units.

The following results will be monitored after each iteration (i) and stored in a table:

Relative Planarity	$rp_{max,i}$
Displacement	$wi_{max,i}$
Face Continuity	$fc_{max,i}$
Edge Continuity	$ec_{max,i}$

In addition we generate a coloured planarity gradient mesh at $rp_{max,i}$ = 0.05, 0.03 and 0.01 in order to trace the improvements visually. The figures 5.36 to 5.38 are showing the planarity scaled respectively to the maximum value of the individual graph.

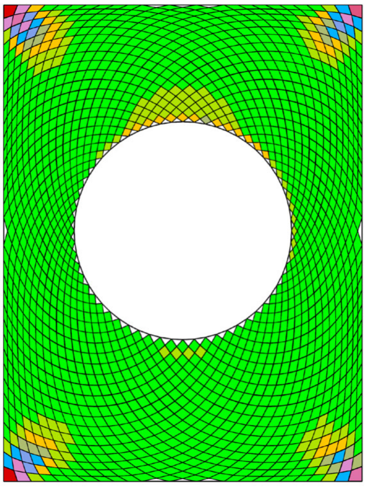


Figure 5.36: BM_Mesh 01, Planarity $rp_{n,max} = 0.578$, i = 0

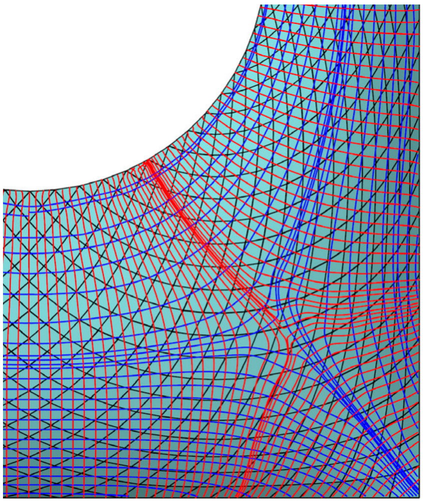


Figure 5.37: BM_Mesh 01 LPC graph overlay

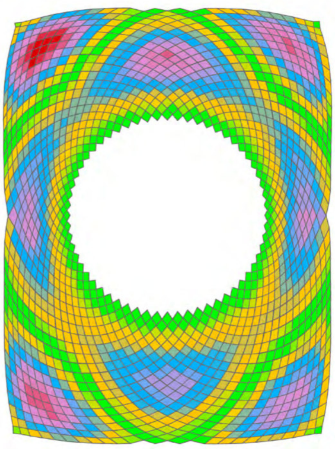


Figure 5.38: BM_Mesh 01, $rp_{i,max} = 0.05$, i = 457

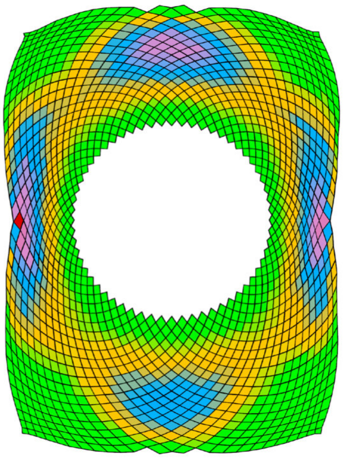


Figure 5.39: BM_Mesh 01, $rp_{i,max} = 0.03$, i = 2601

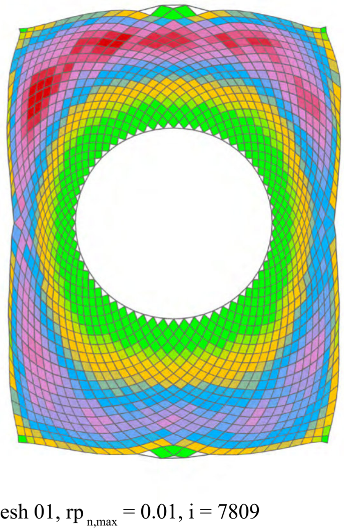


Figure 5.40: BM_Mesh 01, $rp_{n,max} = 0.01$, i = 7809

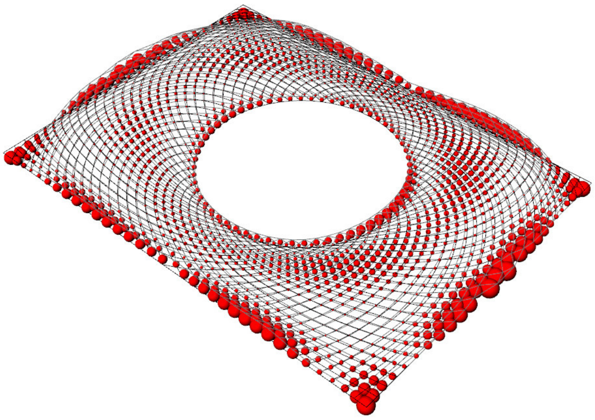


Figure 5.41: BM_Mesh 01, $w_{7809,max} = 5.37$ m

For the full extend of optimization results please refer to Appendix A3 to A6.

As expected the optimization shows slow convergence and requires large deformations from the starting mesh to achieve the planarity target. The edges of the faces are showing very little alignment with the principle curvature lines. Hence this test substantiates the necessity of a starting mesh of which edges are aligned with the principal curvature directions.

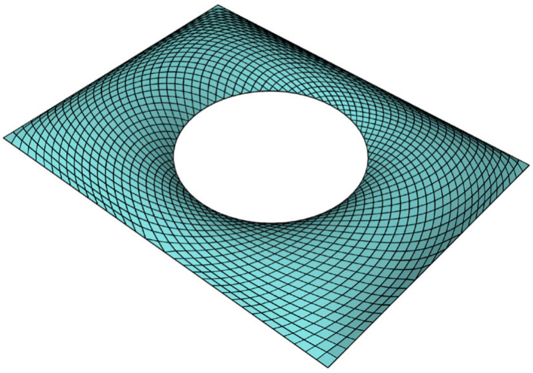


Figure 5.42: BM_Mesh 01, before optimization

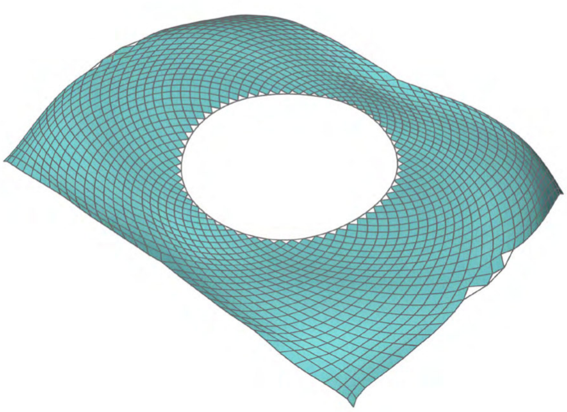


Figure 5.43: BM_Mesh 01, after optimization

5.5.2 BM_Mesh 02: Radial Tessellation free

For the next optimization test we will generate a mesh where most of the mesh edges are aligned with the LPC's oriented in radial direction to the centre of the roof. The overlay with the LPC sketch is shown in Figure 5.44. The overall number of faces counts to 2048 with an average edge length of 2.0 m.

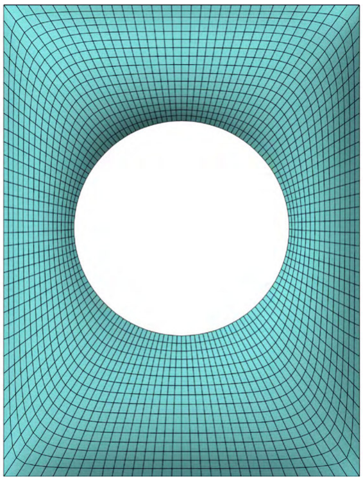


Figure 5.44: BM_Mesh 02, top view

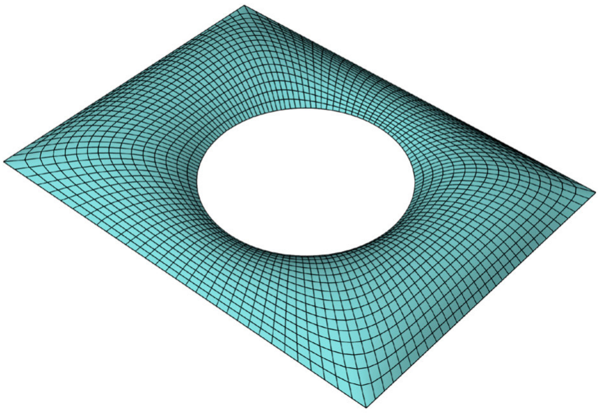


Figure 5.45: BM_Mesh 02, iso view

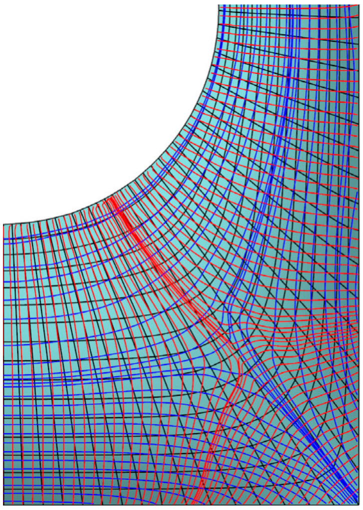


Figure 5.46: BM_Mesh 02, LPC graph overlay

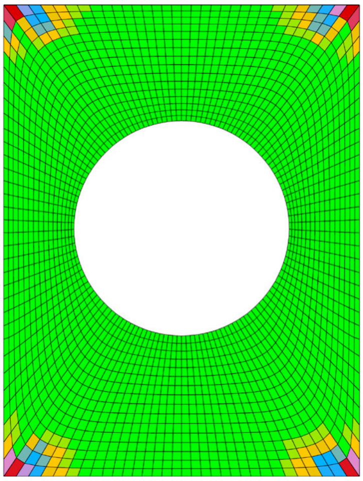


Figure 5.47: BM_Mesh 02, Planarity $rp_{n,max}$ = 0.23, i = 0

The most non - planar faces ($rp_{n,max}$ = 0.23) are again located in the four corners of the roof. Here we need to accept faces with an edge length of 4.0m.

The maximum face continuity angel of the staring mesh counts to $fc_{n,max}$ = 28°. This value should be seen as the reference value in order to judge the mesh smoothness after the optimization.

To understand the general behaviour during the optimization we execute the algorithm first without any edge constraints. The optimization is showing fast convergence requiring only 121 iterations to reach $rp_{121,max}$ = 0.03 relative planarity (Figure 5.49). In order to reach the planarity target of $rp_{i,max}$ = 0.01 the convergence slows down considerably whilst requiring 2664 iteration (Figure 5.50). The maximum deformation $w_{2664,max}$ = 1.97 m (Figure 5.51) occurs in the corners of the mesh which is coherent with the non - planarity distribution. The faces in this area need to move mainly in z - direction forming a spherical shape.

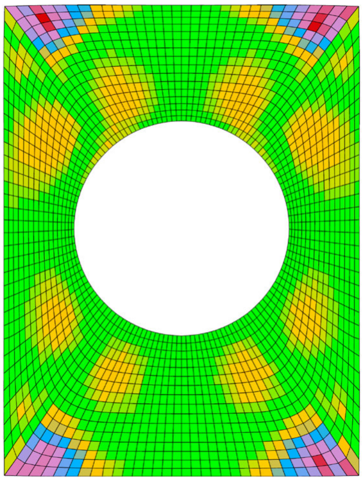


Figure 5.48: BM_Mesh 02, free, $rp_{i,max}$ = 0.05, i = 24

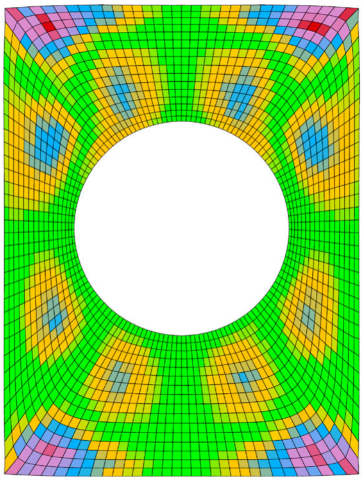


Figure 5.49: BM_Mesh 02, free, $rp_{i,max}$ = 0.03, i = 121

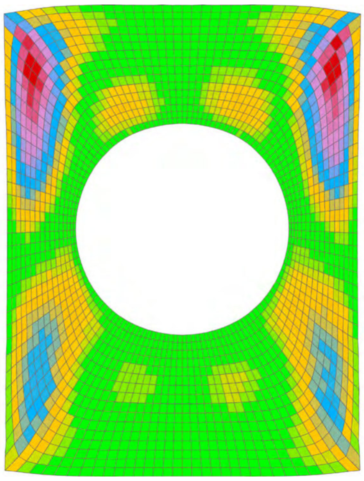


Figure 5.50: BM_Mesh 02, free, rp $_{n,max}$ = 0.01, i = 2664

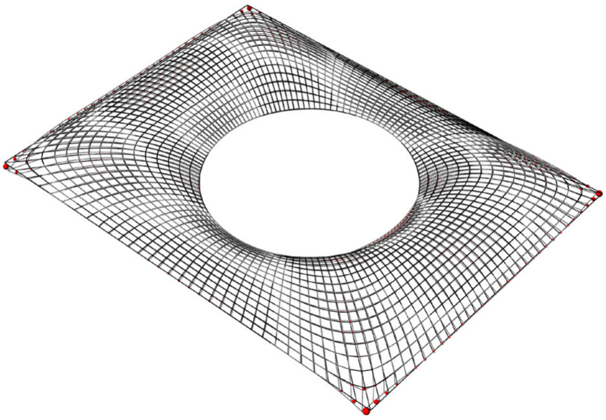


Figure 5.51: BM_Mesh 02,free, displacements, w $_{2664,max}$ = 1.97 m

In plan view the mesh keeps almost the original square perimeter layout. The face continuity improves considerable from previous fc$_{0,max}$ =28° to now fc$_{2264,max}$ =9° . Edge continuity degrades slightly from previous ec$_{0,max}$ $_{=}$40° to now ec$_{2264,max\ =}$ 45°.

However from practical reasons the displacements in the corners may not be acceptable. Therefore we perform a further optimization with the same mesh considering a fixed support along the inner circular and outer rectangular perimeter.

5.5.3 BM_Mesh 02: Radial Tessellation fixed

The optimization is showing very slow convergence requiring 13476 iterations to reach the target relative planarity value rp $_{13476,max}$ = 0.01 (Figure 5.54). The maximum deformation w $_{13476,max}$ = 3.00 m (Figure 5.55) occurs in the corners in upwards direction.

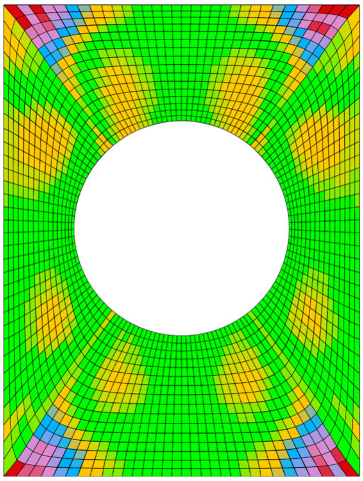


Figure 5.52: BM_Mesh 02, fixed, rp $_{i,max}$ = 0.05, i = 176

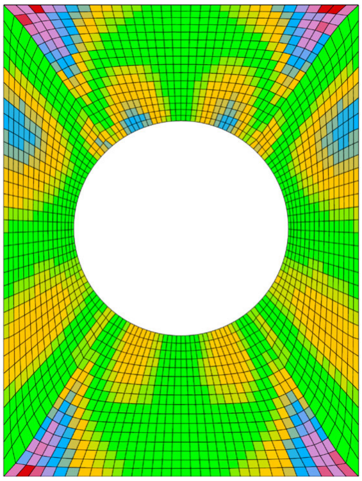


Figure 5.53: BM_Mesh 02, fixed, rp $_{i,max}$ = 0.03, i = 1119

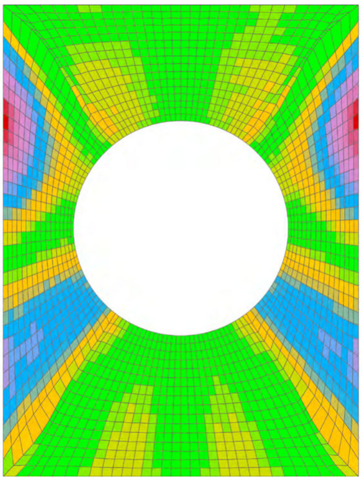


Figure 5.54: BM_Mesh 02, fixed, rp $_{n,max}$ = 0.01, i = 13746

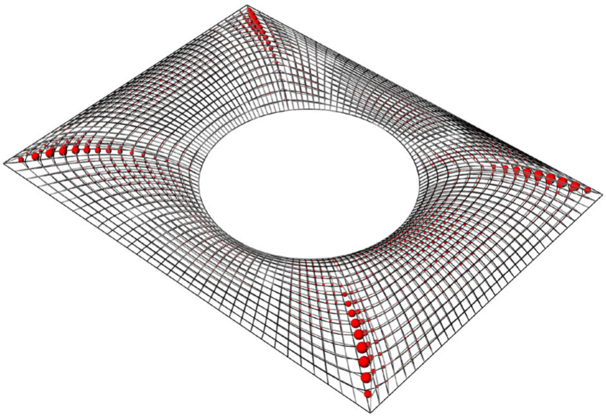


Figure 5.55: BM_Mesh 02, fixed, displacements, w $_{13476,max}$ = 3.00 m

The faces in this area are forming a ridge from the corner towards the centre of the circular opening.

There are significant distortions of the faces along the ridge to be observed.
Hence the face and edge continuity degrades from previous $fc_{0,max}$ = 28° to after $fc_{13476,max}$ = 45° . Edge continuity also degrades significantly from previous $ec_{0,max}$ = 40° to after $ec_{13476,max}$ = 130°. As the visual appearance of the distorted ridges is not acceptable and we will therefore focus on the a better solution for these areas in the next example.

5.5.4 BM_Mesh 03: LPC Tessellation

For the last meshing test we will use the LPC network as a guide for the meshing. This can be performed quite literally as the LPC's show a ordered distribution with only four singular points.

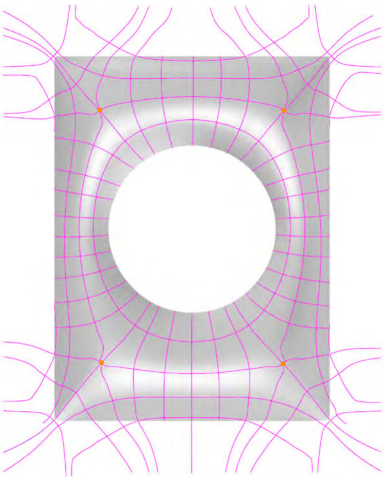


Figure 5.56: BM_Mesh 03, Guide LPC's

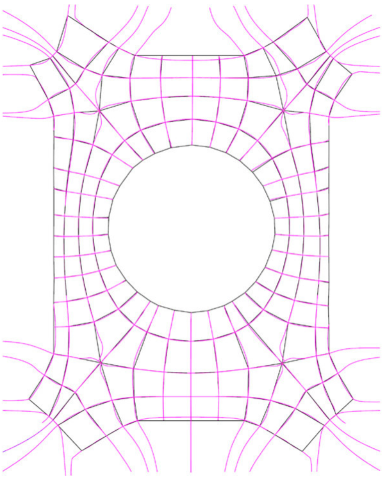


Figure 5.57: BM_Mesh 03, Subdivision 01

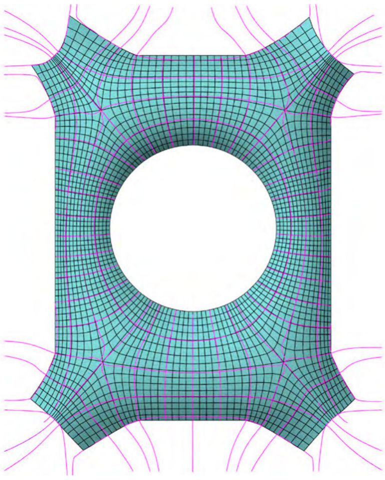


Figure 5.58: BM_Mesh 03, subdivided 3 iterations

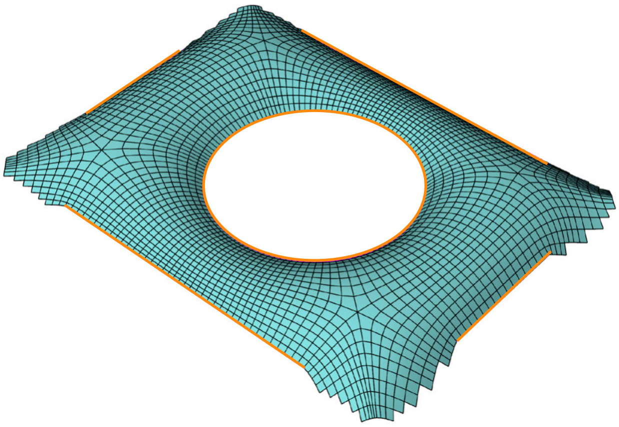


Figure 5.59: BM_Mesh 03, edge constraints

From the flock of LPC's we chose and/or generate LPC's with even spacing. The four umbilical points are from type Star and located symmetrically in the four corner areas.

In Figure 5.54 the guide LPC's are displayed which are used to layout the crude meshing. In order to locate the vertices with appropriate accuracy we apply one subdivision (Figure 5.57) to the initial topology mesh and relocate the vertices thereafter to match the intersections of the guide curves.
In the four corners we need to extend the starting mesh beyond the original perimeter as the LPC's are crossing the perimeter diagonally. To keep the mesh as close as possible to the target surface we need to strictly align the face edges especially in this area and truncated them after the optimization.
Two further subdivision iterations are applied in order to receive an average mesh size of 2.00 m (Figure 5.58).

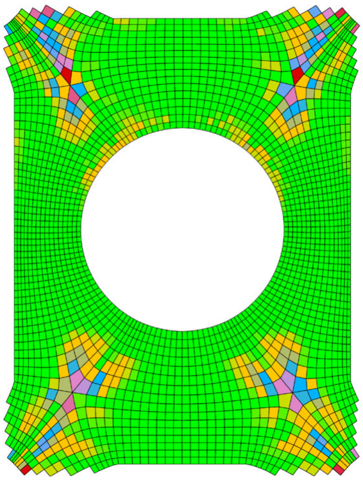


Figure 5.60: BM_Mesh 03, $rp_{i,max} = 0.083$, i = 0

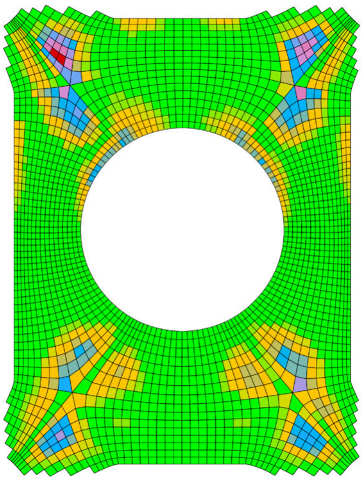


Figure 5.61: BM_Mesh 03, $rp_{i,max} = 0.03$, i = 25

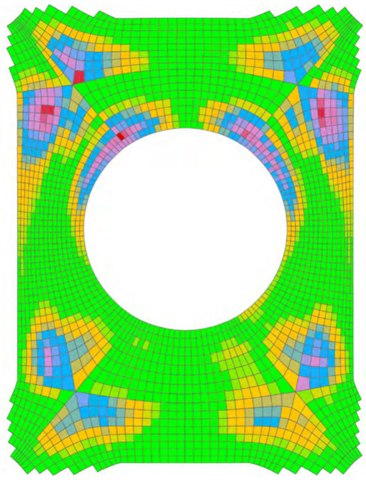


Figure 5.62: BM_Mesh 03, rp $_{i,max}$ = 0.01, i = 593

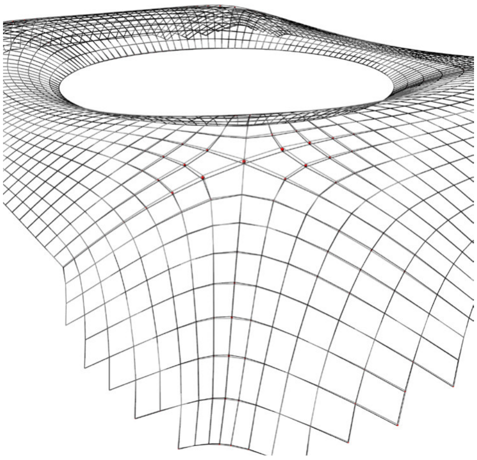


Figure 5.63: BM_Mesh 03, displacements, w $_{593,max}$ = 0.404 m

We constrain the inner circular perimeter and a portion of the rectangular outer perimeter leaving the corners free (Figure 5.59).
The most non - planar faces (rp$_{n,max}$ = 0.083) are located around the umbilical points as displayed i Figure 5.60. This is mainly due to the larger sized faces in this area.

The maximum face/edge continuity angel of the starting mesh counts to fc$_{n,max}$ =11.4°/ fe$_{n,max}$ =33.9°. These value should be seen as the reference value in order to judge the mesh smoothness after the optimization.

The optimization is converging very fast requiring only 593 iterations to reach the target relative planarity value rp $_{593,max}$ = 0.01 (Figure 5.62). The maximum deformation w $_{593,max}$ = 0.40 m (Figure 5.63) occur at the umbilical points in upwards direction.

The face and edge continuity degrade from previous fc$_{0,max}$ = 11.4° to after fc$_{593,max}$ = 18.2° . Edge continuity improves slightly from previous ec$_{0,max}$ = 33.9° to after ec$_{593,max}$ =32.5°.

The meshing which edges are closest aligned with the Lines of Principle Curvature delivers indeed the best results. The deflections are fairly small and continuity values change only slightly which are marks for a good starting mesh quality.

Finally we will summarize the results of the previous optimization meshing test in the following table:

	Mesh 01: Rhombic	Mesh 02: Radial	Mesh 02: Radial	Mesh03: LPC
Edge condition	free	free	fixed	fixed
No of faces	1618*	2048		2459**
Max face edge [mm]	3288	4759		4183
Min face edge [mm]	1139	622		332
Planarity subdiv [mm]	1305	452		148
Planarity subdiv [%]	57,85	22,95		**8,33**
Iterations 5%	457	24	176	**3**
Iterations 3%	2601	121	1119	**25**
Iterations 1%	7809	2664	13746	**593**
Displacement 5% [m]	4,765	0,715	0,863	0,094
Displacement 3% [m]	6,954	1,268	1,801	0,178
Displacement 1% [m]	9,23	1,97	3,000	0,40
Face Discontinuity [°]	21,60	28,20	28,20	11,43
Face Discontinuity 5% [°]	13,86	21,06	46,03	11,16
Face Discontinuity 3% [°]	12,17	14,96	51,77	11,30
Face Discontinuity 1% [°]	17,75	**9,17**	52,72	18,22
Edge Discontinuity [°]	-	40,06	40,06	34,47
Edge Discontinuity 5% [°]	-	38,42	56,21	34,44
Edge Discontinuity 3% [°]	-	37,64	122,59	33,97
Edge Discontinuity 1% [°]	-	45,71	130,49	31,55

Table 5.1: BM _Mesh 01 to 03 optimization summary

* Excluding triangular faces
** Including extended faces

5.6 Westfield Shopping Mall

The undulating roof surface of the Glazed Roof at the shopping centre Westfield/London is inspired by the wave movements of water in nature. Designed by the Buchan Group International pty Ltd and Benoy the roof shell covers the area of approx. 5,500m². The triangulated steel structure is based on a plan grid of equilateral triangles which is then projected on the doubly curved roof surface. The result of this meshing process are 2250 unique glazing units, 12,000 individual beams and 7,000 individual nodes. The beams are formed by welded box sections with common outer cross section dimension and varying plate thicknesses. The nodes are fabricated using CNC milling and composed of 20 individual parts in order to form a bolted connection.

In plan the roof is C - shaped with 72m long and 24m wide horizontal legs. The upright long side measures 120m and is 24m wide as well. The surface comprised of a series of waves joined with dome shaped forms at the corners.

In the following chapters we will investigate alternative PQ meshes for the roof glazing.

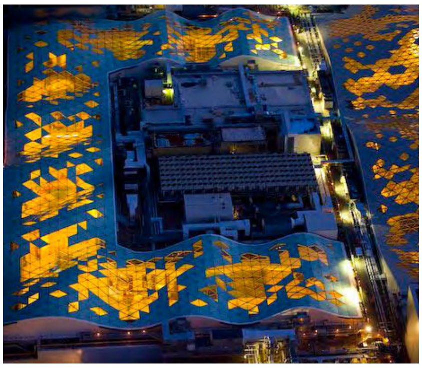


Figure 5.64: Westfield London [Image by Seele]

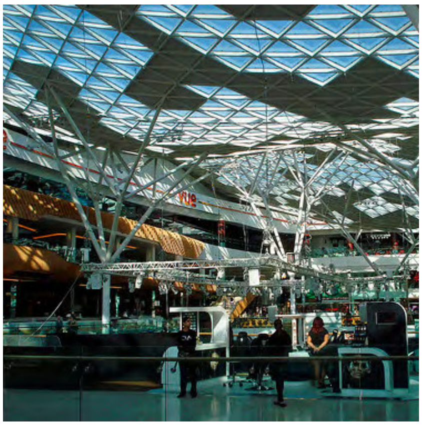


Figure 5.65: Westfield London [Image by Seele]

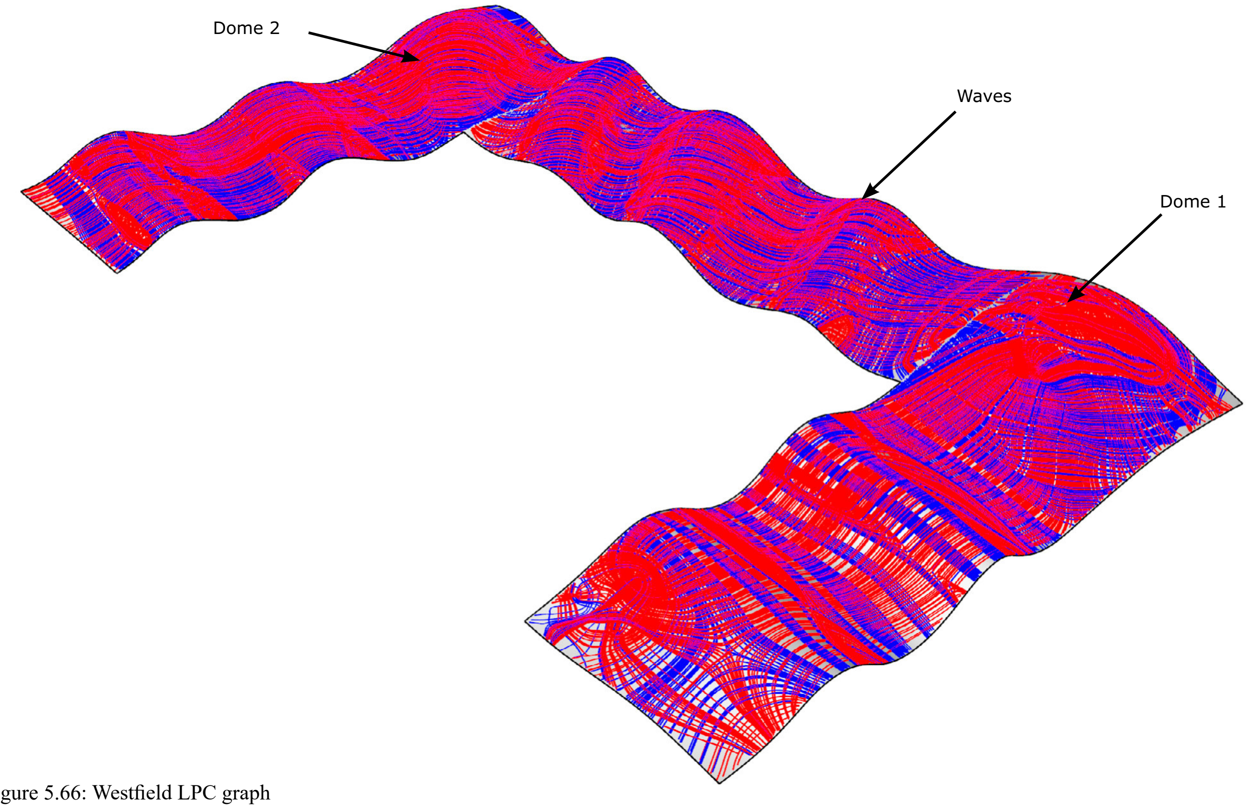


Figure 5.66: Westfield LPC graph

5.6.1 LPC Graph

In order to inform the layout of the topology mesh we compute the LPC graph of the roof surface (Figure 5.66).

We can observe a similar behaviour of the LPC's in the wave parts of the surface (Figure 5.67). The minimum (red) LPC's are generally following the valleys and ridges of the waves. Whilst the maximum LPC's (blue) then necessarily are follwoing the waves and crossing the valleys and ridges orthogonally. Along the ridges several umblical points of type lemon star and star are located which disturbs the in principal clear order of the LPC's. Whilst omitting the local distortion caused by the umbilical points this suggest that this part of the surface is fairly suitable for a meshing which is very close to the LPC graph.

In the dome areas we can observe a different behaviour of the LPC's. At the central area of dome 1 (Figure 5.68) the LPC's show a very noisy behaviour. Several umbilical points distort the graph so that is is not suitable as a literal guide for the topology meshing. Near to the lower part of the dome the LPC'c are more organized whilst showing a circular layout in maximum direction.

Dome 2 is characterized by a ridge spanning diagonally between the inner and outer corner of the perimeter. The maximum LPC's are running parallel here and are flanked by umbilics.

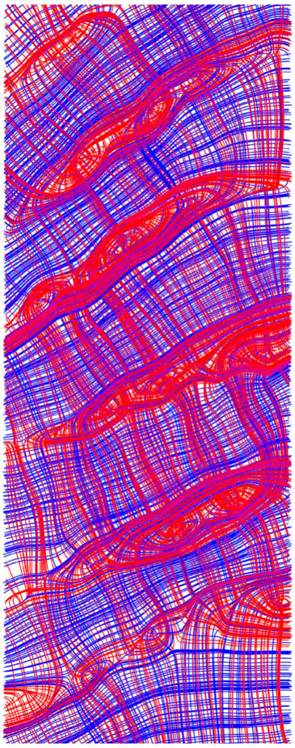


Figure 5.67: Typical wave LPC graph

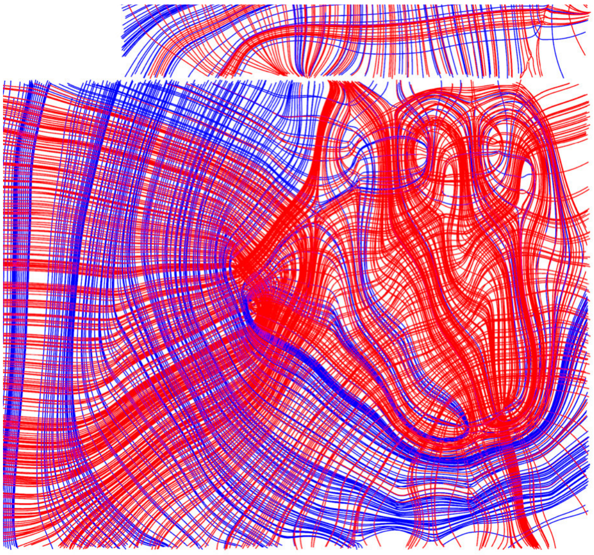


Figure 5.70: LPC graph Dome 1

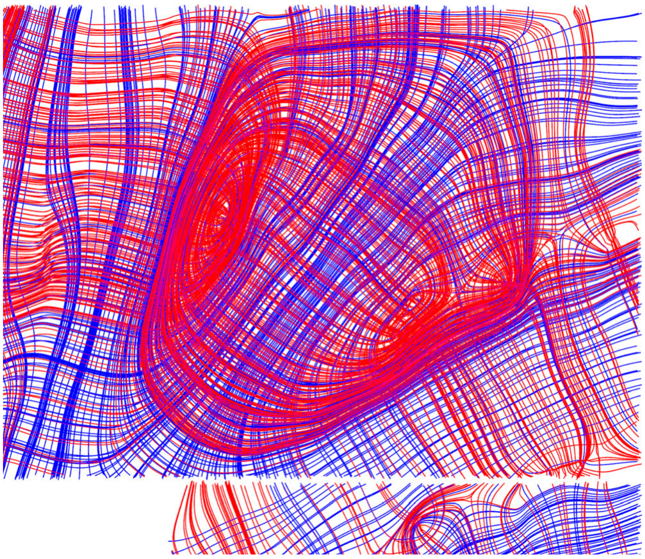


Figure 5.71: LPC graph Dome 2

5.6.2 West_Mesh 01: Rhombic Tessellation

Similar to the case study of the British Museum Court Yard Roof we explore a quad meshing version of the initial triangulated mesh. Two triangular mesh faces are joined to receive a rhombic shaped quad face. In order to control the perimeter in plan during the optimization we introduce a vertical constraint surface along the outline of the roof. All vertices located at the perimeter will stay on this surface during optimization. We exclude the triangular faces along the perimeter for the optimization process.
The most non - planar facets ($rp_{n,max}$ = 0.47) are located along one of the wave ridges. Consequently this occurs where the mesh edges show the least alignment whit the curvature lines (Figure 5.72).

The optimization target will be again set to 1% ($rp_{max,i}$ = 0.001) which is sufficient when using double insulating glazing units. The maximum face/ edge continuity angel of the starting mesh counts to $fc_{n,max}$ =33.4°/ $fe_{n,max}$ =25.7°.

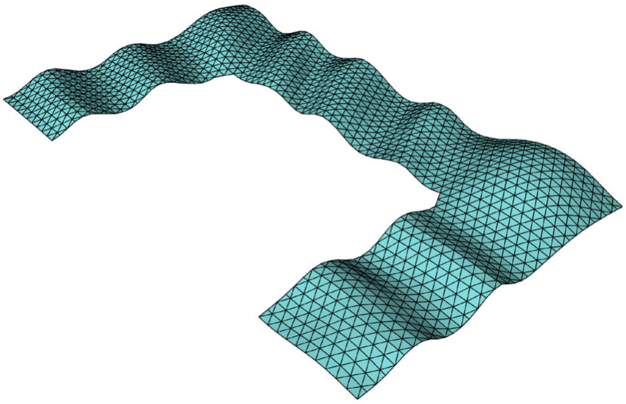


Figure 5.68: Original triangulated tessellation

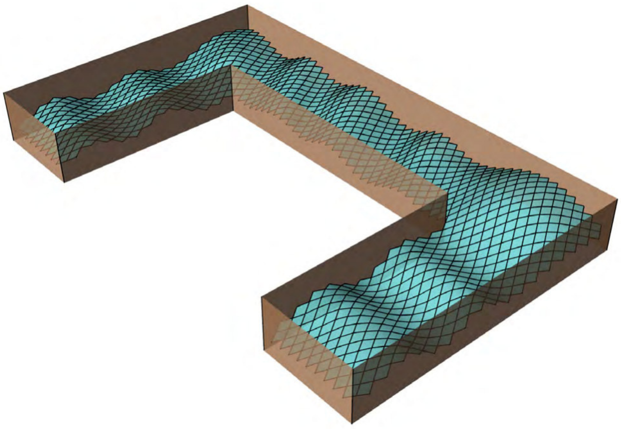


Figure 5.69: Rhomic Mesh with constraint surface

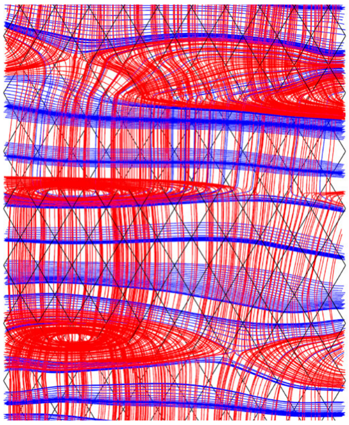


Figure 5.72: Mesh edge to LPC misalignment

The process shows average convergences in respect to the high non - planarity value of the starting mesh and the applied perimeter constraint.

To achieve the target planarity the mesh faces get highly distorted in the some areas. The edge continuity value decreases by 260% from fe $_{0,max}$ =25.7° to fe$_{2945,max}$ =69.2°.

The visual quality of the optimized mesh may not be acceptable for realization. The deflections from the target surface (Figure 3.77) is also fairly large ranging up to 3.00 m.

Hence we will explore more curvature line correspondent meshes in the following chapters.

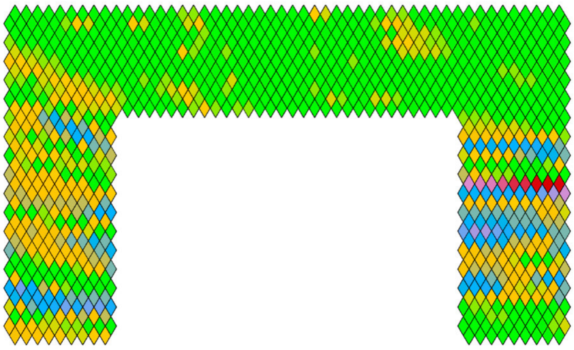


Figure 5.73: West_Mesh 01, Planarity rp$_{n,max}$ = 0.47, i = 0

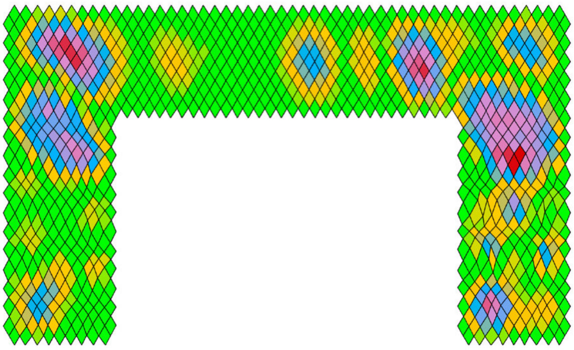


Figure 5.74: West_Mesh 01, rp $_{i,max}$ = 0.05, i = 163

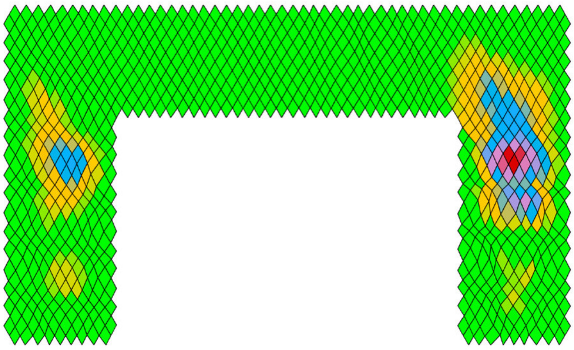


Figure 5.75: West_Mesh 01, rp $_{i,max}$ = 0.03, i = 813

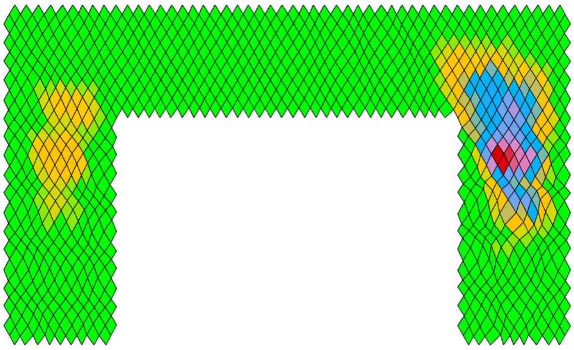


Figure 5.76: West_Mesh 01, rp $_{i,max}$ = 0.01, i = 2945

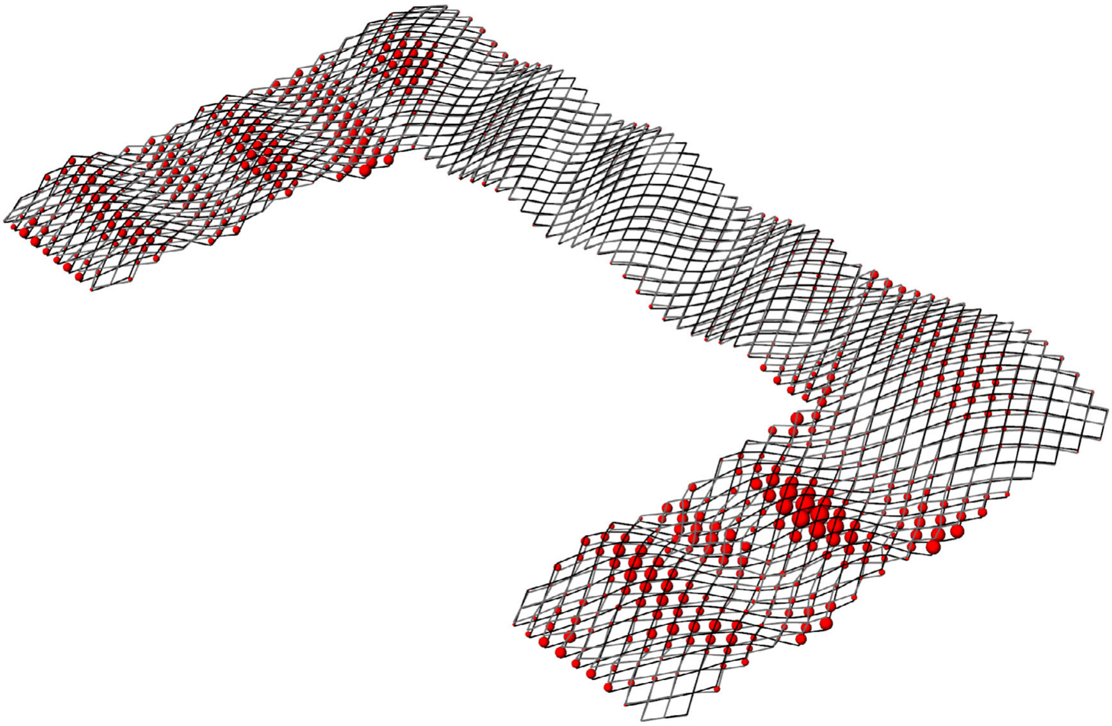


Figure 5.79: West_Mesh 01, displacements, w $_{2945,max}$ = 3.04 m

5.6.3 West_Mesh 02: Orthogonal Wave Tessellation

For the generation of the next mesh we will pick a series of maximum LPC's as guides. As shown in Figure 5.78 the guides are running in equal distance parallel to the wave ridges or valleys. The vertices of the crude topology mesh are the located at the ends and mid points of the guides. At the two domed areas we locate the vertices more freely in a centred position in order to receive an almost orthogonal layout of the mesh faces. We apply 3 subdivision iterations to receive 2304 faces with an average edge length of 2.00 m. We then again apply the perimeter surface constraint during the optimization.

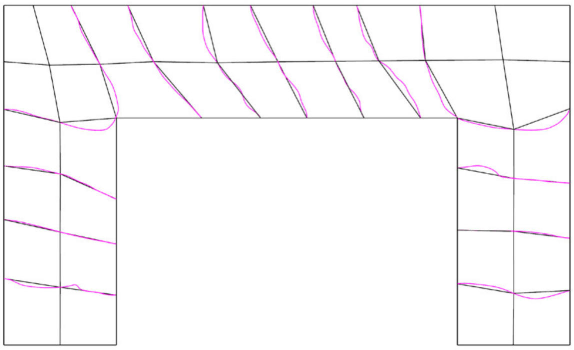


Figure 5.77: Guide LPC and Topology Mesh 02

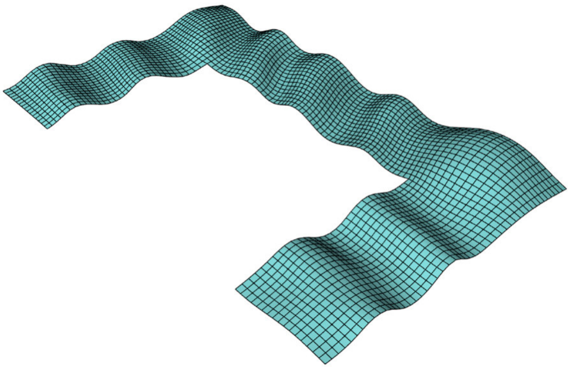


Figure 5.78: West_Mesh 02, n = 2304

The most non - planar faces ($rp_{n,max}$ = 0.114) are located scattered along the inner perimeter as displayed Figure 5.80. The maximum face/edge continuity angel of the starting mesh counts to $fc_{n,max}$ =27.5°/ $fe_{n,max}$ =27.5°.

The optimization shows average convergence requiring 1566 iterations to reach the target relative planarity value $rp_{593,max}$ = 0.01 (Figure 5.83). The scattered faces with the initial maximum non-planarity along the perimeter disappear fast. The non-planarity hot spot thereafter occur at the dome areas.

The face continuity improves to $fc_{1556,max}$ =19.6° whilst the edge continuity decreases only insignificantly to $fe_{n,max}$ =27.9°. The improvement of the face continuity is mainly related to the flattening of dome 2. As in this area the faces edge are crossing the LPC's diagonally large deformations are required to achieve the planarity criteria.

The maximum deformation $w_{593,max}$ = 1.08 m (Figure 5.84) occur along the ridge of dome 1 and in the centre of dome 2 in downwards direction. The visual appearence of the optimized mesh is quite acceptable and stays close to the original intent of the wave form. The explicit results of the optimization runs are documented in Appendix A7 to A9.

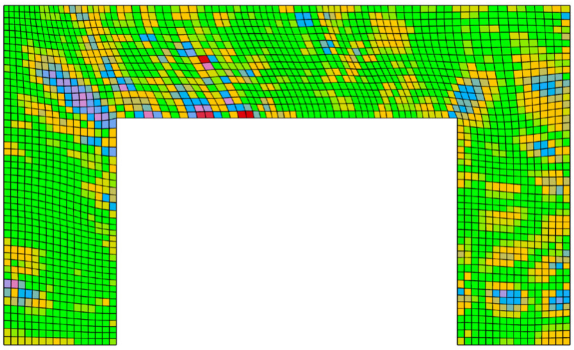


Figure 5.80: West_Mesh 02, Planarity $rp_{n,max}$ = 0.114 ,i = 0

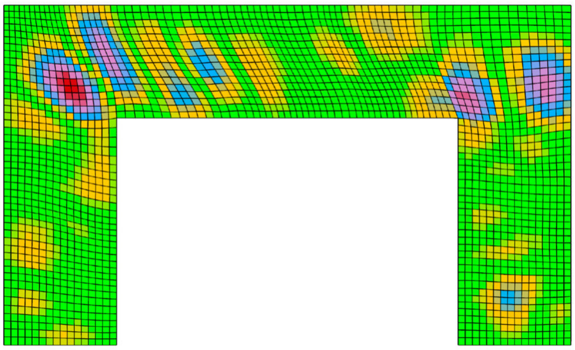


Figure 5.81: West_Mesh 02, $rp_{i,max}$ = 0.05, i = 124

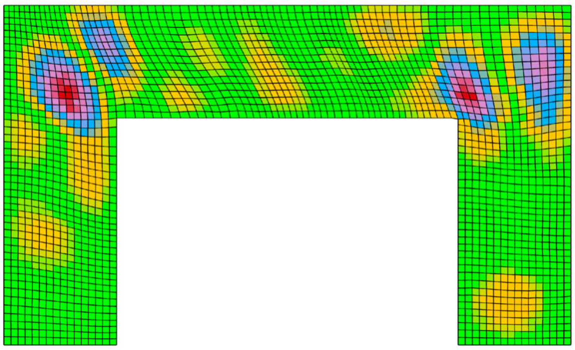


Figure 5.82: West_Mesh 02, $rp_{i,max}$ = 0.03, i = 566

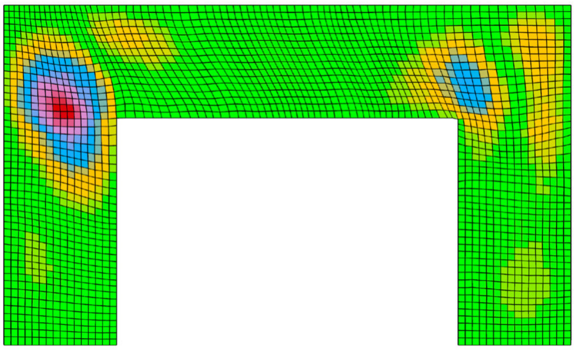


Figure 5.83: West_Mesh 02, $rp_{i,max}$ = 0.01, i = 1566

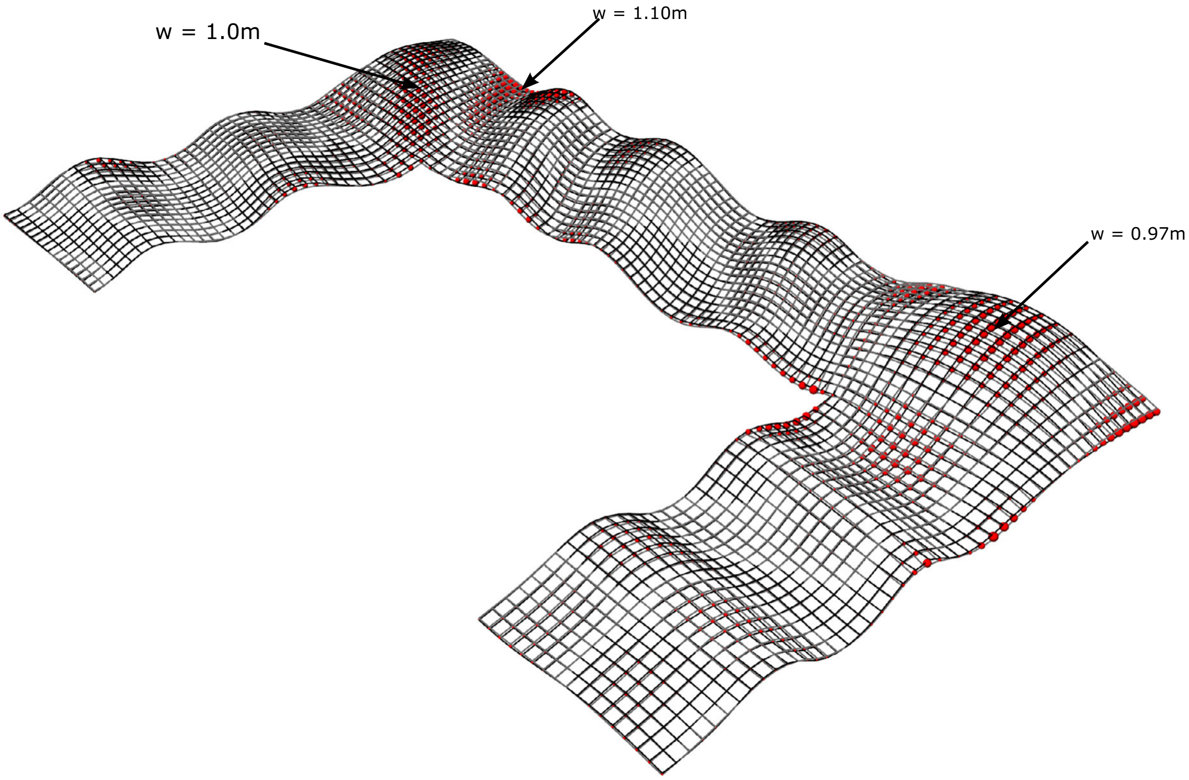


Figure 5.84: West_Mesh 02, displacements, w $_{1566,max}$ = 1.10 m

5.6.4 West_Mesh 03: LPC Tessellation

For the last meshing option we will aim to stay as close as possible to the LPC graph. Due to the fact that the LPC's are crossing the boarder of the surface at any angel we need to extend the starting mesh beyond the perimeter. The unnecessary faces may be deleted or trimmed after the optimization. The network of guide LPC's is displayed in figure 5.85. At the wave areas we aim to choose LPC's at similar spacing in order to receive a homogeneous mesh. The umblical points located along the ridges of the waves are diverting most of the maximum LPC's. Hence we cannot use an entire LPC as a guide. However, if we omit the distorted areas we can compose a interrupted but in general continuous guide from several LPC's as shown in Figure 5.86.

For dome 01 we use the LPC's which are running tangential to the centre of the dome until they start getting distorted by the multiple umblical points in this area. At the inner dome area we continue the mesh in an orthogonal layout.

The dome 2 is characterized by a ridge spanning across between the inner and outer corner of the perimeter. Here we can use the diagonally running LPC's as guides whilst ignoring the distorted regions. As these LPC's are fanning out towards the outer corner we may correct the mesh vertices after one subdivision iteration (Figure 5.88).

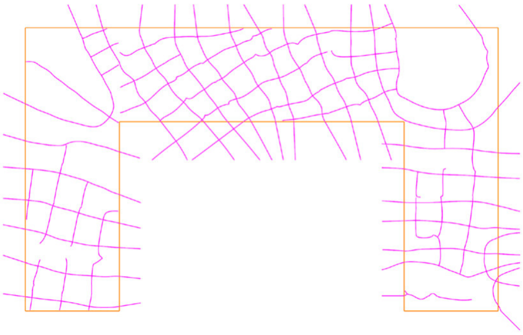


Figure 5.85: Guide LPC network

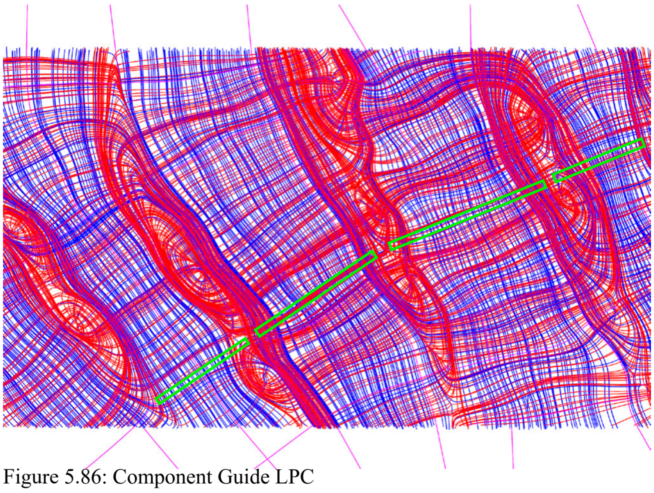


Figure 5.86: Component Guide LPC

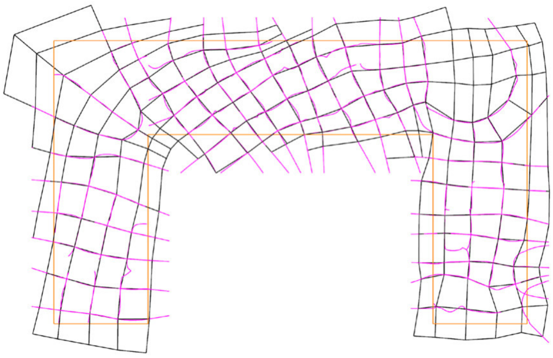


Figure 5.87: Crude mesh

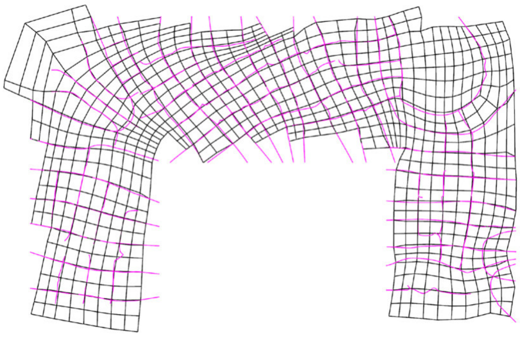


Figure 5.88: Subdivison 1 with corrections

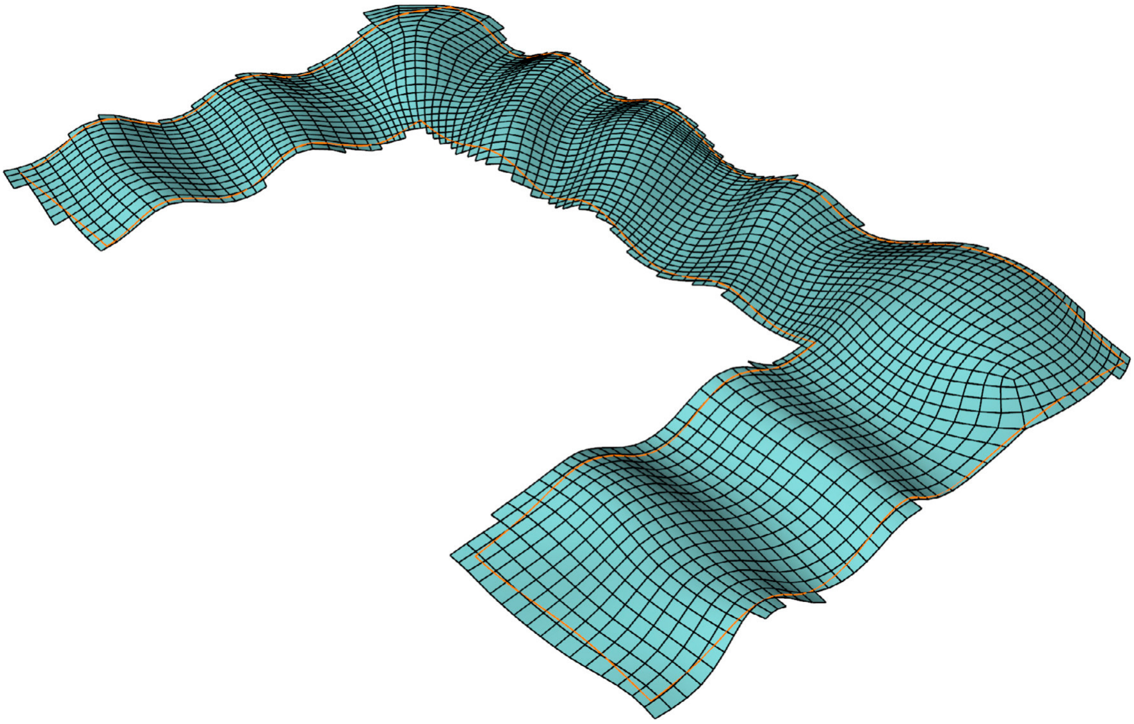


Figure 5.89: West_Mesh 03, n = 2358

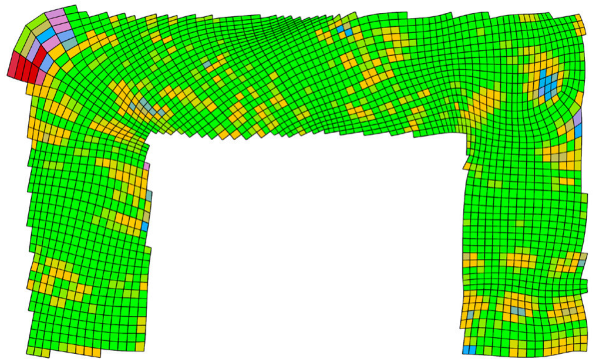


Figure 5.90: West_Mesh 03, rp $_{i,max}$ = 0.161, i = 0

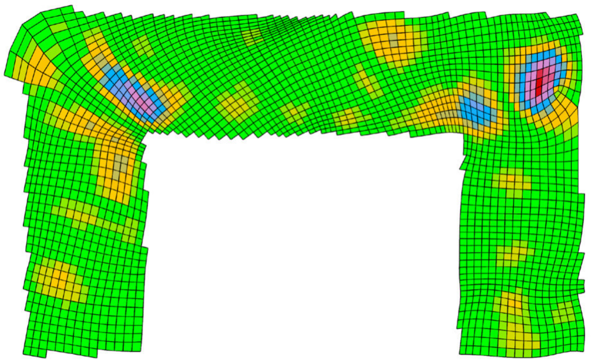


Figure 5.91: West_Mesh 03, rp $_{i,max}$ = 0.05, i = 128

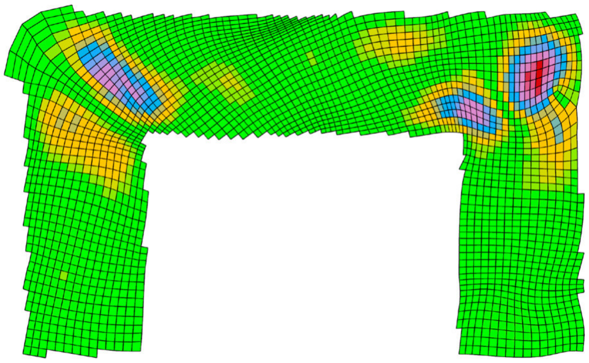


Figure 5.92: West_Mesh 03, rp $_{i,max}$ = 0.03, i = 700

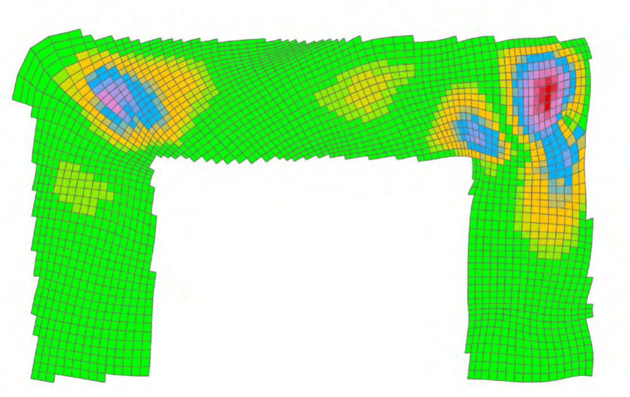


Figure 5.93: West_Mesh 03, rp $_{i,max}$ = 0.01, i = 3149

The most non - planar faces (rp$_{n,max}$ = 0.161) are located at the outer corner of dome 02. This is mainly causes by their elongated shape. As the LPC's are fanning apart towards this corner the shorter face edge follows respectively. The maximum face/edge continuity angel of the starting mesh counts to fc$_{n,max}$ =27.8°/ fe$_{n,max}$ =34.9°.

Due to the jagged perimeter of the mesh we can not apply sensible edge constraints. Here the approach will be that all faces crossing the original perimeter will be truncated after the optimization.

In order to reach the relative target relative planarity value of rp $_{128,max}$ = 0.05 (Figure 5.91) only 128 iteration steps are required.
this is mainly caused by the fact that the corner faces at dome 02, which show the maximum non-planarity, can move fairly free. The non-planarity hot spot the shift to more constraint areas at the centre of dome 01 and side of dome 02.

The speed of convergence the slows down in order to reach the 0.03 and 0.01 target values. The hot spots then consistently stay at the domed areas.

The face continuity improves to $fc_{700,max}$ =21.7° until the optimization reaches $rp_{700,max}$ = 0.03. During the iterations towards $rp_{700,max}$ = 0.01 the face continuity value rises back to the level of the staring mesh: $fc_{3419,max}$ =27.2°. This is caused by a slight ridge developing along the seam of dome 02 connecting to the horizontal wave area (Figure 5.92).

The edge continuity decreases slightly to $fe_{3419,max}$ =36.7° which mainly occurs along the jagged edge faces. As these faces will be truncated afterwards the deterioration can be ignored.

The maximum deformation in the mesh $w_{593,max}$ = 1.19 m occurs along a edge at dome 2 in downwards direction.

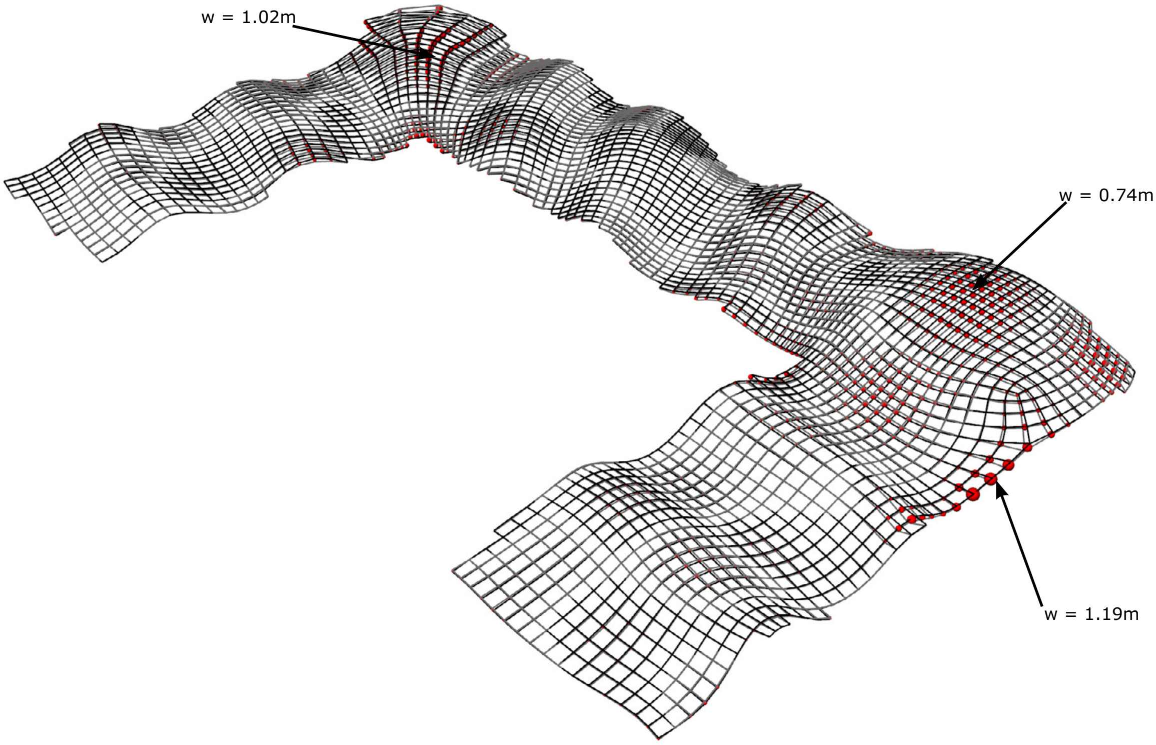


Figure 5.94: West_Mesh 03, displacements, $w_{3149,max}$ = 1.10 m

As previously mentioned we may trim all faces which extend over the original perimeter as shown in Figure 5.95. This approach obviously generates very small faces or faces with sharp angels. Some the resulting shapes may not be realizable due to technical limitations.

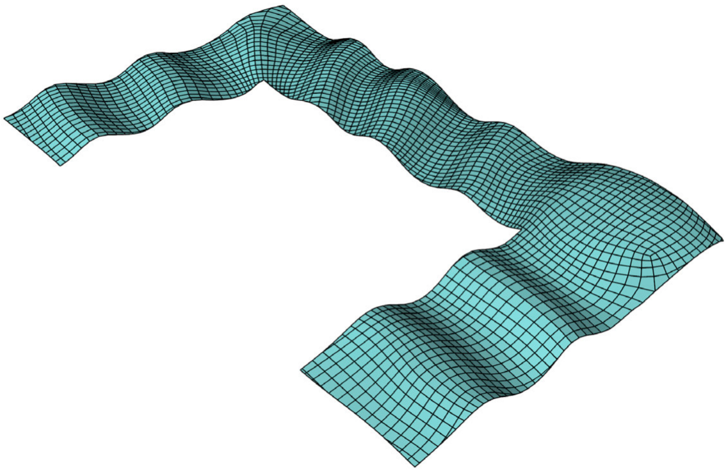


Figure 5.95: West_Mesh 03, truncated mesh

In Table 5.2 we summarize the results of the three meshing approaches for the Westfield Roof.
The best results delivered West_mesh 02 although there was not the most consequent alignment withe the LPC graph.
The much stricter approach in West_mesh 03 did not converge in better optimization result.
This is mainly because in the domed areas the LPC's showed a unsuitable fanning or distorted layout.

	Mesh 01: Rhombic	Mesh 02: Wave	Mesh03: LPC
Edge condition	fixed	fixed	free
No of faces	1011	2304	2358
Max face edge [mm]	2241	2149	6117
Min face edge [mm]	2743	1055	442
Planarity subdiv [mm]	1102	176	256
Planarity subdiv [%]	47.35	**11.75**	16.14
Iterations 5%	163	**124**	128
Iterations 3%	814	**566**	700
Iterations 1%	**2945**	3949	3419
Displacement 5% [mm]	2353	511	920
Displacement 3% [mm]	2794	786	1061
Displacement 1% [mm]	3042	**1103**	1197
Face Discontinuity [°]	34.40	27.86	27.83
Face Discontinuity 5% [°]	25.85	23.70	22.51
Face Discontinuity 3% [°]	30.16	22.68	21.74
Face Discontinuity 1% [°]	35.97	**19.25**	27.20
Edge Discontinuity [°]	25.74	27.86	34.85
Edge Discontinuity 5% [°]	34.89	23.64	36.88
Edge Discontinuity 3% [°]	29.75	22.62	37.74
Edge Discontinuity 1% [°]	69.17	**19.94**	38.68

Table 5.2: West _Mesh 01 to 03 optimization summary

Chapter 6 Conclusion

6 Conclusion

The motivation for this thesis arose from the practical experience in the building and construction business that to date no universal method is available which allows tessellating arbitrarily generated 3D free-form surfaces with quadrilateral planar elements (chapter 1 - Buildability of free-form architecture and its limitations). Available methods are restricted to either a very limited number of surface topologies or may require the manipulation of the input data.

Therefore this dissertation focused on the development of a computational tool which can process the input data provided by any given surface into data output comprising solely of planar quads suitable for the subsequent processing of building components.

This final chapter shall give an overview of the approach taken and the solution provided -what has been achieved and what could be future perspectives. In addition, limitations of the developed method are discussed and an alternative approach presented.

Finally a prospect is given on a number of questions that may be addressed with the further development of the optimization process in order to address specific requirements of the manufacturing industry.

5.7 Methodology and solution

The thesis introduces a novel generation procedure for the approximation of arbitrary surfaces with a PQ mesh and documents with a number of case studies how successfully the developed tool can perform surface approximations.

The process starts with an imperfect mesh consisting of planar and non-planar facets whose vertices are located exactly on the target surface.
In chapter 3 a manipulated version of the Dynamic Relaxation (Planar Relaxation) was proposed which optimises the mesh vertices towards their planar state. It could be proven that the iterative application to any unconstraint mesh will eventually find a solution consisting of almost planar faces.
If interior or exterior constraints are applied to the mesh during the opti-

misationthe process will be slowed down. Depending on the constraints under very rare circumstances it might be possible that no solution can be achieved. During the non-linear optimisation process the vertices will gradually move away from the target surface. Depending on the number of iterations the mesh faces may distort with the overall appearance of the mesh eventually degrading.

The best results can be expected if the geometry of the starting mesh is close to its planar version hence the quality of the starting mesh defines how satisfactory the result of the optimisation process will be. Ideally the edges of the starting mesh can be aligned with a network of conjugate curves.

In chapter 2 the mathematical properties of PQ facets and meshes are outlined in order to explain the nature of a ‘high-quality’ starting mesh. The meshing process starts with a topology mesh which consists of fairly large hand-sculpted faces. In order to achieve a refinement a subdivision algorithm (Catmull Clarke) is iteratively applied until the mesh faces could be reduced to the desired size. It could be proven that the CC subdivision is suitable for PQ meshing and delivers smooth and continuous starting meshes which are suitable as start geometry for the subsequent optimization process. Due to the algorithm’s nature the mesh face population has an exponential development. Hence the size of the individual faces of the crude mesh and the number of subdivision iterations need to be chosen carefully in order to achieve the desired mesh density. If the target surface comprises of a non-compact footprint where some areas require a higher number of crude faces(e.g. due to high curvature areas or umbilical points), the desired mesh density can only be achieved with workaround methods: the mesh may be extended over the target surface perimeter and then truncated after subdivision and optimization.

For future research the algorithm may be manipulated in such a way that other sequences of numbers than the pre-defined subdivision steps of 4, 16, 64, 256can be computed. In chapter 4.3 topology mesh layouts are proposed which reflect the four different types of umbilical points. In general umbilical points represent visually unpleasing features which should be avoided. For visual and meshing specific reasons it is recommended to sidestep the exact re-meshing of umbilical features: Especially in close proximity of the star type point the CC subdivision tends to generate mesh faces which are elongated and too large in size. Stretching of the faces can be avoided by locating the adjacent vertices closer to the umbilical point than proposed by the LPCs as the newly generated vertices will move away during subdivision and smoothing.

Minimizing of the distortion caused by the subdivision algorithm near umbilical points may also be an interesting topic for further research.

For the PQ meshing of the Westfield Roof in London a successful option was proposed which does not consider the umbilical features of the target surface at all. Only the general direction of LPC’S is considered for the topology mesh layout.

As demonstrated in chapter 2.5 the Lines of Principle curvature represent a suitable guide for the layout of the topology mesh. In chapter 4.2 a robust method for the generation of Principle Curvature Lines of any NURBS surfaces is proposed. It is understood that the generated curves

are only approximate and cannot withstand the exact mathematical definition of a Principle Curvature Line. Occasionally it can occur that the curvature information provided by the 3D CAD Software (Rhino) may locally be inaccurate in terms of the correct sign (Min, Max – curvature direction). In order to generate continuous curves these direction signs will require correction during the approximation process. Despite of the inaccuracies the LPC maps have proven to provide sufficient data to inform the sculpting of the topology mesh. The Umbilical points on the target surface already appear roughly in the right location if a reasonably dense grid of Principle Curvature Lines has been generated. For a more accurate detection a robust method using adoptive quad tree composition is proposed in chapter 4.2 and proves to deliver meaningful results.
Chapter 2.4.6 states that any surface comprises of infinite conjugate curve networks. As the Principle Curvature Directions are always pointing in orthogonal directions to each other it can be concluded that the LPC network is the first choice as a guide for PQ meshing.
However, for surface areas with a 'noisy' behavior of the LPC's the investigation of alternative conjugate networks should be the focus of further research: if we compare the generating curves of a translation surface with their Principle Curvature Lines it can be observed that the curve networks do not necessarily match. Especially for spherical shaped regions which naturally comprise of umbilical points an alternative conjugate network may provide a visually more pleasing and simpler PQ mesh. For the dome shaped surface areas of the Westfield Roof case study in (chapter 5.4.3) the orthogonal mesh layout provided better approximation results than the strict re-meshing of umbilical features.
In chapter 3 an overview of surfaces with inherent PQ meshing principles was compiled. With the knowledge about these already well researched PQ meshing surfaces the designer can use them as a kind of construction kit to compose a desired shape from different PQ mesh types or integrate one of them as a patch /patches of them into an otherwise free sculpted surface. The patch transition will have to be smoothed out and planarized using the proposed tools (CC- Smoothing, Planar Relaxation).

5.8 Future directions

Finally it is suggested that the investigation of a 'bottom up' method could be an interesting topic for further research:
The vertices of the starting mesh are not located on the target surface. In contrary to the suggested 'top down' method we start with a perfect PQ mesh whose vertices are located within a certain distance from the target surface. Each vertex is then attracted (pulled) towards the surface whilst applying a user-defined 'force'. By being moved/forced onto the desired shape the initial mesh faces might lose their planarity. The Planar Relax Algorithm can be applied to re- establish the PQ mesh status. To improve the appearance a CC- smoothing iteration can be applied before planarization. By repetitively executing the 3 steps of 'Attraction, Smoothing and Planarization' the starting mesh will be sequentially moulded into the target surface which is acting as the die cast while retaining planarity of its faces independent of the number of iterations. Once the desired shape

has been accomplished the process can be stopped at any time.
As displayed in chapter 5.6 umbilical points of the input surface can cause an irregular appearance of the approximation mesh. This problem can be avoided with the bottom-up method for which the umbilical points of the input surface do not have to be taken into consideration when generating an approximation mesh. Yet the optimization process applied to the approximation mesh may deliver a satisfying result.

5.9 Final reflections

It has been proven that the developed method can be applied successfully to any arbitrary surface. Its application is numerically robust and without exception will deliver a result which can be used for subsequent processes.
Furthermore the case studies conducted have confirmed that the algorithm proposed for the application performs well and achieves good results in terms of planarization of the approximation mesh.
However to improve the properties of the output geometry in terms of buildability and cost a number of properties could be implemented into the optimization process:

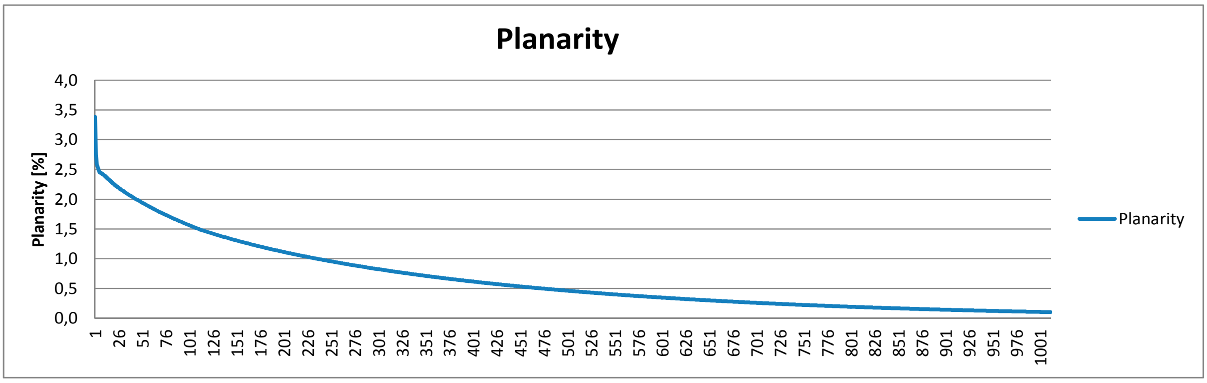
-reduction of torsion within the mesh nodes
-faces with similar edge length
-repetitive elements
-nearly rectangular corner angles

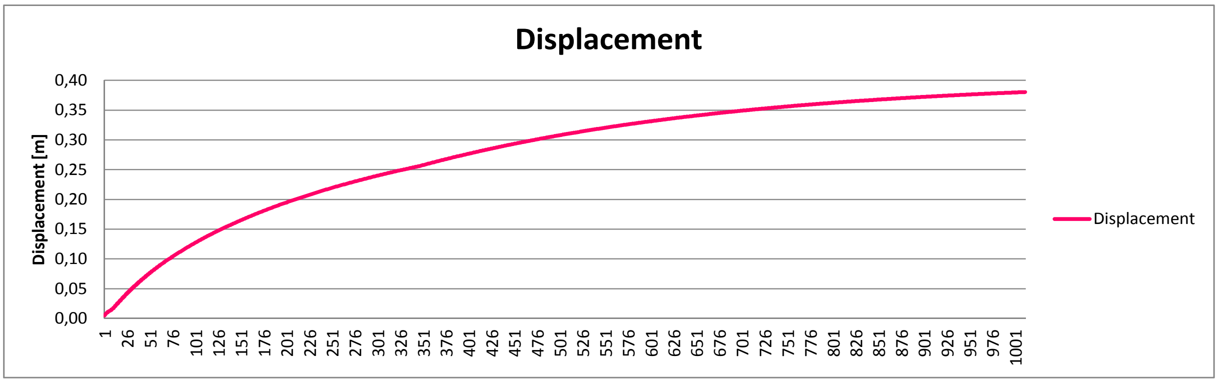
In conclusion, it can be said that the proposed method is not meant to be an end result, but intended to provoke interest and ambitions for continued studies and research in the general field of PQ meshing as well as its translation into the built environment.

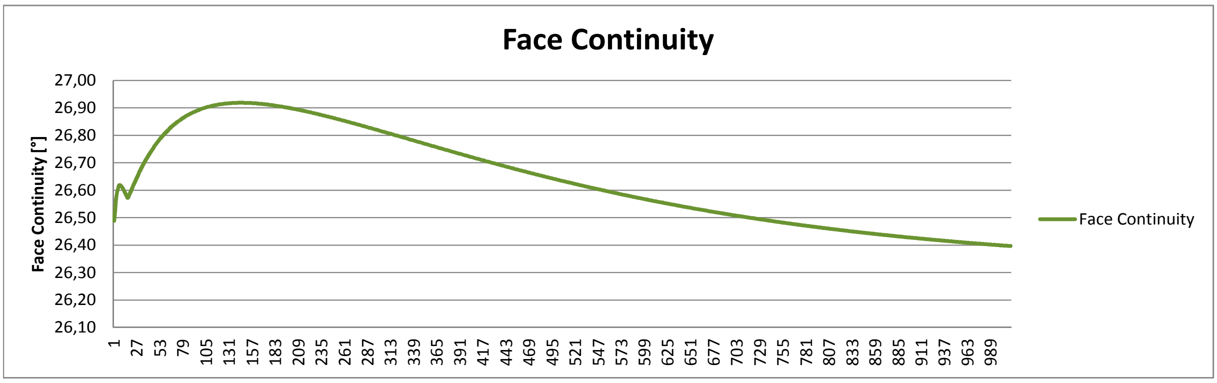
7 Appendix

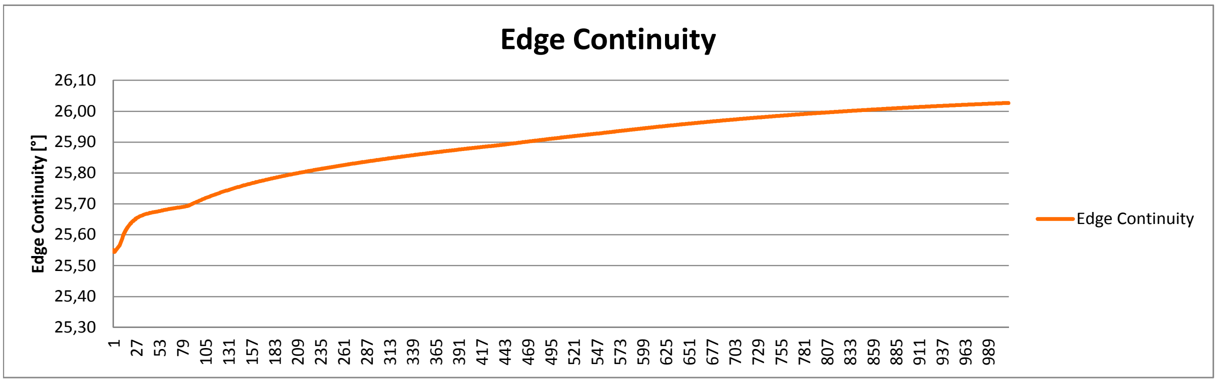
7.1 Graphs

Ellipsoid Mesh 01

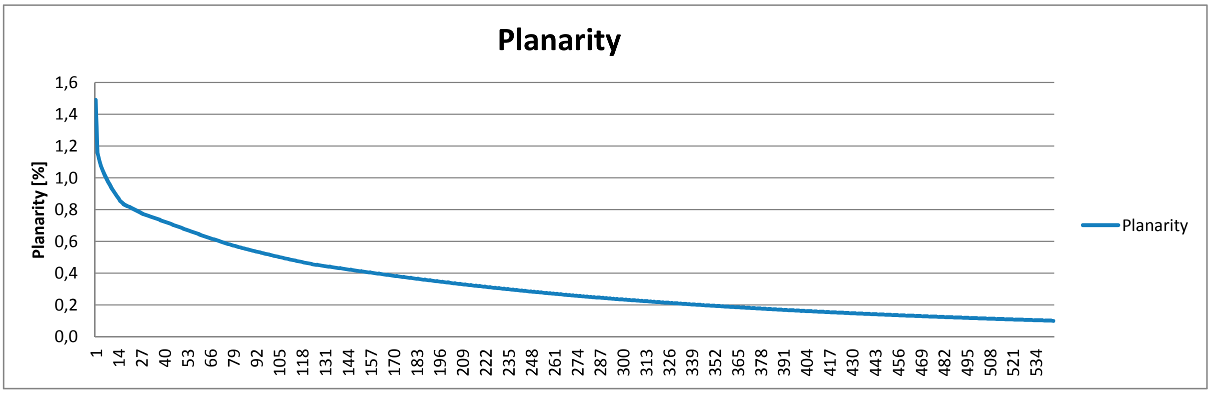


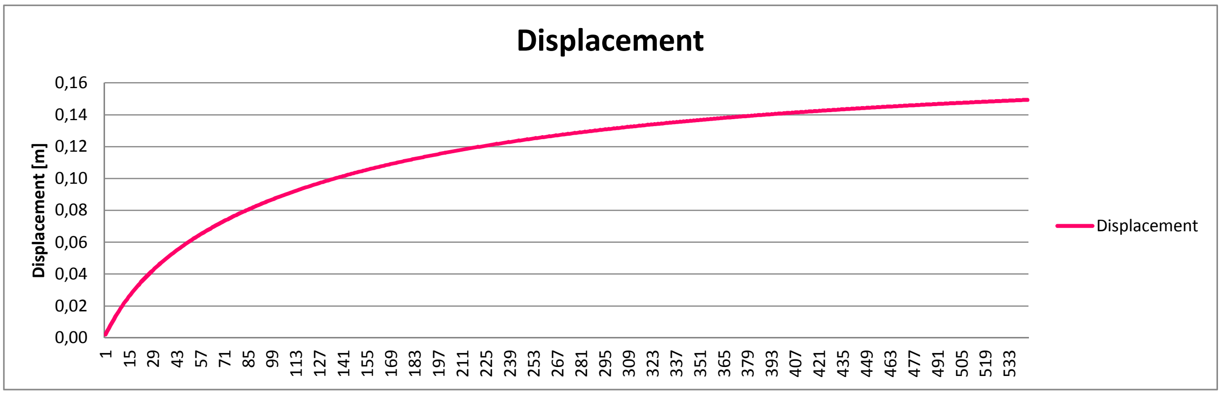


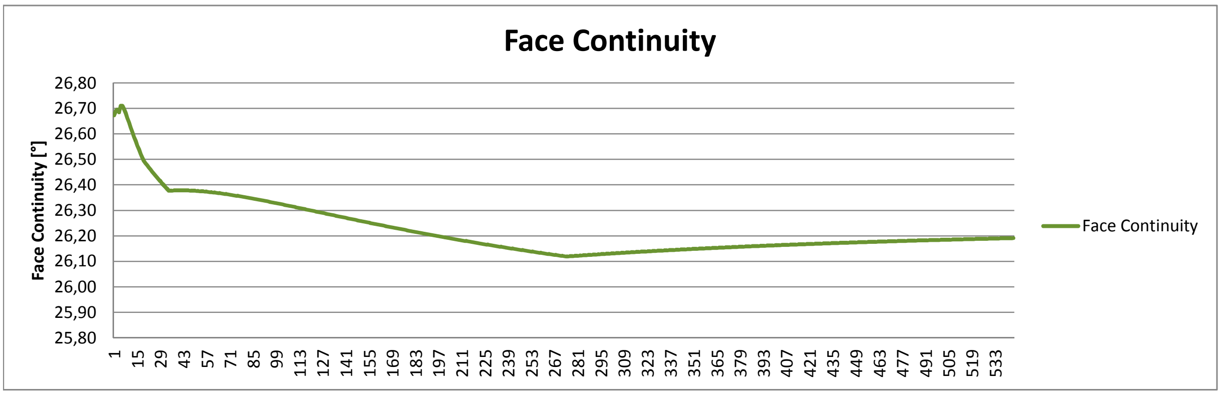


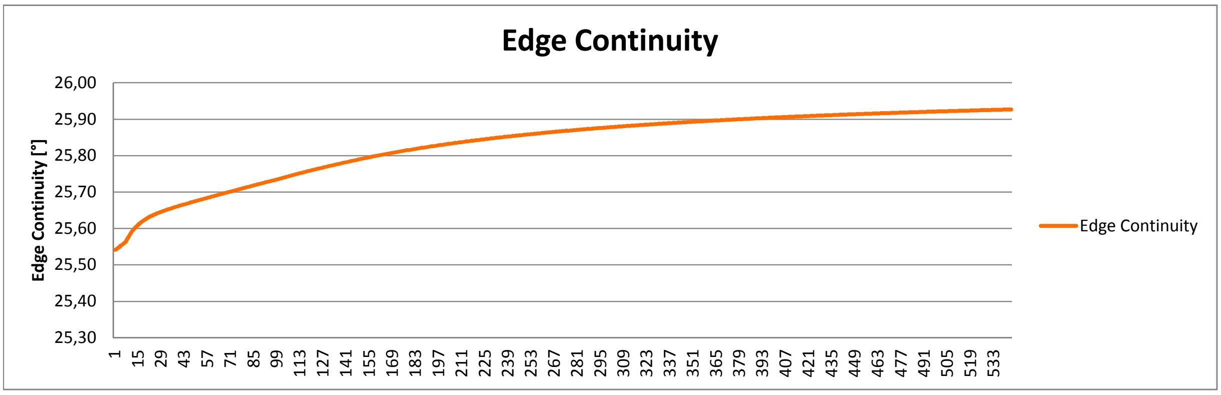


Ellispsoid Mesh 02

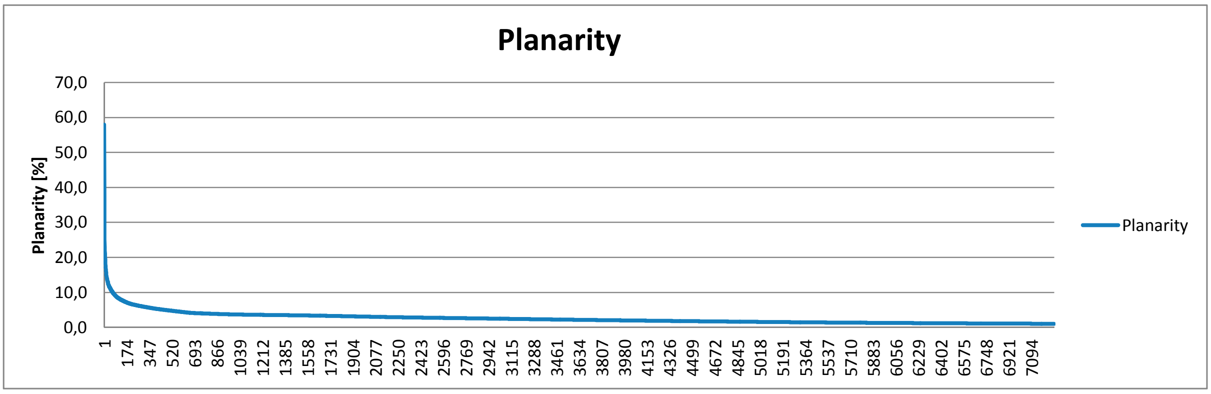


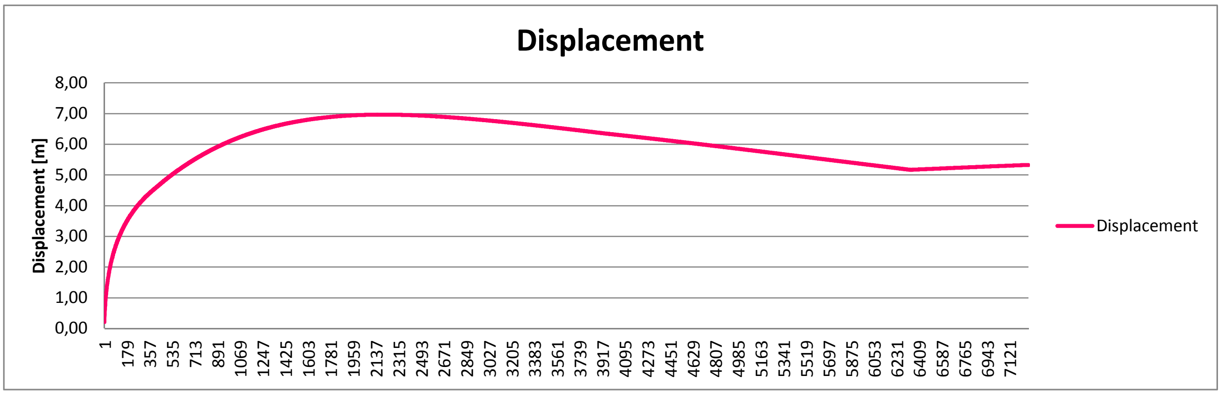


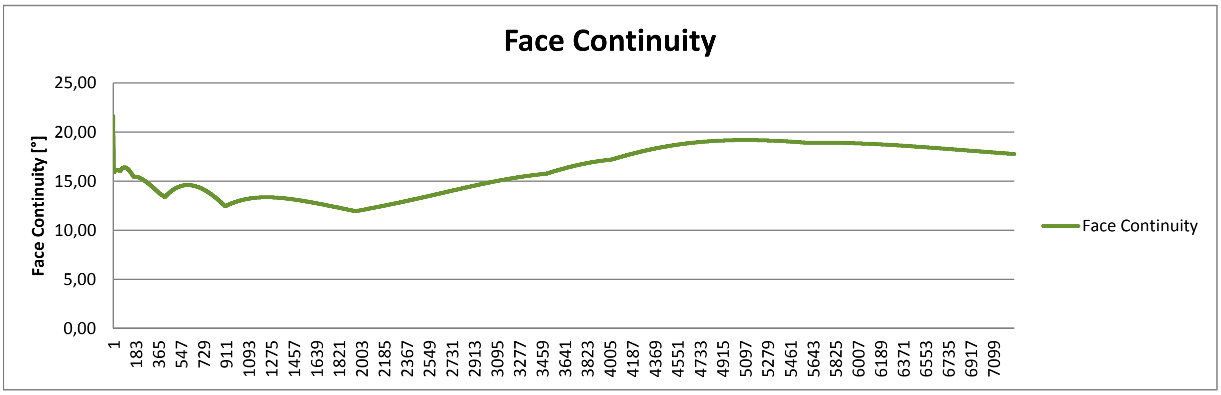


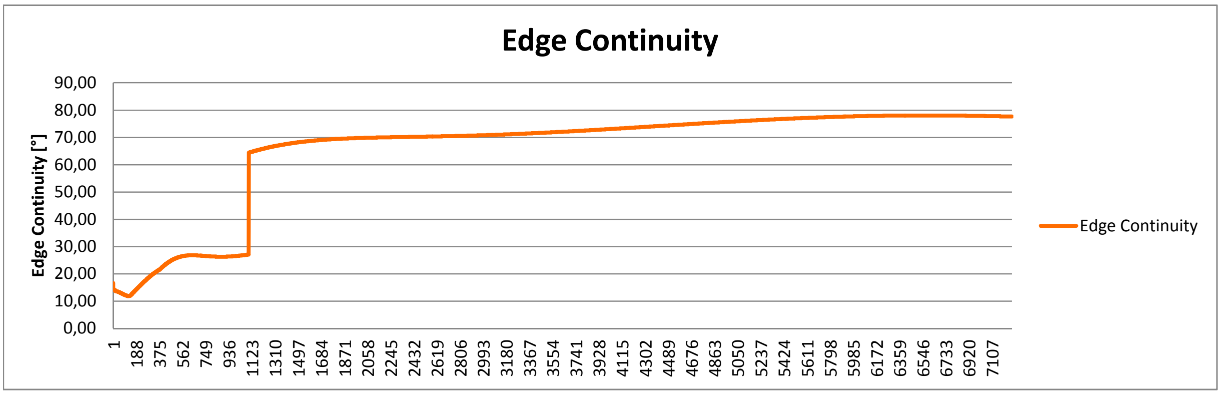


BM Mesh 01 (Rhombic)

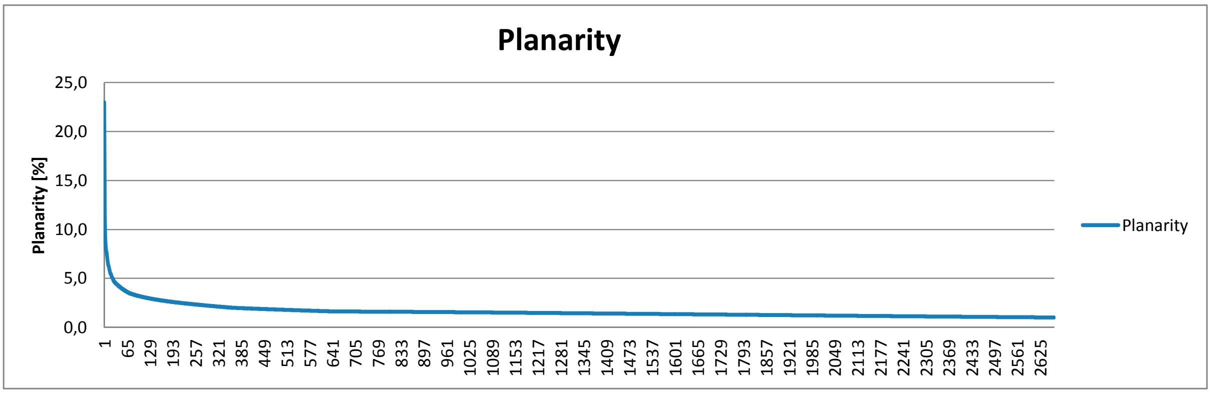


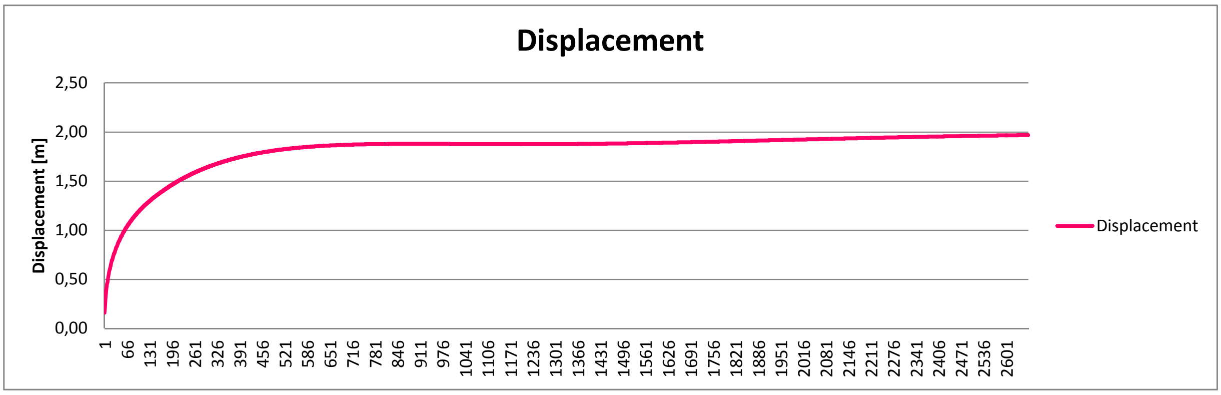


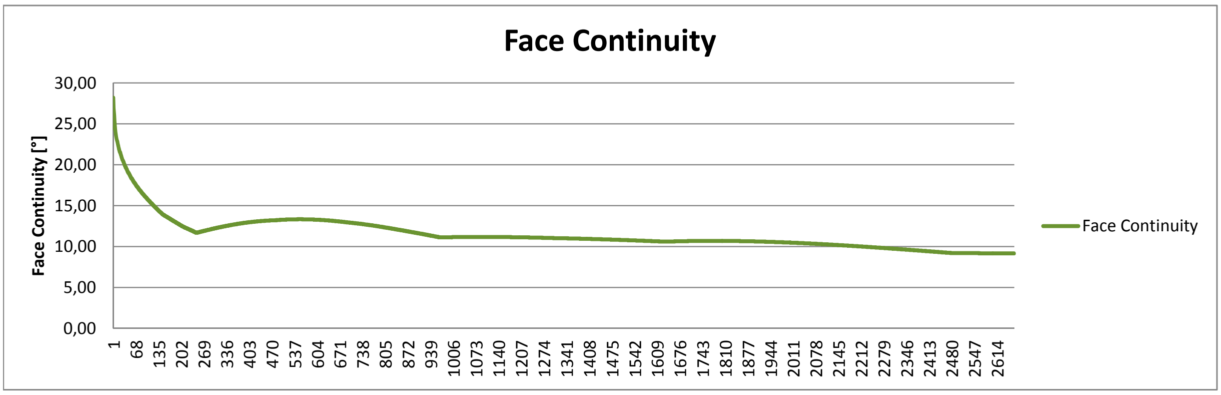


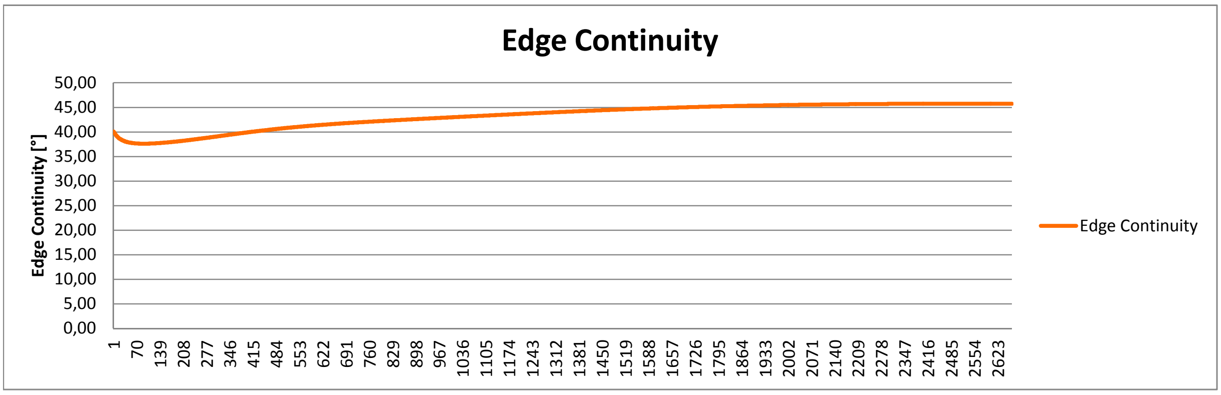


BM Mesh 02 free (Radial)

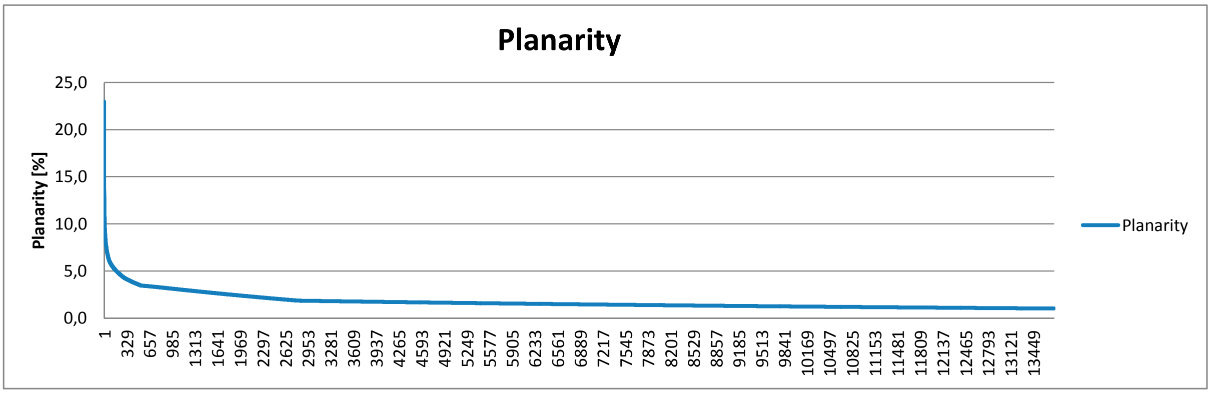


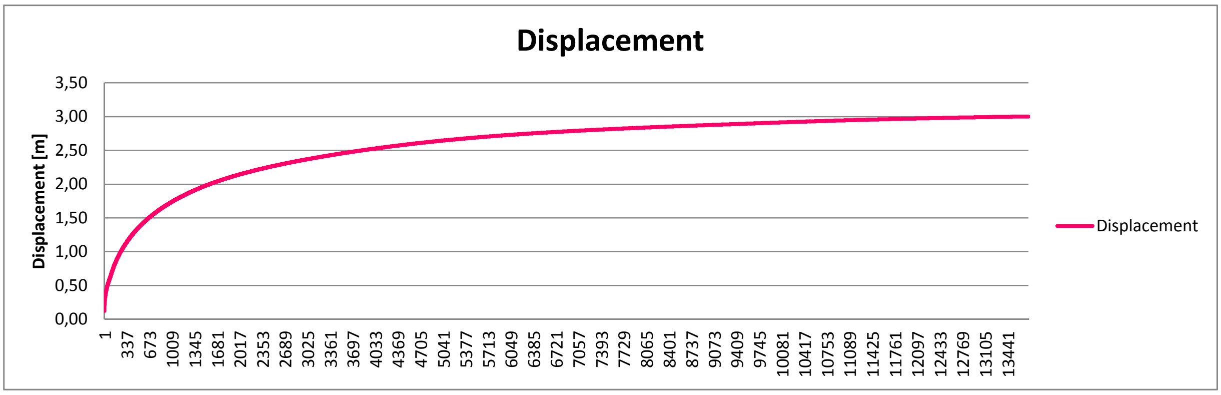


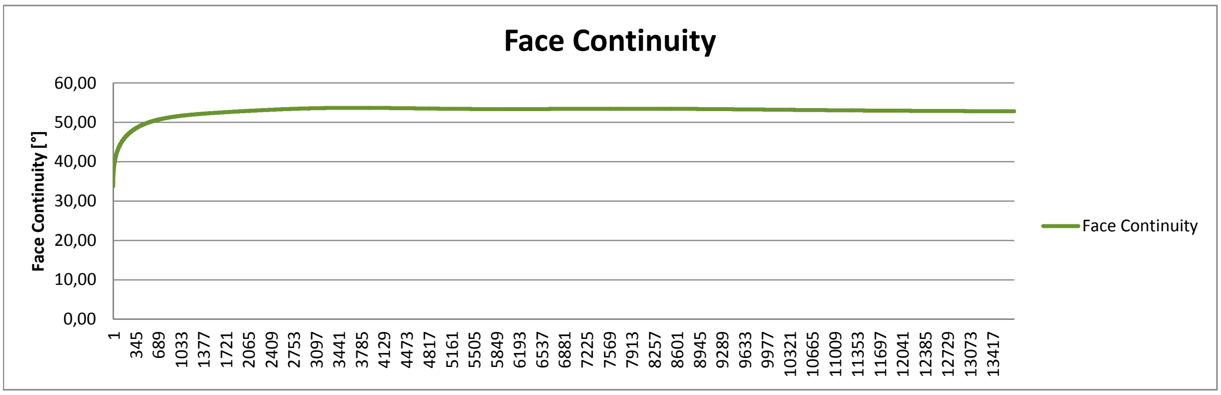


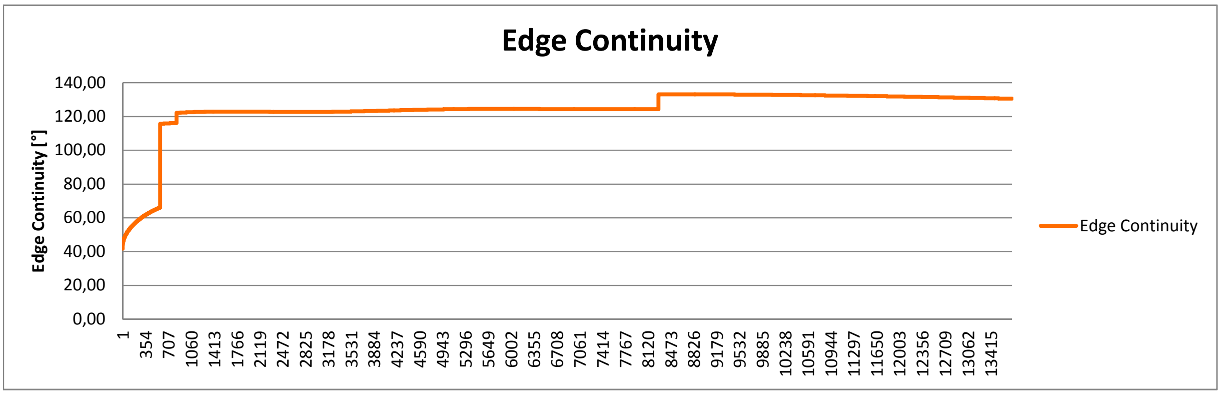


BM Mesh 02 fixed (Radial)

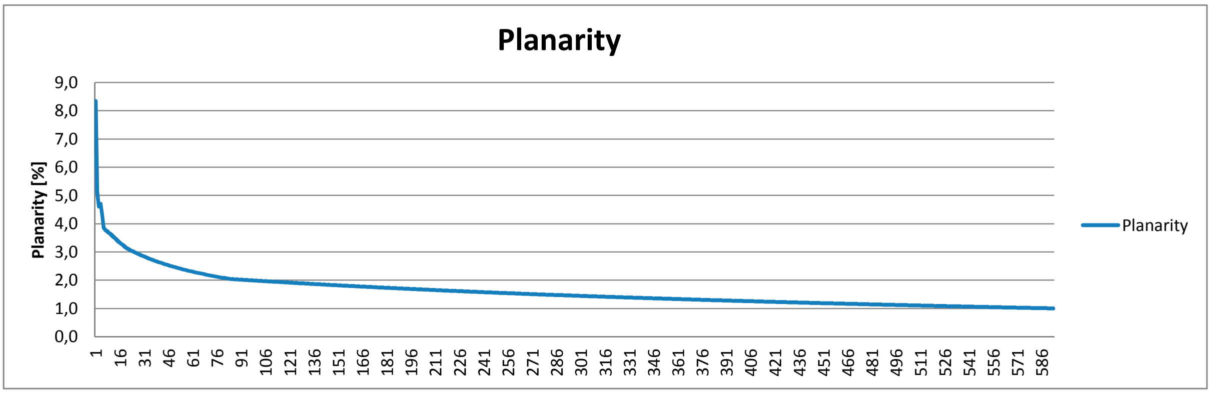


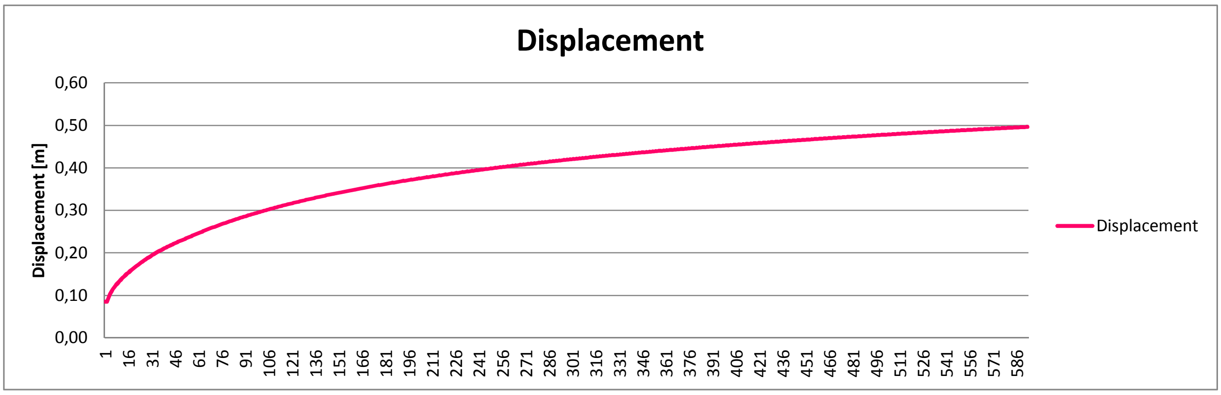


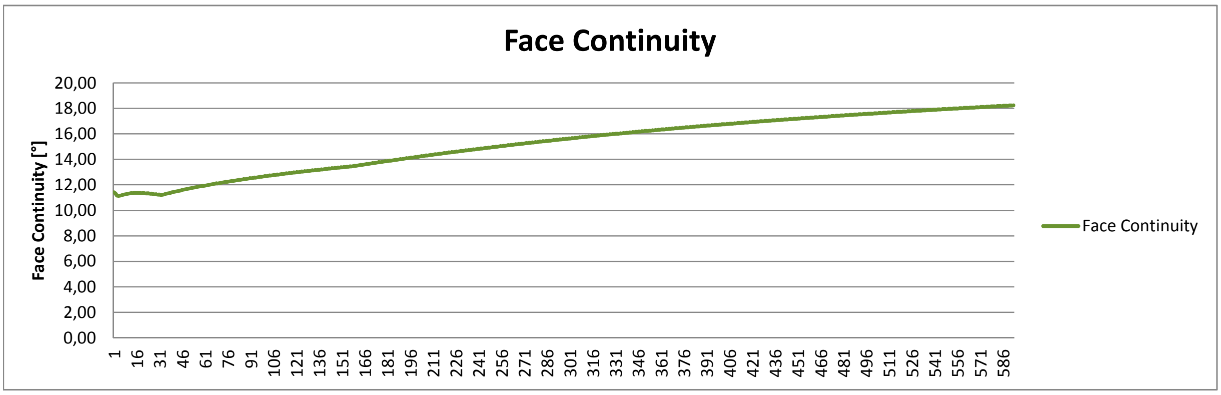


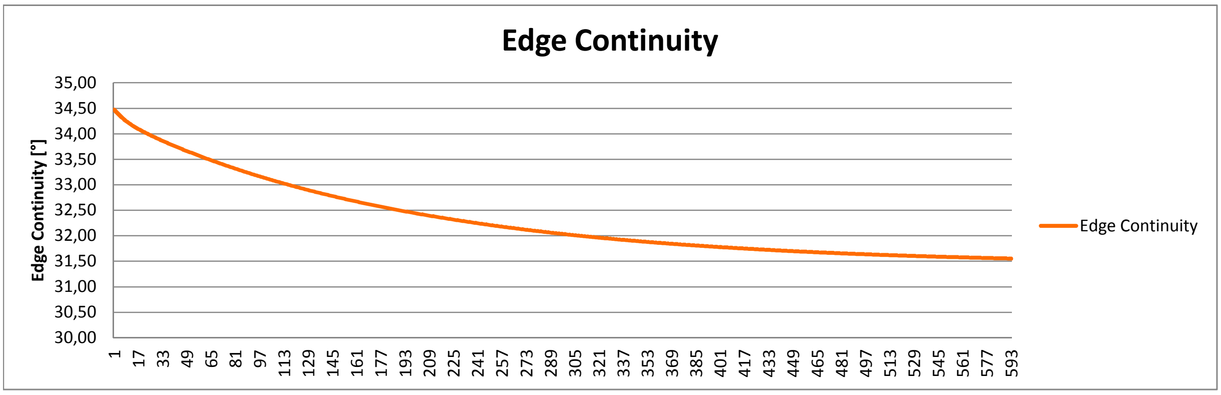


BM Mesh 03 fixed (LPC)

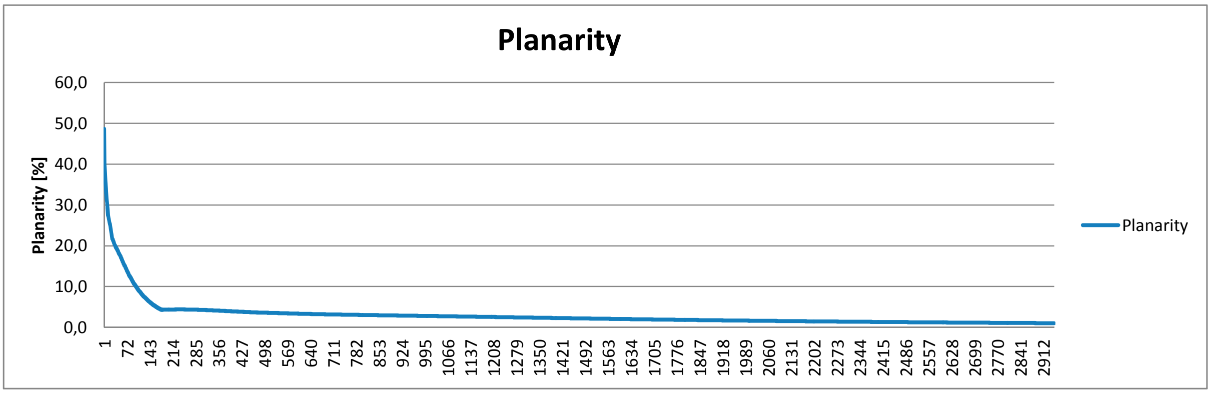


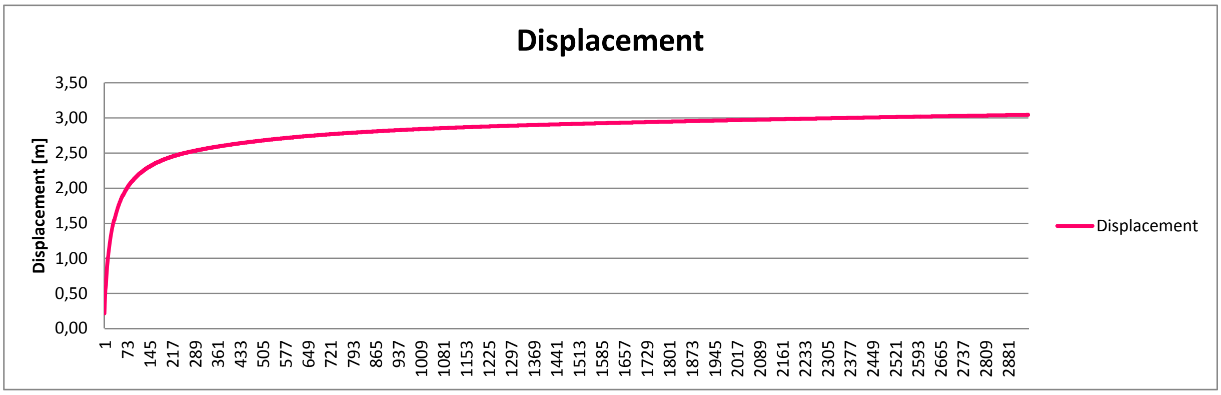


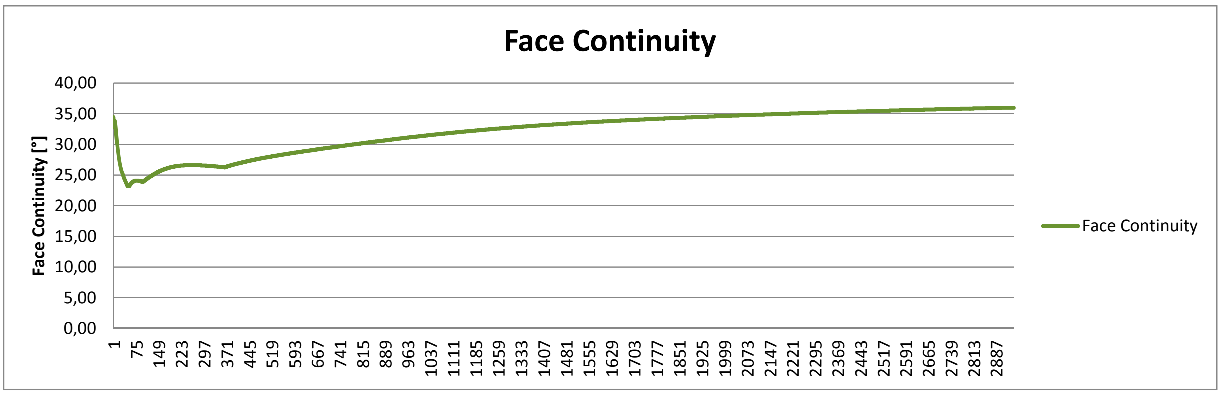


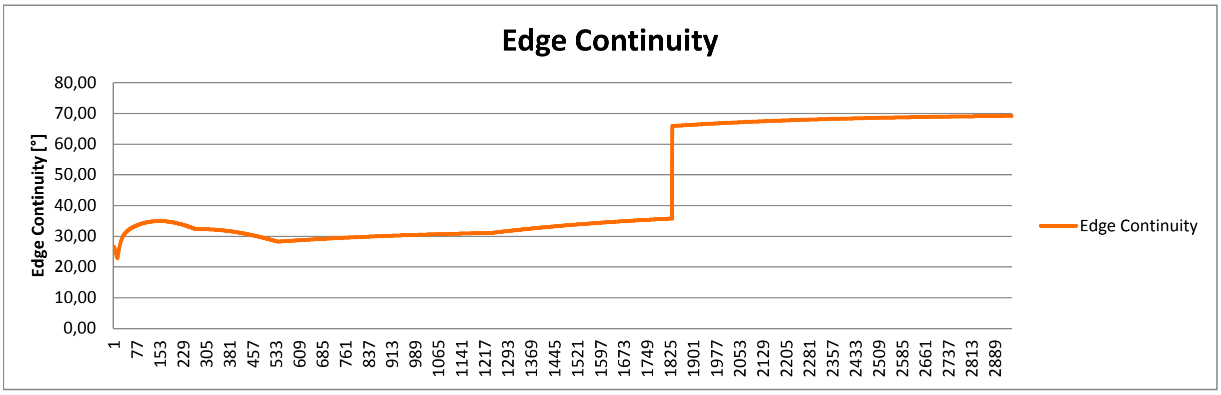


West Mesh 01 Rhombic

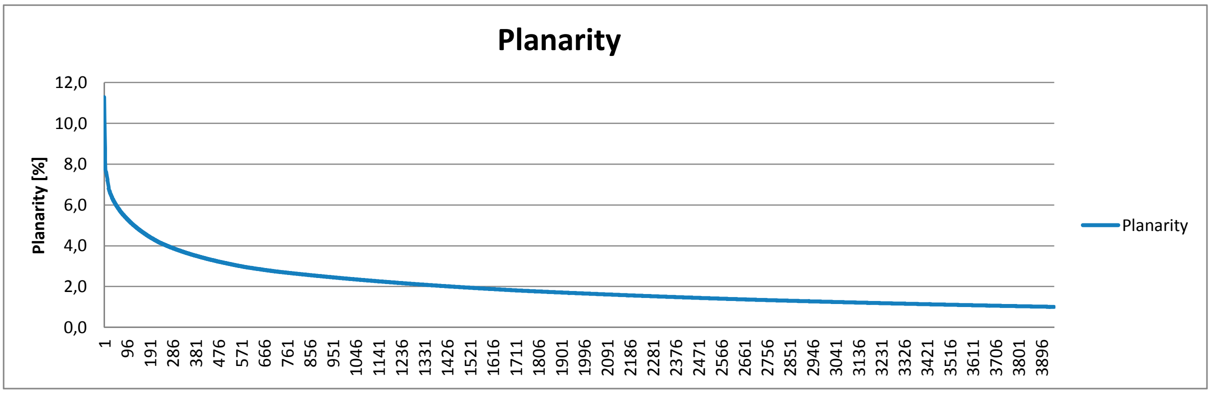


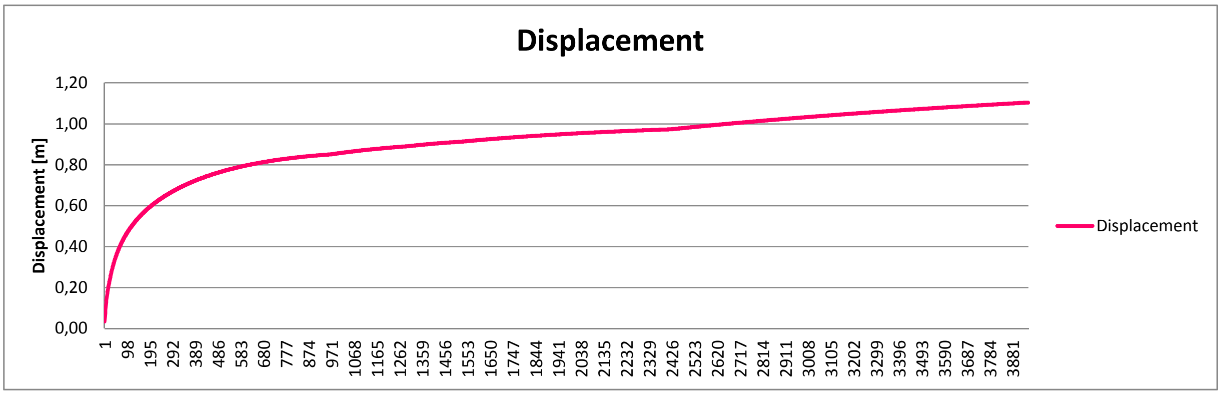


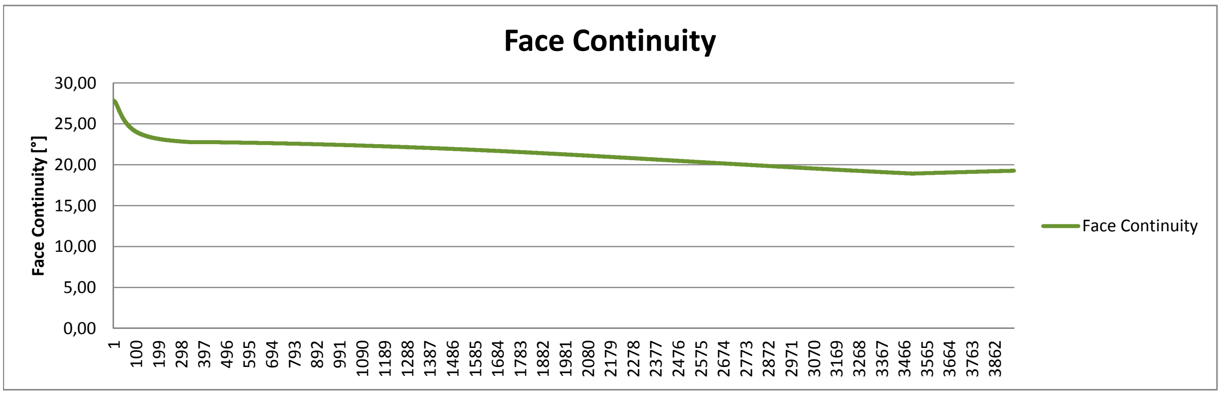


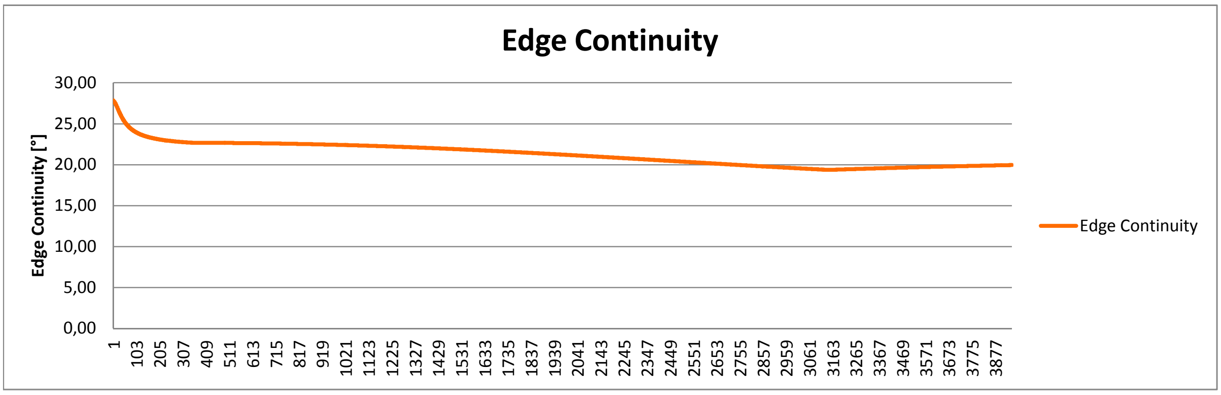


West Mesh 02 Wave

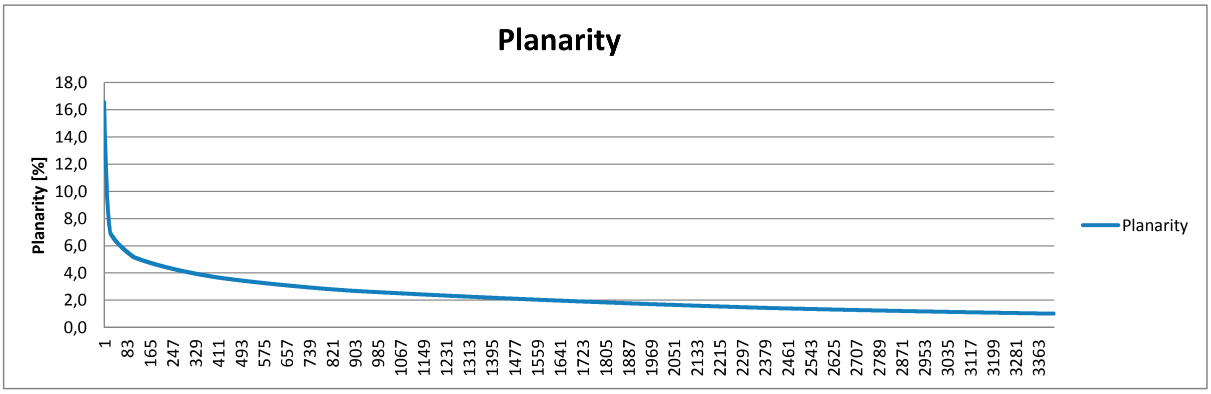


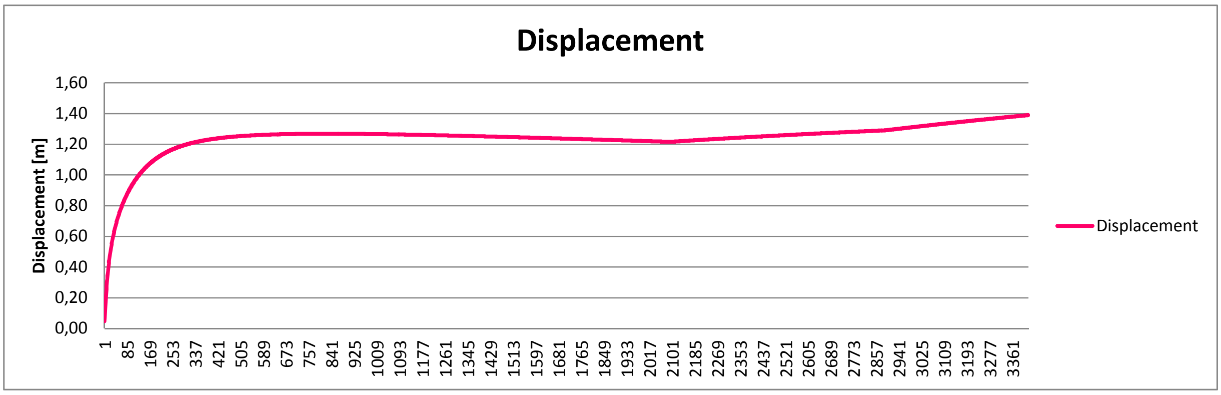


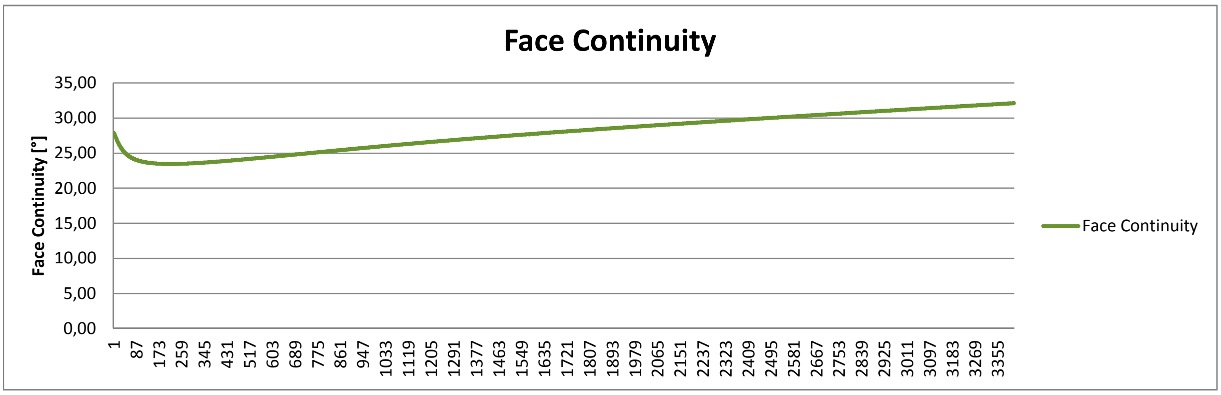


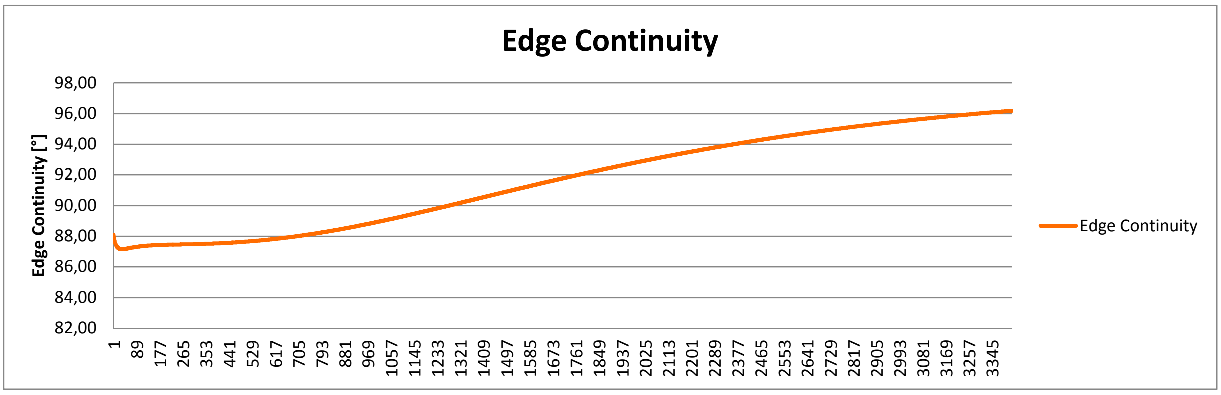


West Mesh 03 LPC

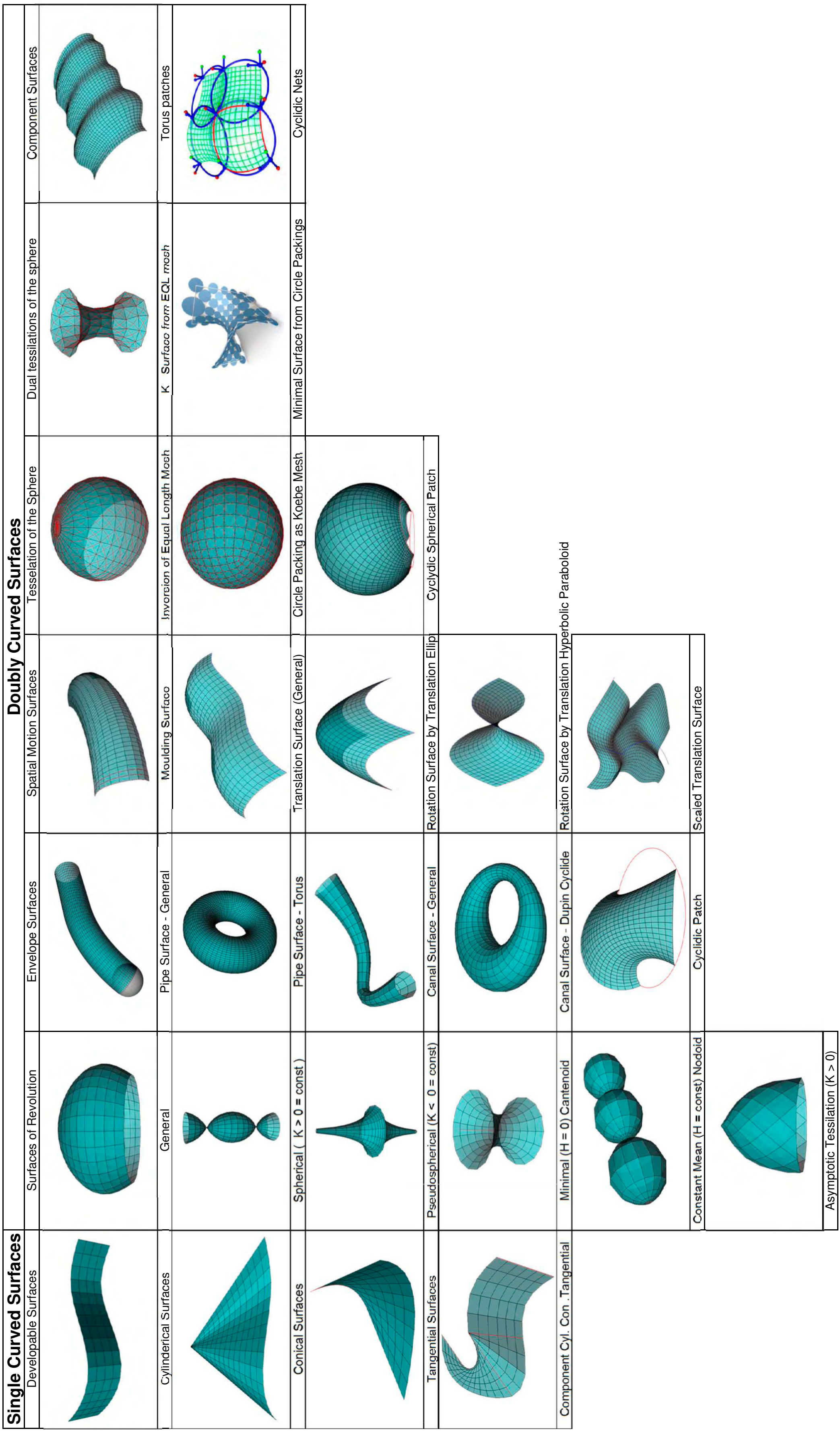








7.2 PQ Surfaces

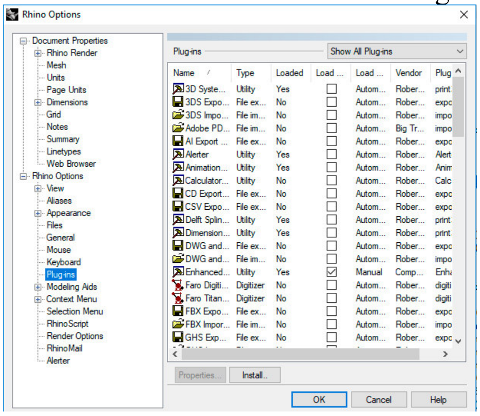

Single Curved Surfaces
Doubly Curved Surfaces
Developable Surfaces
Surfaces of Revolution
Envelope Surfaces
Spatial Motion Surfaces
Tesselation of the Sphere
Dual tessilations of the sphere
Component Surfaces
Cylinderical Surfaces
General
Pipe Surface - General
Moulding Surface
Inversion of Equal Length Mesh
K Surface from EQL mesh
Torus patches
Conical Surfaces
Spherical (K > 0 = const)
Pipe Surface - Torus
Translation Surface (General)
Circle Packing as Koebe Mesh
Minimal Surface from Circle Packings
Cyclidic Nets
Tangential Surfaces
Pseudospherical (K < 0 = const)
Canal Surface - General
Rotation Surface by Translation Ellip
Cyclydic Spherical Patch
Component Cyl. Con. Tangential
Minimal (H = 0) Cantenoid
Canal Surface - Dupin Cyclide
Rotation Surface by Translation Hyperbolic Paraboloid
Constant Mean (H = const) Nodoid
Cyclidic Patch
Scaled Translation Surface
Asymptotic Tessilation (K > 0)

7.3 The ‚Enhanced Meshing‘ Plugin

In order to perform the methods described in chapter 4 PQ mesh approximation a ‚*PlugIn*` application for the 3D-CAD environment Rhinoceros 4.0/5. ® has been developed. The functionality of the ‘Enhanced Meshing’ plugin is outlined in the following chapter.

The programming of the ‚plugin’ has been performed using the VB.Net language. As a programming environment the Visual Studio Version 2008 Standard is used. McNeels Rhino provides a Software Developer Kit (SDK) for Rhino 4.0 / 5.0 which allows to use all command functions available in the standard software for the development of bespoke applications - called Plugins.

In order to run the Enhanced Meshing Plugin it needs to be installed using the Rhino Plugin Manager:


Rhino Options

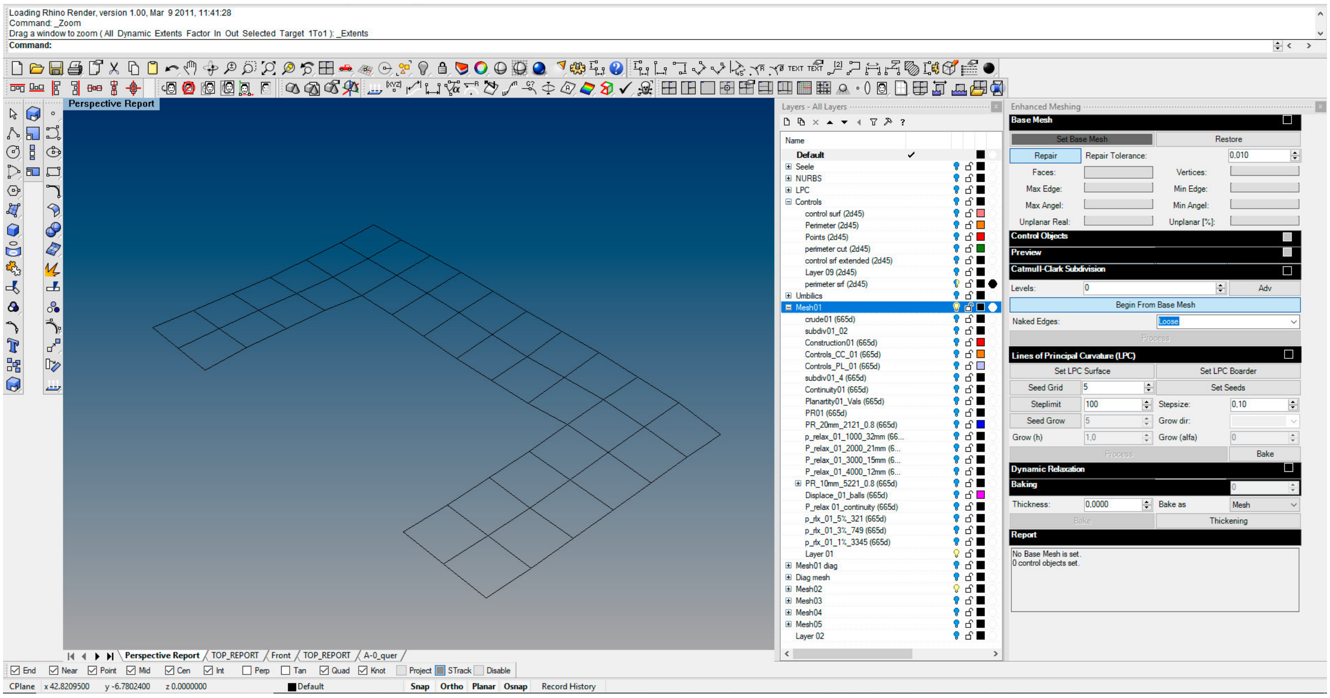
Plug-ins — Show All Plug-ins

Name	Type	Loaded	Load ...	Load ...	Vendor	Plug
3D Syste...	Utility	Yes	☐	Autom...	Rober...	print
3DS Expo...	File ex...	No	☐	Autom...	Rober...	expo
3DS Impo...	File im...	No	☐	Autom...	Rober...	impo
Adobe PD...	File im...	No	☐	Autom...	Big Tr...	impo
AI Export ...	File ex...	No	☐	Autom...	Rober...	expo
Alerter	Utility	Yes	☐	Autom...	Rober...	Alert
Animation...	Utility	Yes	☐	Autom...	Rober...	Anim
Calculator...	Utility	No	☐	Autom...	Rober...	Calc
CD Export...	File ex...	No	☐	Autom...	Rober...	expo
CSV Expo...	File ex...	No	☐	Autom...	Rober...	expo
Delft Splin...	Utility	Yes	☐	Autom...	Rober...	print
Dimension...	Utility	Yes	☐	Autom...	Rober...	print
DWG and...	File ex...	No	☐	Autom...	Rober...	expo
DWG and...	File im...	No	☐	Autom...	Rober...	impo
Enhanced...	Utility	Yes	☑	Manual	Comp...	Enha
Faro Digiti...	Digitizer	No	☐	Autom...	Rober...	digiti
Faro Titan...	Digitizer	No	☐	Autom...	Rober...	digiti
FBX Expo...	File ex...	No	☐	Autom...	Rober...	expo
FBX Impor...	File im...	No	☐	Autom...	Rober...	impo
GHS Exp...	File ex...	No	☐	Autom...	Rober...	expo

If the Load box in the plugin manger is ticked the plugin will load automatically when Rhino is opened. Alternatively the plugin will appear when ‘Enhanced_Meshing’ is typed into the command line.

```
Command: Enhanced_Meshing
```

The Enhanced Meshing Toolbox will appear on right side of the screen. The box can be undocked and replaced similar to the standard Rhino tool boxes.


```
Loading Rhino Render, version 1.00, Mar  9 2011, 11:41:28
Command: _Zoom
Drag a window to zoom ( All  Dynamic  Extents  Factor  In  Out  Selected  Target  1To1 ): _Extents
Command:
```

Enhanced Meshing

Base Mesh
- Repair Tolerance: 0,010

Catmull-Clark Subdivision
- Levels: 0
- Naked Edges: Loose

Lines of Principal Curvature (LPC)
- Seed Grid: 5
- Steplimit: 100
- Stepsize: 0,10
- Seed Grow: 5
- Grow (h): 1,0
- Grow (alfa): 0

Baking
- Thickness: 0,0000
- Bake as: Mesh

Report

No Base Mesh is set.
0 control objects set.

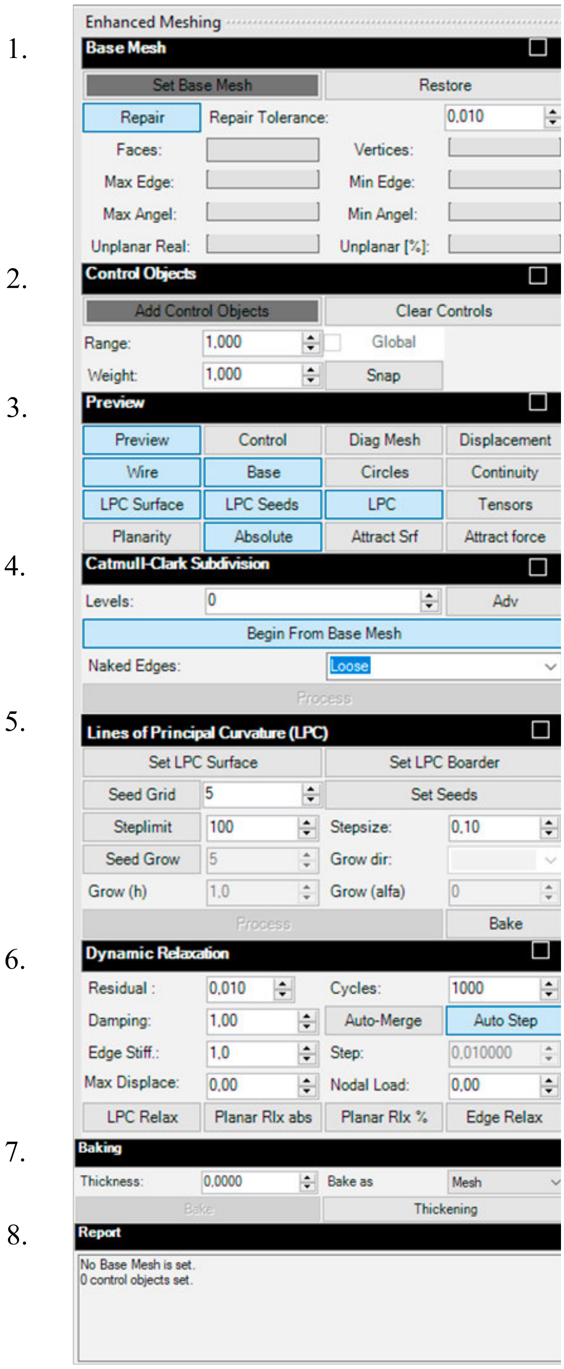
CPlane x 42.8209500 y -6.7802400 z 0.0000000 Default

The toolbox contains the following selection boxes:

1. Base mesh
2. Control Objects
3. Preview
4. Catmull Clark Subdivision
5. Lines of Principal Curcature (LPC)
6. Dynamic Relaxation
7. Baking
8. Report

The Selection boxes can be be expanded and collapsed using the top right black button in the title block of each selection box.

The functionality of the individual tool boxes will be described in the following .



Repair Tolerance: 0,010

Range: 1,000

Weight: 1,000

Levels: 0

Naked Edges: Loose

Seed Grid 5

Steplimit 100

Stepsize: 0,10

Seed Grow 5

Grow dir:

Grow (h) 1,0

Grow (alfa) 0

Residual : 0,010

Cycles: 1000

Damping: 1,00

Edge Stiff.: 1,0

Step: 0,010000

Max Displace: 0,00

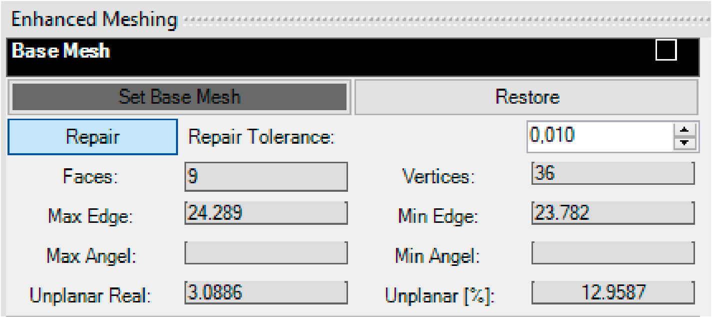
Nodal Load: 0,00

Thickness: 0,0000

Bake as Mesh

No Base Mesh is set.
0 control objects set.

1. Base mesh



Repair Tolerance: 0,010

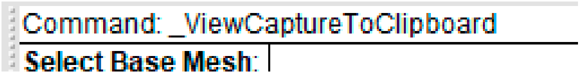
Faces: 9 Vertices: 36

Max Edge: 24.289 Min Edge: 23.782

Max Angel: Min Angel:

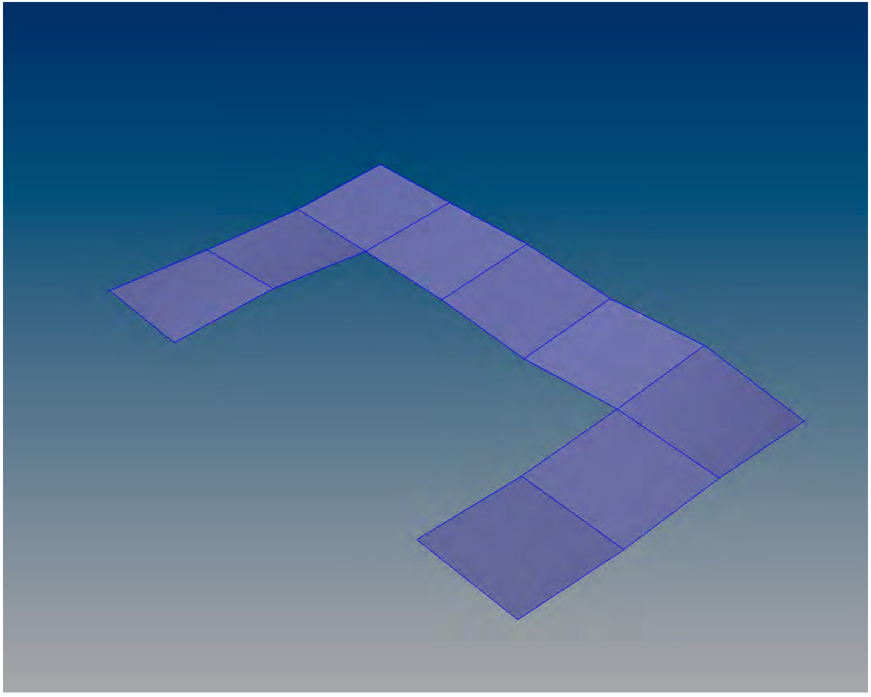
Unplanar Real: 3.0886 Unplanar [%]: 12.9587

The ‘Set Base Mesh’ button will prompt in the Command Line:
‚Select Base Mesh‘



```
Command: _ViewCaptureToClipboard
Select Base Mesh:
```

When the Base Mesh is selected a blue colored preview mesh appears



Mesh preview

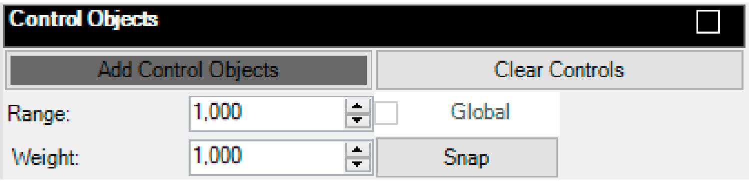
If ‚Repair‘ button is enabled all mesh vertices which are closer together then the defined repair tolerance (absolute value in the rhino file units) will be collapsed into one vertex.

‚Restore‘ button will restore the original state of the mesh when it was selected
All operation performed with the ‚Catmull-Clark Subdivison‘ and/or ‚Dynamic Relaxation‘ functions will be lost.

The Info Panel will give information about the selected base mesh:

No of Faces, Vertices
Max/Min Edge Lenght
Max/Min Angel – not used
Absolute Planrity in rhino file units (usually: mm or m)
Relative Planarity as described in chapter 5.2.1

2. Control Objects



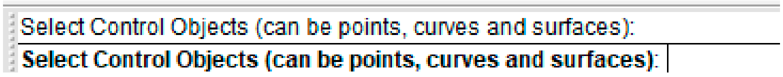
Control Objects □

Range: 1,000 □ Global

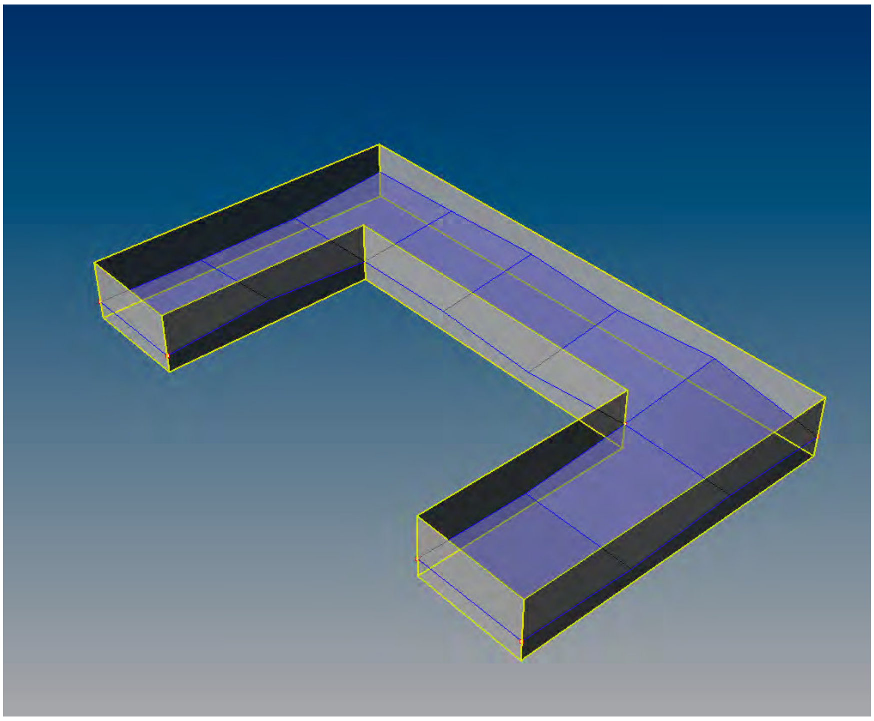
Weight: 1,000

In the ‚Control Objects' box the drawing objects can be selected which will constrain the operations executed with function in the 'Catmull- Clark Subdivion' and ‚Dynamic Relaxation' boxes.

Add Control Objects Button:
Point, Curves and surfaces to be selected – Command line

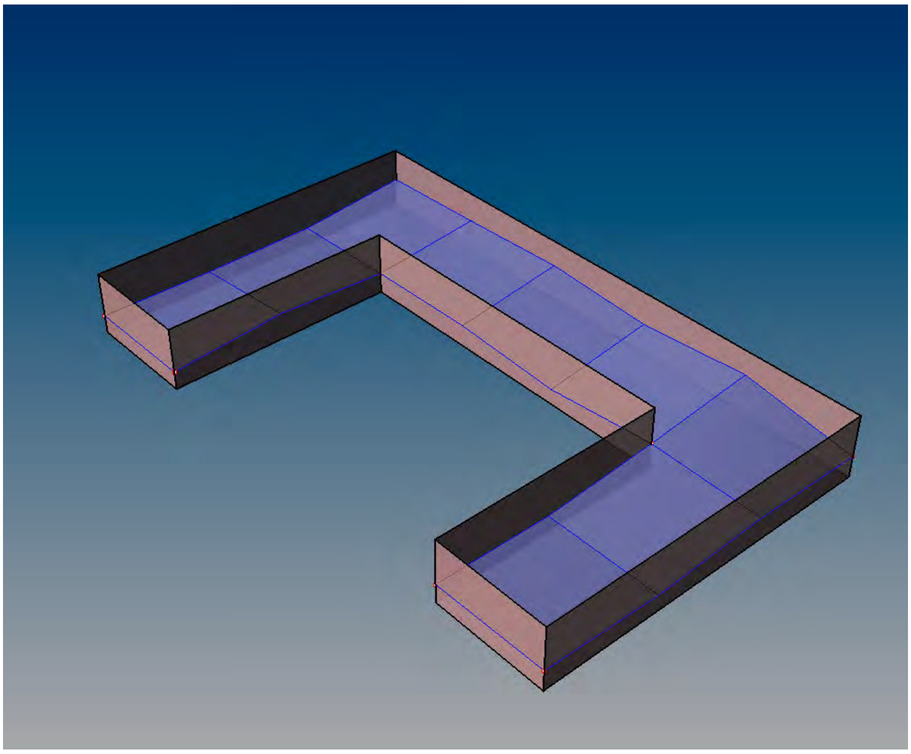


```
Select Control Objects (can be points, curves and surfaces):
Select Control Objects (can be points, curves and surfaces):
```



Selected control objects: boundary surface

Selected control objects will be highlighted in light red after their selection (only if preview button is enabled for control objects)



Control objects - when selected

If ‚Snap‘ button is enabled the vertices snaps directly to all control object within the given range
Hence range to be chosen in relation to mesh faces for curves and points

‚Range‘ within a vertex snaps or attracts to the closest control object

If ‚Snap‘ is disabled the value ‚Weight‘ defines the distance between the vertex and the control object within the snap range. A higher value for ‘weight’ the vertex will be pulled closer to the control object.

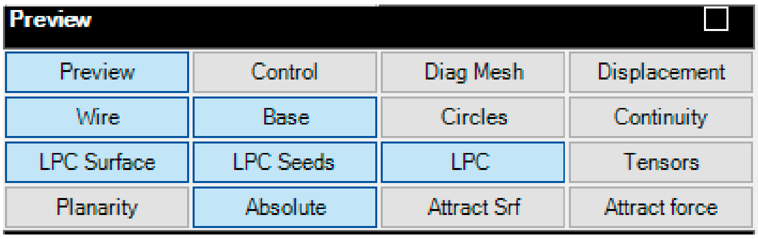
If ‚Global‘ is ticked all properties (Range, weight) of previously added control objects will be overridden and set to the current values
Control objects can be selected successively and given an individual range or weight for snapping or attracting
‚Clear Controls‘ will delete all control objects

This function will work for the components ‚Catmull Clark Subdivision‘ and ‚Dynamic Relaxation‘

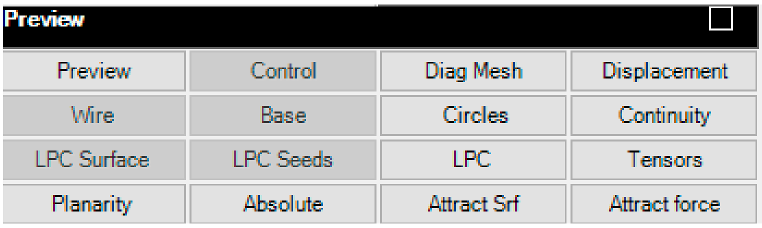
3. Preview

The ‚Preview’ box controls the items which are shown in the Graphics conduit. The displayed geometry only exist as a preview and are not rhino drawing objects. Only after executing the baking function the chosen elements become real drawing object.

Buttons within ‚Preview‘ box will enabled/disables preview of selected and generated objects



When button appears light blue the specific function is enabled.

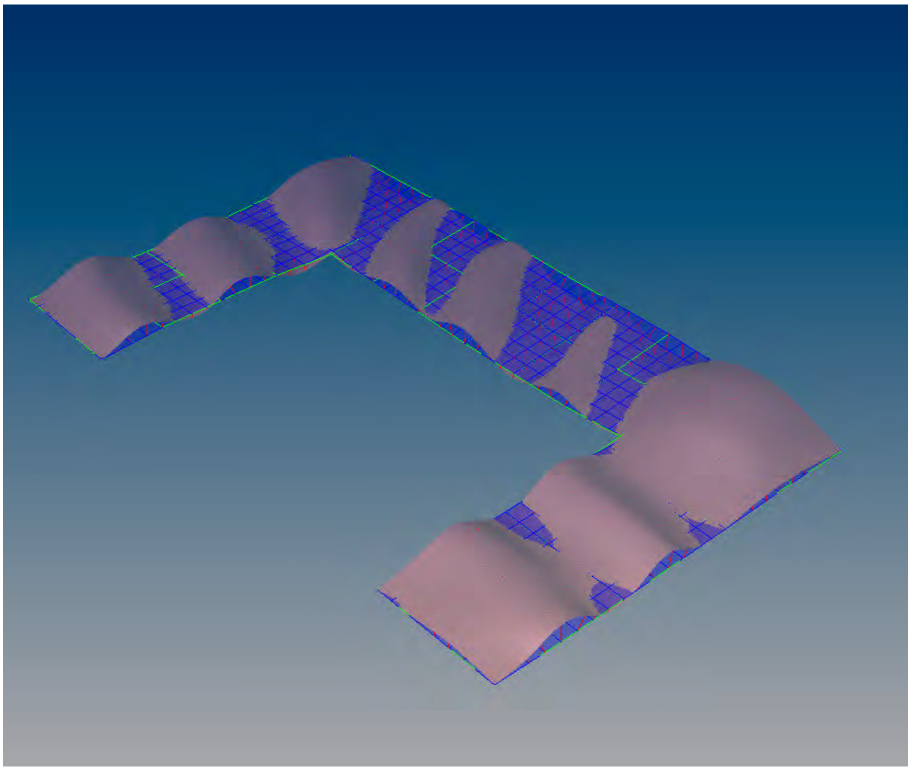


If button is grey the specific function is disabled.

To visualize the change in of setting the ‚Preview’ button may need to be switched on/off or the display need to be zoomed in/out.

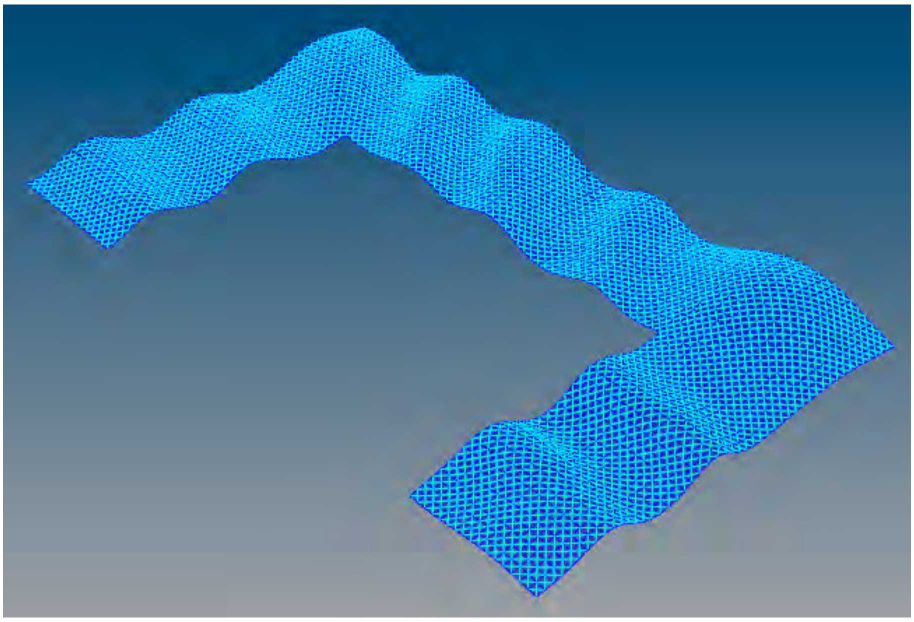
‚Preview‘ button enables/disables preview of objects in general

‚Control‘ button enables/disables preview of control objects and will show/hide control objects coloured light red. Red connection lines will be shown between a mesh vertex and all control objects within the given Range

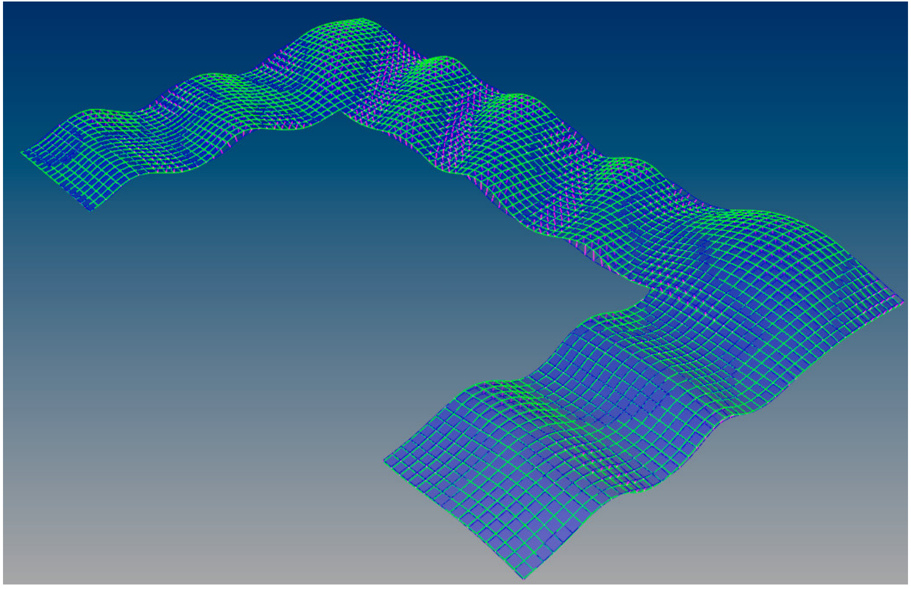


Vertices with control objects within range

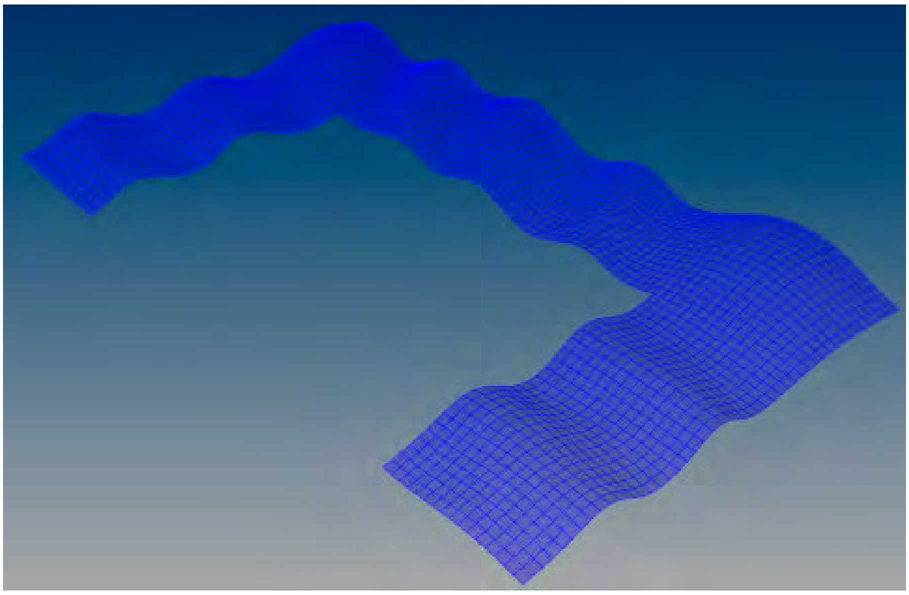
‚Diag Mesh‘ button showing the diagonal lines connecting the opposite vertices of each face of the selected mesh



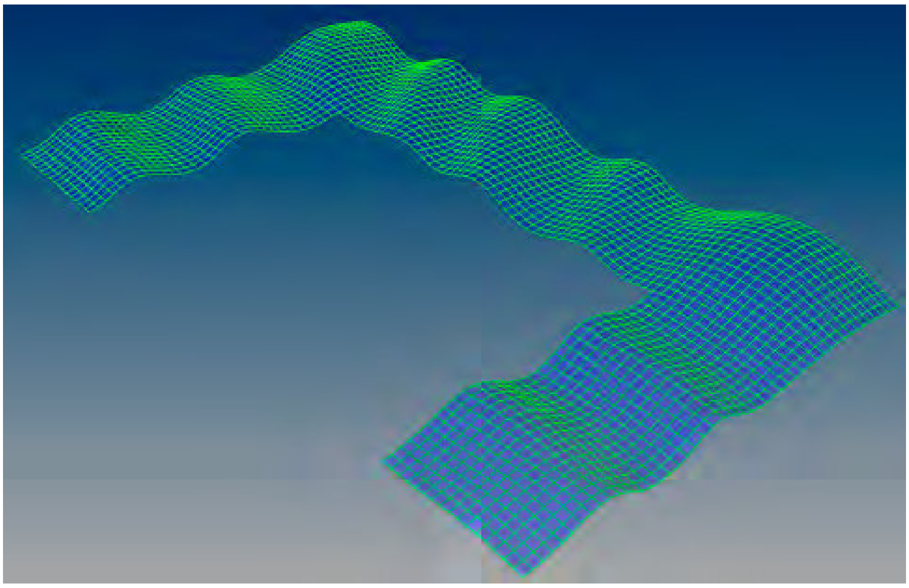
‚Displacement' button shows pink connection lines between each vertex of the base mesh and the corresponding vertex of the optimized mesh after applying ‚Dynamic Relaxation' operations. Functionality only active for mesh deformations not for mesh populations such as Catmull-Clark operations.



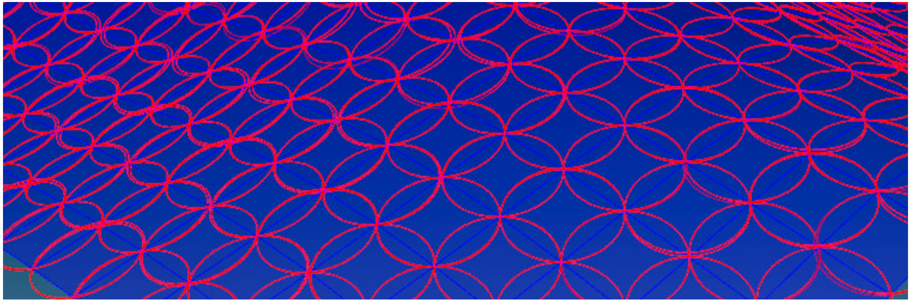
´Wire' button enables the display of all face edges of the current mesh as blue lines



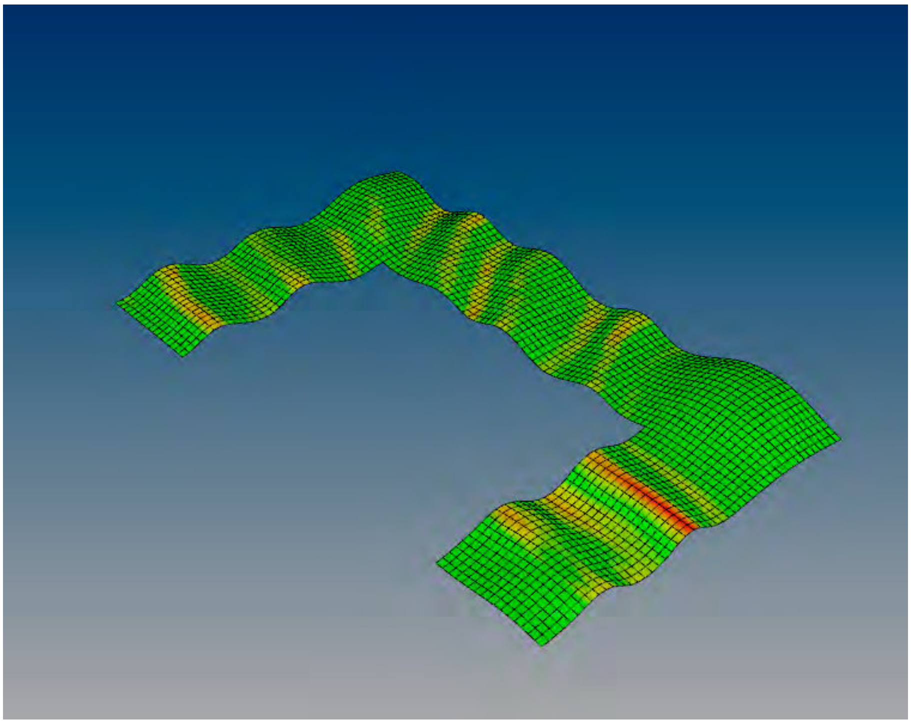
‚Base' button enables the display of the base mesh face edges as green lines



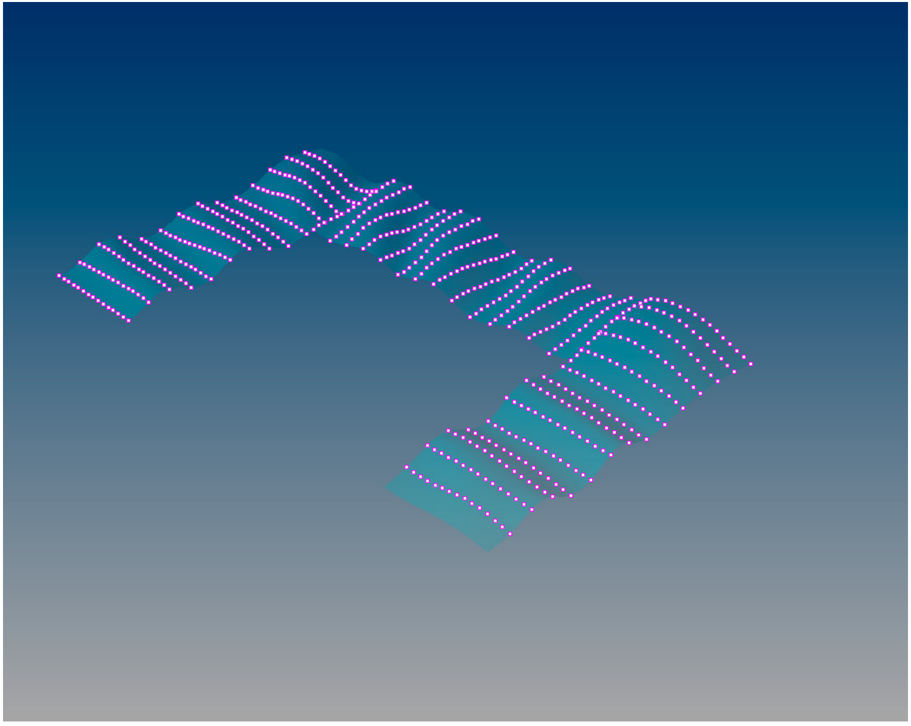
'Circles' button shows four red coloured circles each defined by three vertices of a mesh face in consecutive order. If all circles are identical the mesh faces is called to be circular. Function allow a visual judgment if the chosen mesh is suitable for ‚Circular' Dynamic Relaxation.



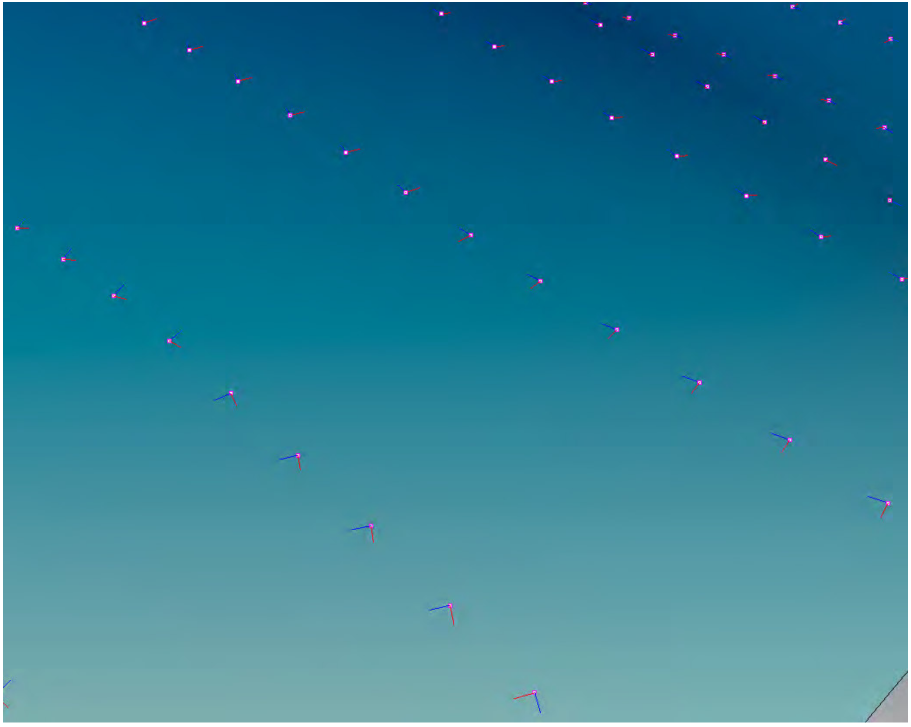
‚Continuity' button shows the colour gradient mesh informed by the continuity values computed at each vertex of the current mesh.



‚LPC Surface' button shows the selected surface for lines of principle curvature generation in light cyan colour
‚LPC Seeds' button shows the seed points selected for lines of principle curvature generation in pink colour

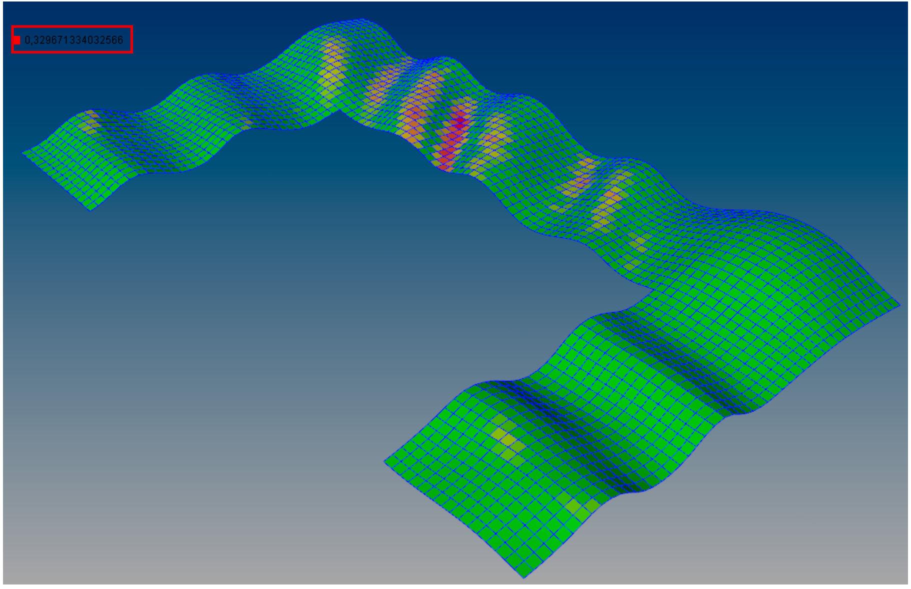


‚Tensors' button is showing the principal curvature vectors of the selected LPC surface at each seed point. For minimum curvature direction in blue colour and for maximum curvature direction in red colour. The vector display only appears after the LPC generation procedure was performed previously.

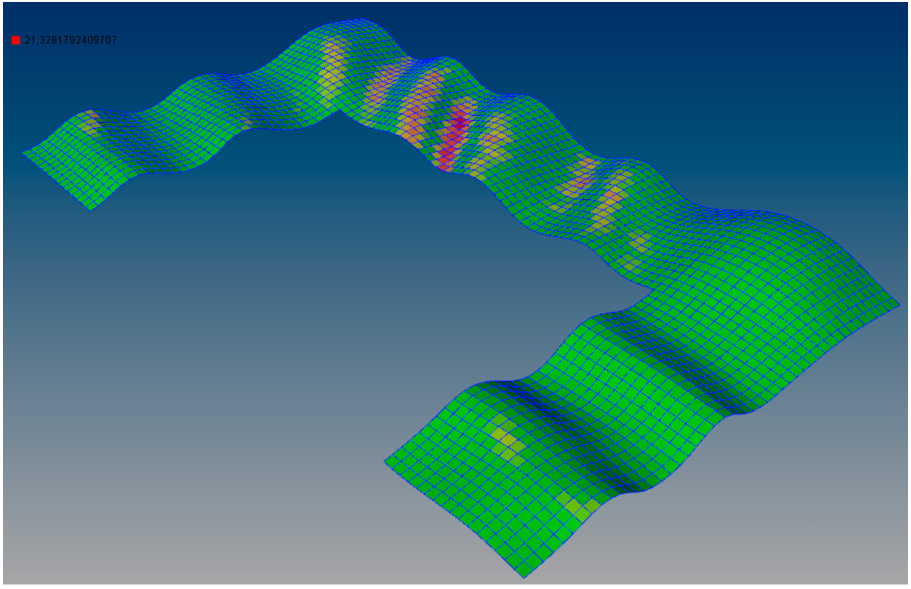


‚Planarity' button shows a colour gradient mesh informed by the planarity values of each face

If the ‚Absolute' button is enabled the gradient colour mesh is informed by the absolute planarity values. Maximum value is displayed numerically in the top left corner of the screen.

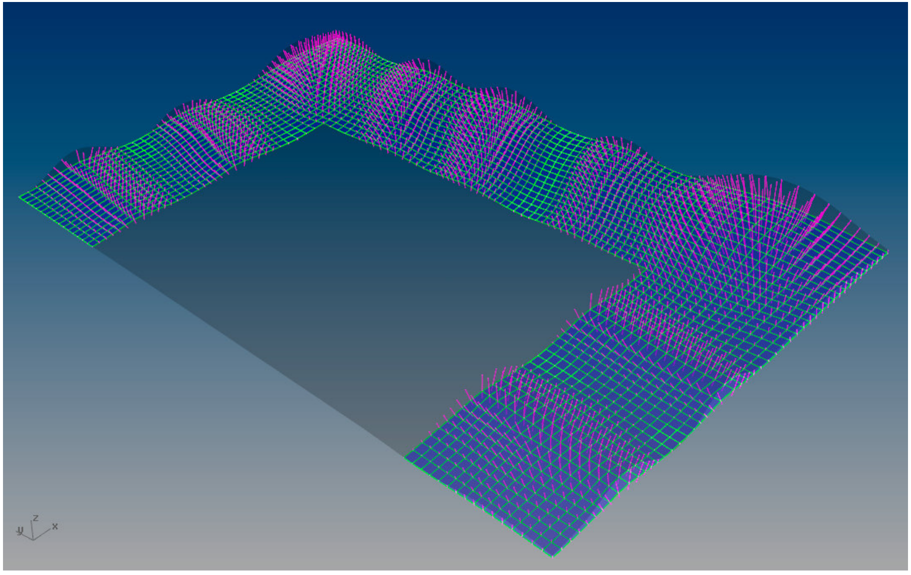


If the ‚Absolute' button is disabled the gradient colour mesh is informed by the relative planarity values. Maximum value is displayed numerically in the top left corner of the screen.



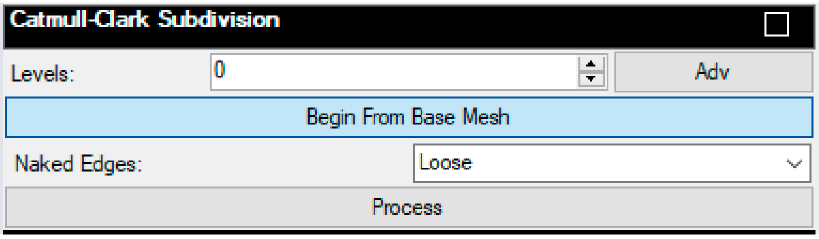
‚Attract Srf' button shows the surface selected if the ‚Planar Attract' Optimization in 'Dynamic Relaxation' is chosen.

‚Attract force' button shows the force vectors as a pink line between each vertex and the attract surface (Similar to the ‚Control' display).



4. Catmull Clark Subdivision

In the ‚Catmull-Clark Subdivision' box the input data and type of execution of the CC Subdivision process is defined.



Levels: 0

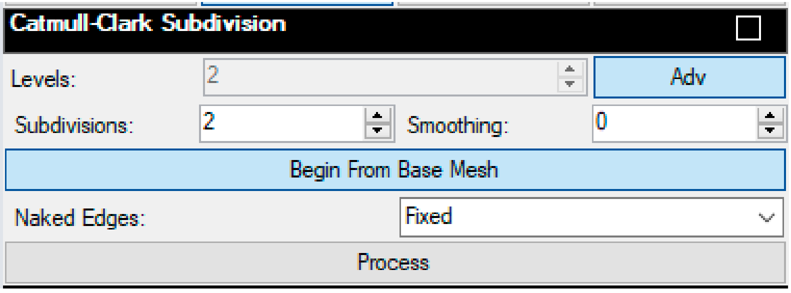
Naked Edges: Loose

In the ‚Levels:' selector box the number iterations for subdivision and smoothing operation can be chosen

If 'Adv' is disabled for each iteration both subdivision and smoothing will be performed simultaneously

If 'Adv' is enabled the no of iteration for subdivision and smoothing operations can be chosen individually

Only subdivisions are performed:

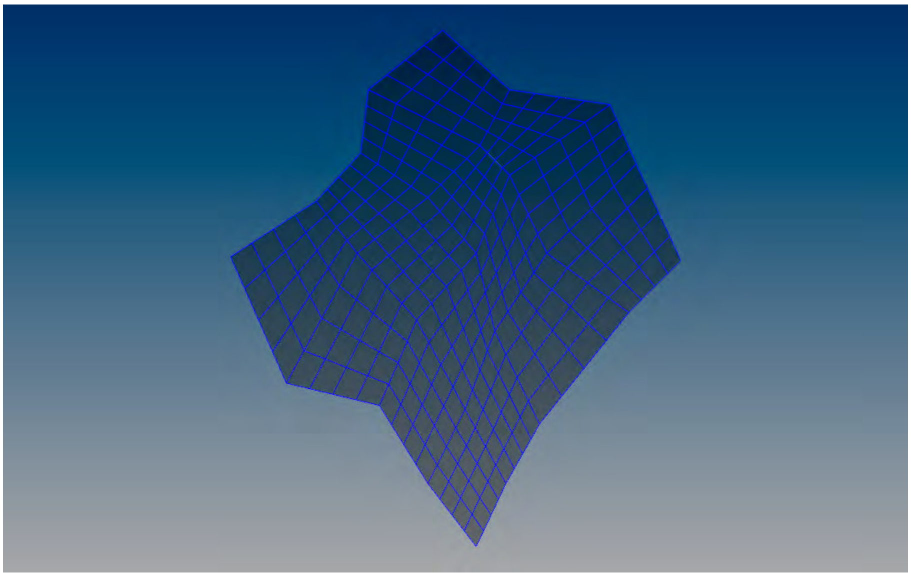


Levels: 2

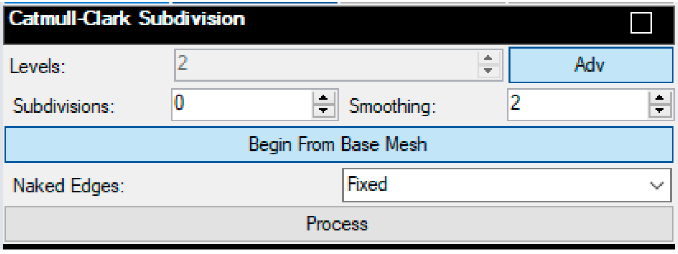
Subdivisions: 2

Smoothing: 0

Naked Edges: Fixed



Only smoothing is performed:

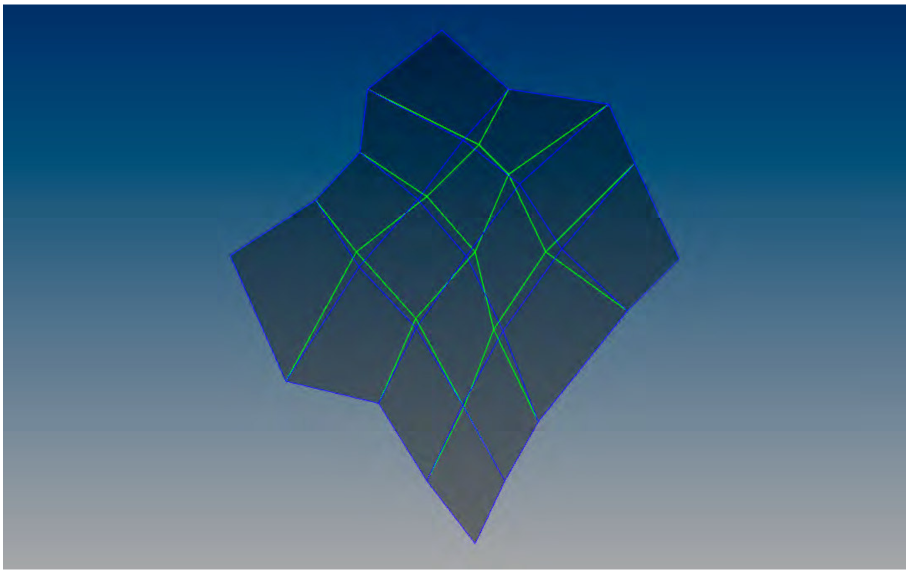


Levels: 2

Subdivisions: 0

Smoothing: 2

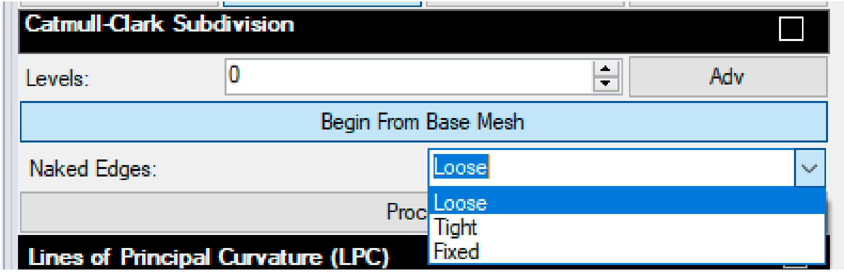
Naked Edges: Fixed



If ‘Begin From Base Mesh’ is enabled all previously performed operations will be restored and currently chosen operations start from the base mesh

If ‘Begin From Base Mesh’ is disabled all operations will be performed using the current mesh to start from. This would allow to change control conditions for each iteration.

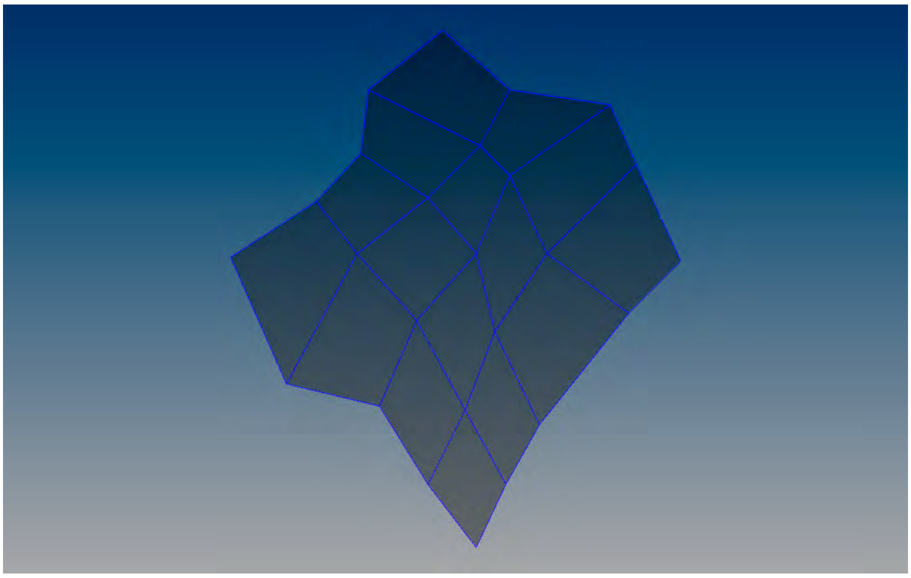
In the ‘Naked Edges’ selector box the following restrain condition for naked edges and their belonging vertices can be chosen: Loose, Tight or Fixed:

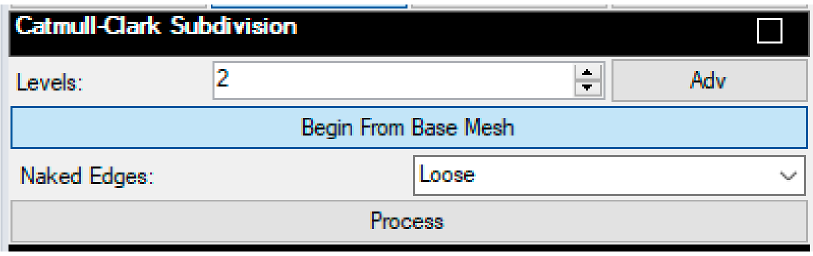


Levels: 0

Naked Edges: Loose

For the ‘Loose’ condition all edge vertices will be unrestrained. Hence the mesh will shrink along the edge vertices due to the subdivision and averaging procedure as described in chapter

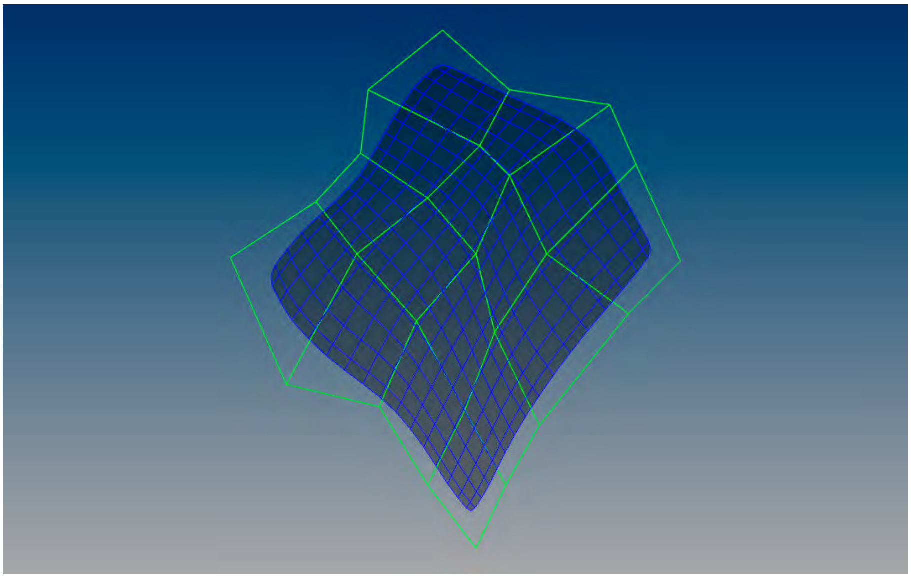




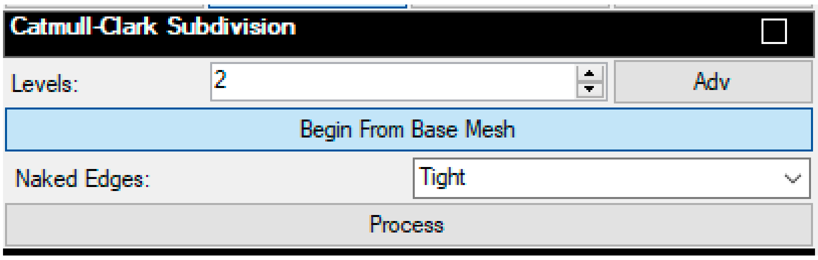
Catmull-Clark Subdivision

Levels: 2

Naked Edges: Loose



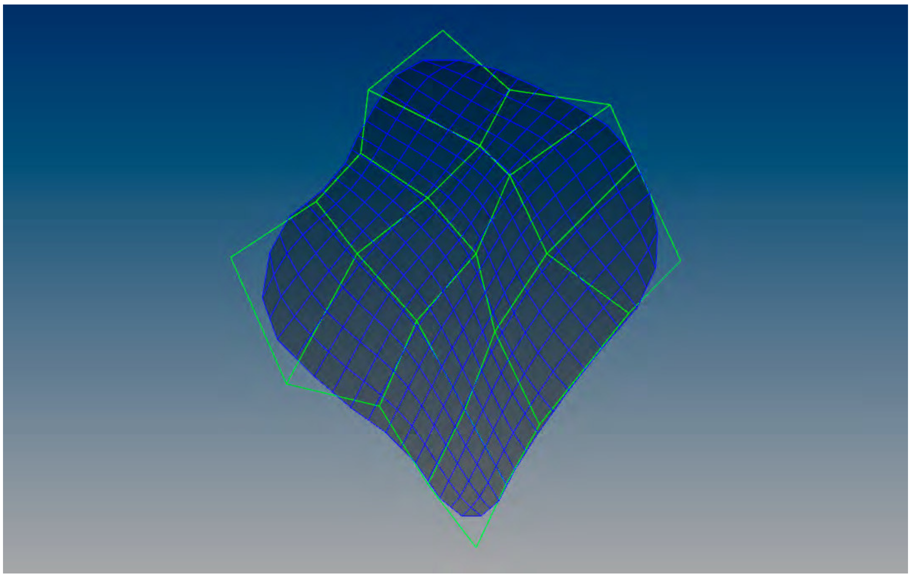
For the ‘Tight’ condition all edge vertices will be dependently restraint and move into the average position between the ‘Fixed’ and the ‘Loose’ condition.



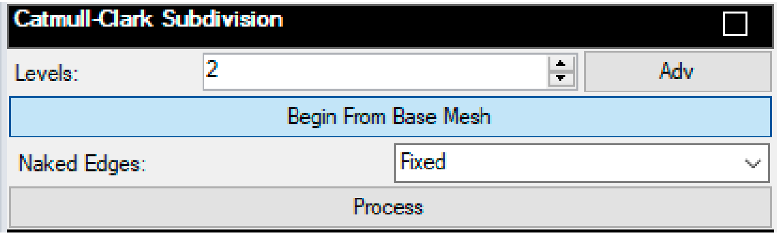
Catmull-Clark Subdivision

Levels: 2

Naked Edges: Tight



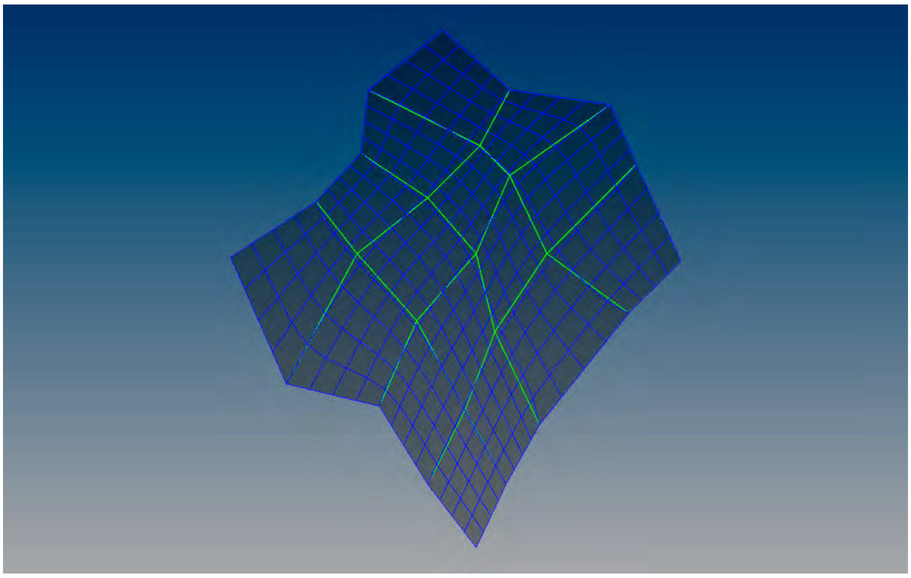
For the ‘Fixed’ condition all corner and edge vertices will stay at their position according to the base mesh. All newly generated vertices are located on the polygonal perimeter of the base mesh.



Catmull-Clark Subdivision

Levels: 2

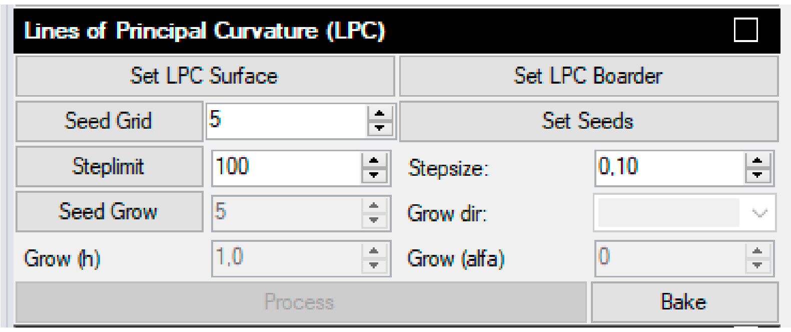
Naked Edges: Fixed



The ‚Process’ button will perform the operations with the chosen settings.

5. Lines of Principal Curvature (LPC)

In the ‚Lines of Principal Curvature (LPC)' box the input data and type of execution of the LPC generation process is defined.



Lines of Principal Curvature (LPC) □

Set LPC Surface

Set LPC Boarder

Seed Grid: 5

Set Seeds

Steplimit: 100

Stepsize: 0,10

Seed Grow: 5

Grow dir:

Grow (h): 1,0

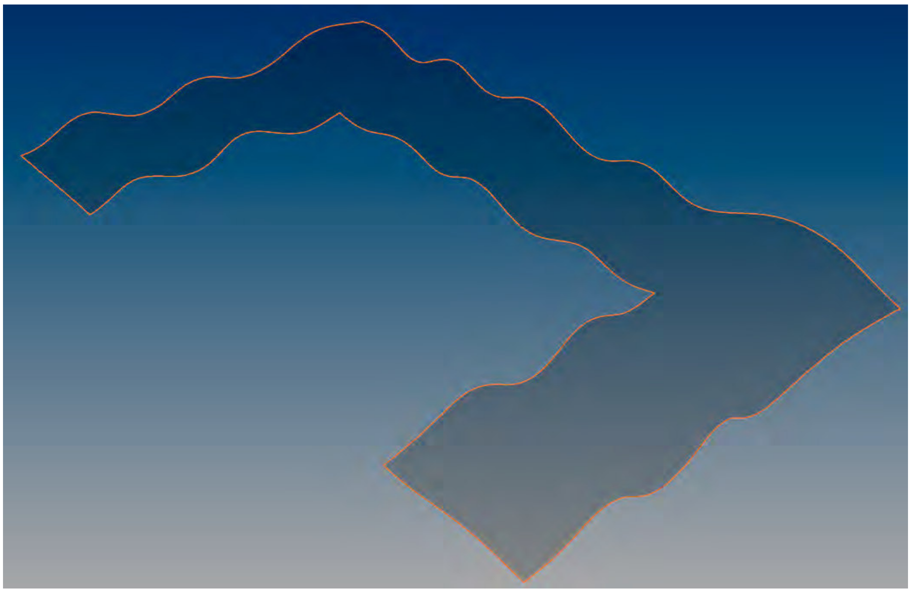
Grow (alfa): 0

Process

Bake

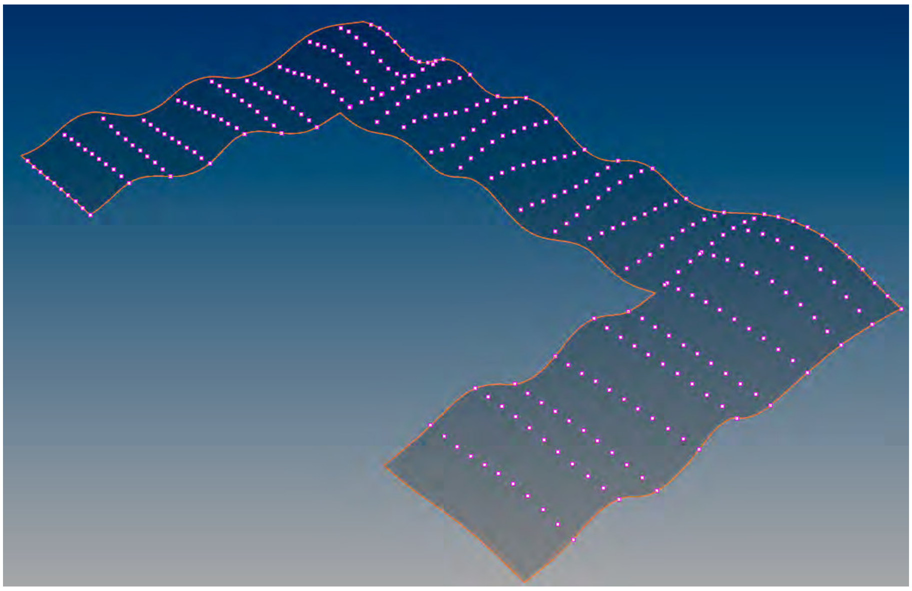
‚Set LPC Surface' button will prompt in the command line to select the surface for LPC generation.

‚Set LPC Boarder' button will prompt in the command line to select the surface boarder of the LPC surface.



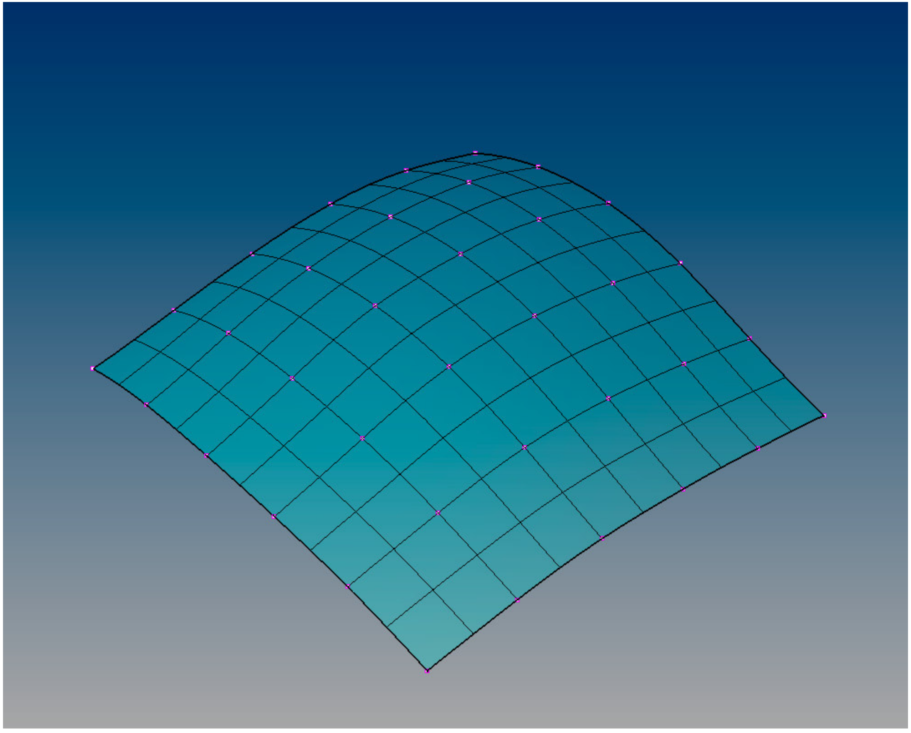
Note: When a trimmed Brep is selected the trimming boundary will not be detected automatically by the algorithm. Hence the explicit definition of the surface boarder is required. As an option a smaller area within the chosen surface can be defined by closed curve.

‚Seed Grid' will set a grid of seed points on the selected surface with the defined density. Both the U and V domain of the surface patch will be divided by the number given and a seed point located respectively.



In order to control the seed point density it is recommend to rebuilt the surface (cut off trimmed areas, change u/v density) using the standard Rhino Tools.

Nurbs patch with rebuilt uniform u/v- density:



With the ‘Set Seeds’ Button’ individual seed point can be selected to start the LPC generation from

‚Set LPC Boarder’ button will set the boundary within the LPC generation is performed. The boarder not necessarily has to coincide with the surface patch boarder but cannot cross the surface boarder.
In the section box ‚Steplimit’ the number of approximation steps can be chosen. For details of the approximation process please refer to Chapter 4.2.3

The ‚ Stepsize’ defines the length of the vector in the Rhino files units for the approximation procedure. Value should be chosen according the surface curvature: Small value for highly curved surfaces – larger value for less curved surfaces. Recommended value: 100mm to 500mm

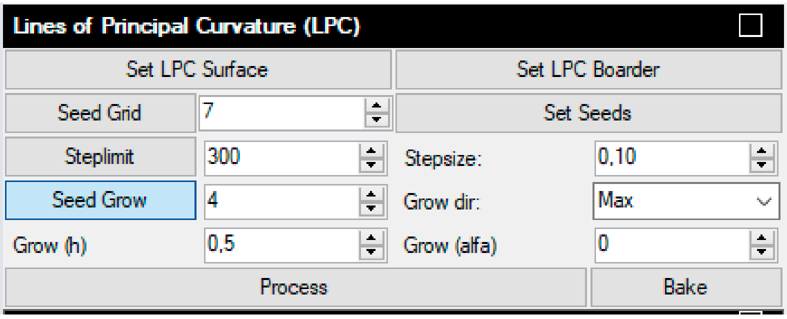
Note: Values for ‚Steplimit’ and ‚Stepsize’ need to chosen in order to allow long enough LPC’s to run over the entire surface patch. Generated lines will automatically stop at the defined LPC boarder.

With the ‘Seed Grow’ button individual LPC’s can generated using the seed point sampling definition as outline in chaper 4.2.4

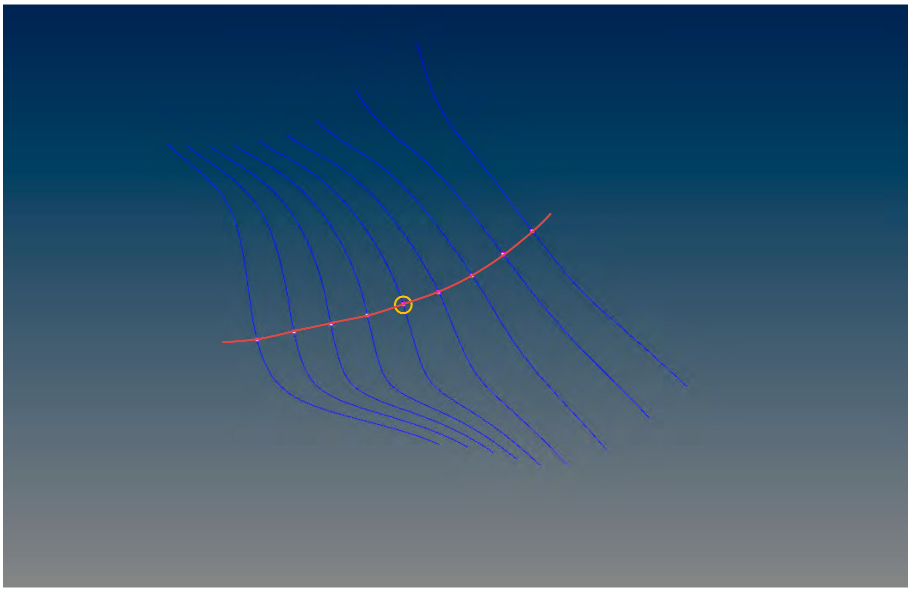
The value set for ‘Seed grow’ will define the number of LPC’s are generated for the selected Growing direction.

The ‘Grow dir’ can be set to ‚Max’ or ‚Min’.
For ‚Max’ the new generated Lines are Minimal LPC’s places in the specified number and distance along the corresponding Maximum LPC. The process will start at all selected seed points in both direction parallel to the initial Minimum LPC.
For ‚Min’ the procedure is performed respectively.

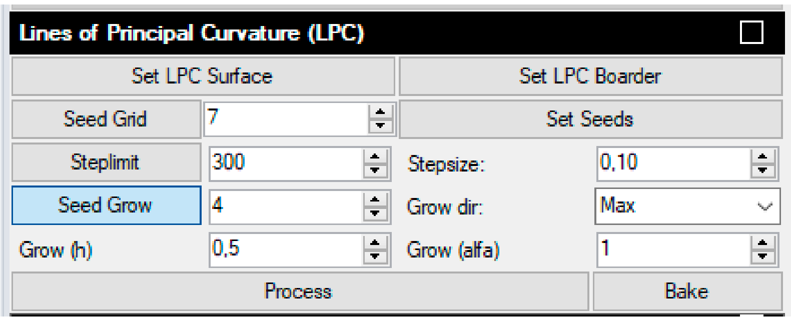
The value ‚Grow (h) specifies the reference sampling distance according to Equation 4.4 chapter 4.2.4.
The value for ‚Grow (alfa)’ influences the valued sampling distance (Equation 4.4).
For the value ‚0’ the newly generated LPC’s are spaced in reference distance along the LPC in growing direction



Seed Grid 7
Steplimit 300
Stepsize: 0,10
Seed Grow 4
Grow dir: Max
Grow (h) 0,5
Grow (alfa) 0



For values (alfa) higher then ‚0' the valued sampling distances (h1, h2) at all step points are evaluated. The new sampling point is then located at the highest value for h1 or h2 in the evaluated distance. Hence the location of the new sampling points may not lay on a particular growing direction LPC.



Seed Grid: 7

Steplimit: 300

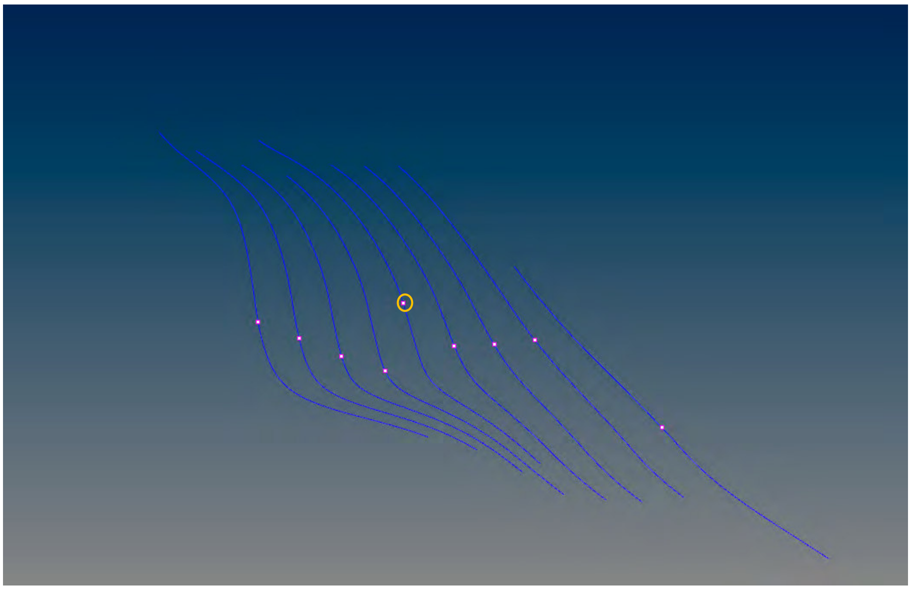
Stepsize: 0,10

Seed Grow: 4

Grow dir: Max

Grow (h): 0,5

Grow (alfa): 1



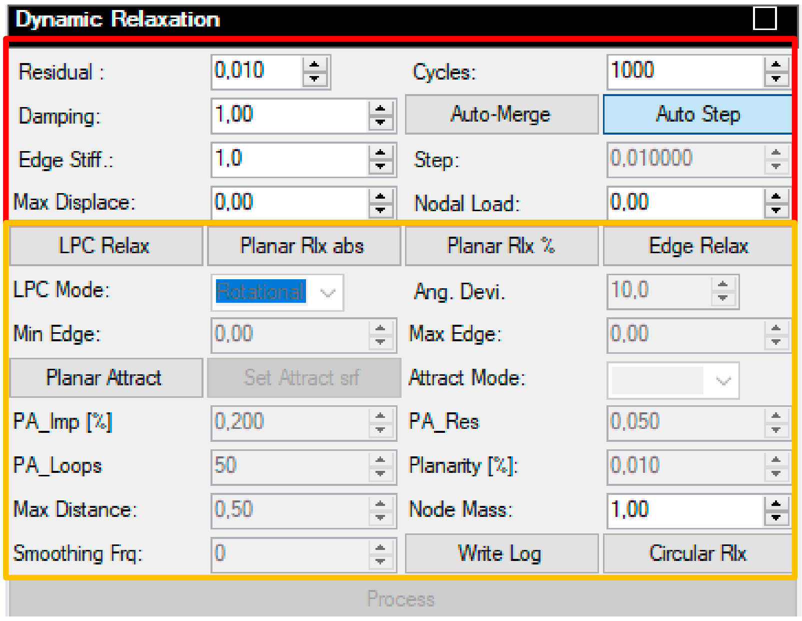
The 'Process' button will run the generation procedure for the given settings.

The 'Bake' button will add the LPC's as curve objects to the current layer of the rhino file: Blue coloured spline curves for Minimum LPC's and Red coloured splines for Maximum LPC's

6. Dynamic Relaxation

In the ‚Dynamic Relaxation' box the input data and type of execution of the Dynamic Relaxation Process is defined.

In the upper part (red) the settings for general DL processing can be defined. In the lower part (orange) for the bespoke DL processing functionality can be defined.



Residual : 0,010
Cycles: 1000
Damping: 1,00
Edge Stiff.: 1,0
Step: 0,010000
Max Displace: 0,00
Nodal Load: 0,00
LPC Mode: Rotational
Ang. Devi. 10,0
Min Edge: 0,00
Max Edge: 0,00
Attract Mode:
PA_Imp [%] 0,200
PA_Res 0,050
PA_Loops 50
Planarity [%]: 0,010
Max Distance: 0,50
Node Mass: 1,00
Smoothing Frq: 0

In the ‚Residual’ selector the numeric value will be until the DR process will be executed.

In the ‚Cycles’ selector a maximum limit of iteration can be defined. Please note that the DL procedure will stop when the given residual is reached despite the maximum no. of cycles are reached. As against the given residual may not be reached for defined no. of cycles.

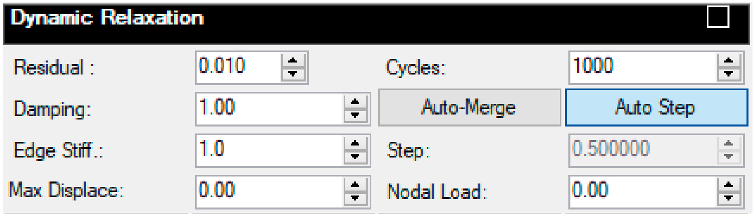
The value for ‘Damping:’ can be chosen in the corresponding selection box. Note: For the default value = 1.00 the DL process will executed as an undamped system. Only values larger the 1.00 will consider damping. The application of damping is only recommended for instable systems as the damping will slow down the convergence of the system. Usually damping may not be required.

When ‘Auto Merge’ is enabled the algorithm will merge nodes automatically which are closer then the ‚Repair Tolerance’ value as defined in box ‘Base Mesh’

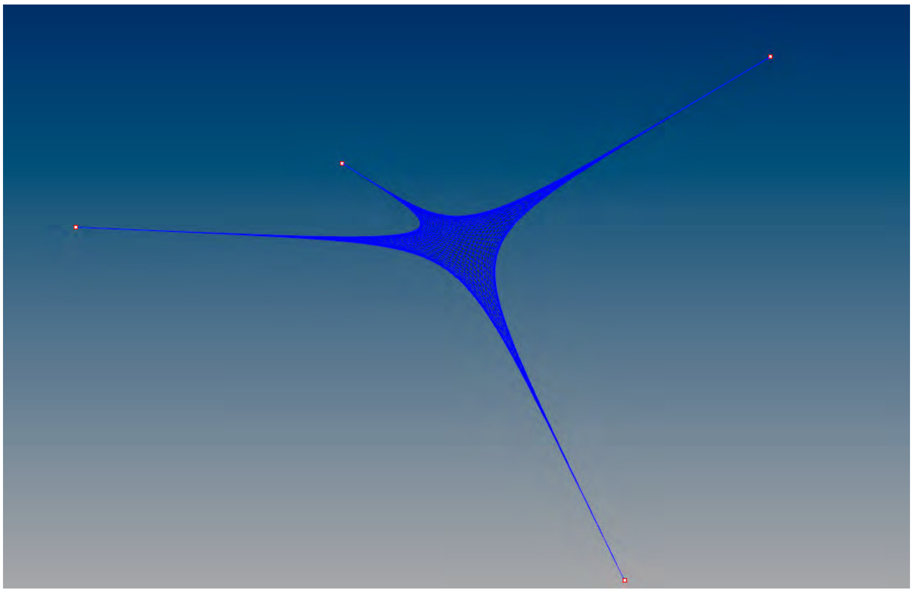
If the selector ‘Auto Step’ the program will automatically evaluate a step value which ensures a stable process. The chosen values is displayed in the disabled selector box below. I is recommended to run the DR process with ‚ Auto Step’ enabled.

In the ‚Edge Stiff:’ selector the cable stiffness of all faces edges along the mesh perimeter can be set. For the default value 1.00 all edge cables will have the same stiffness then the cables within the net/mesh. All other values will amplify/reduce the edges stiffness in relation to the inner members.

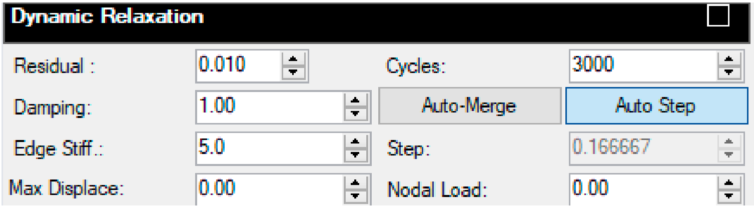
Edge Stiffness = 1.0



Residual : 0.010
Cycles: 1000
Damping: 1.00
Edge Stiff.: 1.0
Step: 0.500000
Max Displace: 0.00
Nodal Load: 0.00

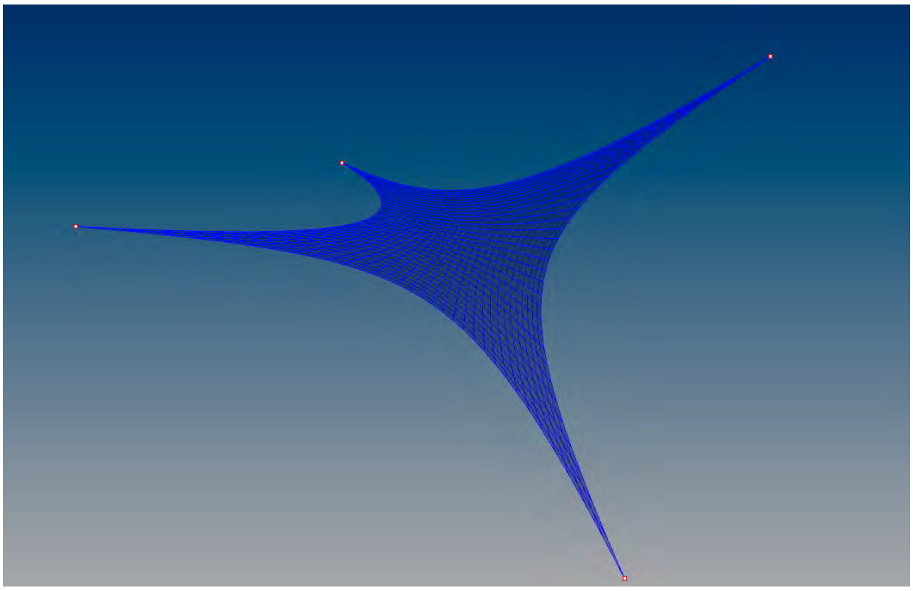


Edge Stiffness = 5.0



Dynamic Relaxation

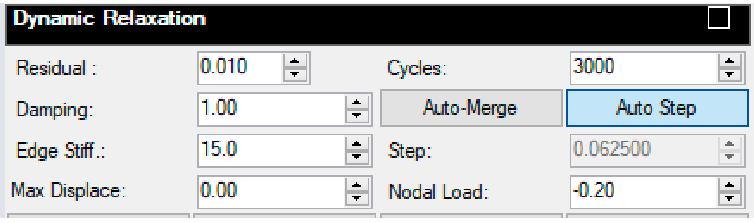
Residual :	0.010	Cycles:	3000
Damping:	1.00	Auto-Merge	Auto Step
Edge Stiff.:	5.0	Step:	0.166667
Max Displace:	0.00	Nodal Load:	0.00



The ‚Max Displace:' selector will limit the displacement of any non – constraint vertex to the specified value in relation to its original position. For the default value 0 no displacement restriction will be considered.

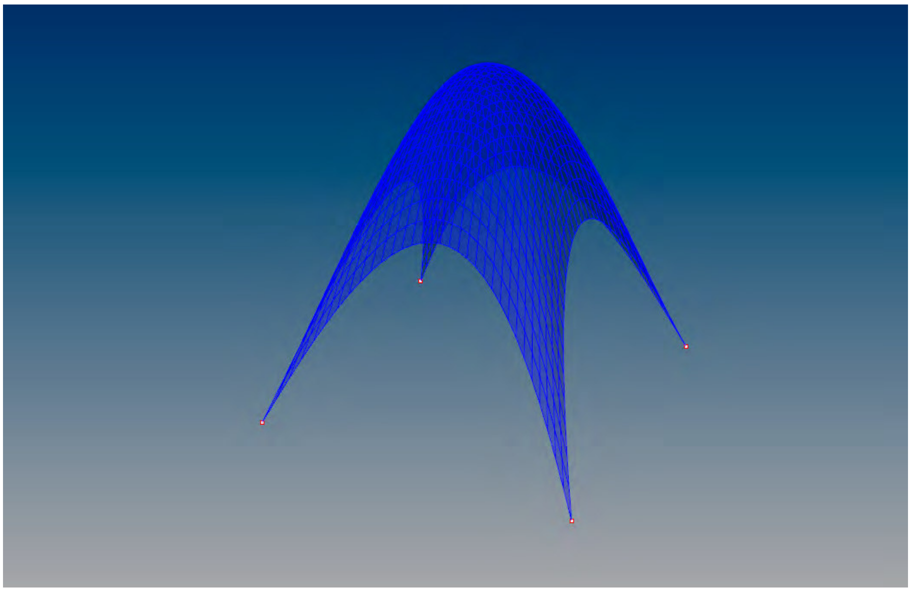
In the ‚Nodal Load' selector external loads can be specified which are applied in global z – direction to each node of the mesh. The loads value is dimensionless and needs to be set in relation to the mesh size. A trail run will show the effect of the applied loading.

For the form finding of shell structure a negative value for the nodal load must be applied.

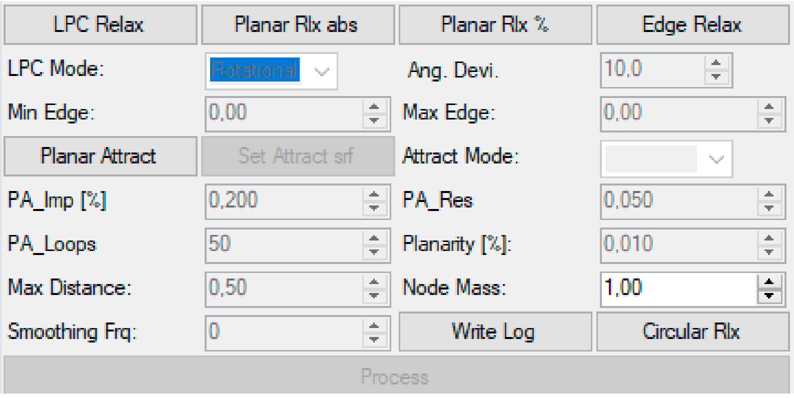


Dynamic Relaxation □

Residual :	0.010	Cycles:	3000
Damping:	1.00	Auto-Merge	Auto Step
Edge Stiff.:	15.0	Step:	0.062500
Max Displace:	0.00	Nodal Load:	-0.20



In the lower part of the box the input parameter for the bespoke DL processing functionality can be defined.



LPC Relax	Planar Rlx abs	Planar Rlx %	Edge Relax
LPC Mode:	Rotational	Ang. Devi.	10,0
Min Edge:	0,00	Max Edge:	0,00
Planar Attract	Set Attract srf	Attract Mode:	
PA_Imp [%]	0,200	PA_Res	0,050
PA_Loops	50	Planarity [%]:	0,010
Max Distance:	0,50	Node Mass:	1,00
Smoothing Frq:	0	Write Log	Circular Rlx

Process

'LPC Relax' – Development still in progress

'Planar Rlx abs' will activate the planar relaxation optimization considering the residual target as an absolute value of planarity for each mesh face.

'Planar Rlx %' will activate the planar relaxation optimization considering the residual target as the relative value of planarity for each mesh face. The following general selection boxes are enabled for the ‚Planar Rlx abs, and ‚Planer Rlx%' : Damping, Auto Merge, Auto Step, Max Displace and Nodal Load.

'Edge Relax' – Development still in progress

In the ‘Planarity [%]: ‘ selector box the residual value both for ‚Planar Rlx abs, and ‚Planer Rlx%’ must be specified.

In ‚Node Mass’ the mass value may be modified which is used in for the DR algorithm – please refer to chapter 4.5.2, Equation 4.2.4. A higher value for m will amplify the internal forces.

Note: ‚Node Mass’ selector is also available for general DR processing.

The ‚Smoothing Frq:’ selector defines after how many DR cycles a CC smoothing cycle will be performed (optional).

If the ‘Write Log’ button is enabled a log file monitoring each cycle of the optimization will be written to the following location: *c:\temp\planar_rlx_log.txt*. Existing files will be overwritten.

Note: Please ensure that the designated folder *c:\temp* exists on the computers hard drive.

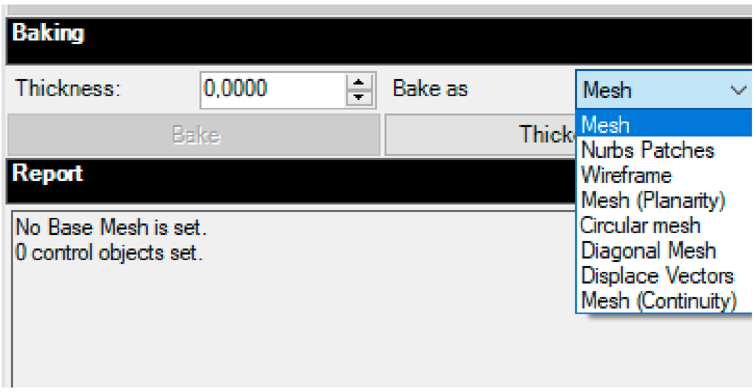
‘Circular Relax’ – Development still in progress

The ‘Process’ button will run the Dynamic Relaxation procedure for the given settings.

7. Baking

In the ‚Baking’ box the geometric objects generated or manipulated with/by the processes and functions described above can be baked as drawings objects into the current rhino file. The following options are available.

Mesh	2D mesh
Nurbs Patches	mesh surface approximated with nurbs patches
Wireframe	mesh edges discretised with lines
Mesh (Planarity)	gradient coloured mesh showing planarity values
Circular mesh	no. 4 circles approximating each mesh face
Diagonal Mesh	diagonal inversion of mesh with triangular faces along perimeter
Displace Vectors	lines objects connecting verticies of base mesh and optimised mesh
Mesh (Continuity)	gradient coloured mesh showing continuity values



Thickness: 0,0000

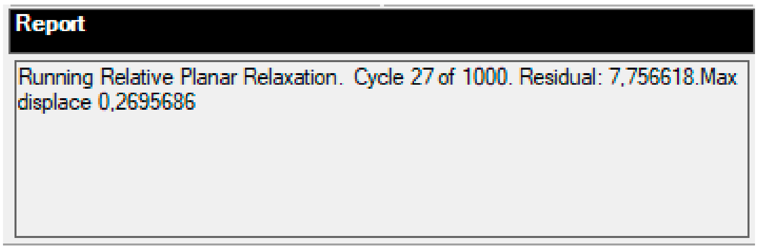
Bake as: Mesh

No Base Mesh is set.
0 control objects set.

In the selection box ‚Thickness’ the option is available to bake a mesh with a specified thickness. For the default value 0.00 a mesh with no thickness will be baked.

8. Report

In the ‘Report box’ information is given about the current selection, optimization or baking process.



Running Relative Planar Relaxation. Cycle 27 of 1000. Residual: 7,756618.Max displace 0,2695686

7.4 The PQ Mesh Approximation Process

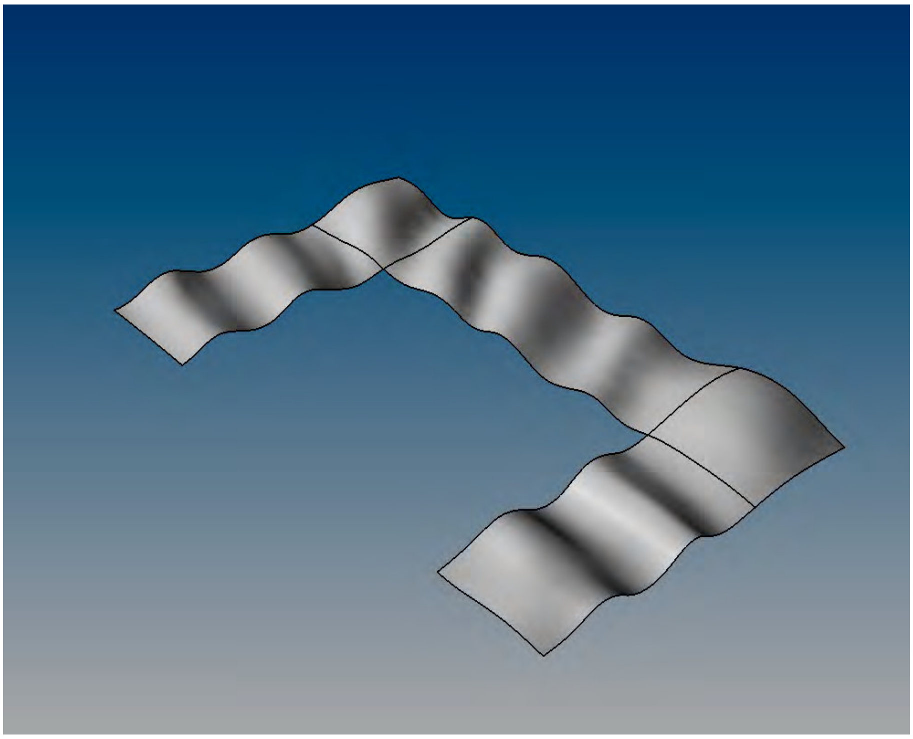
In addition to the theoretical background and procedure described in chapters 4.2 to 4.5 in the following the associated Rhino[(c)] modelling and Enhanced Meshing plugin operations are outlined in further depth. We recall the approximation steps:

Step 1: Principle Curvature Line Sketch
Step 2: Topology Mesh
Step 3: Subdivision Mesh
Step 4: Optimization

For their demonstration the case study Westfield Shopping Mall as summarized in chapter 5.6 is used and to be seen as an extension of the content already provided in this chapter.

Step 1: Principle Curvature Line Sketch

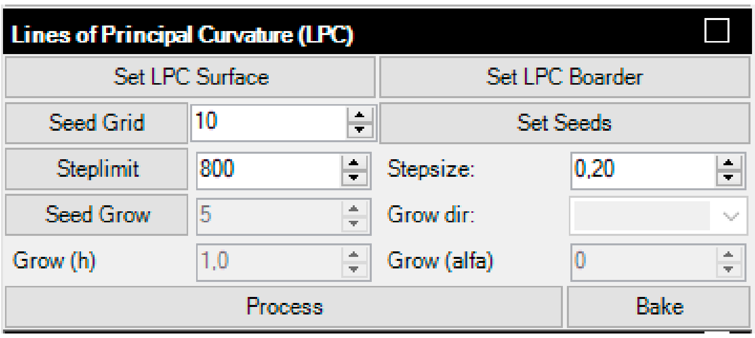
For the processing of the principle curvature line sketch the target surface needs to be represented as a nurbs surface. It is also possible to use a trimmed and joined Brep for the LPC Generation as the algorithm runs over seams. In our example the target surface consist of no. 5 trimmed Breps. One can generate the LPC's for each patch individually or as a whole.



Nurbs surface

Especially if he 'Seed Grid' function is use the generation in individual patches may be preferred. As described previously the 'Seed Grid' function subdivides the u / v - domain by a fixed number chosen by the user and dependant on u/v - density of the particular patch. With ‚Rebuilt Surface' function provided by Rhino the u/v - density can be adjusted in order to match the patch size and desired density of the seeds. Alternatively the seeds can be manually chosen using the 'Set Seeds' function of the Enhanced Meshing plugin. Please make sure that seeds are located on the surface.
For our example we use the ‚Seed Grid' function with the density of 10 processing the entire surface. The sur-

face comprised of the outline dimensions of 120m x 72m. The ‚Stepsize' for LPC generation is set to 0.2m. The ‚Steplimit' is set to 800 which would allows the generation of a single LPC with the a total maximum length of approximately 800 x 0.2m = 160m. Hence a LPC could theoretically run over the entire length of the surface.



Lines of Principal Curvature (LPC)

Seed Grid 10

Steplimit 800

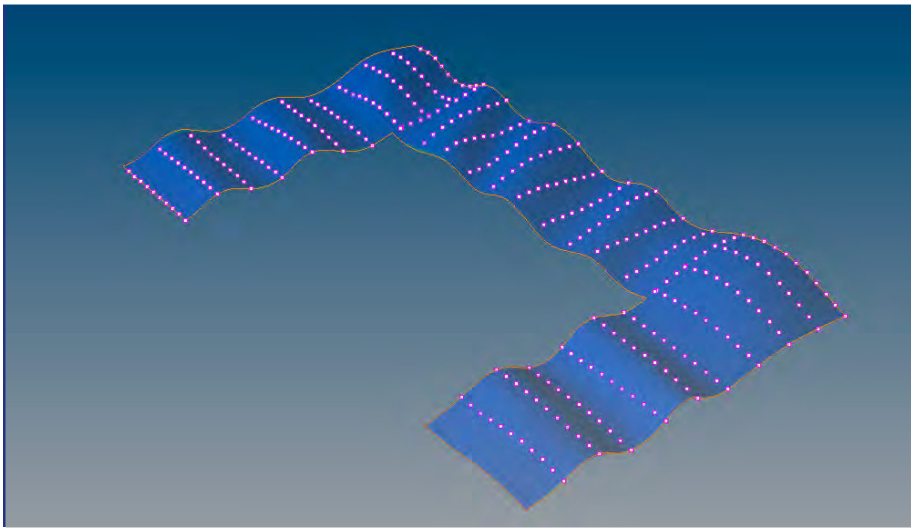
Stepsize: 0,20

Seed Grow 5

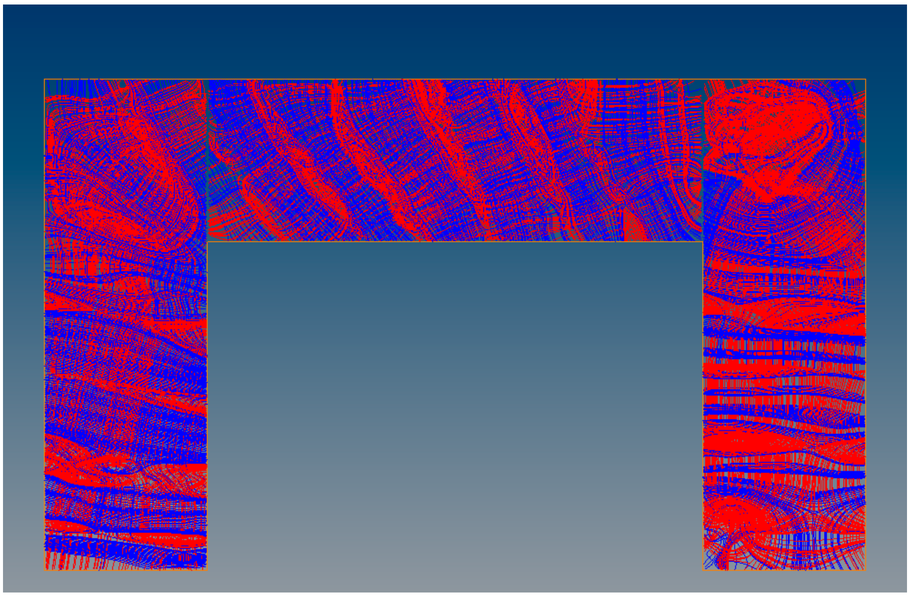
Grow dir:

Grow (h) 1,0

Grow (alfa) 0



LPC surface with LPC border and seeds

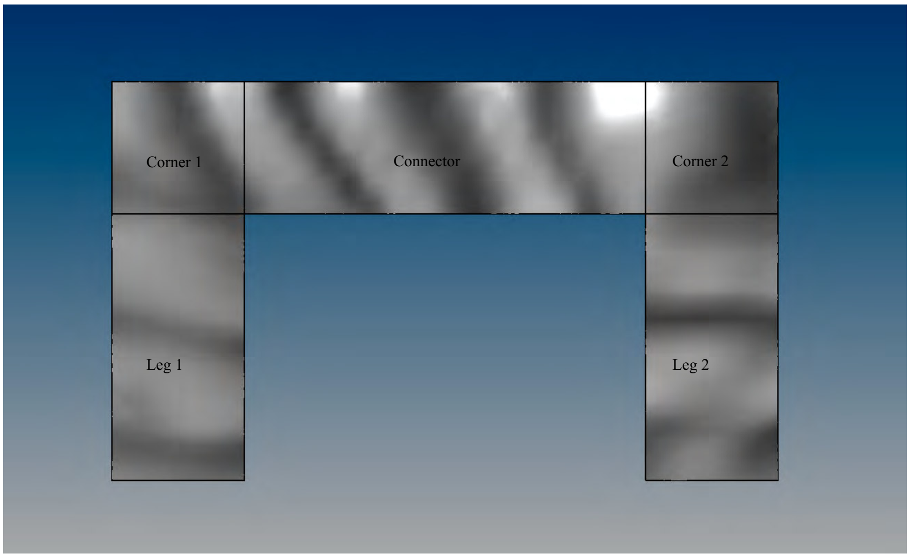


LPC sketch

Step 2: Topology Mesh

As described in chapter 4.3 for the sculpting of the topology mesh a selection of LPC need to be chosen. There location relates to the desired face size and the no of CC - subdivisions required to gain this mesh size. The target surface can be broken down into no 5 patches:

Leg 1 : 23.8m x 48.0m
Corner 1: 23,8m x 23,8m
Connector: 72.0m x 23,8m
Corner 2: 23,8m x 23,8m
Leg 2 : 23.8m x 48.0m



Nurbs patches

If we set the desired face size to approximately 1.5m the width of leg 1/2 needs to subdivided by 23.8m /1.5m = 15.8 faces. The leg length needs to be subdivided by 48m / 1.5m = 32 faces.
In Equation 4.10 a formula is given to estimated the topology mesh size:

$$f_{lc} = f_{lt}\ 2^{i} \qquad \text{Equation 4.10}$$

f_{lc} ... face edge length crude
f_{lt} ... face edge length target
i ... number of subdivisions

In the following table the options for leg 1/2 are summarised:

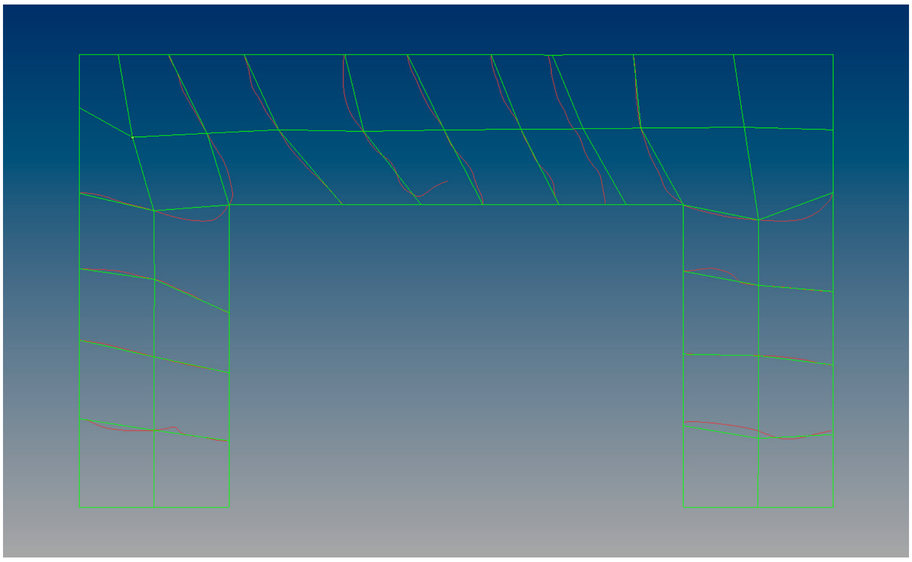
No of subdivisions [i]	face edge length target (flt) [m]	face edge length crude (flc) [m]	Leg 1/2 width [m]	No of crude faces	Leg 1/2 length [m]	No of crude faces
1	1.5	3	24	8	48	16
2	1.5	6	24	4	48	8
3	1.5	12	24	2	48	4
4	1.5	24	24	1	48	2

The option chosen for patch leg 1/2 surface is marked red.

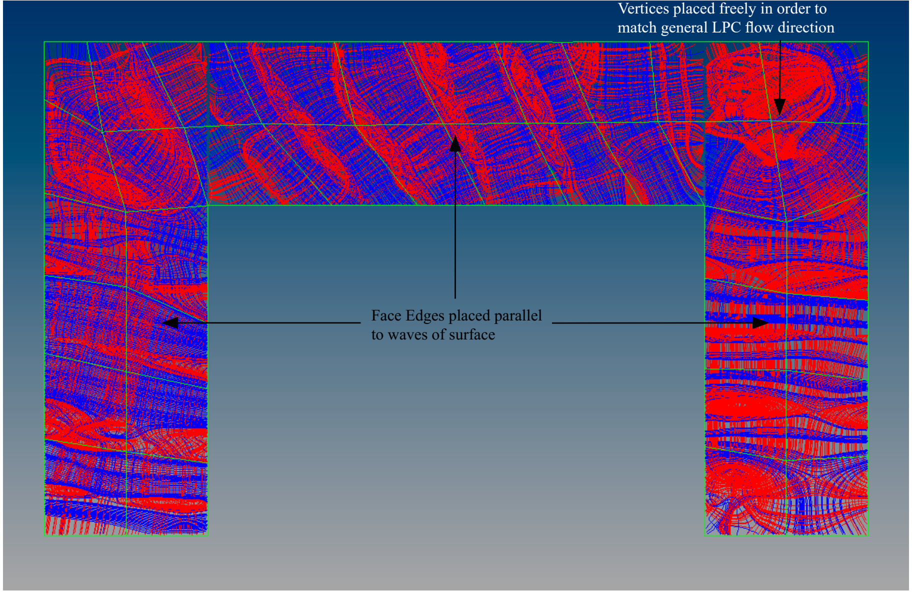
For the corner 1/2 and connector part we proceed respectively:
The count of the topology mesh layout can be summarized to:

Patch	No of Crude faces
Leg 1	2 x 4
Corner 1	2 x 2
Connector	6 x 2
Corner 2	2 x 2
Leg 2	2 x 4

For the geometric location of the mesh vertices we use the LPC sketch. As proposed in chapter 5.6.3 we will pick a series of maximum LPC's as guides. The guides are running in equal distance parallel to the wave ridges or valleys and the vertices of the crude topology mesh are the located at the ends and mid points of the guides. At the two domed areas we locate the vertices more freely in a centred position in order to receive an almost orthogonal layout of the mesh faces.



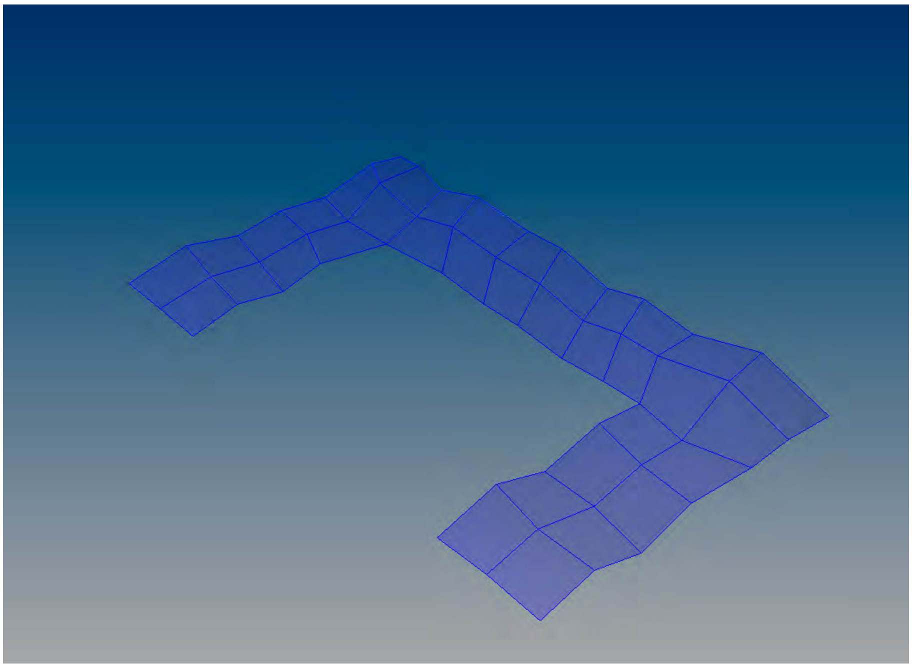
Guide LPC's/Topology Mesh overlay



Vertices placed freely in order to
match general LPC flow direction
Face Edges placed parallel
to waves of surface

LPC Sketch/Topology Mesh overlay

For the placement of the mesh vertices the LPC sketch also can be activated in order to have additional positions available. Any position along a LPC curve is suitable and can be snapped.



Isometric view Topology Mesh

Step 3: Subdivision Mesh

In order to populate the topology mesh we apply 3 iterations of the Catmull Clarke subdivision algorithm. As all newly generated vertices are desired to be exactly be located on the target surface we need to consider the following control objects:

Target surface	All other then edge or corner vertices are snapped to surface
Corner points	Stay in there original position
Perimeter curve	All edge vertices are located on the target perimeter

When selecting the individual control objects the ‚Snap' selector needs to be enabled and an appropriate snap range needs to selected:

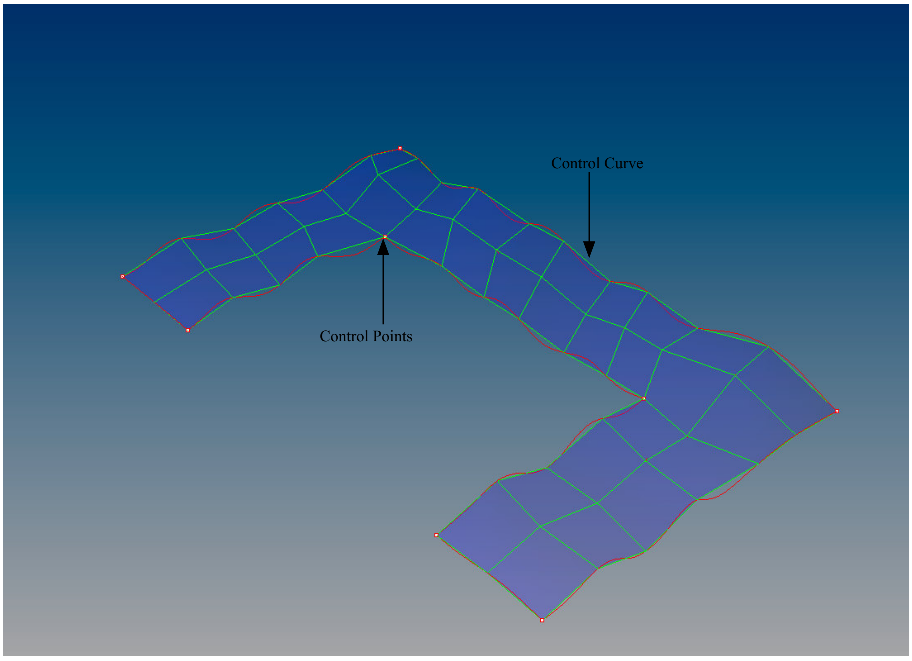
	Snap Range	Explanation
Target surface:	5.0m	Distance of target surface to topology mesh face plane less the 5.0m
Corner points:	1.0m	Desired mesh face larger then 1.0m
Perimeter curve:	1.0m	Desired mesh face larger then 1.0m

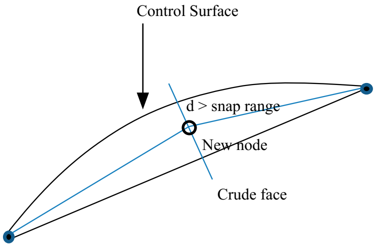
The snap range for control *surfaces* can be chosen more generously as the snap direction is orthogonal to the mesh and hence the vertices are unlikely to collapse into each other during subdivision. However the largest distance between mesh vertex and target surface is governing. As the position of the new generated vertices only can be guessed a trail run my be required to evaluate the adequate snap range.

For control *points* and the snap range needs to be chosen as small as possible in order avoid snapping to vertices in the next row of the mesh. It needs to be less then estimated final mesh face size. In or case the corner control points are already located in their final position and should exactly stay their after subdivision. Hence the snap range can be chosen smaller then the estimated final mesh size - say 1.0m

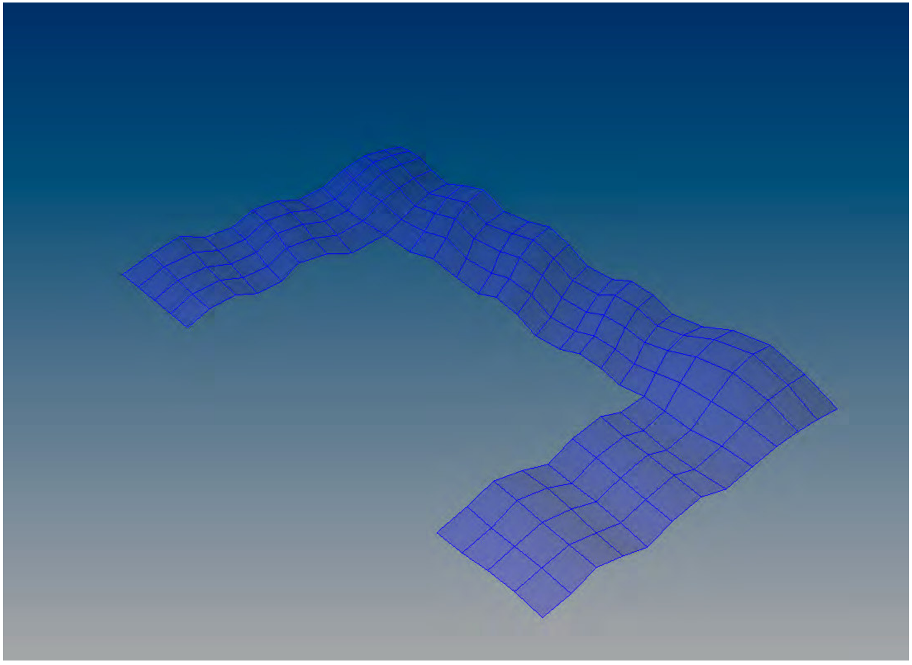
For the *curve* control objects similar conditions apply then for points objects.

However, it could be case that the distance between the control curve and closest vertex is larger then the estimated mesh size. These vertices then will not snap to the control curve. If the snap range is chosen to be larger then the estimated mesh size unwanted vertices will also be snapped to the control curve. In this case the snap range needs to be set for each subdivision iteration individually.

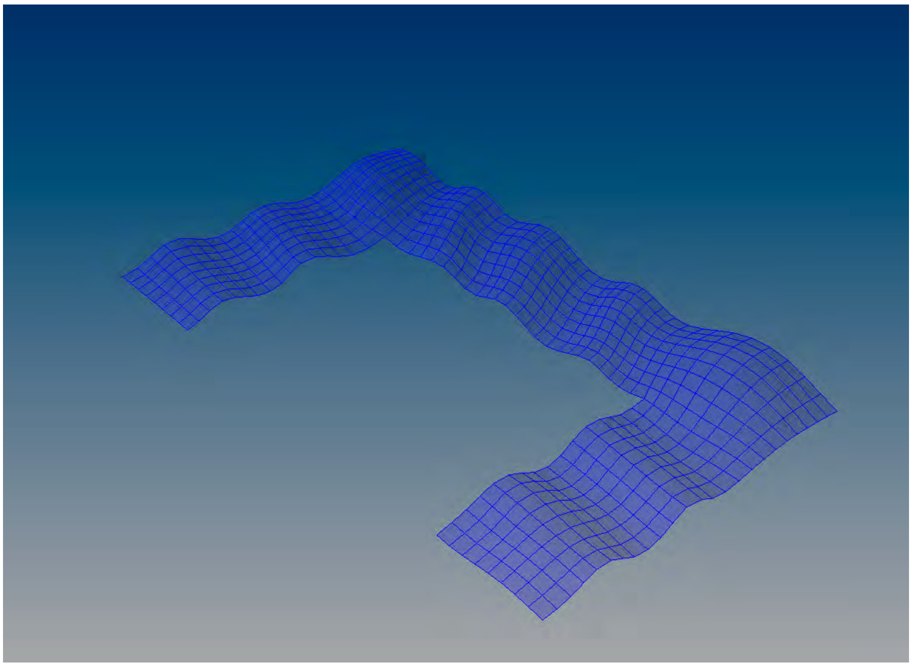




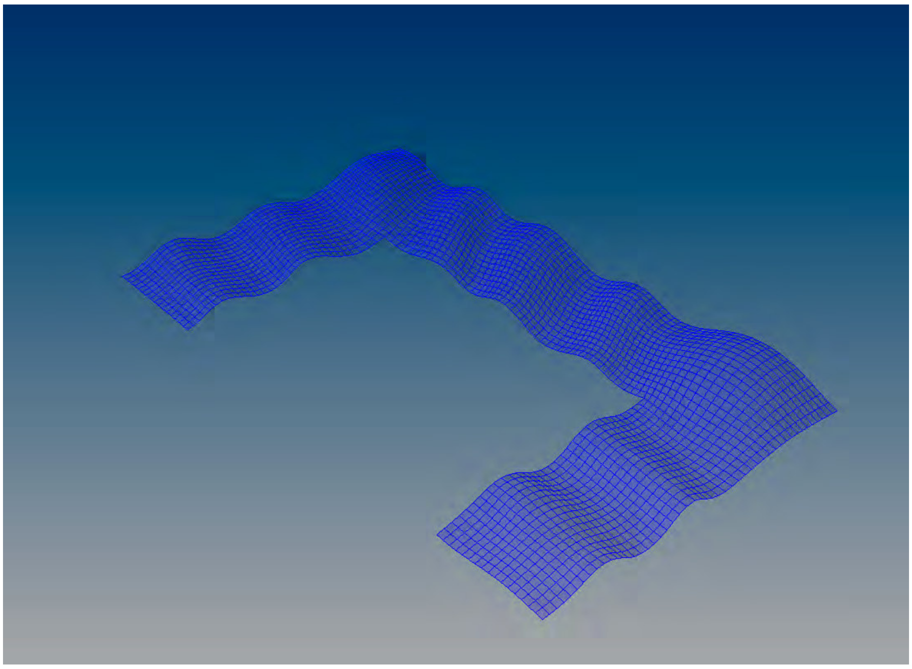
Control objects: Points, Perimeter Curve (Control Surface not shown)



Subdivision 1

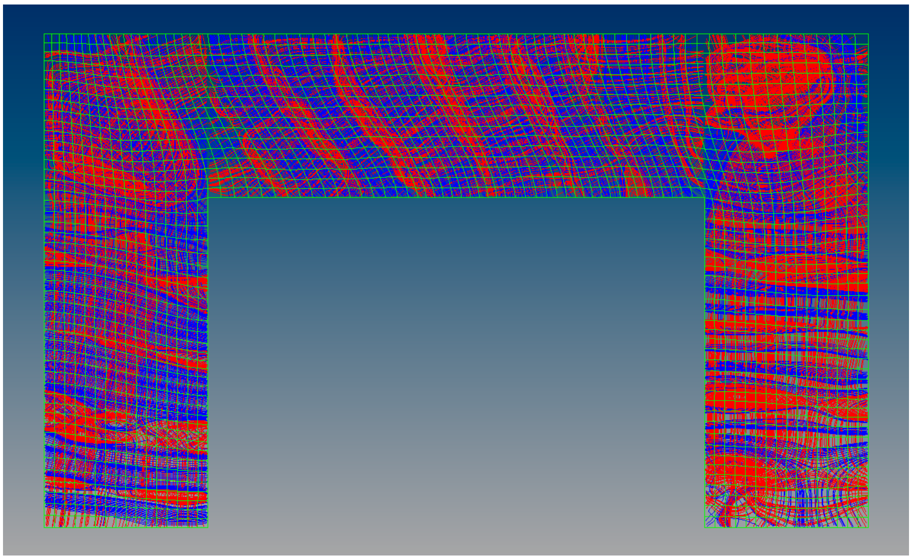


Subdivision 2



Subdivision 3

If we overlay the final mesh with the LPC sketch we can observe a fairly good alignment in most parts of the leg 1/2 and connector patch. For this meshing approach we don not consider the umbilical points.

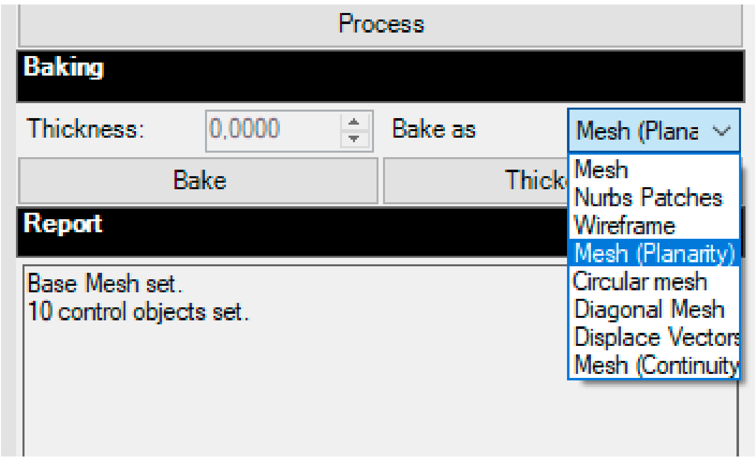


Overlay Subdivision 3 / LPC sketch

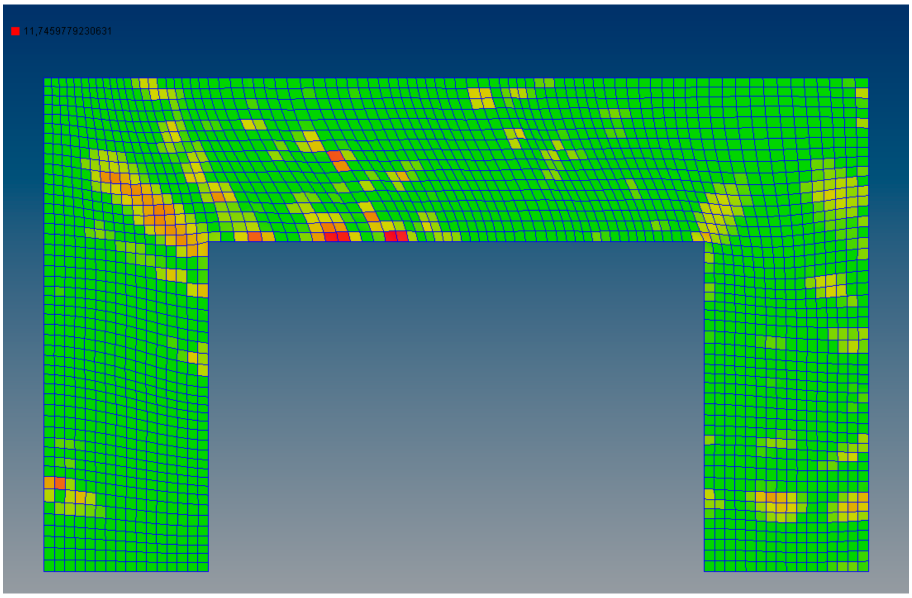
It is recommended to bake the final subdivided mesh to the current rhino file before sating with the Planer Relaxation optimization.

Step 4: Optimization

As the final step we apply the Planar Relaxation optimization to the subdivided smooth mesh. First we analyse the mesh for planarity and mesh continuity using the specific buttons in the Preview Tool box. The gradient coloured meshes can be baked as drawing objects to the rhino file using the choices available in the ‚Baking, tool box.

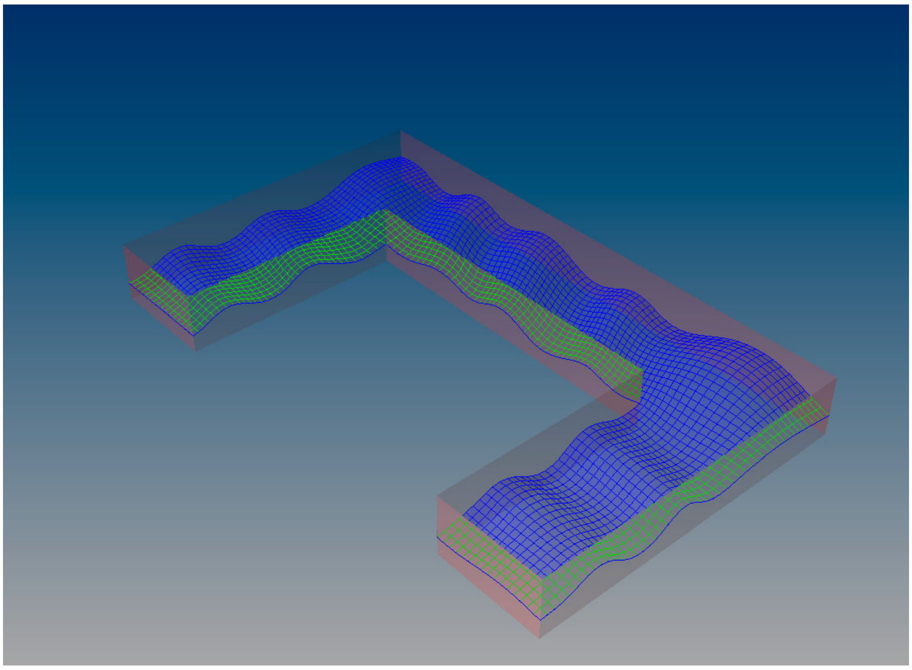


Base Mesh set.
10 control objects set.



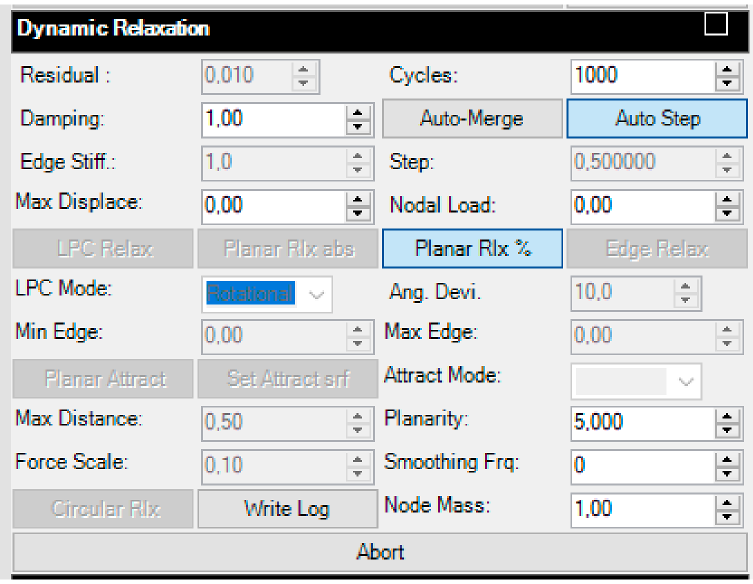
Planarity Gradient mesh, $rp_{n,max} = 0.114$

For the optimization the subdivided mesh from the previous process can be used. However, it is recommended to start with a baked mesh. In case of unexpected numerical problems the results of the subdivision may get lost and needs to performed again. Further this give us more flexibility to remove existing or add new control objects. The best optimization results are expected without consideration of any control object. For our example we only apply a perimeter surface.



Perimeter control surface, snap range 0.5m

We run the, Dynamic Relaxation' with the ‚Planar Rlx %' mode. In order to track intermediate results the optimization target is tighten in the 3 steps: 5%, 2%,1%. In order to receive numerical results of the optimization process at each cycle we enable the ' Write Log' option. The results are written to: *c:\temp\planar_rlx_log.txt.* Existing files will be overwritten. Please make sure that the folder is existing on the harddrive.



Dynamic Relaxation

Residual : 0,010

Cycles: 1000

Damping: 1,00

Auto-Merge

Auto Step

Edge Stiff.: 1,0

Step: 0,500000

Max Displace: 0,00

Nodal Load: 0,00

LPC Relax

Planar Rlx abs

Planar Rlx %

Edge Relax

LPC Mode: Rotational

Ang. Devi. 10,0

Min Edge: 0,00

Max Edge: 0,00

Planar Attract

Set Attract srf

Attract Mode:

Max Distance: 0,50

Planarity: 5,000

Force Scale: 0,10

Smoothing Frq: 0

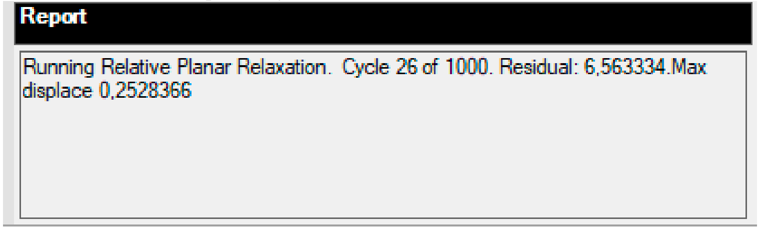
Circular Rlx

Write Log

Node Mass: 1,00

Abort

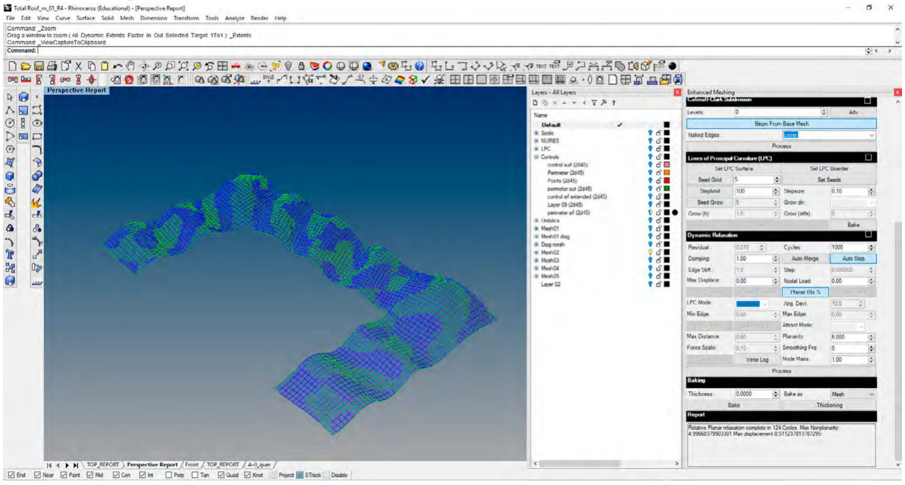
Planar Rlx % mode 5% planartity



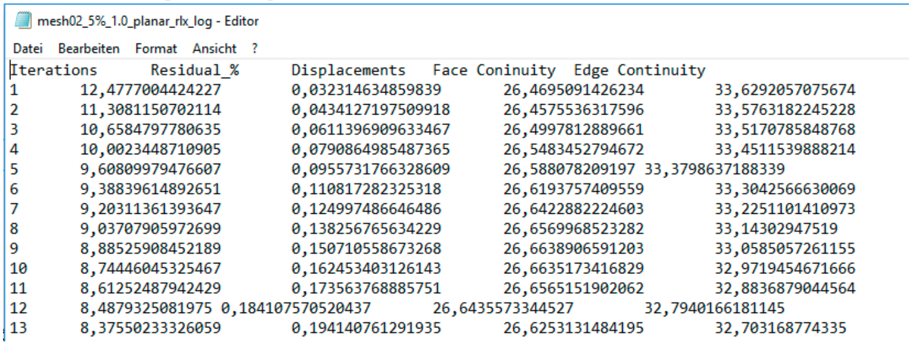
Report

Running Relative Planar Relaxation. Cycle 26 of 1000. Residual: 6,563334.Max displace 0,2528366

Report box display



Visual control of Optimization process



mesh02_5%_1.0_planar_rlx_log - Editor

Datei Bearbeiten Format Ansicht ?

```
Iterations        Residual_%        Displacements    Face Coninuity  Edge Continuity
1         12,4777004424227        0,032314634859839        26,4695091426234         33,6292057075674
2         11,3081150702114        0,0434127197509918       26,4575536317596         33,5763182245228
3         10,6584797780635        0,0611396909633467       26,4997812889661         33,5170785848768
4         10,0023448710905        0,0790864985487365       26,5483452794672         33,4511539888214
5         9,60809979476607        0,0955731766328609       26,588078209197 33,3798637188339
6         9,38839614892651        0,110817282325318        26,6193757409559         33,3042566630069
7         9,20311361393647        0,124997486646486        26,6422882224603         33,2251101410973
8         9,03707905972699        0,138256765634229        26,6569968523282         33,14302947519
9         8,88525908452189        0,150710558673268        26,6638906591203         33,0585057261155
10        8,74446045325467        0,162453403126143        26,6635173416829         32,9719454671666
11        8,61252487942429        0,173563768885751        26,6565151902062         32,8836879044564
12        8,4879325081975 0,184107570520437        26,6435573344527         32,7940166181145
13        8,37550233326059        0,194140761291935        26,6253131484195         32,703168774335
```

Log file (text file)

8 Bibliography

8.1 Table of figures

* Images generated with software ‘Cyclidic Nets’ by E. Huhnen-Venedey http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml

** Images generated with software ‘Kobe Polyhydron Editor’ by S. Echelmann http://www3.math.tu-berlin.de/geometrie/lab/ps.shtml

8.2 References

[Al07] H. Almegaard, The Curvature Coordinate System, 2007, Technical Univeraity of Denmark Lyngby, IASS 2007 Venice Italy : Shell and Spatial Structures: Structural Architecture - Towards the future looking to the past

[AlCo03] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, M. Desbrun, Anisotropic Polygonal Remeshing, 2003, ACM SIGGRAPH '03

[Ba] M. R. Barnes, Non Linear Numerical Solution Methods for Static and Dynamic Analysis of Tension Structures

[Ba99] M. R. Barnes. Form finding and analysis of tension structures by dynamic relaxation. International journal of space structures, 14(2):89–104, 1999. 2.4.1

[Ba02] C. Balmond, Informal, 2002, Prestel, München;Berlin;London;New York, ISBN 3-7913-2400-4

[BeFo97] Daniela Bertol, David Foell, Designing Digital Space, 1997, John Wiley & Sons, ISBN-10: 0471146625, ISBN-13: 978-0471146629

[BHSp05] Alexander I. Bobenko, Tim Hoffmann, Boris A. Springborn, Minimal surfaces from circle patterns:Geometry from combinatorics , Reprint, 2006, Technisches Universitat Berlin, arXi:math.DG/0305184 v3, 231-264

[Bo07] A. I. Bobenko, Surfaces from Circles, 2007, Technisches Universitat Berlin, arXiv:0707.1318v1 [math. DG] 9 Jul 2007

[BoHu12] A. I. Bobenko, E. Huhnen-Venedey, Curvature line parametrized surfaces and orthogonal coordinate systems: discretization with Dupin cyclides, 2012, Geometriae Dedicata ,August 2012, Volume 159, Issue 1, 207-237

[BoPi94] A. Bobenko, U. Pinkall, Discrete Isothermic Surfaces, 1994, Technisches Universitat Berlin, Sfb 288 Preprint No. 143

[BoPo09] A. I. Bobenko, H. Pottmann, J. Wallner, A curvature theory for discrete surfaces based on mesh parallelity, 2009, Technisches Universitat Berlin, arXiv:0901.4620v1 [math.DG] 29 Jan 2009

[BoSo02] A. I. Bobenko, B. A. Springborn, Variational principles for circle patterns and Koebe's theorem, 2002 Technische Universit¨at Berlin, arXiv:math.GT/0203250 v2 3 Jun 2002

[BoSu08] A. I. Bobenko, Y. B. Suris, Discrete Differintial Geometry: Integrateable structures, 2008, American Mathematical Society, ISBN 978-0-8218-4700-8, 32-42

[BoSu08_1] A. I. Bobenko, Y. B. Suris, Discrete Koenigs nets and discrete isothermic surfaces, 2008, Technisches Universitat Berlin, arXiv:0709.3408v1 [math.DG] 21 Sep 2007

[Br91] I.N. Bronstein, Taschenbuch der Mathematik, 1991, B.G. Teubner Stuttgart/Leipzig, ISBN 3-8154-2000-8

[CaCa78] E. Catmull, J.Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, 1978, Computer Aided Geometric Design, 10(6), 350-355

[CaFa06] F. Cazals, J.-Ch. Faugère, M. Pouget, F. Rouillier, Ridges and umbilics of polynomial parametric surfaces, 2006, Springer Verlag, Computational Methods for Algebraic Spline Surfaces II, 141-159

[CaPo05] F. Cazals and M. Pouget, Topology driven algorithms for ridge extraction on meshes, 2005, INRIA, Technical Report RR-5526

[CoMo03] David Cohen-Steiner and Jean-Marie Morvan, Restricted delaunay triangulations and normal cycle, In SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry, pages 312–321, New York, NY, USA, 2003. ACM Press.

[CFPR05] F. Cazals, J.-C. Faugere, M. Pouget, and F. Rouillier, Topologically certified approximation of umbilics and ridges on polynomial parametric surface, 2005, INRIA, Technical Report 5674

[Da65] A. S .Day, An Introduction to Dynamic Relaxation, 1965, The Engineer, Vol. 219, 218-221

[DoKi05] S. Dong, S. Kircher and M. Garland, Harmonic functions for quadrilateral remeshing of arbitrary manifolds, 2005, Computer Aided Geometric Design, 22(5), 392–423

[DoSa78] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, 1978, Computer Aided Geometric Design, 10(6), 356-360

[Fa93] G. Farin, Curves and surfaces for computer, 1993, Acadamic Press, Boston

[Ga97] F.Gauss, Faccetierungen von Flächen, Diplomarbeit, 1997, Institut für Leichte Flächentragwerke Technische Universität Stuttgart

[GlSh04] J. Glymph, D. Schelden, C. Ceccato, J. Mussel, H. Schober, A Parametric Strategy for Freeform Glass Structures Using Quadrilateral Planar Facets, 2004, Elsevier, Automation in Construction 13 (2004), 187– 202

[Ho00] T. Hoffmann, Discrete Curves and Surfaces, Dissertation, 2000, Technischen Universität Berlin

[Ho97] A. Holgate, The Art of Structural Engineering, 1997, Axel Menges, Stuttgart/London, 103-123

[HoGr00] K. Hormann, G. Greiner, Quadrilateral Remeshing, 2000, Proceedings of the 2000 Conference on Vision Modeling and VisualizationNovember 22-24, 2000

[Hu07] E. Huhnen-Venedey, Curvature line parametrized surfaces and orthogonal coordinate systems Discretization with Dupin cyclides, Diploma Thesis, 2007, Technische Universität Berlin Fakultät II - Institut für Mathematik

[KäNi07] F. Kälberer, M. Nieser, K. Polthier, QuadCover - Surface Parameterization using Branched Coverings, 2007, The Eurographics Association and Blackwell Publishing 2007

[Kl97] B.Klotzek, Einführung in die Differentialgeometrie, 1997, Harri Deutsch, Frankfurt am Main, ISBN-10: 3817115490

[Ko03] B. Kolarevic, Architecture in the Digital Age, 2003, Spon Press, ISBN 0-415-38141-x

[KoMa03] K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, F.-E. Wolter, Shape Intrinsic Fingerprints for

Free-Form Object Matching, 2003, Prodeedings of the Eighth ACM Symposium on Solid Modeling and Applications, 196-207

[Kr59] E. Kreyszig, Differential Geometry, Reprint, 1991, Dover Publications Inc., ISBN 0-486-66721-9

[LeTu04] N. Leach, D. Turnbull, C. Williams, Digital Tectonics, 2004, WILEY A-ACADEMY, ISBN 0-470-85729-3

[Li01] B. Lindsey, Digital Gehry Material Resistance Digital Construction, 2001, Birkhäuser Verlag Basel, ISBN 3-7643-6562-5

[LiPo06] Y. Liu, H. Pottmann, J. Wallner, Y. -L. Yang, W. Wang, Geometric Modeling with conical meshes and Developable Surfaces, Reprint, 2006, ACM Transactions on Graphics 25(3) 2006, Proceedings of SIGGRAPH 2006, 681 - 689

[MaKo04], M. Marinov, L. Kobbelt, Direct Anisotropic Quad-Dominant Remeshing, Proceedings, 2004, Proccedings of the 12th Pacific Conference on Computer Graphics and Applications 2004, 207–216

[Ot65] J. R. H. Otter, Computations for Prestressed Concrete Reactor Pressure Vessels Using Dynamic Relaxation, 1965, Nuclear Structural Engineer, Vol. 1, 61-75

[Pa81] M. Papadrakakis, A method for the automatic evaluation of dynamic relaxation parameters and Engineering, Vol. 25, 1981, Computer Methods in Applied Mechanics, 35-48

[Pi08] U. Pinkall, Designing Cylinders with Constant Negative Curvature, 2008, Birkhäuser Verlag Basel, Discrete Differential Geometry, Oberwolfach Seminars, Vol 38, 55-66

[PoAs07] H. Pottmann, A. Asperl, M. Hofer, A. Kilian, Architectural Geometry, 2007, Bentley Institut Press, ISBN 978-1-934493-04-5

[Pol02] K. Polthier, Polyhedral Surfaces of Constant Mean Curvature, Habilitationsschrift, 2002, Technisches Universitat Berlin

[PoLi07] H. Pottmann, Y. Liu, J. Wallner, A. I. Bobenko, W. Wang, Geometry of multi-layer freeform structures for architecture, 2007, ACM Trans, Graphics 26 (2007), no. 3, #65, 11pp

[PoSchi08] H. Pottmann, A. Schiftner, P. Bo, H. Schmiedhofer, W. Wang, N. Baldassini, and J. Wallner, Freeform surfaces from single curved panels, 2008, ACM Trans. Graphics, 27/3, Proc. SIGGRAPH (2008), Graphics, 27/3, Proc. SIGGRAPH (2008)

[PoSchi08_1] H. Pottmann, A. Schiftner, J. Wallner, Geometry of Architectural Freeform Structures, 2008, WILEY-VCH Verlag GmbH & Co. KGaA, Internat. Math. Nachrichten Nr. 209 (2008), 15-28

[PoWa01] H. Pottmann, J. Wallner, Computational Line Geometry, 2001, Springer Verlag, ISBN 978-3-642-04017-7

[PoWa06] H. Pottmann, J. Wallner, The focal geometry of circular and conical meshes, Revised Version, 2006, Technische Universität Wien, Geometry preprint 163

[RaWa06] N. Ray, W. Chiu Li, B. Levy, A. Sheffer, P. Alliez, Periodic global parameterization, 2006

[ReUm06] J. Reiser, N. Umemoto, Atlas of Novel Tectonics, 2006, Princeton Architectural Press, New York, ISBN 978-1-56898-554-1

[Sche02] D. R. Shelden, Digital Surface Representation and the Constructability of Gehry's Architecture, PhD Thesis, 2002, MIT

[Schi07] A. Schiftner, Planar quad mesh from relative principal curvature lines, Diploma Thesis, 2007, Vienna University of Technology

[SchlScho98] J. Schlaich, H. Schober, Glaskuppel für die Flußpferde im Zoo Berlin, 1998Verlag Ernst & Sohn Bautechnik, vol. 67, 1998 April, 307-318

[Scho02] H. Schober, Geometrie-Prinzipien für wirtschaftliche und effiziente Schalentragwerke, 2002, Verlag Ernst & Sohn, Bautechnik, vol. 79, 2002 January, 16-22

[SchoBe05] D. Schodek, M. Bechthold, K. Griggs, K. M. Kao, M. Steinenberg, Digtial Design and Manufacturing CAD/CAM Applications in Arcitecture and Design, 2005, John Wiley & Sons, ISBN 0-471-45636-5

[Schu04] P. Schumacher, Digital Hadid: Landscapes in Motion, 2004, Birkhäuser Verlag Basel, ISBN 3-7643-0172-4

[Se07] S. Sechelmann, Discrete Minimal Surfaces, Koebe Polyhedra, and Alexandrov’s Theorem. Variational Principles, Algorithms,and Implementation, Diploma Thesis, 2007, Technische Universität Berlin Fakultät II - Institut für Mathematik

[Sp03] B. A. Springborn, Variational Principles for Circle Patterns, PhD Thesis, 2003, Fakultät II – Mathematik und Naturwissenschaften der Technischen Universität Berlin

[St01] J. Steele, Architecture and computers, 2001, Laurence King Publishing, ISBN 1-85669-220-5

[St05] K. Stephenson, Introduction to Circle Packing The Theory of Discrete Analytic Functions, 2005, Cambridge University Press, ISBN-0-521-82356-0

[Ste03] K. Stephenson, Circle Packing: A Mathematical Tale, 2003, NOTICES OF THE AMS, VOLUME 50, NUMBER 11, 1376-1388

[Str61] D. J. Struik, Lectures on Classical Differntial Geometry: Second Edition, Reprint, 1988, Dover Publications Inc., ISBN 0-486-65609-8, 58-61, 73-76

[StSaKn04] S. Stephan, J. Sanchez-Alvarez, K. Knebel, Stabwerke auf Freifomflächen, 2004, Verlag Ernst & Sohn, Stahlbau, vol 73, Heft 8, 562 - 572

[Sz05] P. Szalapaj, Contemporary Architecture and the Digital Design Process, 2005, Architectural Press, Oxford, ISBN 0-7506-5716-2

[Th97] W. Thurston, Three - dimensional geometry and topolgy Vol 1, Reprint, 1997, Princeton University Press Princeton Mathematical Series 35, ISBN 978-0-691-08304-9

[Tr01] V. Travi, Advanced Technlogies Building in the Computer Age, 2001, Birkhäuser Verlag Basel, ISBN

3-7643-6450-5

[WaPo0] J. Wallner and H. Pottmann, Infinitesimally flexible meshes and discrete minimal surfaces, 2008, Springer Verlag, Monatshefte für Mathematik 153, 347-365
[WaWa07] W. Wang, J. Wallner, Y. Liu, An Angle Criterion for Conical Mesh Vertices, 2007, Journal for Geometry and Graphics 11(2), 199-208

[Wi01] C. J. K. Williams, The analytic and numerical definition of the geometry of the British Museum great court roof, 2001

[WIKI] Wikipedia the free Encyclopedia, https://en.wikipedia.org/wiki

[ZaSch10] M. Zadravec, A. Schiftner, J. Wallner, Designing Quad-dominant Meshes with Planar Faces, 2010, Eurographics Symposium on Geometry Processing 2010, Volume 29 (2010), Number 5