CURVILINEAR MESH GENERATION FOR CAD MODELS INVOLVING TRIMMED, PERIODIC AND DEGENERATED SURFACES

by RODRIGO DE MIGUEL BÁRCENA

EG-M123 Visiting Research dissertation, Swansea 2020

Student nº:1915329

Advisor: Rubén Sevilla

Abstract

The development and implementation of new capabilities to a linear elastic method for curve high-order 3D mesh generation, is described. The new features include the treatment of degenerated surfaces and an updated definition for periodic surfaces. These surfaces are widely used in engineering designs and are essential characteristics for the adoption of a mesh generator. The surfaces are described by NURBS, whose properties are used to determine whether surfaces are degenerated, periodic or not.

Several examples illustrate the results of the implementation, measuring the quality of the mesh with a distortion method. Finally, in order to prove the utility of the generated meshes, one is used to solve a thermal problem.

Contents

1. Introduction

The finite element method (FEM) can be traced back to the 17th century followed by an initial slow development. Nevertheless, the birth of the computer opened a new era in the development and implementation of FEM. As mention in [1], a community of academics such as A. Hrennikoff, R. Courant in the 1940's and J. H. Argyris or O. C. Zienkiewicz in the 1960s and 1970s developed the primary FEM, and stablished the basis of the modern era of FEM.

As the problems solved by the FEM gained in complexity, and the required accuracy of the results increased, high-order methods appeared. In 1978 the high-order method was proposed to solve fracture mechanics problems [2] and in 1981 it was published the first theoretical paper addressing the p-version of the FEM [3].

In the last years the high-order methods have grown exponentially, in fields such as computational fluid dynamics, acoustics or electromagnetics which are computationally expensive problems.

There are some erroneous beliefs regarding the high-order method that were addressed in [4]. It is known that, for the same number of elements, high-order methods take more CPU time than linear FEM but, it is important to remember that FEM does not give an exact solution to a problem, but an approximation with a certain error. For the same error, high-order methods need a lower number of degrees of freedom (DOFs) than linear methods, what makes high-order methods faster than the traditional ones.

In order to exploit the capabilities of high-order methods, different high-order curved finite elements techniques have been created. A comparison of four of them can be found in [5].

Further developments in the high-order method have been made to improve its accuracy and velocity. For example, the h-p version, which combines increasing the order of the mesh and the reduction of the elements size where it is needed. More recent methods like NEFEM [6] have been proven to be superior in problems such as wave-scattering, where h refinement implies reducing the time steps, resulting on a higher number of steps, which increases the overall computing time. Development in these and other methods is been made.

A series of setbacks need to be addressed before the widespread adoption of the high-order method in the design process including (i) the robustness and convergence rate, (ii) the memory requirements, and (iii) the high-order curved mesh generation.

As previously mention, several high-order curved finite element methods can be found, some of them in process of development to alleviate different problems such as the convergence rate. Improving the algorithms will reduce the memory requirements.

The third problem, the availability of high-order curved mesh generators, is also been studied. The discretization of the domain with a high-order mesh is the first step in high-order finite element analysis. There exist several procedures for generating the mesh that can be divided in *direct* and *a-posteriori* approaches as explained in [[7], [8]].

The *direct* approach begins with the discretization of the curvilinear boundary of the model followed by the interior discretization of the domain. This method ensures a valid mesh, but it has a high computational cost.

A-posteriori approaches begin with the generation of a linear mesh. Then, the entities laying on the curved boundary are modified to match the boundary of the domain using different methods such as, *optimisation* [9], *local modification of geometric entities* [10] or the *solid mechanic analogies* [11] proposed for the first time in 2009. Mesh modifications are localized in the boundary regions, this coupled with the already available well-developed linear mesh generators, made *a-posteriori* approaches to be preferred over the direct ones.

The *optimisation method* and the *modification of geometric entities method* only modify the external edges/faces of the elements while the interior ones remain unchanged. This can lead to poor quality boundary elements, or even invalid ones if the deformation of the exterior edge/face is big enough to intersect with the interior parts, as mentioned in [[11], [9]] and shown in the following figure.

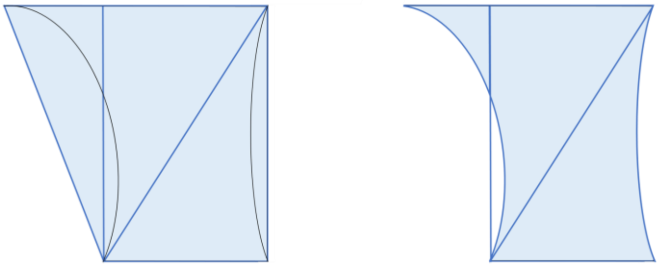

Figure 1 Example of a mesh curved using direct methods

These methods required a postprocess procedure to correct the invalid elements, such as the algorithm explained in [9], by curving also the internal entities of the domain.

When using *solid mechanics* procedures, both boundary and internal edges are deformed. In this case, choosing the correct material properties is a key ingredient to achieving good quality elements. Poisson ratios near the incompressible limit, are demonstrated to give the best quality elements with a proper deformation of the interior edges as depicted in [8].

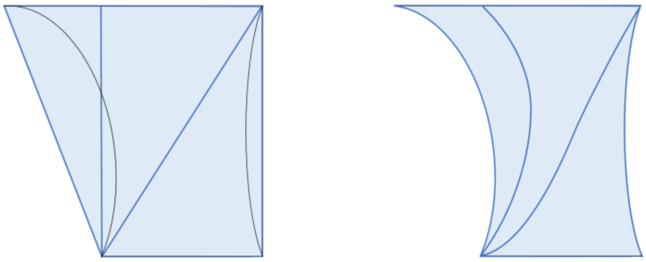

Figure 2 Example of a mesh curved using a posteriori methods

The first proposed *solid mechanics method* [11], make use of the non-linear elasticity problem, but later proposed procedures based on linear elasticity [12], using incremental steps to deal with large deformations, are being prove to deliver high quality meshes with a lower computation cost [8].

1.1. Objectives

This thesis is focused on the third of the above-mentioned problems, the high-order curved mesh generation. The following pages describe the development and implementation of new capabilities to the linear elastic mesh generation method explained in [12].

At the beginning of this project, the implementation of [12] is not automatically able to deal with degenerated surfaces and, although a procedure to treat periodic surfaces is already developed, the definition of periodic surfaces has to be changed to include a wider variety of surfaces. Degenerated and periodic surfaces commonly appear in engineering designs, making the implementation of these surfaces a key upgraded to the software, if it is to be used for finite element analysis.

The work is divided in three main parts, (i)the development and implementation of the new capabilities to the software, (ii)the creation and study of high-order meshes involving the upgrade and (iii) the use of one of the meshes for solving a thermal problem and review its performance.

2. Geometric modelling

2.1. Introduction to the NURBS

This chapter is a brief introduction to the Non-Uniform Rational B-Splines. The NURBS are a type of piecewise parametric curves commonly used in engineering. Their characteristics are making NURBS spread from CAD software to finite element analysis.

Parametric form of a curve

The mathematical representations of a curve or surface is generally done using either an implicit or a parametric description.

An implicit representation of a curve consists of a function of several variables that define the equation that the coordinates must satisfy to be on the curve. For instance, the implicit expression of the unit circle is

$$x^2 + y^2 - 1 = 0 \tag{1}$$

In a parametric representation each coordinate is defined as a function of a common independent variable or parameter. The parametric expression of the unit circle is

$$x(t) = \cos(t) \qquad\qquad y(t) = \sin(t) \tag{2}$$

It shall be noticed that, unlike the parametric curve, in the implicit form, the same value of one of the coordinates can lead to several points on the curve. In the example of the circle, for $x = 0$, the point of the curve can be either in the upper or in the lower part of the circle, depending on the value of y that can be 1 or -1.

On the other hand, the parametric form of a curve can be understood as the mapping of a straight line of value t, the independent variable, into a curve.

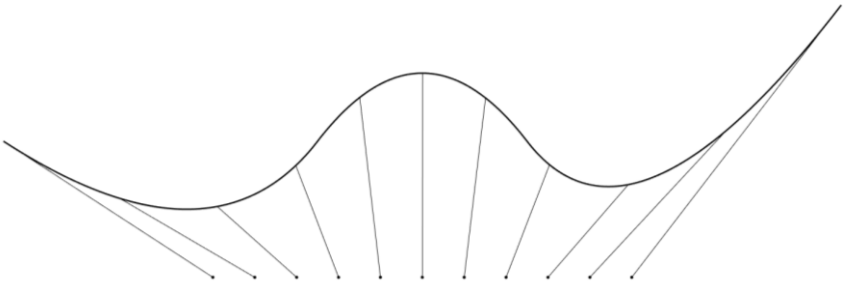

Figure 3 Mapping of a parametric curve from its independent variable t

When it comes to representation, each method has its advantages over the other, but there are several reasons why the parametric method is the most common in CAD software. Among these reasons stands out (i) its more geometrical flavour, which gives the designer a more natural way of interacting with the CAD model, (ii) the ease to expand from 2D to 3D and (iii) a faster computation of points in the curve.

There are also some disadvantages such as the difficulty to determine if a point is on the curve/surface or not and the existence of parametric anomalies that does not relate to true geometry like the poles in a sphere.

These issues are treated later in this document both in the point projection and degenerated points sections.

Bezier curves

There are several ways of representing the same curve in parametric form. For the sake of simplicity, the class of functions are restricted.

The basis functions need to have a correlation with the geometrical space in order to be easily used when creating the designs.

Bezier curves are formed from a set of polynomials known as the Bernstein polynomials that are defined as

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} B_{i,n}(u)\boldsymbol{P_i} \qquad 0 \le u \le 1 \qquad (3)$$

where $\{B_{i,n}(u)\}$ are the Bernstein polynomials of n-*th* degree and $\{\boldsymbol{P_i}\}$ are the so-called control points. As it will be illustrated further down, the control points are coefficients of the curve with a geometrical meaning.

Bernstein basis functions $\{B_{i,n}(u)\}$

Bernstein basis functions have certain properties that determine the characteristic of Bezier curves. Some of these properties can be noticed in the following figure.

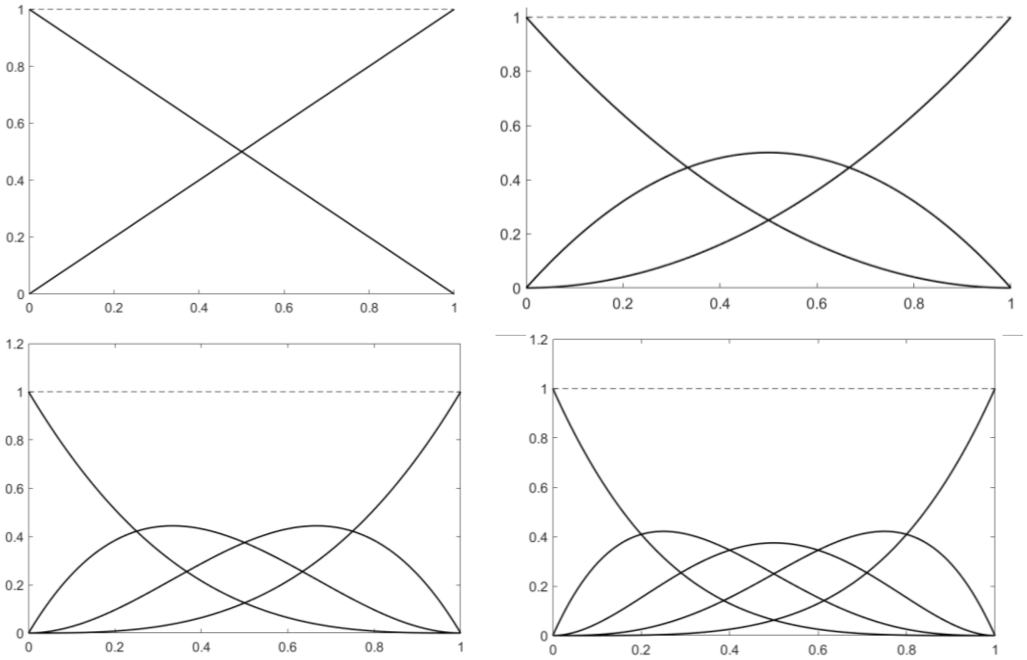

Figure 4, Bernstein basis functions for n=1,n=2, n=3 and n=4. The dashed line is $\sum_{i=0}^{n} B_{i,n}(u) = 1$

- Nonnegativity: $B_{i,n}(u) \geq 0$ for all i, n and $0 \leq u \leq 1$
- Partition of unity: $\sum_{i=0}^{n} B_{i,n}(u) = 1$ for all $0 \leq u \leq 1$.
- Value 1 at one edge of the first and last polynomials $B_{0,n}(0) = B_{n,n}(1) = 1$.
- Each polynomial attains one maximum in [0,1] at $u = i/n$.
- Symmetry for any n with respect to $u = 1/2$.

Control points $\{\boldsymbol{P_i}\}$

The control points are the geometric coefficients. The $\{\boldsymbol{P_i}\}$ points form the *control polygon* that determines the path the curve is following. Some important properties are:

- The control polygon approximates the shape of the curve
- The initial and final points of the curve lie in the initial and final control points.
- The curve is tangent to the control polygon at both corners.
- The change of a control point modifies the entire curve.

These properties can be seen in the following figure.

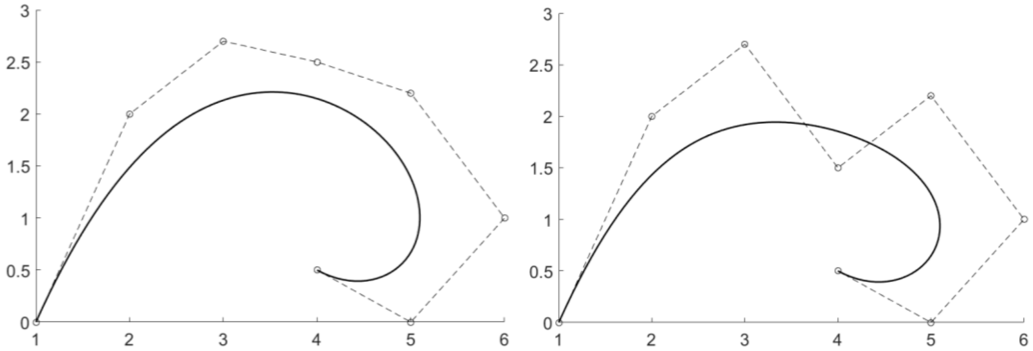

Figure 5 Bezier curves for two similar control points distribution

Among the advantages of using Bezier curves, it can be mentioned (i) the geometrical oriented algorithms, which help the designers, (ii) the numerical stability and (iii) the geometric invariance which allows the rotation, translation and scaling of the curve.

On the other hand, the degree of the Bernstein polynomials is $n-1$, equal to the number of control points, leading to a high degree polynomial when the number of points increases. Also, if one of the control points is modified, the entire curve changes.

B-spline curves

Spline functions consist in piecewise polynomials constructed with m-*th* degree segments. This type of functions gives similar results to polynomials but maintaining a lower degree.

B-splines or basis splines are a type of spline functions that, as well as Bezier curves, consists on a basis function multiplied by a set of control points which give the geometrical meaning to the curve.

To overcome the disadvantages of Bezier curves B-splines are used. The expression for the B-spline curves is

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P_i} \qquad a \leq u \leq b \tag{4}$$

The *control points* are equal to the ones in Bezier curves whereas the basis functions are changed.

B-spline basis functions

A knot vector $\boldsymbol{U}$ is created. This vector consists of $m-1$ non-decreasing values u_i called knots. The $\boldsymbol{i}$*-th* B-spline basis functions of p-degree are

$$N_{i,0}(u) = \begin{cases} if\ u_i \leq u \leq u_{i+1} \\ 0\ otherwise \end{cases} \tag{5}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u) \tag{6}$$

Looking at the equation it can be seen that, $N_{i,p}(u) = 0$ if *u* is outside the interval $[u_i, u_{i+p+1})$. This yields to local support property meaning that, modifying a control point, only modifies certain segments of the curve, solving one of the problems found in Bezier curves.

In this report the knot vectors $\boldsymbol{U}$ will be of the form

$$\{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$$

Having $p+1$ times the initial and final knot and *m*+1 total number of knots.

If the knot vector is of the form $\boldsymbol{U} = \{0, \dots, 0, 1 \dots, 1\}$, resulting B-spline basis functions are the Bernstein polynomials.

Figure 6 shows the B-spline basis functions of degree p=2 for two different knot vectors.

$$\boldsymbol{U} = \{0,0,0,0.1,0.5,0.7,0.7,1,1,1\}$$

$$\boldsymbol{U} = \{0,0,0,0.2,0.4,0.6,0.8,,1,1,1\}$$

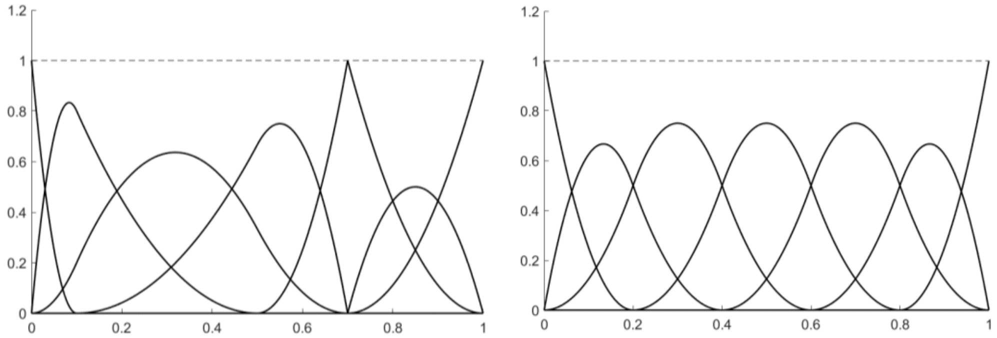

Figure 6 B-spline basis functions of degree p=2 for two different knot vectors

When compared with Bernstein polynomials, the nonnegativity and partition of unity is still satisfied in the whole domain, but symmetry is no longer a needed characteristic of the B-spline basis functions.

The other problem of the Bezier curves is solved by the piecewise definition. The order of the curve can be chosen by the designer and no longer be set by the number of control points.

Increasing the degree increases the continuity. Continuity is also affected by the repetition of knots, the more times a knot is repeated, the lower is the continuity. This can be noticed in *Figure 7* which shows the B-spline curves constructed with the control points $\boldsymbol{P}$ and with a similar knot vector but with different knot repetition.

$$\boldsymbol{P} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 0 & 2 & 0 & 2 & 0 \end{bmatrix}$$

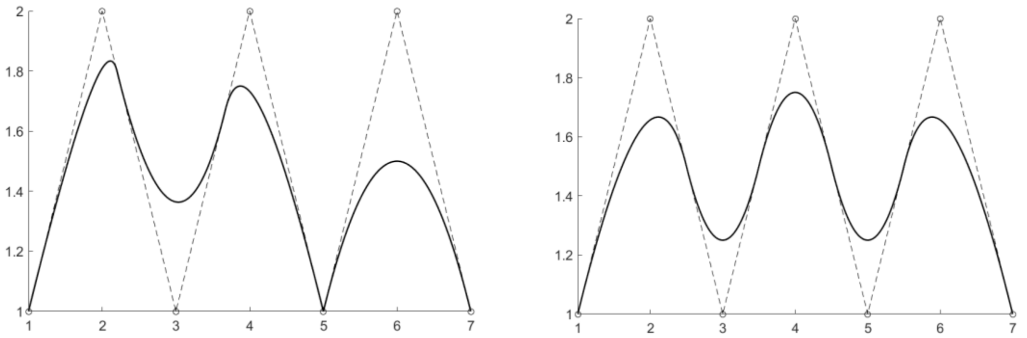

Figure 7 B-splines for same control points distribution but different knot repetition

Rational Curves

Polynomials are versatile to represent many curves, but they cannot represent conic curves nor surfaces such as circles, hyperbolas, spheres, etc. which are widely used in mechanical engineering design.

Rational functions are defined as, the division of a polynomial by another polynomial. This type of functions can represent conic curves and are used as basis functions for representing conic curves and surfaces.

The rational B-splines are defined as

$$\boldsymbol{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \boldsymbol{P_i}}{\sum_{i=0}^{n} N_{i,p}(u) w_i \boldsymbol{P_i}} = \sum_{i=0}^{n} R_{i,p}(u)\boldsymbol{P_i} \qquad a \leq u \leq b \qquad (7)$$

where the basis functions are

$$R_{i,p} = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^{n} N_{j,p}(u) w_i} \qquad (8)$$

It is worth noting that the rational functions introduce the so-called weights, w_i. These values are related to each control point creating the weighted control matrix. They can be understood as the value of the gravity of the control points. The higher the value the closer the curve will get to the control point. The next figure shows a comparison of two curves with the same control points but different weights.

$$P_1^w = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad P_2^w = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 5 & 1 & 1 & 1 \end{bmatrix}$$

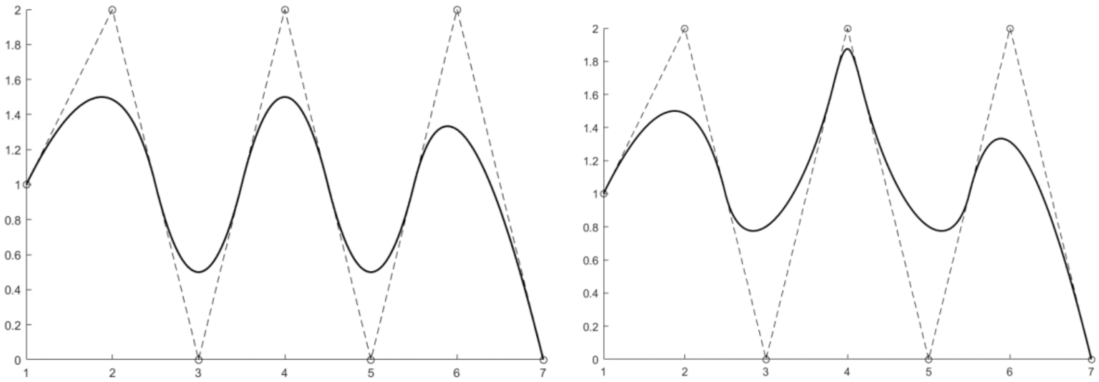

Figure 8 Rational curves for the same control points distribution but different weights

Uniform vs Non-Uniform

In a uniform knot vector, the distance between knots is always the same.

$$U = \{0,2,4,6,8\}$$

In a Non-uniform knot vector, the distance between knots can change.

$$U = \{0,2,3,7,9\}$$

Non- uniform vectors can be used for setting knot multiplicity higher than one in case the curve continuity in a region wants to be changed.

Uniform knot vectors are a special case of the non-uniform knot vectors, so if non-uniform vectors can be computed the uniform ones are also included, that is the reason why the algorithms are developed considering non-uniform vectors.

2.2. NURBS surfaces

A surface is treated as an extension of the parametric curve, it is driven by two independent values (u, v) instead of one. The parametric form of a surface is a vector-value function which represents the mapping from a 2D rectangular plane to a 3D surface. The expression of NURBS surfaces is as follows.

$$\boldsymbol{S}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u, v) \boldsymbol{P_{i,j}} \tag{9}$$

where $R_{i,j}(u, v)$ are the basis functions.

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^{n} \sum_{l=0}^{m} N_{k,p}(u) N_{l,q}(v) w_{k,l}} \tag{10}$$

Curves can be of different order in the u and v directions and the orders are denoted by p and q respectively. The weighted control points $\boldsymbol{P_{i,j}}$ form a control net with i points in u direction and j points in v direction. In *Figure 9* two surfaces have been constructed using the same net of control points but, with different orders in the u direction.

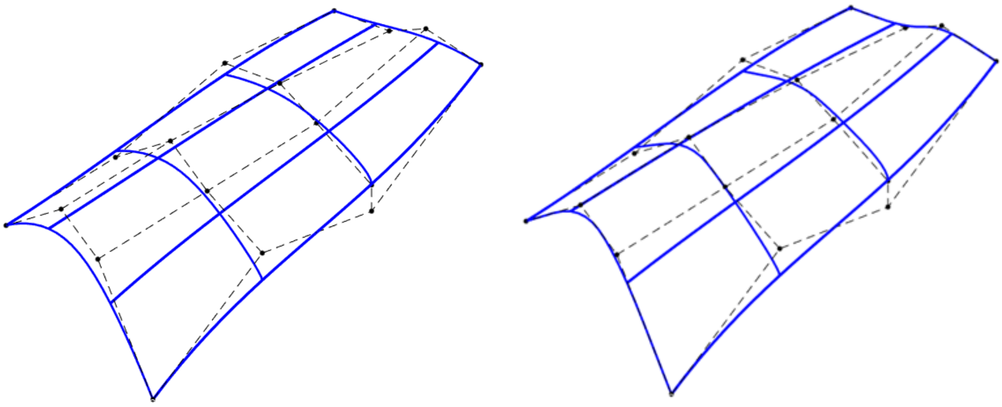

Figure 9 NURBS surfaces for the same control points net. For the surface of the left p=4, q=3 and for the surface on the right p=2 q=3

The properties of NURBS surfaces, derived from its basis functions, are the same as the NURBS curves.

2.3. NURBS operations

Point projection

As previously explained, computing a point of a parametric curve, given the independent parameter u is an easy task but, to determine whether a point is in the curve or not is cumbersome.

In this regard a Newton iteration method is used to build the algorithm explained below.

A point $\boldsymbol{P}$ is considered to be in the curve if the closest distance to the curve is under previously specify tolerances (ϵ_1, ϵ_2).

1. Compute n equally spaced (in the parametric space) points of the NURBS curve, these are the test points. The closest a test point is to $\boldsymbol{P}$**,** the fastest will be the convergence of the method.

2. Determine which test point is the closest one to $\boldsymbol{P}$**.** This test point is called u_o.

3. The following dot product function is constructed to find the minimum distance from $\boldsymbol{P}$ to $\boldsymbol{C}(u)$.

$$f(u) = \boldsymbol{C}'(u) \cdot (\boldsymbol{C}(u) - \boldsymbol{P}) \tag{11}$$

   Notice that only two possibilities make this equation equal 0. The first one, if $\boldsymbol{C}(u)$= $\boldsymbol{P}$ which means that the point $\boldsymbol{P}$ lies on the curve. The second one is a property of dot product functions. When $\boldsymbol{C}^{'}(\boldsymbol{u})$**,** the derivative of the curve on the point u, is perpendicular to the difference $\boldsymbol{C}(u) - \boldsymbol{P}$, the result will be $f(u) = \boldsymbol{0}$**.**

4. Now the newton iteration is set.

$$u_{i+1} = u_i - \frac{f(u_i)}{f^{'}(u_i)} = \frac{\boldsymbol{C}^{'}(u_i) \cdot (\boldsymbol{C}(u_i) - \boldsymbol{P})}{\boldsymbol{C}^{''}(u_i) \cdot (\boldsymbol{C}(u_i) - \boldsymbol{P}) + \left|C^{'}(u_i)\right|^2} \tag{12}$$

   u_i is the value of the parameter at the $\boldsymbol{i}$-*th* Newton iteration.

5. The iteration is halted when the following convergence criteria are satisfied.
   a. Point coincidence.

$$|\boldsymbol{C}(u) - \boldsymbol{P}| \leq \epsilon_1 \tag{13}$$

   b. Zero cosine.

$$\frac{|\boldsymbol{C}^{'}(ui) \cdot (\boldsymbol{C}(ui) - \boldsymbol{P})|}{|\boldsymbol{C}^{'}(ui)| \cdot |(\boldsymbol{C}(ui) - \boldsymbol{P})|} \leq \epsilon_2 \tag{14}$$

   c. The parameter does not change significantly from the previous iteration.

$$|(u_{i+1} - u_i) \cdot \boldsymbol{C}^{'}(u_i)| \leq \epsilon_1 \tag{15}$$

6. An extra step is needed in case the parameter is outside the valid range given by the knot vector of the curve $\boldsymbol{U}$.

   If the curve is not closed

$$if\ (u_{i+1} < \boldsymbol{U}(1)) \rightarrow u_{i+1} = \boldsymbol{U}(1)$$
$$if\ (u_{i+1} > \boldsymbol{U}(end)) \rightarrow u_{i+1} = \boldsymbol{U}(end)$$

   If the curve is periodic.

$$if\ (u_{i+1} < \boldsymbol{U}(1)) \quad \rightarrow u_{i+1} = \boldsymbol{U}(end) - (\boldsymbol{U}(1) - u_{i+1})$$

$$if\ (u_{i+1} > \boldsymbol{U}(end)) \ \rightarrow \ u_{i+1} = \boldsymbol{U}(1) + (u_{i+1} - \boldsymbol{U}(end))$$

The point projection for a surface is analogous to the one explained here for the curve, but it needs to be extended to consider two independent parameters u, v.

2.4. Degenerated and periodic surfaces

In this sections two important cases of NURBS are studied due to their importance in mechanical engineering design, namely closed NURBS curves and surfaces.

Periodic curves and surfaces

A periodic or closed curve is a curve that starts and ends at the same point. Looking at the parametric space of a closed curve, the first and last points of it refers to the same point in the real space. This can be shown in *Figure 10* with a line in the middle representing the parametric space and the mapping to the real space.

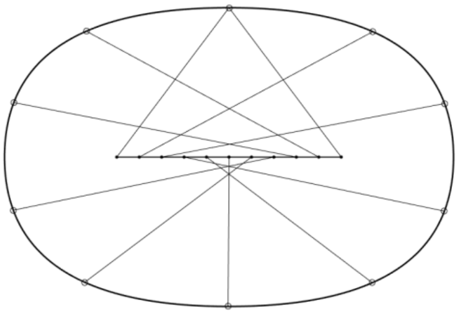

Figure 10 Periodic curve and it's parametric space

In a surface, the periodicity occurs when one edge of the parametric space is the same as the opposite edge. This can be seen in *Figure 11*.

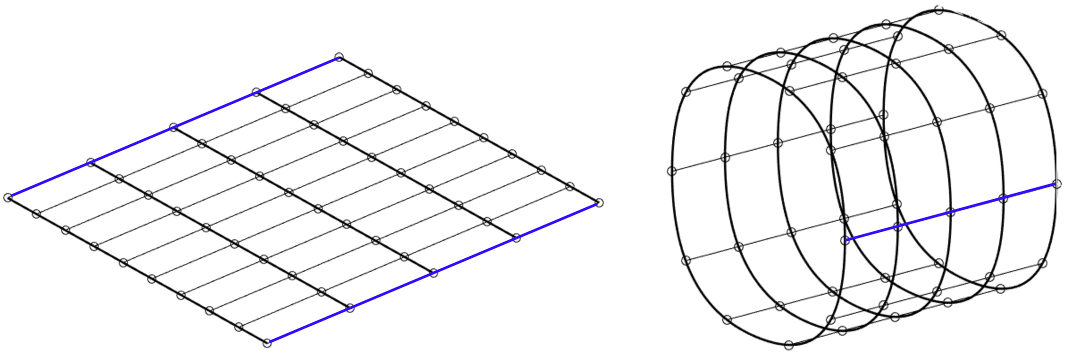

Figure 11 Periodic surface and its parametric space surface, periodic line in blue

Some surfaces can be periodic without been completly closed. An example of such surfaces can be seen in *Figure 12*, where the surface is closed in one side but open in the other one.

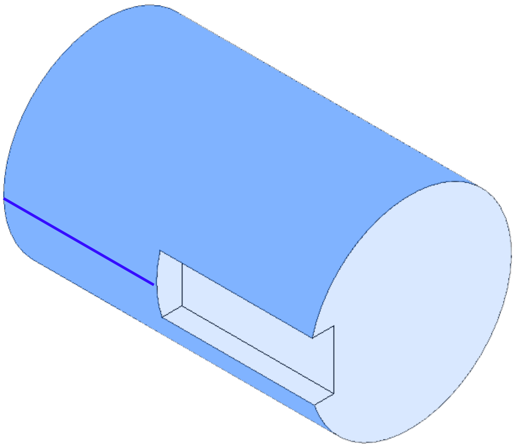

Figure 12 Periodic open surface

Degenerated surfaces

When every point of an edge of the parametric space, yields to only one point in the real space, the surface is degenerated in that edge. In *Figure 13* it can be seen the parametric and real spaces of a cone. It is degenerated in one side, colored in red, and periodic in one direction, blue lines.

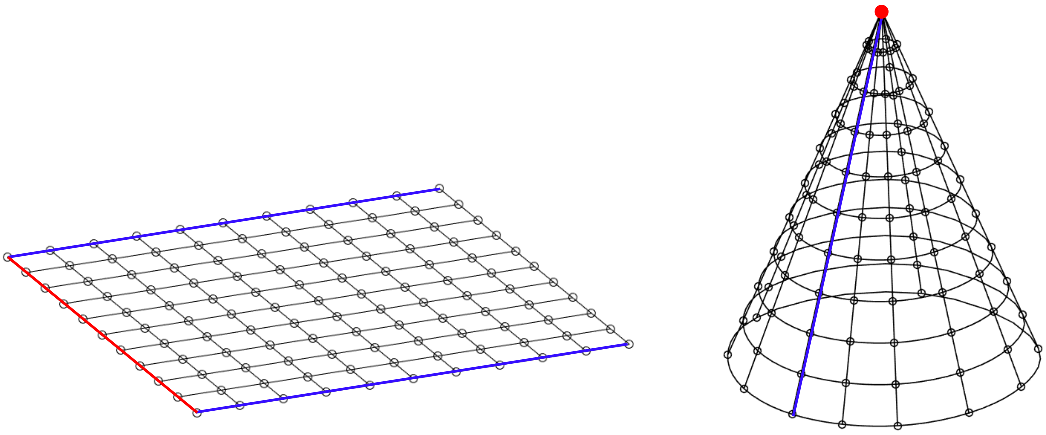

Figure 13 Degenerated and periodic surface and its parametric space surface. Periodic edge in blue, degenerated edge in red

Other examples of degenerated surfaces can be a three edges surface or a sphere.

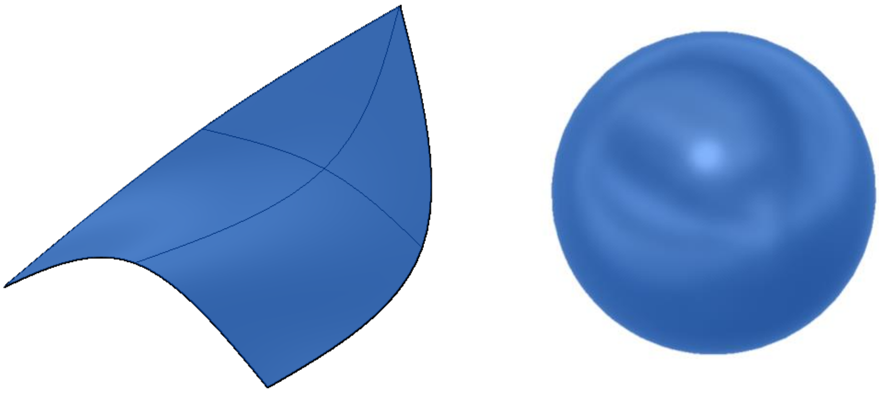

Figure 14 Examples of degenerated surfaces

3. High-order finite element analysis

In this section a general high-order finite element 2D heat flow analysis will be described to assess the implications of using high-order elements. The explanation will be conducted using a quadratic Lagrangian element. Afterwards, the Fekete nodal distribution for optimal interpolation, will be introduced.

3.1. High-order 2D heat flow

The heat conduction law describes the transfer of thermal energy within a body. In two dimensions it is expressed as

$$\frac{\partial q_x(x,y)}{\partial x} + \frac{\partial q_y(x,y)}{\partial y} - Q(x,y) = 0 \qquad (16)$$

where $Q(x,y)$ is the heat source and q_x , q_y are the heat fluxes in x and y directions. These heat fluxes relate the gradient of temperature while moving along the cartesian directions, and the properties of the material, yielding to the following constitutive law, the Fourier's law,

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = -\begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \qquad (17)$$

where k_{xx} and k_{yy} coefficients are the conductivity of the material with respect to x and y direction respectively. An isotropic material is considered, meaning that the conductivity is the same in every direction. Putting together the equilibrium (16) and constitutive (17) equations, the differential equation of heat conduction problems in strong form appears

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + Q(x,y) = 0 \qquad (18)$$

Boundary conditions

In the heat flow problem, it is mandatory that every boundary has an imposed condition either Dirichlet, Neumann or Cauchy. The boundary of the domain is referred as Γ and is divided in three types, depending on the condition set in each point.

Dirichlet: Prescribed temperature in the boundary Γ_{T}.

$$T = \overline{T}(x,y) \qquad for\ x,y\ \epsilon\ \Gamma_{\mathrm{T}} \qquad (19)$$

Neumann: Prescribed normal boundary flux Γ_{q}, is the heat entering or scaping the body.

$$q_n = \overline{q_n}(x,y) \qquad for\ x,y\ \epsilon\ \Gamma_q \qquad (20)$$

Cauchy: Convective boundary conditions Γ_{c}, relate the convective heat transfer coefficient h and the difference of temperatures between the boundary T and the ambient temperature $T\infty$.

$$q_n = h(T - T\infty) \qquad for\; x, y\, \epsilon\, \Gamma_c \tag{21}$$

Weak formulation

The conditions impose in the strong formulation (18) require a solution that satisfy equilibrium in every point of the domain. To ease that strong condition, the differential equation is multiplied by a weighting function Ø and then integrated resulting in the expression

$$\int_{\Omega} \emptyset \left[\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + Q\right] d\Omega \; = 0 \tag{22}$$

that allows to calculate the solution as an average on the entire domain, instead of satisfying the equilibrium at every single point.

The next step is to go from second order to first order derivatives and to extract the boundary terms. Integration by parts is made making use of the divergence theorem.

After integrating and rearranging terms, the following expression appears

$$\int_{\Omega} \left(\frac{\partial \emptyset}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial \emptyset}{\partial y} k \frac{\partial T}{\partial y}\right) d\Omega + \int_{\Gamma_T} \emptyset q_n d\Gamma + \int_{\Gamma_q} \emptyset q_n d\Gamma + \int_{\Gamma_c} \emptyset q_n d\Gamma = \int_{\Omega} \emptyset Q d\Omega \tag{23}$$

where the heat fluxes q_n terms depend on the boundary conditions.

The first boundary term Γ_T , refers to the Dirichlet condition. The weighting function in this part of the boundary is set to 0 and will be calculated later as a reaction flux.

The second boundary term Γ_q, corresponds to the Neumann condition. The heat flux is the already known boundary flux $q_n = q_0$.

The last boundary term Γ_c is the Cauchy condition, $q_n = h(T - T\infty)$. It is important to notice here that the temperature T is unknown at the Cauchy boundary.

After these steps, an easier expression for the original steady state 2D heat flow problem, the weak formulation, appears.

$$\int_{\Omega} \left(\frac{\partial \emptyset}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial \emptyset}{\partial y} k \frac{\partial T}{\partial y}\right) d\Omega + \int_{\Gamma_q} \emptyset q_o d\Gamma + \int_{\Gamma_c} \emptyset h(T - T\infty) d\Gamma \; = \int_{\Omega} \emptyset Q d\Omega \tag{24}$$

Finite element discretization

Once the weak form is obtained, the Galerkin method is used to go from a continuous to a discrete problem, dividing the domain in triangular quadratic elements which will be mapped to the reference element.

The reference element, the basis functions and its derivatives are presented below.

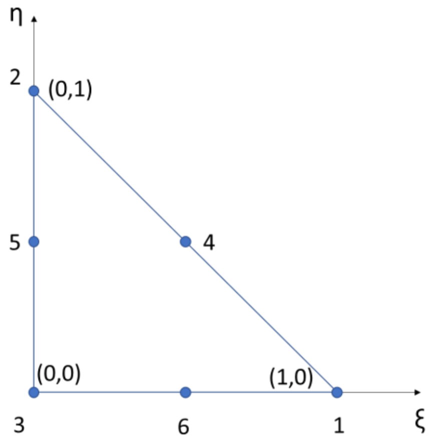

Figure 15 Quadratic reference element

$$N_1 = 2\xi^2 - \xi \tag{25}$$

$$N_2 = 2\eta^2 - \eta \tag{26}$$

$$N_3 = 2(\xi + \eta)^2 - 3(\xi + \eta) + 1 \tag{27}$$

$$N_4 = 4\xi\eta \tag{28}$$

$$N_5 = -4\eta^2 - 4\xi\eta + 4\eta \tag{29}$$

$$N_6 = -4\xi^2 - 4\xi\eta + 4\xi \tag{30}$$

$$\frac{\partial N_1}{\partial \xi} = 4\xi - 1 \tag{31}$$

$$\frac{\partial N_1}{\partial \xi} = 0 \tag{32}$$

$$\frac{\partial N_2}{\partial \xi} = 0 \tag{33}$$

$$\frac{\partial N_2}{\partial \xi} = 4\eta - 1 \tag{34}$$

$$\frac{\partial N_3}{\partial \xi} = 4\xi + 4\eta - 3 \tag{35}$$

$$\frac{\partial N_3}{\partial \xi} = 4\xi + 4\eta - 3 \tag{36}$$

$$\frac{\partial N_4}{\partial \xi} = 4\eta \tag{37}$$

$$\frac{\partial N_4}{\partial \xi} = 4\xi \tag{38}$$

$$\frac{\partial N_5}{\partial \xi} = -4\eta \tag{39}$$

$$\frac{\partial N_5}{\partial \xi} = -4\xi - 8\eta + 4 \tag{40}$$

$$\frac{\partial N_6}{\partial \xi} = -8\xi - 4\eta + 4 \tag{41}$$

$$\frac{\partial N_6}{\partial \xi} = -4\xi \tag{42}$$

Isoparametric mapping is used, meaning that the same shape functions are used to interpolate nodal temperatures and coordinates. To do so, the values in every node of the element are (i) multiplied by their corresponding basis function, evaluated in the reference coordinates, and (ii) added together. For the sake of simplicity, it can be expressed in vector notation as

$$T^{(e)}(\xi,\eta) = T_1^{(e)} N_1(\xi,\eta) + \cdots + T_6^{(e)} N_6(\xi,\eta) = \boldsymbol{N}\boldsymbol{T}^{(e)} \tag{43}$$

$$\boldsymbol{x}^{(e)}(\xi,\eta) = \boldsymbol{x}_1^{(e)} N_1(\xi,\eta) + \cdots + \boldsymbol{x}_6^{(e)} N_6(\xi,\eta) = \boldsymbol{N}\boldsymbol{x}^{(e)} \tag{44}$$

where $\boldsymbol{N}$ is a vector containing the basis functions evaluated at the point of interest, $\boldsymbol{T}^{(\boldsymbol{e})}$ contains the temperatures at every node of an element and $\boldsymbol{x}^{(\boldsymbol{e})}$is formed by the real space coordinates (x, y), of every node of a given element.

Using the isoparametric transformation, the weak form (24) can be expressed in terms of the reference element.

First, the derivatives of the temperature term in respect to the reference coordinates are computed as

$$\frac{\partial T^{(e)}}{\partial \xi} = T_1^{(e)}\frac{\partial N_1}{\partial \xi} + \cdots + T_6^{(e)}\frac{\partial N_6}{\partial \xi} =; \qquad \frac{\partial T^{(e)}}{\partial \eta} = T_1^{(e)}\frac{\partial N_1}{\partial \eta} + \cdots + T_6^{(e)}\frac{\partial N_6}{\partial \eta} \tag{45}$$

what can be expressed in a more compact way as follows

$$\frac{\partial \boldsymbol{T}^{(e)}}{\partial \boldsymbol{\xi}} = \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}}\boldsymbol{T}^{(e)} \tag{46}$$

where the terms of $\frac{\partial \boldsymbol{N}}{\partial \xi}$ are as presented below.

$$\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} 4\xi - 1 & 0 & 4\xi + 4\eta - 3 & 4\eta & -4\eta & -8\xi - 4\eta + 4 \\ 0 & 4\eta - 1 & 4\xi + 4\eta - 3 & 4\xi & -4\xi - 8\eta + 4 & -4\xi \end{bmatrix} \tag{47}$$

The derivative of the temperature must be written on respect to the global coordinates. By using the chain rule, the expression can be transformed from the local to the global coordinates

$$\begin{bmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \tag{48}$$

where $\boldsymbol{J}$ is the Jacobian, the matrix that allows the transformation from the reference space to the global space. The terms are computed using the derivatives of the basis functions, similarly to equation (45). Having a look at the expressions,

$$\frac{\partial x^{(e)}}{\partial \xi} = 4\left(x_{16}^{(e)} + x_{36}^{(e)}\right)\xi + 4\left(x_{45}^{(e)} + x_{36}^{(e)}\right)\eta - \left(x_{16}^{(e)} + 3x_{36}^{(e)}\right) \tag{49}$$

$$\frac{\partial x^{(e)}}{\partial \eta} = 4\left(x_{35}^{(e)} + x_{46}^{(e)}\right)\xi + 4\left(x_{25}^{(e)} + x_{35}^{(e)}\right)\eta - \left(x_{25}^{(e)} + 3x_{35}^{(e)}\right) \tag{50}$$

$$\frac{\partial y^{(e)}}{\partial \xi} = 4\left(y_{16}^{(e)} + y_{36}^{(e)}\right)\xi + 4\left(y_{45}^{(e)} + y_{36}^{(e)}\right)\eta - \left(y_{16}^{(e)} + 3y_{36}^{(e)}\right) \tag{51}$$

$$\frac{\partial y^{(e)}}{\partial \eta} = 4\left(y_{35}^{(e)} + y_{46}^{(e)}\right)\xi + 4\left(y_{25}^{(e)} + y_{35}^{(e)}\right)\eta - \left(y_{25}^{(e)} + 3y_{35}^{(e)}\right) \tag{52}$$

it is important to mention that, for straight edges elements with equally space nodal distribution, the Jacobian is reduced to a constant. In fact, the result is the same as the one for a linear element. A constant Jacobian term implies no deformation along the elements yielding to a good mesh quality. A proper representation of the domain boundaries might imply the use of curved elements, resulting in not constant Jacobians. This fact reduces the quality of the element. A nodal distribution that minimize the deformation of the elements will result in higher quality meshes.

Using the isoparametric transformation the determinant of the Jacobian can be computed as

$$A^{(e)} = \int_{\Omega^{(e)}} 1 d\Omega = \int_{\Omega^{\mathrm{ref}}} 1\left|\boldsymbol{J}^{(e)}\right| d\Omega = \left|\boldsymbol{J}^{(e)}\right| A^{ref}$$

$$\left|\boldsymbol{J}^{(e)}\right| = 2A^{(e)} \tag{53}$$

resulting in two times the area of the element.

Putting together equations (47) and (48), the derivatives of the temperature in respect to the global coordinates are

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \boldsymbol{J}^{(e)^{-1}} \begin{bmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \end{bmatrix} = \boldsymbol{J}^{(e)^{-1}} \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \boldsymbol{T}^{(e)} = \boldsymbol{B}^{(e)} \boldsymbol{T}^{(e)} \tag{54}$$

where $\boldsymbol{B}^{(e)}$ can be expressed on terms of the element area using equation (53).

$$\boldsymbol{B}^{(e)} = \frac{\mathbf{1}}{\left|\boldsymbol{J}^{(e)}\right|} (\boldsymbol{J}^{*})^{T} \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} = \frac{1}{2A^{(e)}} (\boldsymbol{J}^{*})^{T} \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \tag{55}$$

Each term of the weak form is now expressed for every element in terms of the reference element as follows.

$$\int_{\Omega_{\mathrm{e}}} \left(\frac{\partial \emptyset}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial \emptyset}{\partial y} k \frac{\partial T}{\partial y} \right) d\Omega = \emptyset^{T} \left[k^{(e)} \int_{\Omega_{\mathrm{ref}}} \boldsymbol{B}^{(e)^{T}} \boldsymbol{B}^{(e)} \left|\boldsymbol{J}^{(e)}\right| d\Omega \right] \boldsymbol{T}^{(e)} = \emptyset^{\boldsymbol{T}} \boldsymbol{K}^{(e)} \boldsymbol{T}^{(e)} \tag{56}$$

$$\boldsymbol{K}^{(e)} = k^{(e)} \int_{0}^{1} \left(\int_{0}^{1-\eta} \boldsymbol{B}^{(e)^{T}} \boldsymbol{B}^{(e)} \left|\boldsymbol{J}^{(e)}\right| d\xi \right) d\eta \tag{57}$$

$$\int_{\Omega_{\mathrm{e}}} \emptyset Q d\Omega = \emptyset^{\boldsymbol{T}} Q^{(e)} \int_{\Omega_{\mathrm{e}}} \boldsymbol{N}^{\boldsymbol{T}} d\Omega = \emptyset^{\boldsymbol{T}} \boldsymbol{r}_{Q}^{(e)} \tag{58}$$

$$\int_{\Gamma_{\mathrm{q}}^{\mathrm{e}}} \emptyset q_{o} d\Gamma = \emptyset^{\boldsymbol{T}} q_{o}^{(e)} \int_{\Gamma_{\mathrm{q}}^{\mathrm{e}}} \boldsymbol{N}^{T} d\Gamma = \emptyset^{\boldsymbol{T}} \boldsymbol{r}_{q}^{(e)} \tag{59}$$

$$\int_{\Gamma_{\mathrm{c}}^{\mathrm{e}}} \emptyset h (T - T_{\infty}) d\Gamma = \emptyset^{\boldsymbol{T}} \left[h^{(e)} \int_{\Gamma_{\mathrm{q}}^{\mathrm{e}}} \boldsymbol{N}^{T} \boldsymbol{N} \, d\Gamma \right] \boldsymbol{T}^{(e)} - \emptyset^{\boldsymbol{T}} h^{(e)} \boldsymbol{T}_{\infty}^{(e)} = \emptyset^{\boldsymbol{T}} \boldsymbol{h}^{(e)} \boldsymbol{T}^{(e)} - \emptyset^{\boldsymbol{T}} \boldsymbol{r}_{\infty}^{(e)} \tag{60}$$

After assembling the contributions of every element, the global system of finite element equations can be written as

$$(\boldsymbol{K}_T + \boldsymbol{H}_T)\boldsymbol{T} = \boldsymbol{R}_\infty + \boldsymbol{R}_Q + \boldsymbol{R}_q \tag{61}$$

It is important to mention that, numerical integration is needed. Notice that $\boldsymbol{B}^{(\boldsymbol{e})}$ contains the inverse of the Jacobian, whose terms are polynomial expressions. This yields to a non-polynomial expression that can be approximated by using a quadrature, but cannot be exactly computed.

The accuracy of the finite element solver depends on the distribution of the interpolation points and the numerical quadrature. In the next section, the Fekete nodal distribution is introduced to compute appropriate interpolation points.

3.2. High-order elements with Fekete points nodal distribution

As it has been already stated, the position of the nodes within the reference element will affect the final solution. There are different strategies available to distribute the nodes.

The accuracy of different distributions can be measured using the Lebesgue constant as explained in [13]. The smaller the Lebesgue constant, the better.

For low order functions, the use of equally spaced nodal distribution gives low Lebesgue constant values but, as the degree increases, the Lebesgue constant grows exponentially with it.

The Fekete points are closer to the optimal interpolation points yielding to a smaller Lebesgue constant. In [13], Fekete points have been proved to maintain a better performance than other methods while increasing the order of the element, and an algorithm has been developed to compute the points in a triangle.

The Fekete points for a triangular face in the reference element triangle for different degrees are shown *Figure 16*.

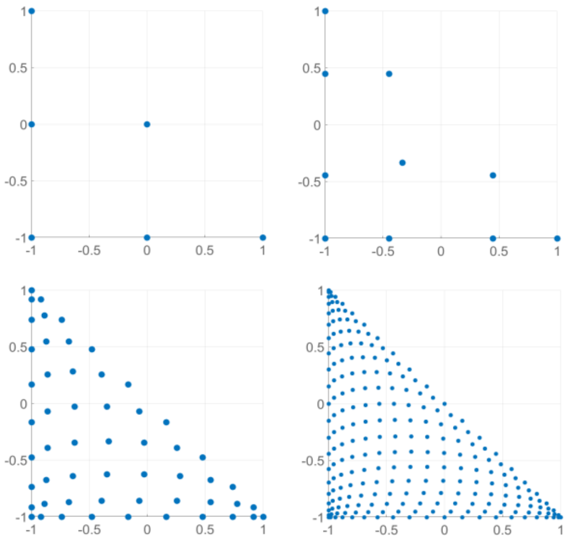

Figure 16 Fekete nodal distribution in the reference element for p=2, p=3, p=9 and p=20

4. High-order mesh generation with periodic and degenerated surfaces

In this essay the incremental linear elastic method proposed in [12] has been used. A brief summary of the method can be read bellow.

First an initial linear mesh is constructed making use of the well-developed linear mesh generators available.

With the aim of creating the high-order mesh, internal nodes laying on the low order elements are added. These nodes are located following the Fekete nodal distribution explained in *section 3.2*

To create the high-order curved mesh, these newly created nodes have to be moved to their corresponding location in the real domain, which is given by the geometrical model described using NURBS.

Finding these points in the NURBS surface is not trivial. For doing so, the point projection algorithm, presented in *section 2.3*, is used.

Once both the nodes in the mesh and the corresponding points on the surface are known, the distance between them is computed. The distance is imposed as a Dirichlet boundary condition and the linear elastic problem governed by equation

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0} \tag{62}$$

is solved. The solution is a displacement field that when applied to the original mesh will result in a high-order curved mesh. It is important to mention that at this point, the NURBS definition of the boundary is no longer used.

A modified scaled Jacobian is used to measure the quality of the resulting elements. As previously mentioned in *section 3.1*, the fact that curved elements are used yields to a non-constant Jacobian which means that the element is deformed. The location of the nodes will impact the amount of deformation along the element, which can be measured using

$$I = \frac{\boldsymbol{min}_{\xi \epsilon R} |\boldsymbol{J}(\xi)|}{\boldsymbol{max}_{\xi \epsilon R} |\boldsymbol{J}(\xi)|} \tag{63}$$

where a negative value of I indicates that the element is not valid. The closer I is to 1 the better the quality of the element is. When looking at the expression of the Jacobian on equation (53), it can be noticed that, the scaled Jacobian is nothing else than a quotient of areas (in 2D). The same area is place in different locations of the reference element, then the projection of those areas in the real space can be compared. If the resulting areas in the real space are the same, it has occurred no distortion, the value of I is one, but if the areas are different from each other the value decreases.

Low quality elements are mainly related to an inappropriate distribution of the nodes in the curved elements and to large deformations of the elements.

High-order nodes are placed in the linear element following the Fekete nodal distribution and then projected to the curved boundary. If the curvature of the boundary inside an element is too high, the projection gives back a bad nodal distribution. When a polynomial going through these nodes is constructed, the boundary is not well represented.

Take a look at *Figure 17* where this has been illustrated using an equally spaced nodal distribution. The projection of the high-order nodes to the curved boundary is represented in black and the polynomial going through the nodes is represented in blue. In the lower side of the picture there is a straight line representing the location of the nodes along a normalize line. Notice that the distribution of the nodes is not even, the segments on the sides are smaller than the middle one. This means a large deformation of the element yielding to a low I value.

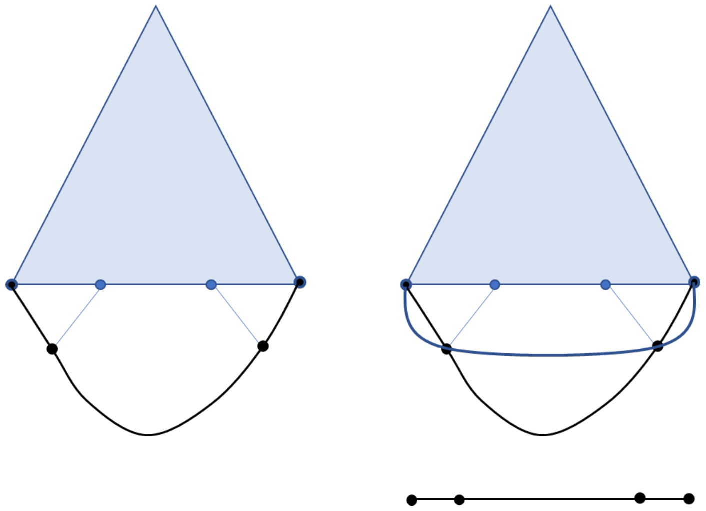

Figure 17 Curved element and the domain boundary for p=2

A better representation of the boundary can be achieved by increasing the order of the mesh. Another option is to relocate the nodes, finding a better distribution along the boundary. The two approaches are represented in *Figure 18.*

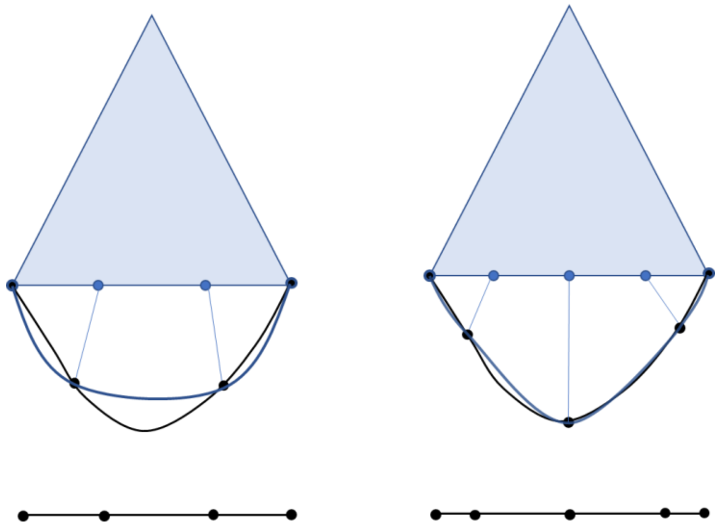

Figure 18 Curved element and the domain boundary for p=2 with corrected high-order nodes in the left, and p=3 with non-corrected nodal distribution

On the right side of *Figure 18*, where the mesh degree has been changed from $p = 2$ to $p = 3$, the representation of the boundary is more accurate. Nevertheless, looking at the normalize line, the segments size is still uneven. If the scaled Jacobian is computed, its value will still be low due to the high deformation suffered by the element. On the other hand, the figure on the left, where the nodes have been relocated, presents a better nodal distribution and will have a higher I value.

Increasing the order of the mesh will provide a better representation of the domain, but not necessarily a higher I value. The main factor affecting the quality of the elements is the nodal distribution, which at the same time can affect on the accuracy of the boundary representation. Some work about a proper nodal distribution has been made in [12] using geodesics. Other possible approach is the use of optimisation methods on those elements with not enough quality.

In some cases, where the curvature within the elements is too high, the best approach is to reduce the element size of the initial mesh on those elements.

Large deformations related errors can be corrected using a multistep solving process. Divide the total deformation into several steps, can alleviate the errors introduced by the usage of a linear elastic model that does not account for large displacements.

4.1. Periodic surfaces

As explained in *section 2.4*, periodic surfaces can go from cylindrical like surfaces to semi-periodic ones such as the one in *Figure 19.*

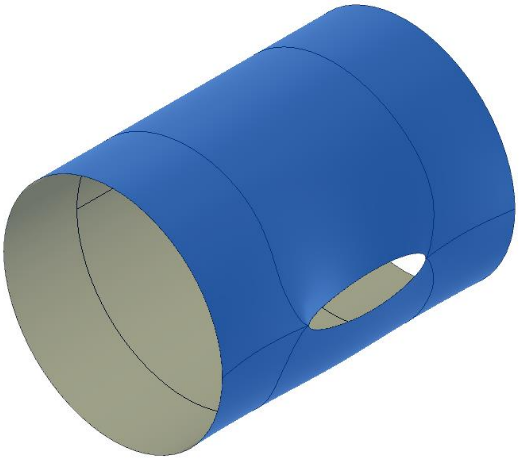

Figure 19 Periodic surface

The implementation of the method explained in [12] needs a definition of periodicity that englobe all these surfaces. A surface is considered periodic if it is formed by at least one closed curve, in other words, if two points in opposite edges of the surface's parametric space correspond with the same point in the real space, the surface is periodic.

Checking the surface periodicity requires selecting several points on the parametric space edges and computing their coordinates in the real space. If the real coordinates of more than one point are the same, the surface is treated as a periodic surface.

4.2. Degenerated surfaces

Degenerated surfaces are implemented for the first time and are the main development of this work.

A definition must be given for such surfaces. A surface is considered degenerated if any of their edges in the parametric space is projected to a single point in the real space.

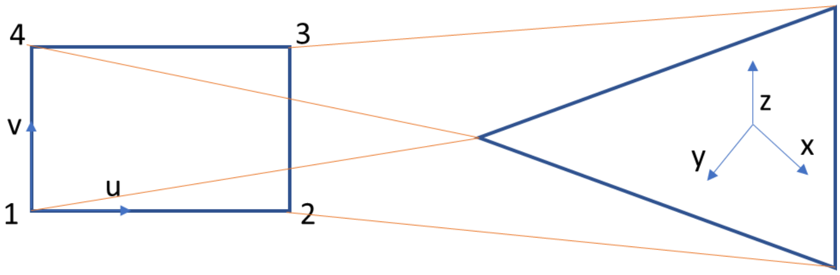

Figure 20 Parametric and real space of a degenerated surface

Some CAD software provide an inappropriate parametric space when dealing with degenerated surfaces. If the given parametric space does not have rectangular shape it has to me modified to assure that no edge crosses the parametric space. *Figure 21* shows the parametric space reconstructed, from the information given by a CAD software, before and after fixing it.

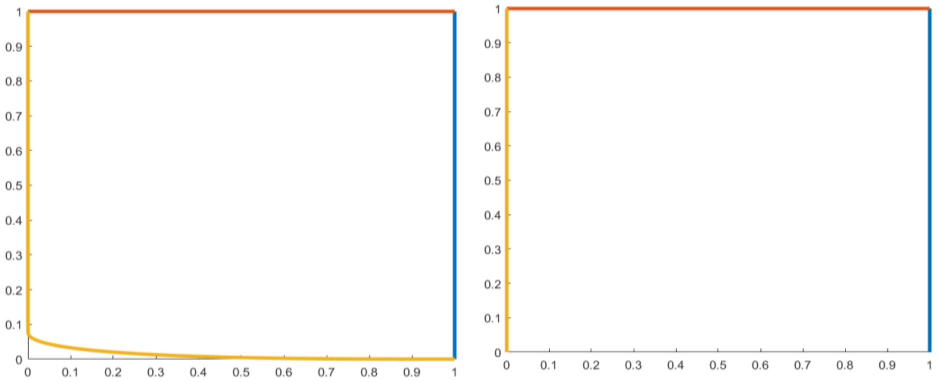

Figure 21 . Cone parametric space before and after correction

The final step on the categorization of the degenerated surface, is to identify the faces of the elements laying on the degenerated edge. If one of the face's nodes is placed on the degenerated edge, that face is classified as degenerated.

Remeshing the parametric space

The mesh is built in the real space and afterwards projected to the parametric space of the NURBS. As a result, there is only one node in the degenerated edge. This node is shared by every element in that side, what leads to a parametric space that is not completely mesh.

.

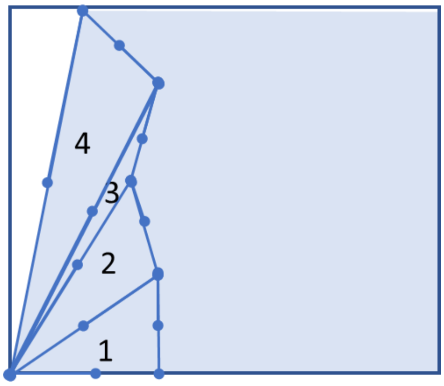

Figure 22 Mesh projection into the parametric space

A new type of element is created to be able to mesh the entire parametric domain. The node in the degenerated edge can be unfold in several nodes laying all of them in the degenerated edge. The projection of all these nodes in the real space, will have the same coordinates, maintaining the same DOF in the real space.

This new element will be named *collapsed quadrilateral*. It is a four-sided element, containing $p + 1$ nodes on one side, 1 node on the opposite one and p nodes on the rest of the edges. Inner nodes are placed in line with the p nodes sides. In total the *collapsed quadrilateral* element is formed by the same number of nodes as the initial triangular element. *Figure 23* shows the *collapsed quadrilateral* elements for several degrees.

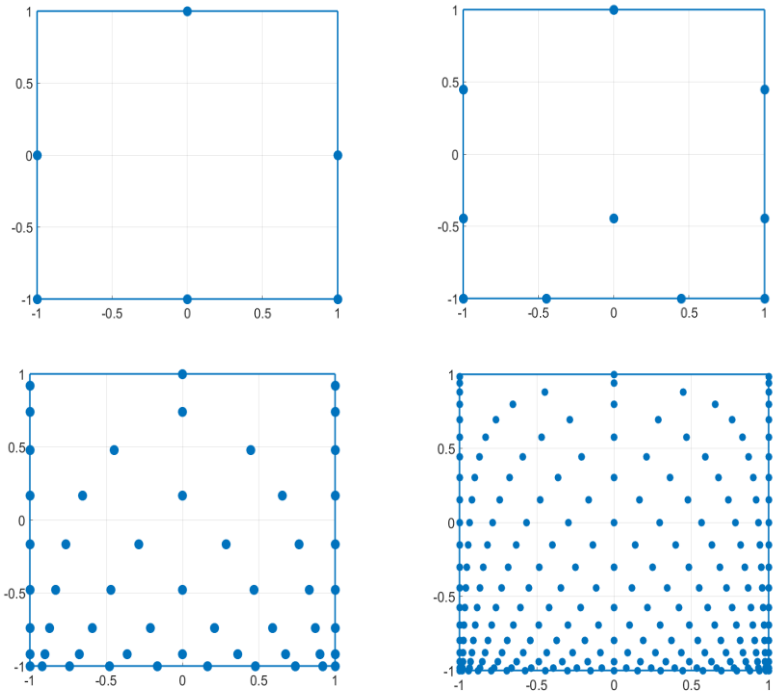

Figure 23 Reference collapse quadrilateral element for p=2, p=3, p=9 and p=20

By changing the triangular elements for the new ones, the mesh on the degenerated edge covers the entire parametric space as can be seen in *Figure 24.*

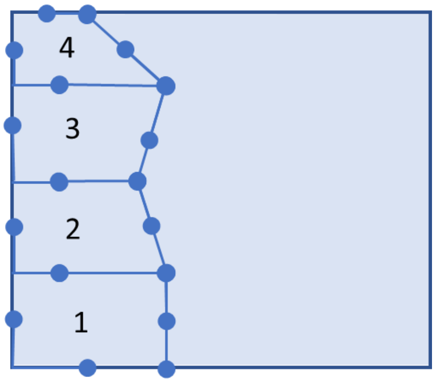

Figure 24 Corrected parametric space using collapse quadrilaterals

Certain surfaces are trimmed in their interior or in the perimeter. In this case an elasticity problem must be solved in the parametric space to relocate the mesh nodes. *Collapsed quadrilateral* are not suitable for FEM calculations so they need to be modified. A hybrid mesh is created changing the collapsed elements by *regular quadrilateral* elements as shown in *Figure 42*.

The location of the nodes in the quadrilateral element has been proved to have an important influence on the final quality of the mesh. The nodal distribution on the *regular quadrilateral* elements, is made by using the blending function method explained in [14].

After solving the elasticity problem in the parametric space, the quadrilateral elements are again changed into the *collapsed quadrilateral* elements. The rest of the meshing process is conducted in the same manner as for other surfaces.

5. Numerical examples

In this chapter the different steps carried out for the generation of the high-order mesh are explained. A series of examples are introduced, gradually increasing the complexity of the geometry and explaining how the mesh generator deals with certain especial cases.

5.1. Deformed Cube

The first geometry is a cube with four deformed faces for which a $p = 3$ triangular mesh is generated. The first input to the process is the geometrical definition of the boundary surfaces, given using NURBS.

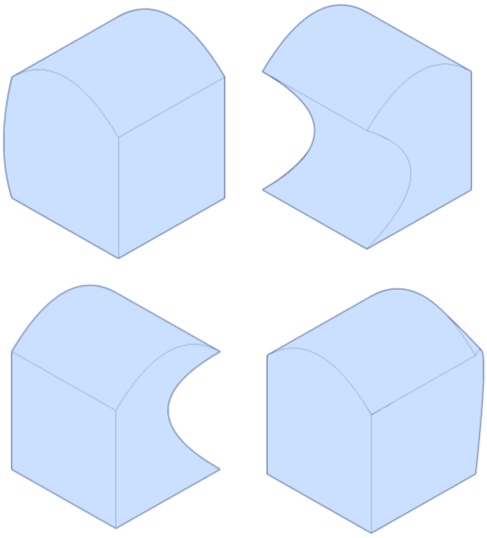

Figure 25 Deformed cube geometry

The triangular linear elements mesh is created thanks to a linear mesh generator.

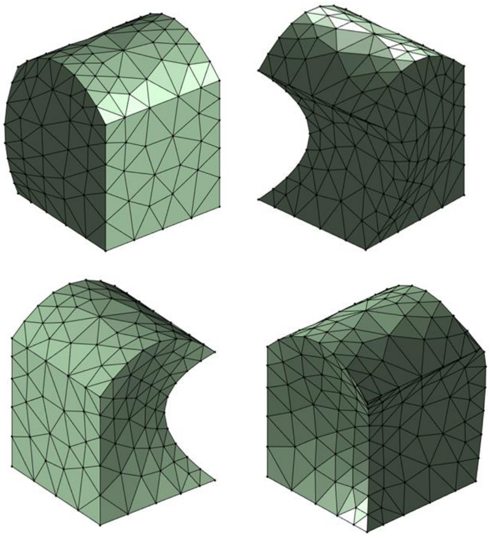

Figure 26 Deformed cube linear mesh

The high-order meshing process requires the creation of a parametric space for every boundary surface of the domain. The top surface can be seen in *Figure 27*, it is formed by 4 curves in the real space that are projected to the parametric space. The area delimited by the four projected lines formed a closed planar surface that corresponds with the projection of the curve in the parametric space.

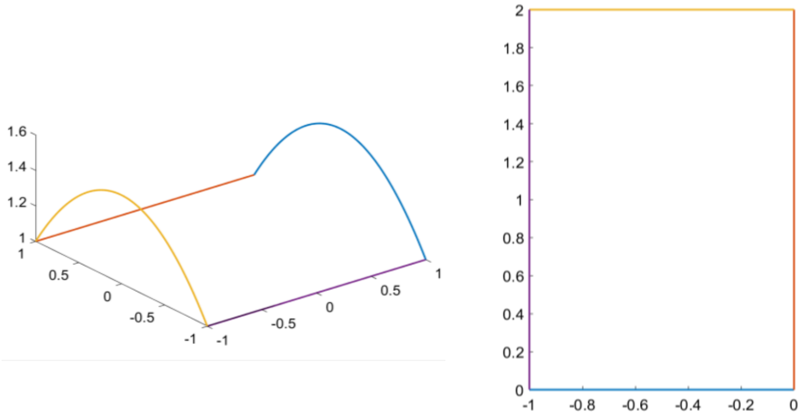

Figure 27 Curves of the upper surface of the cube on the left and the surface parametric space on the right

In a similar way, the linear mesh nodes can be projected in the parametric space yielding to the mesh of the parametric space shown in *Figure 28.*

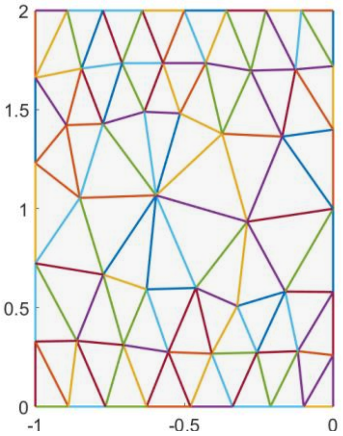

Figure 28 Parametric space mesh

Continuing with the meshing process, the high-order nodes are added on the linear real space elements following the Fekete distribution explained in *section 3.2*. The result can be seen in *Figure 29.*

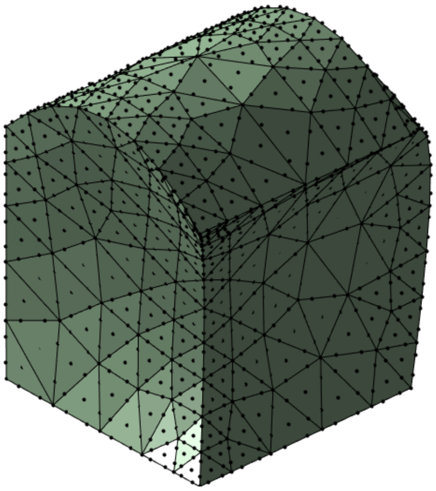

Figure 29 Non-curved high-order mesh

It shall be noticed that, these high-order elements are still straight-sided elements, the newly created nodes have yet to be moved to the boundary of the domain.

The initial nodes, created in the generation of the linear mesh, are laying on the NURBS. That is the reason why they have a known position on the surface parametric space. Nevertheless, the high-order nodes have been added on the linear elements and are not laying in the NURBS.

Using the NURBS point projection algorithm, explained in *section 2.3,* for every high-order node the closest point laying on the NURBS surface is found. The parametric space mentioned before is necessary to complete this step.

The distance between the high-order nodes and their correspondent point in the NURBS surface is measured and set as a Dirichlet boundary condition for the linear elastic solver. Solving the problem results on a deformation field of the high-order nodes, giving the high-order curved mesh.

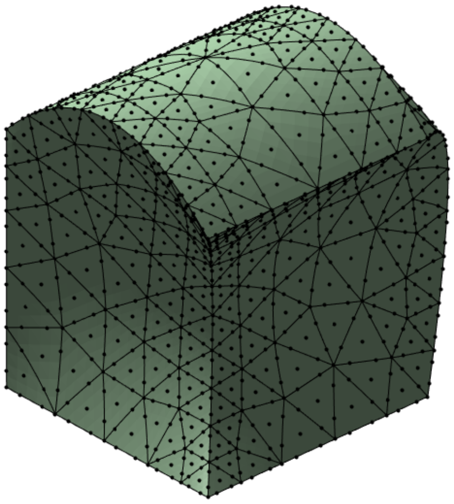

Figure 30 Curved high-order mesh

The resulting mesh accurately represents the boundaries of the domain, but the quality of the elements needs to be checked if the mesh is going to be used for solving a high-order problem. The quality measure used is the scaled Jacobian, explained in *section 4*, which gives the volumetric deformation caused on an element during the high-order mesh creation process.

The maximum value of the scaled Jacobian is 1, meaning not deformation at all, and a value under 0 means a non-valid element. Even if a value of 0 gives a theoretical valid element, the results achieve in the resolution of a problem with such elements will be incorrect. In practice, the value of the scaled Jacobian for most of the elements must be above 0.5 to be considered a proper mesh.

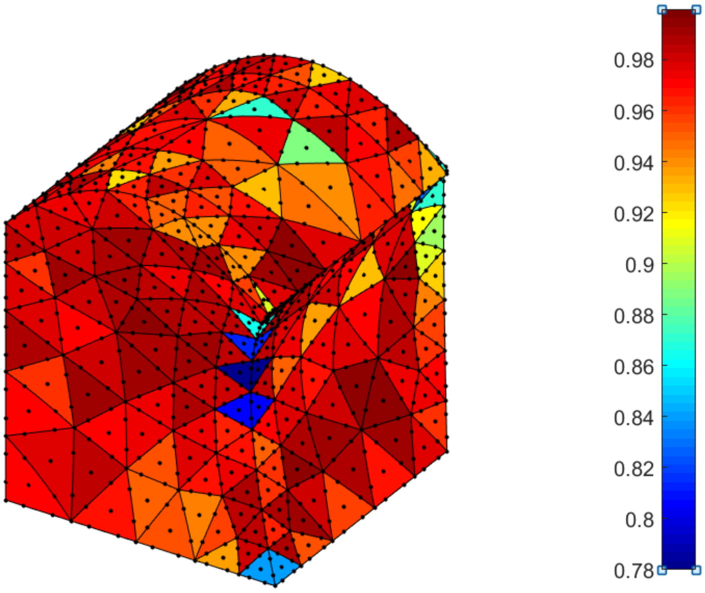

Figure 31 Scaled Jacobian values for the curved high-order mesh

As mention in *section 4*, optimisation methods can be used in the elements with too low I value.

5.2. Cylinder

For this second example a periodic surface is presented. The geometry and linear mesh can be seen in the next figure.

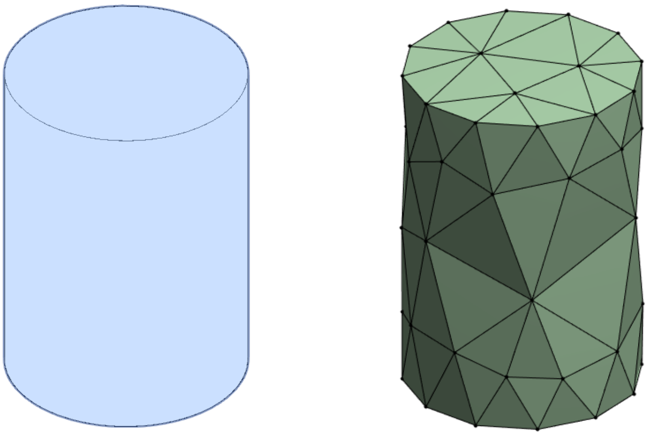

Figure 32 Cylinder geometry and linear mesh

The direct projection of the linear mesh into the parametric space results in an invalid mesh.

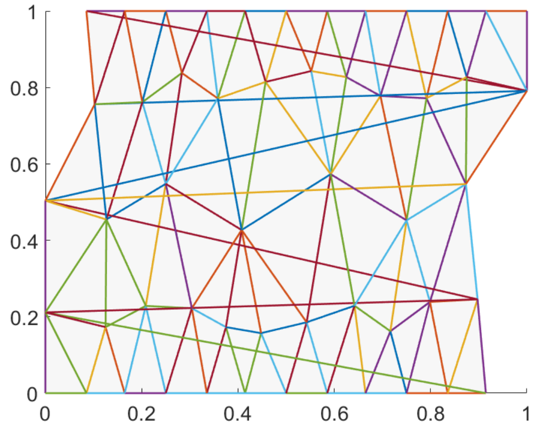

Figure 33 Original mesh of the cylinder on the parametric space

This is produced by the projection of the nodes laying on the periodic edge. A node in that edge corresponds to a single point in the real space, but it has to represent two points in the parametric space. To solve this issue the nodes on the periodic edge are duplicated and placed on both sides of the parametric space.

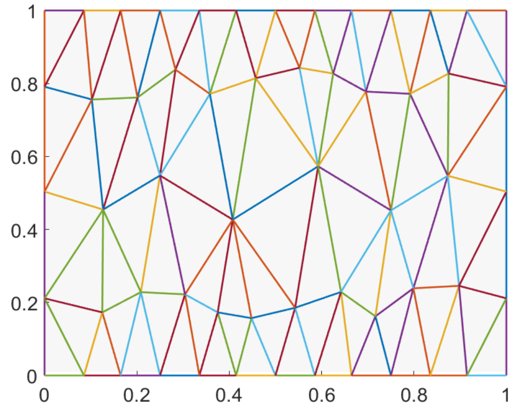

Figure 34 Corrected mesh of the cylinder on the parametric

Once the mesh is corrected, the high-order mesh can be generated following the same procedure as for the deformed cube.

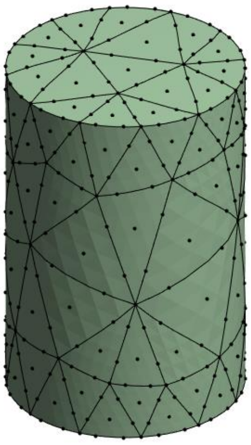

Figure 35 High-order curved mesh of the cylinder

To analyse the behaviour of the mesh quality for different order meshes, six high-order meshes from $p = 2$ to $p = 7$ are generated and compared using the scaled Jacobian.

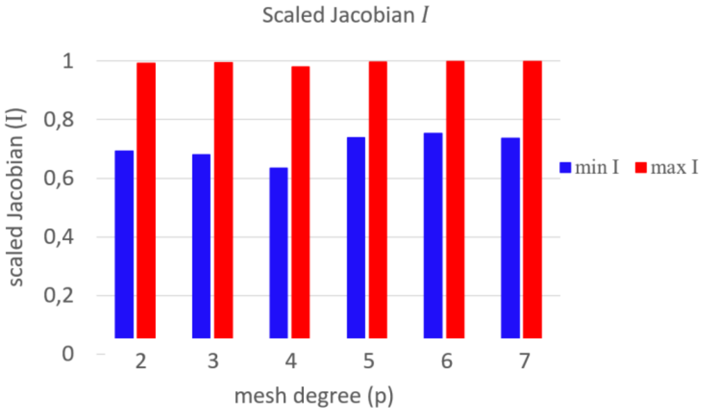

Figure 36 Comparison of the minimum and maximum quality of the elements for different degree high-order meshes

The maximum scaled Jacobian values are similar for every mesh going from 0.98 to 0.99. However, the minimum values fluctuate with the mesh degree. When moving from $p = 2$ to $p = 4$ the minimum value of I decreases. As previously mention in *section 4,* the quality of the elements is not necessarily related to the order of the mesh.

It shall be noticed in *Figure 37,* that the lower quality elements are the same while the degree increases. The elements are those covering larger curvatures. Increasing the quality of these elements, could be done relocating the high-order nodes on the NURBS surface, using optimisation methods.

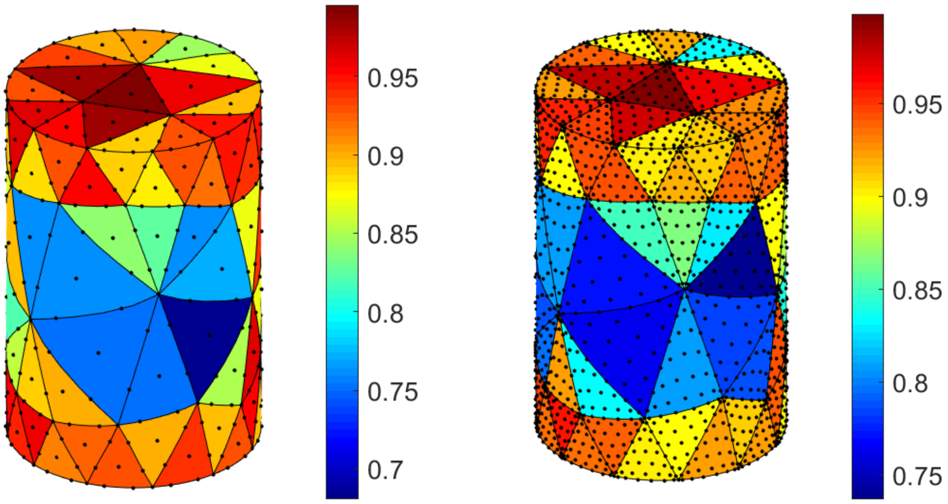

Figure 37 Scaled Jacobian for p=3 mesh on the left and p=7 on the right

5.3. Sphere

The sphere presented here is constructed with only one NURBS surface, it is periodic in one direction and degenerated in the two opposite edges.

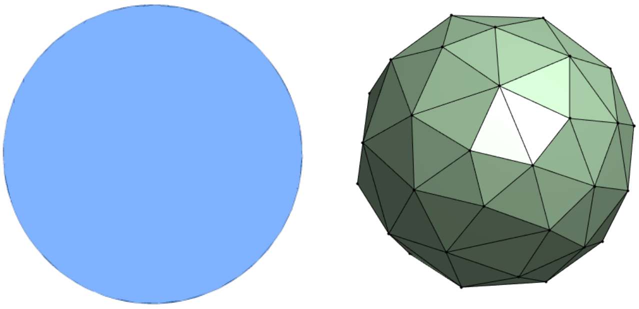

Figure 38 Sphere geometry and linear mesh

The projection of the mesh into the parametric space is affected by the fact that the surface is periodic and degenerated.

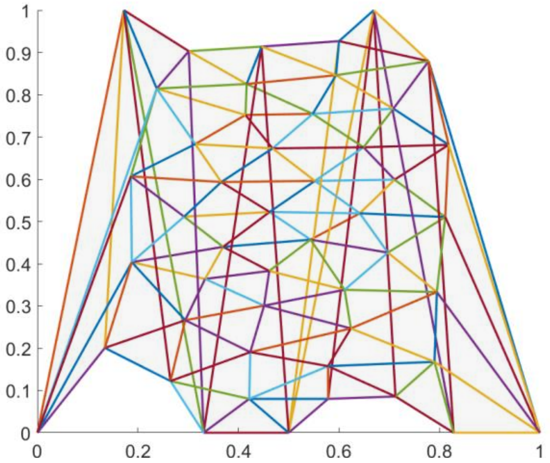

Figure 39 Original mesh of the sphere on the parametric space

First the periodicity is corrected as it was explained in the cylinder example.

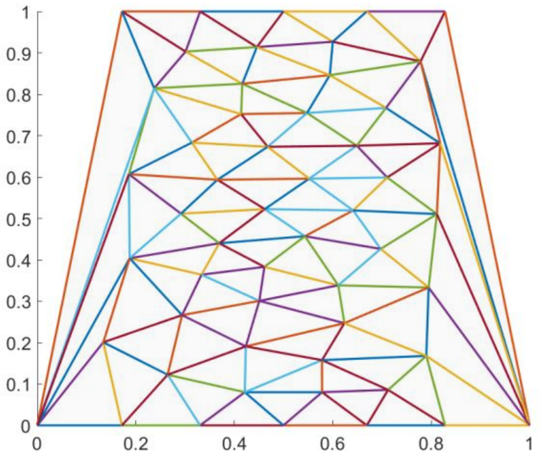

Figure 40 Mesh of the sphere on the parametric space after periodicity correction

Now the elements of the mesh laying on the degenerated edge need to be modified. The process explained in *section 4.2* is implemented. One node is added for each of the degenerated elements and placed in the degenerated edge at a proper location. Using these nodes, the *collapsed quadrilateral* elements are created. *Figure 41* shows how the entire parametric space has been meshed.

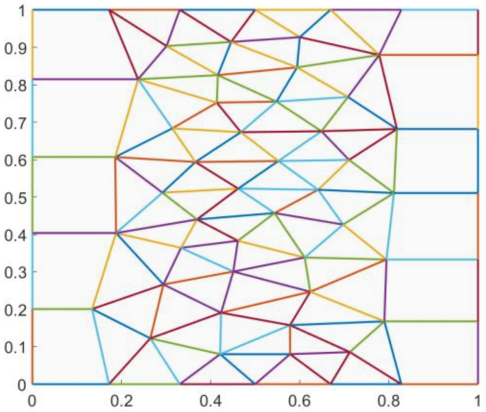

Figure 41 Corrected mesh of the sphere on the parametric space

Remember that the *collapsed quadrilateral* elements, are auxiliary elements with a nodal distribution that cannot be used to solve an elasticity problem in the parametric space. For doing so, the elements must be transformed to regular quadrilateral elements. *Figure 42* shows the distribution of the nodes on the degenerated elements, for a $p = 3$ mesh, using *collapsed quadrilateral* elements and *regular quadrilateral* elements.

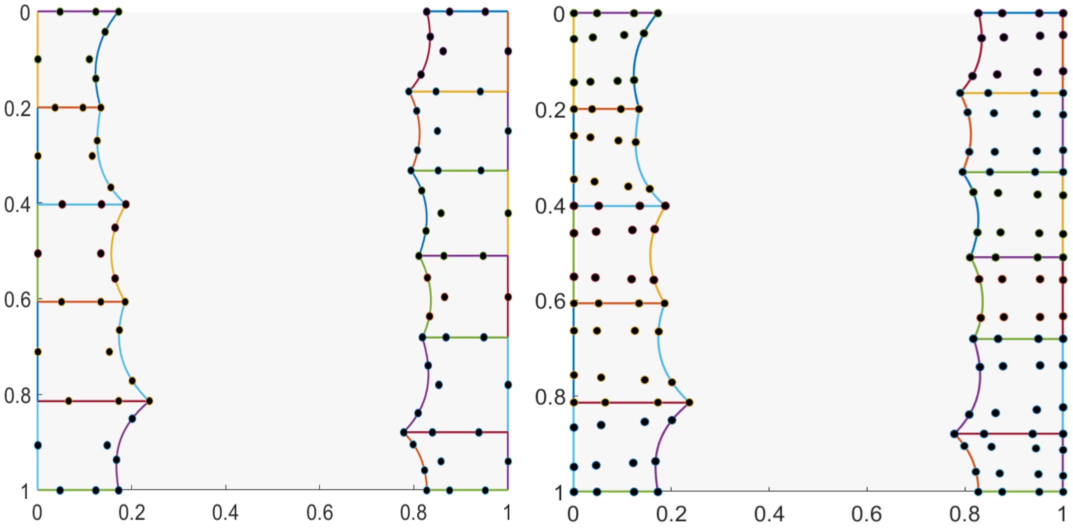

Figure 42 Nodal distribution on the degenerated elements using collapsed quadrilateral elements on the left and using regular quadrilateral elements on the right

After solving the linear elastic problem in the parametric space, which is used to ensure the boundaries of the surface match the domain, the mesh is again transformed into the *collapsed quadrilateral* elements mesh.

Once the parametric space is correctly meshed, the rest of the procedure can be conducted resulting in the following high-order mesh.

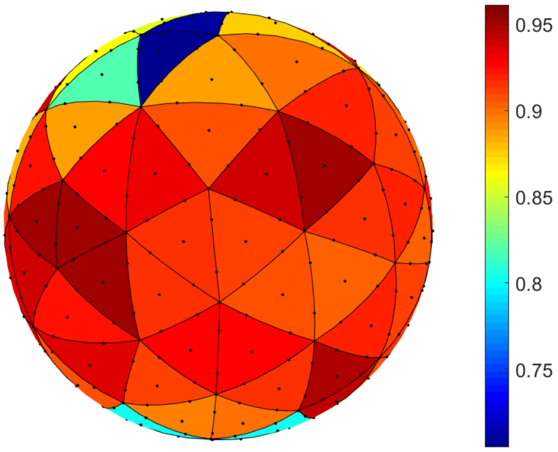

Figure 43 High-order p=3 mesh of the sphere with the values of the scaled Jacobian

The quality of the high-order curved mesh is analyse using the scaled Jacobian. For this example, every element of the high-order mesh has a good quality.

5.4. Sphere intersected by a cylinder

This example is aimed to demonstrate the capacities of the new implementations to deal with trimmed surfaces. The sphere is formed by five surfaces, four surfaces form the shape of the sphere and the fifth one the cylindrical cut.

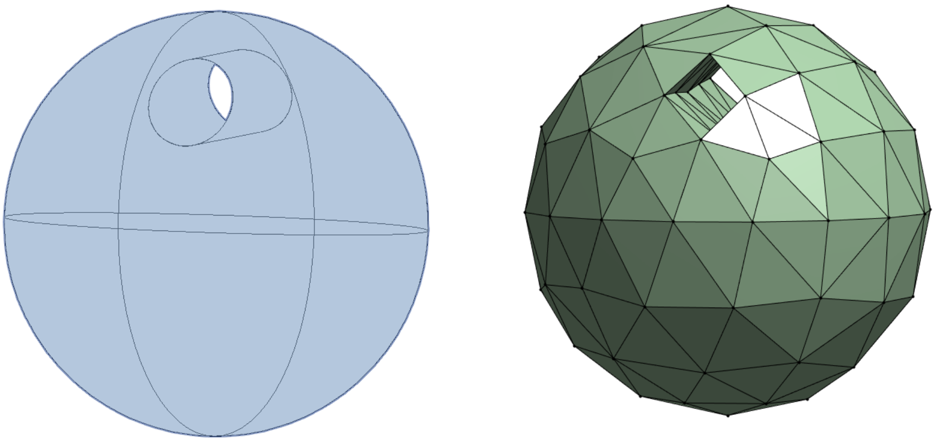

Figure 44 Intersected sphere geometry and linear mesh

Figure 45 shows the corrected parametric space for one of the sphere surfaces intersect by the cylinder.

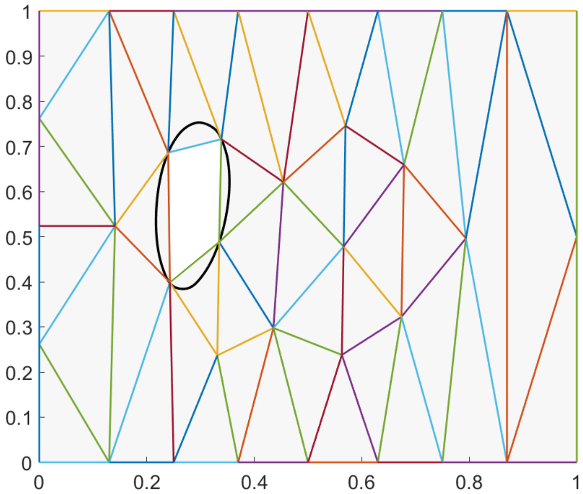

Figure 45 Parametric space of one of the sphere intersected surfaces after been corrected

The black line is the curve produced by the intersection of the sphere surface and the cylinder surface. The quadrilateral shape, shaded in white, is not an element but the hole in the mesh created by the intersection.

A linear elastic problem has to be solved in the parametric space. First, the high-order nodes, on the parametric space, are projected to the hole boundary curve. The distance between the nodes and their projections are set as Dirichlet boundary conditions and the problem is solved.

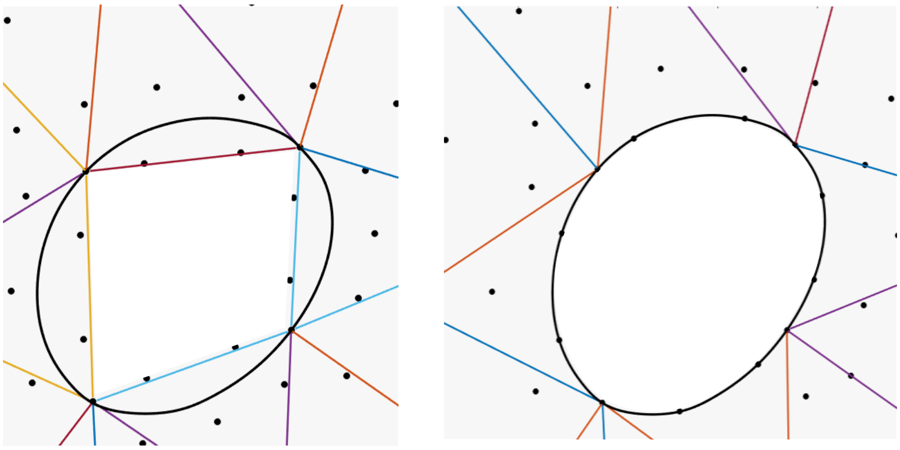

Figure 46 Parametric space mesh of one of the sphere surfaces, before and after applying the deformation to match the boundary, produced by the intersection with the cylinder

Figure 46 shows how the high-order nodes surrounding the hole are moved to meet the boundary curve.

Have a look now at the cylinder parametric surface in *Figure 47.* In this case the surface is cut at both ends, where the cylinder meets the sphere. This time the high-order nodes to be moved are the ones on the boundaries of the parametric space.

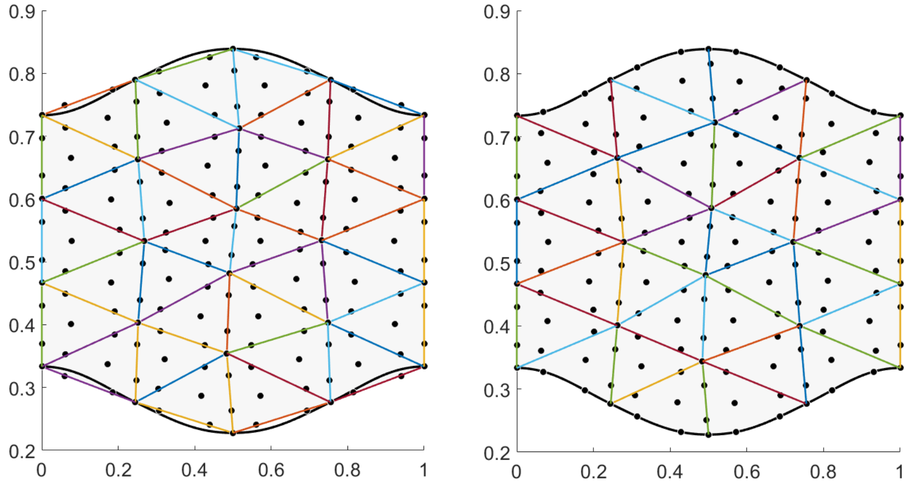

Figure 47 Parametric space mesh of the cylindrical hole before and after applying the deformation to match the boundary

The high-order mesh of the real space for this example can be seen in *Figure 48.*

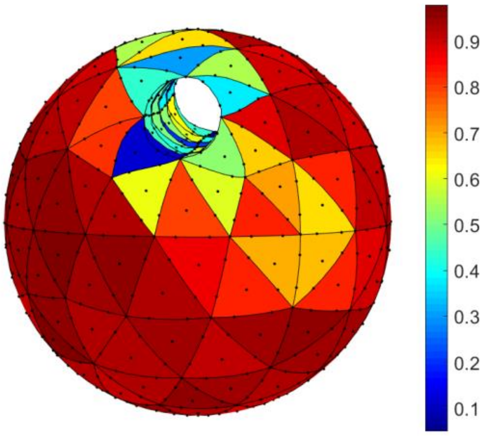

Figure 48 High-order p=3 mesh of the holed sphere with the values of the scaled Jacobian

Some of the elements are too large. It is decided to reduce the size of the elements using a different linear mesh. The result is shown in *Figure 49.*

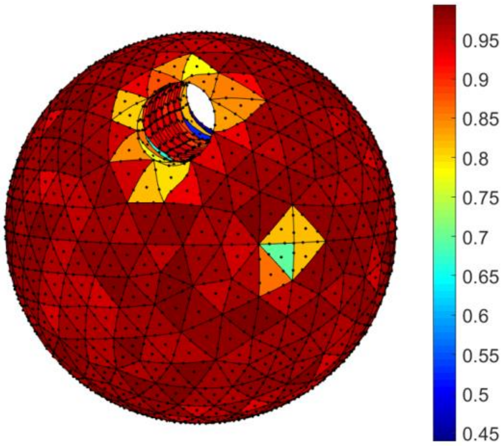

Figure 49 High-order p=3 refined mesh of the holed sphere with the values of the scaled Jacobian

The process of trimming the surface has led to large deformations at some of the elements. Imposing a multistep solving procedure reduces the influence of large deformations on the mesh.

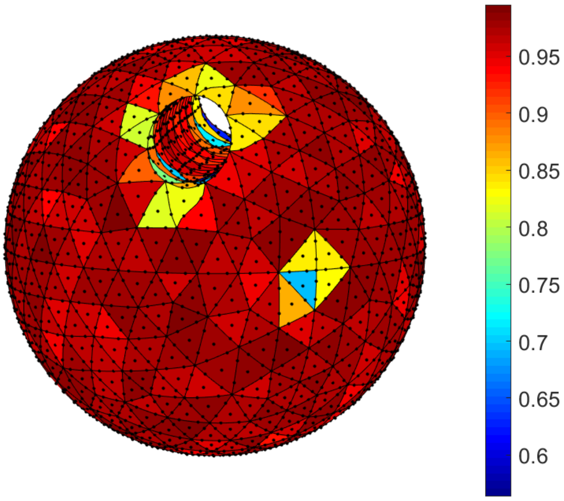

Figure 50 High-order p=3 refined mesh of the holed sphere with the values of the scaled Jacobian, solved using 3 deformation steps

The problem is solved by dividing the total deformation in 3 steps, the result can be seen in *Figure 50*. In this case, large deformations are not caused by large curvatures, that is the reason why the multistep procedure significantly increases the quality of the mesh.

5.5. Cone

The last example exposes one limitation of this method. The geometry to mesh is a cone. Notice that a cone is periodic in one direction and degenerated in one edge.

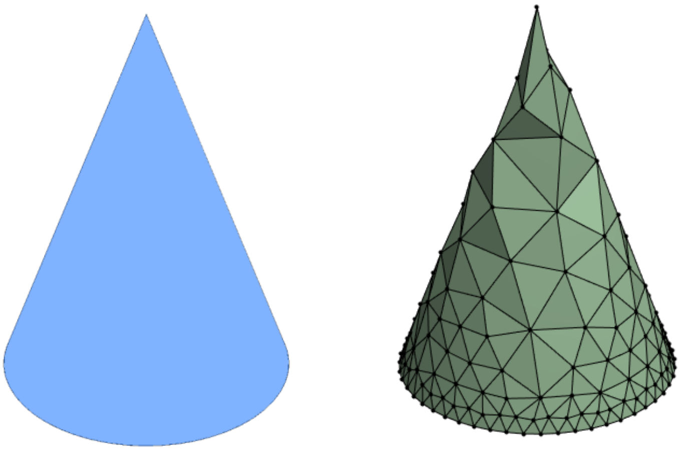

Figure 51 Cone geometry and linear mesh

As previously done, the mesh on the parametric mesh is corrected beginning with the periodic elements following with the degenerated elements.

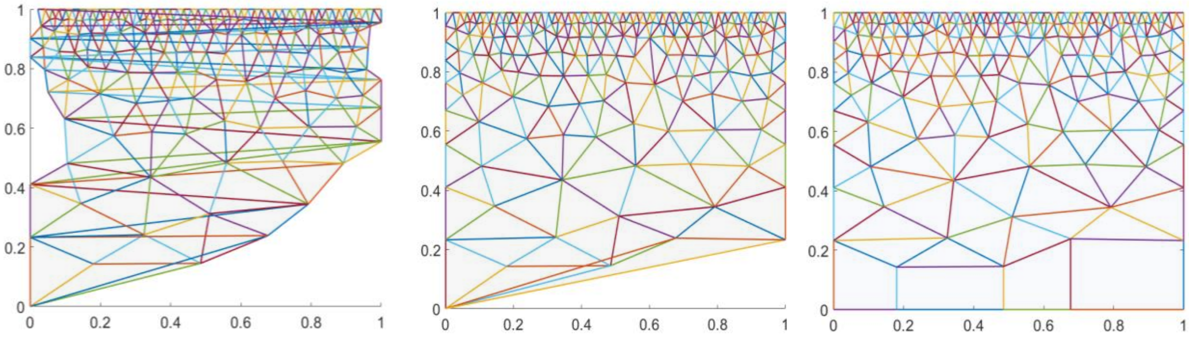

Figure 52 Steps followed to correct the parametric space mesh

Once the parametric space is corrected the high-order mesh can be generated.

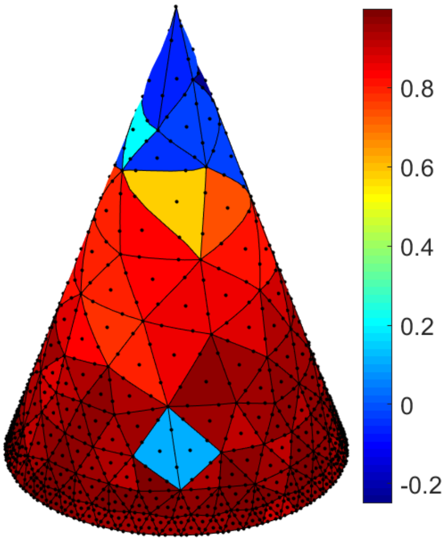

Figure 53 High-order p=3 mesh of the cone

The elements near to the cone tip have negative values of I. Increasing the number of steps used to solve the linear elastic problem improve the quality, but they are still invalid elements. Using higher order meshes do not solve de problem neither.

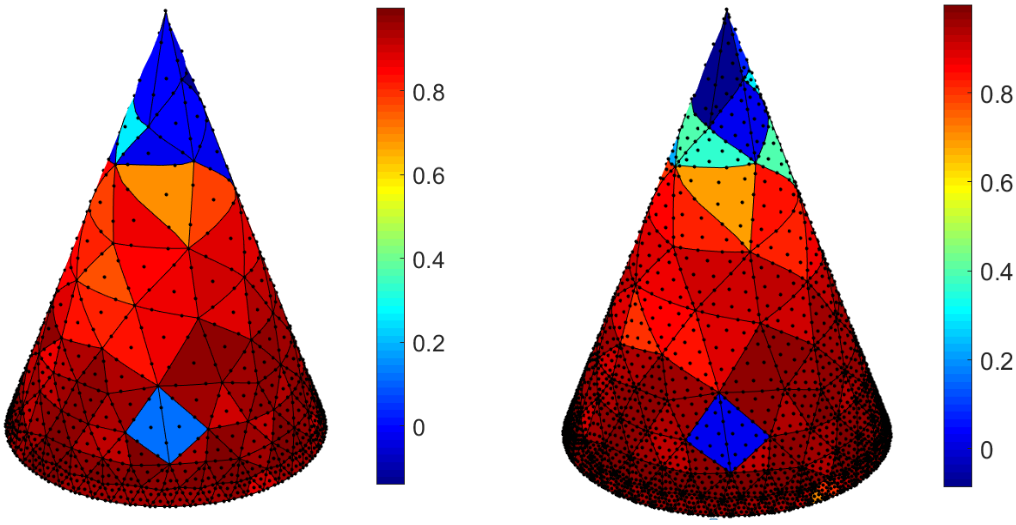

Figure 54 On the left p=3 high-order mesh constructed dividing the deformation in 5 steps. On the right, p=5 high-order mesh

Refining the mesh still gives low quality elements at the cone tip. This is produced by the characteristics of this geometry. If many elements are placed on the tip of the element, the angle of these elements would be too small yielding again to low quality elements.

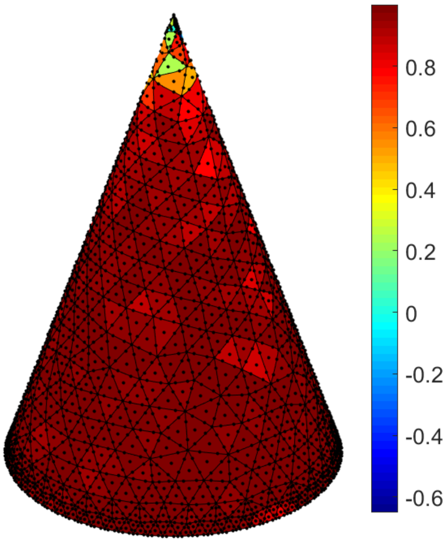

Figure 55 Refine high-order p=3 mesh

This result, again, points out the importance of a correct nodal distribution for a good quality mesh. In these types of geometries, the usage of optimisation methods is compulsory for obtaining valid meshes.

6. Applications

High-order methods make use of high-order curved meshes to compute physical problems, with a higher accuracy than linear meshes for the same number of elements.

Making use of a test problem, the high-order meshes created for the sphere are analysed and compared with the initial linear mesh.

The source term is selected so that the analytical solution is

$$u(x, y) = \exp(\alpha\, sin(ax + cy + ez) + \beta\, cos(bx + dy + fz)) \qquad (64)$$

with $\alpha = 0.1$, $\beta = 0.3$, $a = 5.1$, $b = 4.3$, $c = -6.2$, $d = 3.4$, $e = 1.8$ and $f = 1.7$. Dirichlet boundary conditions, corresponding to the analytical solution, are imposed.

The problem is solved for meshes from $p = 1$ to $p = 5$. The results obtained by solving the problem, making use of the finite element method, are compared with the result given by the analytical equation (64). The accuracy of the results is meshed using the L2 norm.

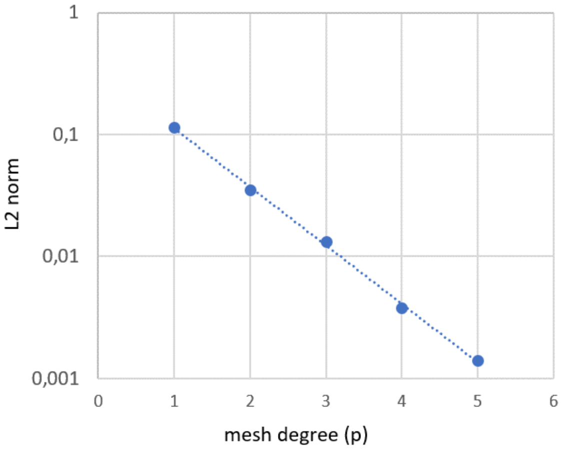

Figure 56 L2 norm for different degree meshes

It can be seen a reduction of the error, beginning with an initial error of 11.49% for the linear mesh and finishing with a 0.1% error for the $p = 5$ mesh. Notice the logarithmic scale. This is consistent with the exposed in [15].

The high-order method used in this test is discontinuous Galerkin DG. In this method, the continuity of the temperature field does not have to be satisfied at the boundaries of the elements. The continuity increases as the error diminishes. This can be observed in *Figure 57* that shows the temperature field on the sphere boundary for different order meshes.

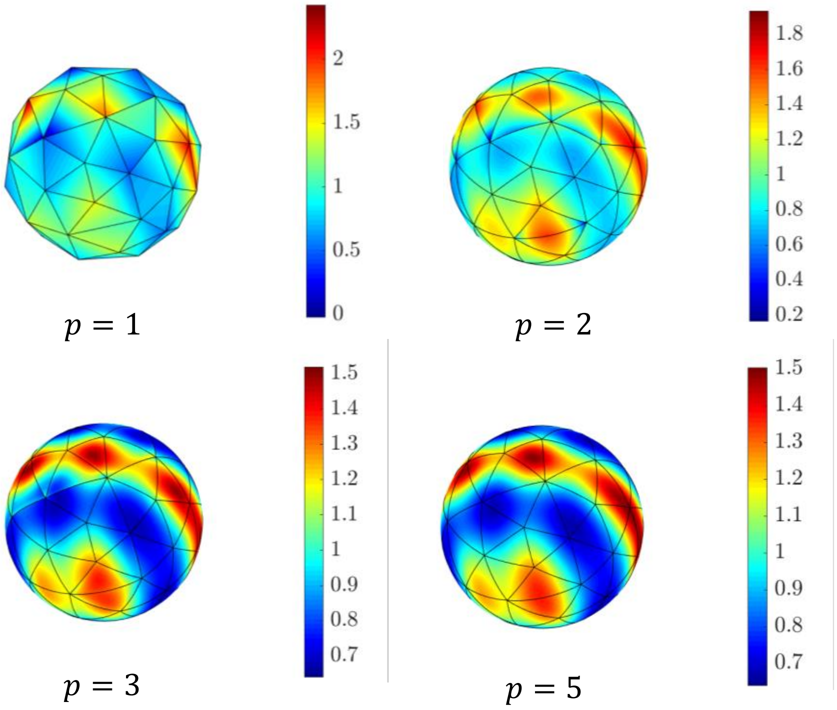

Figure 57 Temperature on the sphere surface for different degree meshes

For the linear and quadratic meshes the discontinuity along the elements' edges can be seen at first sight, while for the $p = 5$ mesh, a smooth temperature distribution has been achieved.

7. Conclusions

The development of a procedure for treating degenerated and periodic surfaces, and the implementation of this process into a linear elastic method for the generation of high-order curved meshes, have been described. The quality of the elements on the resulting meshes was analysed using the scaled Jacobian.

The process begins with the detection of the type of surface to be meshed. When a degenerated surface is detected, the elements laying on the degenerated edge have to be found. Then the parametric space mesh is corrected making use of the *collapsed quadrilateral* elements, explained above. For trimmed surfaces, the *collapsed quadrilateral* elements are again changed by regular quadrilateral elements. This step is needed to solve a linear elastic problem in the parametric space. After solving this problem, regular quadrilateral elements are substituted again for *collapsed quadrilateral* elements and the meshing process continues with the 3D mesh deformation.

In a similar way, when the mesh is periodic the parametric space mesh has to be corrected before solving the 3D elastic problem for generating the high-order mesh.

The implementation was tested by creating several meshes. The geometries contained different combinations of periodicity and degeneration. The proposed process was able to mesh all of them. The quality of the meshes has been related to the high-order nodes distribution and the amount of deformation. Degenerated and periodic elements behaved in a similar manner to the rest of the elements, in terms of quality.

Low quality elements produced by large deformations, can be corrected using a multistep deformation process. The distortion caused by the location of high-order nodes on the NURBS surface, needs to be addressed in the future to improve the quality of the meshes created with this method.

8. Future work

The high-order nodal distribution has been proved to be of great importance in the quality of the curved elements. A new method to project the high-order nodes, from the linear elements to their final location on the NURBS surface, that could reduce the distortion of the curve elements would improve the quality of the final mesh.

Other possible approach is to implement an optimisation algorithm to relocate the nodes on those elements with a quality under a preestablished value.

The refinement of the algorithms created during this project could, potentially, reduce the mesh generation time.

9. Bibliography

1. Stein E. History of the Finite Element Method-Mathematics Meets Mechanics. Hannover: Leibniz University, Institute of Mechanics and Computational Mechanics; 2014.

2. Szabo BA, Mehta AK. p-Convergent finite element approximations in fracture mechanics. St. Louis, Missouri: Washington University; 1978.

3. Babuška I, Szabó, B. A. and Katz, I. N., "The p-version of the finite element method." SIAM Journal on Numerical Analysis 18, pp. 515-545, 1981..

4. Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, et al. High-Order CFD Methods: Current Status and Perspective. ; 2012.

5. Sevilla R, Fernandez-Mendez S, Huerta A. Comparison of high-order curved finite elements. Barcelona: Universitat Politecnica de Catalunya, Departamento de Matematica Aplicada III; 2011.

6. Sevilla R. NURBS-Enhanced Finite Element Method. Barcelona: Universitat politecnica de Cataluña, Departament de Matemática Aplicada III; 2009.

7. S.Dey , O'Bara RM, Shepard MS. Towards curvilinear meshing in 3D: The case of quadratic simplices. ; 2001.

8. Poya R, Sevilla R, Gil AJ. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. Swansea: Swansea University, Zienkiewicz Centre for Computational Engineering; 2016.

9. Toulorge T, geuzaine C, Remacle JF, Lambrechts J. Robust untangling of curvilinear meshes. ; 2013.

10. Sherwin SJ, Peiró J. Mesh generation in curvilinear domain using high-order elements. ; 2002.

11. Persson P, Peraire J. Curved mesh generation and mesh refinement using lagrangian solid mechanics. ; 2009.

12. Xie ZQ, Sevilla R, Hassan O, Morgan K. The generation of arbitrary order curved meshes for 3D finite element analysis. Swansea: Swansea University, Civil and Computational Engineering Centre; 2012.

13. Taylor MA, Wingate BA, Vincent RE. An algorithm for computing Fekete points in the triangle. ; 2000.

14. Duster A, Rank E, Szabó B. The p-Version of the Finite Element and Finite Cell Methods. ; 2017.

15. Szabó B, Babuska I. Introduction to Finite Element Analysis: Formulation, Verification and Validation. ; 2011.

16. Piegl L, Tiller W. The NURBS book London: Springer; 1995.

17. Abgrall R, Dobrzynski C, Froehly A. A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems. ; 2014.