ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

GENERATING TRIANGULAR LATTICES FOR SURFACES WITH IRREGULAR BOUNDARY

Tatiana Sá Marques[1], and Vitor Dias da Silva[2]

[1]Department of Civil Engineering, University of Coimbra
Rua Luís Reis Santos - Pólo II, 3030-788 COIMBRA, Portugal
e-mail: t.marques@cobagroup.com

[2] INESCC, Department of Civil Engineering, University of Coimbra
Rua Luís Reis Santos - Pólo II, 3030-788 COIMBRA, Portugal
e-mail: vdsilva@dec.uc.pt

Keywords: Lattice, Delauney Triangulation, Bézier Curves, Ringwise Triangulation.

Abstract. *In this contribution algorithms for the generation of triangular lattice meshes for surfaces with an irregular boundary are presented. A central objective of this study was to get meshes with the maximum possible number of bars of a given length. Three different approaches are described. In all three the meshing process starts at the boundary and advances to the central part of the domain, by defining successive rings of bars. The differences between the three algorithms lie on the different ways as the problem of node elimination and smoothing is solved, as the rings become smaller. Algorithm A relies on the Delauney criterion for mesh smoothing. Algorithm B uses Bézier curves to approximate the cloud of initially generated points in a new ring, when smoothing is necessary. In algorithm C first an automatic elimination of points is performed, when the points in the new ring ate too close, which is followed by a relocation of interactively chosen points. A representative example is presented, allowing a comparison of the performance of the three approaches.*

1 INTRODUCTION AND LITERATURE REVIEW

The purpose of this work was to develop a pre-processor with a specific automatic mesh generator that can be used to study grid shells. The constraints imposed a priori were: to use a triangulated system to prevent the excessive deformability and resulting instability of non-triangulated systems; to develop a 2D generator based on a sound theoretical footing that could be extended to grid shells; to create a triangulated system of continuous rings that should enable the analysis of different stages of a self-supporting construction, since there is a belief in this potential capacity in these type of structures (double curvature shells); to use as many bars of equal length as possible, thereby offering advantages in terms of both aesthetics and the construction process.

Three separate algorithms were developed that correspond to different approaches of the used method and solve the smoothing of the mesh, which allows a geometric solution without the elements overlapping. Algorithm C corresponds to the semi-automating of a manual pre-processor with automatic generation of the triangulation with bars of equal length, followed by slight automatic smoothing (getting rid of very short bars), and manual smoothing (manual choice of points to be displaced). The heavy reliance on manual smoothing makes it difficult to generate two identical meshes. Algorithm B adjusts the Bézier curves to a series of points. The Bézier curves are widely used in computer graphics, but no report of their use in mesh generation was found in the literature. Bézier curve adjustment is used for smoothing between the rings of the mesh. Coupling the triangulation programmes in algorithm C, the Bézier curves and creating the connection mesh gave algorithm B. Algorithm A is the only wholly automatic generator, based on geometric criteria and it fulfils the objectives set initially.

The efficacy of the computer programs is demonstrated in the numerical applications. Lattices generated on a plane surface with a irregular curved boundary, and obtained with the three algorithms, are presented. The results are compared with respect to the total number of bars, the number of same-size bars and the regularity of the meshes.

References in the literature on the automatic generation of meshes based on geometric criteria are scarce in comparison with references on automatic generation based on adaptive finite elements techniques. No models were found in the literature that could be directly adapted to solve the specific problem of double curvature grid shells.

Of the three types of mesh - structured, unstructured and hybrid - this work uses unstructured meshes because there may be a variable number of adjacent elements in the internal nodes, without explicit connective relations and they are more adaptable to arbitrary and/or irregular geometry domains. Unstructured meshes also allow local and adaptive refinement, impossible to achieve in structured meshes.

With Sibson, 1979 [1] and Watson, 1981 [2], we find triangulation algorithms of domains based on the Delaunay triangulation technique. S.H.Lo, 1985 [3] presents a discretization algorithm of arbitrary planar domains in triangles that generates finite element meshes based on the advancing front and Delaunay triangulation techniques. Chew, 1989 [4] and S.H.Lo, 1989 [5] present restricted Delaunay triangulation versions applied to planar domains of irregular, non-convex boundaries. Zienkiewicz and Wu, 1994 [6] and El-Hamalawi, 2004 [7] introduced the possibility of controlling the direction of mesh elongation.

2 DOMAIN BOUNDARY

The domain boundary is defined by means of a polygon and interpolated by means of Bézier curves, as shown in Figure 1. Each of the vertices in Figure 1 is considered in turn, as vertex

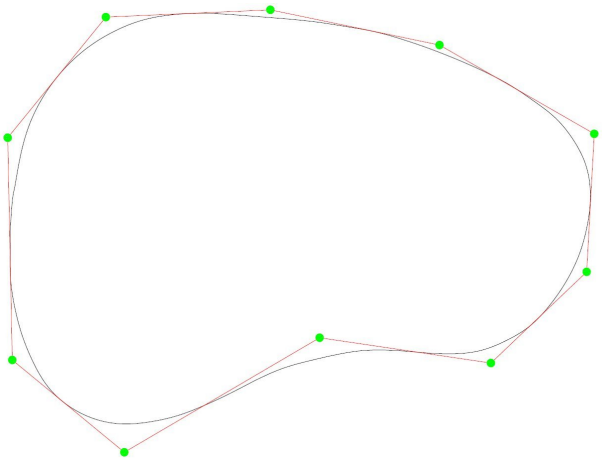


Figure 1: Base polygon and boundary (defined by a set of Bézier curves).

V_i, defined by the adjacent segments designated generically as V_{i-1} and V_{i+1}. In each segment, $V_{i-1}V_i$ and V_iV_{i+1}, the midpoints are designated respectively by 0 and 3. The midpoints will be the endpoints of the Bézier curve, as shown in Figure 2-a. The curve developed between points 0 and 3 and is controlled by means of two more points, 1 and 2, as shown in Figure 2-b, with the coordinates given by means of a parameter α

$$x_1 = (1-\alpha)x_{V_i} + \alpha x_0 \quad (1)$$
$$y_1 = (1-\alpha)y_{V_i} + \alpha y_0 \quad (2)$$
$$x_2 = (1-\alpha)x_{V_i} + \alpha x_3 \quad (3)$$
$$y_2 = (1-\alpha)y_{V_i} + \alpha y_3 \quad (4)$$

In the set of Bézier curves depicted in Figure 1 $\alpha = 0.35$ has been used.

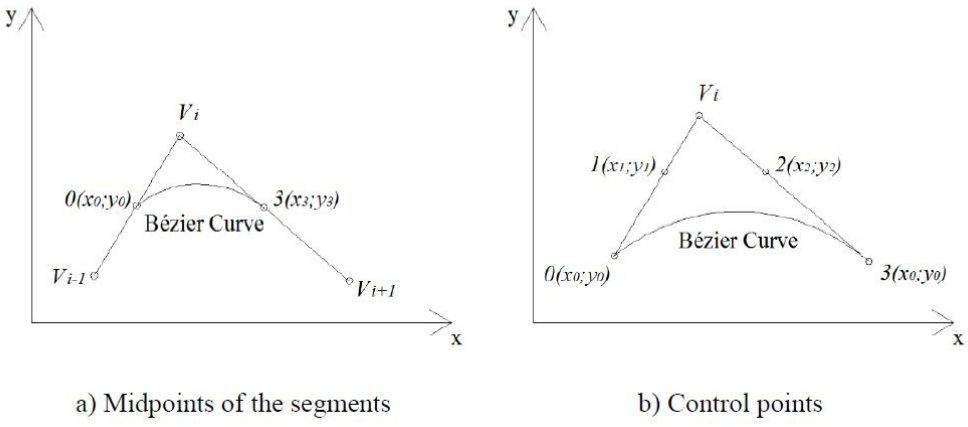


Figure 2: Construction of the Bézier polygon and its curve for each vertex V_i.

3 LATTICE GENERATION

3.1 Common features to the three approaches

In every of the three considered approaches the first step consists of defining the nodal points that lie on the boundary. The distance between two neighbouring points is as close as possible to the targeted length of the bars C_b, under the condition that all these distances have the same value.

Once these points are defined, a new interior ring of points is defined, which lie on the left, as the progression on the boundary points takes place in an anti-clockwise direction. Considering two consecutive points, A and B, on the boundary, the coordinates of a point C of the new ring are obtained by the expressions (Figure 3)

$$x_c = x_a + C_b \cos\gamma \tag{5}$$

$$y_c = y_a + C_b \sin\gamma \tag{6}$$

with $\gamma = \beta + \theta$ and $\theta = \arccos\left(\frac{\sqrt{(x_b-x_a)^2+(y_b-y_a)^2}}{2}\frac{1}{C_b}\right)$.

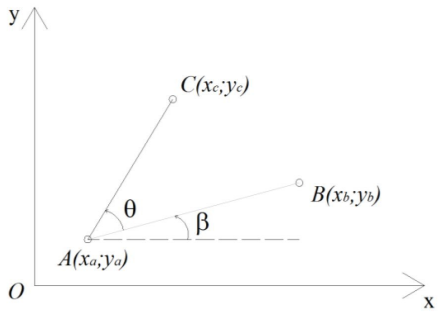


Figure 3: Determination of point C.

After the needed rings have been generated, using one of the three algorithms described in the next sub-sections, the lattice is closed by linking every point on the last ring with a mean point, as shown in Figure 4.

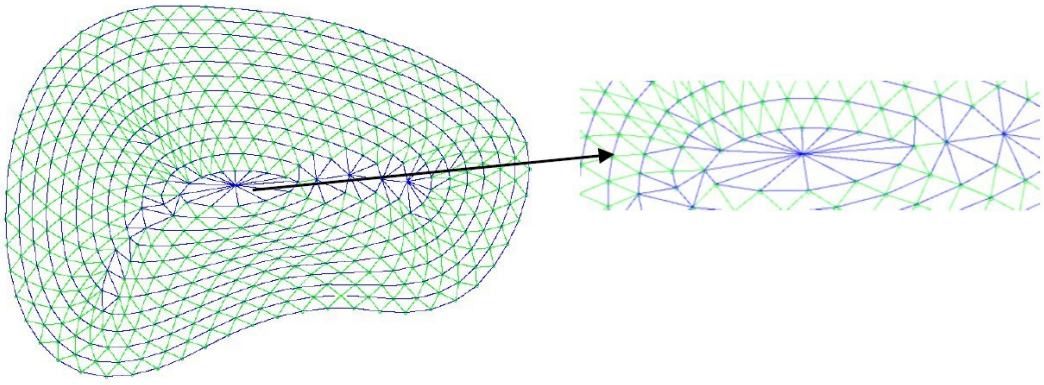


Figure 4: A complete mesh with detail of mesh closure.

3.2 Algorithm A

After the formation of ring we have to update the generation front that serves as the base for the construction of the next ring, i_a. The ring ia will be formed from i points of the ring i_{a-1} and will give k points that will be the generation front of the ring i_{a+1}. The number of k points is equal to i under these conditions:

- the distance between points k and $k+1$ is greater than the value accepted, according to the smoothing criterion established for the distance;
- the triangulation obtained with i and k points are Delaunay triangles.

In the absence of any these conditions the number of k points is smaller than i, and we need to determine which points will constitute the new generation front.

The main property of the Delaunay triangulation or Delaunay criterion is that no point of a triangular element can belong to the interior of the circle defined by the three vertices of another triangle. Figure 5 shows an example in which the criterion is violated by the presence of point P inside the circle that circumscribes triangle ABC.

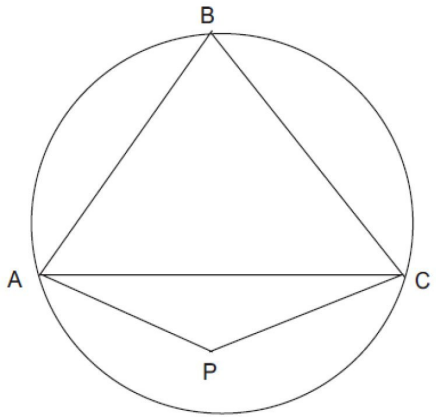


Figure 5: Violation of the Delaunay criterion.

The points that do not obey the Delaunay criterion are detected as follows:

1. For each set of i and $i+1$ points a k point is determined, and the circumference that passes through i, $i+1$ and k. The circumference is defined from the intersection of the perpendicular bisectors of the edges of the triangles.
2. Once the circle that passes through i, $i+1$ and k has been determined, those k_j points formed by all the sets of 2 successive points of the front located inside the circle are verified. The existence of an arbitrary point P inside the circle is verified by comparing the distance of P from the centre of the circle O, d_{PO}, with the radius of the circle: if d_{PO} is the same as or greater than the radius, then P does not violate the Delaunay criterion and thus can be regarded as a valid point. Thus the lack of overlapping elements in the triangulation is ensured.
3. All the points of the generation front that give rise to triangulations that do not meet the Delaunay criterion are included in the i line of the matrix corresponding to the circle formed by i, $i+1$ and k.

4. At the end of the verifications, a matrix $[mp]_{nx\times(n+1)}$ is obtained, in which n is the number of generation front points.

5. After constructing the matrix $[mp]$ it is necessary to define those k points that should be eliminated from future generation front and triangulation to be carried out with the i points that will no longer have two-way correspondence with k. This procedure is called matrix condensation.

Condensation of the detection matrix $[mp]$

The matrix $[mp]$ contains information about overlaps that will occur in the formation of the ring if the triangulation continues with the same number n of front points as were used in the formation of the previous ring. With this information we must decide which points to keep on the generation front, which points to remove, what new points to introduce and how the triangulation should be done when new points are eliminated or created. The method follows a logic similar to Gauss-Jordan elimination. The decisions taken, with regard to the above situations, are as follows:

- If the circle C_i contains point $n + j$, then circle C_j is deemed to contain point $n + i$, and the occurrence is referenced on the matrix $[mp]$. This procedure requires going through each of the columns $j = npc_a + 2, \ldots, npc_a + n + 1$, of $[mp]$ and putting in the first row where the value is not null all values of the points that give rise to the non-violation of the Delaunay criterion.

- Once the sets of points to be removed from the generation front are established, we must check if some of the sets intersect. If they do, the matrix $[mp]$ is transformed so that the points to remove are between the smallest and largest points in the sets, taking into consideration the special case of the set of points that encloses the end and the beginning of the ring.

After the matrix [mp] condensation we know which points to remove, so that there is no overlap of triangulation when processing the ring.

Elimination/creation of new points

Whenever it is necessary to remove generation front points this results in the distortion of the mesh, which is greater the higher the number of points to be eliminated. Criteria must be introduced to lead to the least distortion possible. Several options can be taken. The following strategies were adopted in the algorithm developed:

- when an i point is removed from the generation front, the end of the triangulation segments that would connect to the i point pass to the $i - 1$ point;

- when two or more points are removed from the generation front, a point is created, a midpoint, whose coordinates are the average of the coordinates of the points removed and the two adjacent points (before and after) and which remain on the front. The end of all triangulation segments which would connect to the eliminated points replaced then has its end at the midpoint.

These criteria proved to be effective in the triangulations carried out. Other criteria could be introduced, in particular: the successive use recursive Delaunay criterion; consideration of

the criterion of smoothing angle (angle limitation); introduction more than one midpoint when distances are greater than $2 \times C_b$. The first two options did not show good results in the examples studied.

Distance-based smoothing

Different smoothing methods were developed in this work. One eliminates the bars whose length is less than a percentage value of the length prescribed for the bars (C_b) and this was applied to algorithm A. Whenever the distance between consecutive points is below the present value, the points are removed from the generation front such that the bars converging at those points will converge at the point immediately preceding it. These considerations are valid for all the bars, apart from those converging on the last k point of the ring, which are converging at the first point of the ring. Figure 6 illustrates the application of this method in a mesh and Figure 7 shows local enlargement of the smoothing.

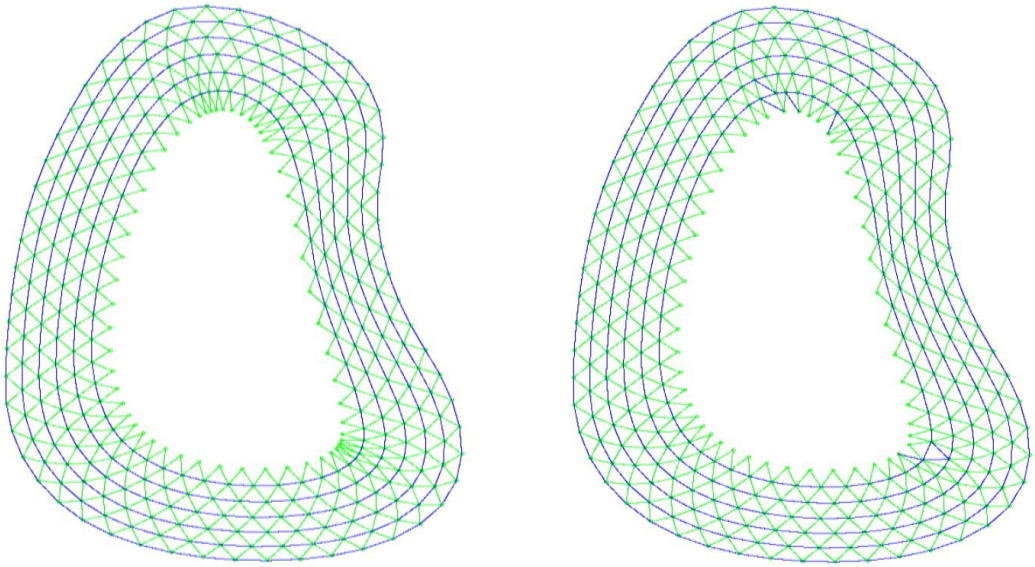
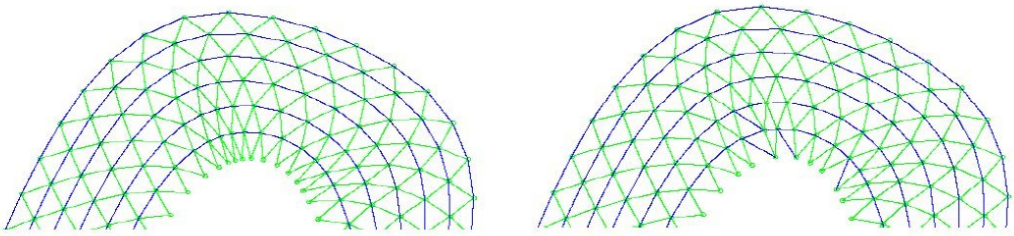


Figure 6: Overall effect of distance-based smoothing process.



a) Before smoothing b) After smoothing

Figure 7: Local enlargement of effect of distance-based smoothing.

3.3 Algorithm B

In each ring we have to update the generation front that serves as the base for the construction of the next ia ring. The ia ring will be formed from npr (number of points of the ring) of the $ia-1$ ring and will give k points that will be the generation front of the ring $ia+1$. In this algorithm, the number of k points is equal to npr, except when the smoothing based on the Bézier curves is applied. In that case, the number of k points is generally different from npr, depending on the number of points given by dividing the curve into segments of equal length, C_b.

As mentioned above, after the formation of a ring, smoothing corresponds to making the initially generated npr points approximate a curve. The goal is to use the Bézier curves to find the curve that best approximates the npr points of the ring. This closed curve can be related to a base polygon formed by a number of previously established points, as was analysed in the definition of the domain boundary. The number of points of the polygon, nv should be chosen/tested to adjust/fit the description of the desired geometry with sufficient accuracy. In the examples presented in this paper five-node polygons ($nv = 5$) have been used. With the number of base points set, we have to find the coordinates of the points corresponding to a given fixed value of the parameter α, which best allows a fit between the curve and the points. The curve is obtained iteratively, using the procedure set out in the following. Once the curve that best fits the points of the generation front, we can recalculate the value of the α parameter which leads to a total length of the curve that is a multiple of the length of the pre-defined bar, C_b. The value of α is also determined iteratively. The iterative method considered in the two approximations was the Newton-Raphson method.

The main steps of this procedure are

1. Using the procedure described in sub-section 3.1 new rings are generated, until any of the interior bars become smaller than a given value, which, in the example presented in Figure 8, occurs in ring 7.

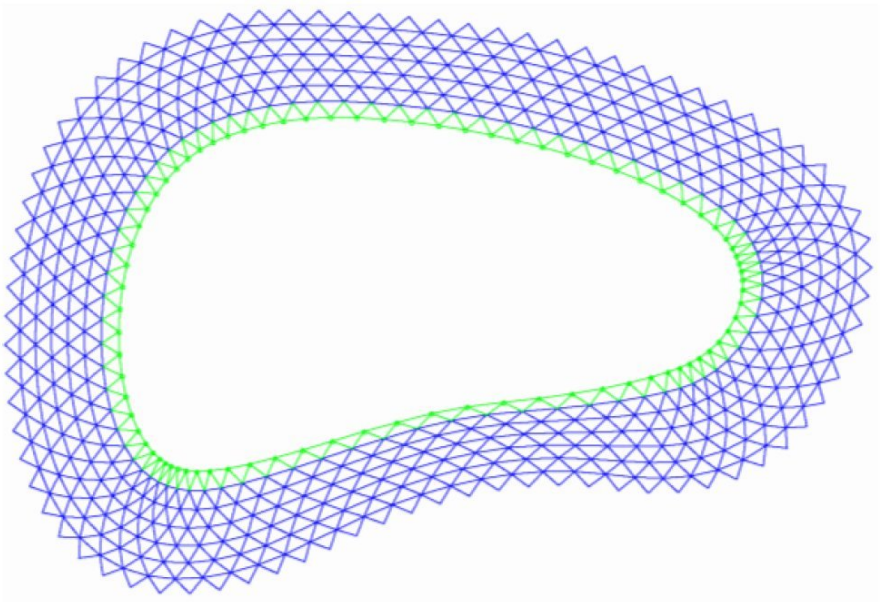


Figure 8: Initially generated nodes and bars for ring 7 of a mesh (green).

2. A starting set of Bézier curves (5, in this case) is defined, taking as vertices 5 of the initially generated npr points, as depicted in Figure 9

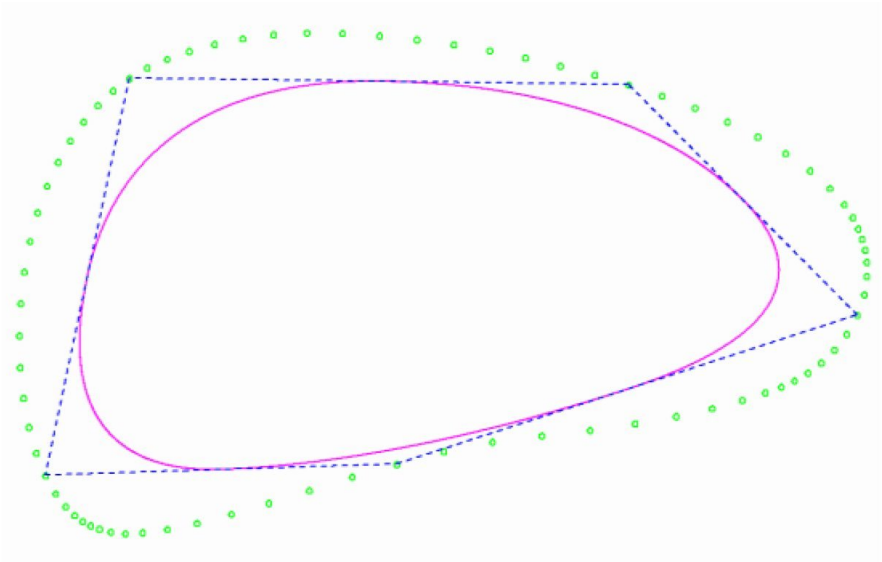


Figure 9: Starting configuration of a 5-vertex approximation set of Bézier curves.

3. An error function E is defined as the sum of the minimum distances of each of the npr points to the set of Bézier curves (Figure 10) [8]

$$E = \sum_{n=1}^{npr} d_{k_n}(x_1, x_2, \ldots, x_{nv}, y_{nv}) = \sum_{n=1}^{npr} d_{k_n}(c_1, c_2, \ldots, c_{nc}) \tag{7}$$

where $nc = 2nv$ is the number of unknowns to be iteratively approximated (10, in this case).

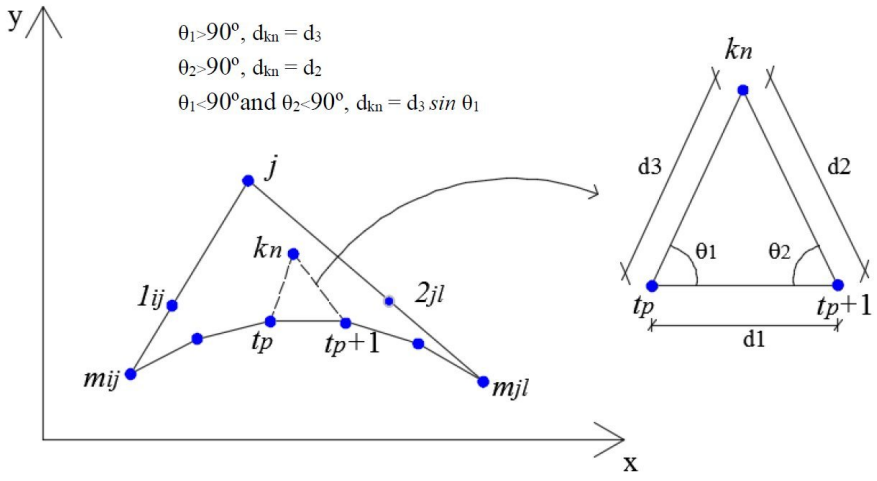


Figure 10: Minimum distance from point kn of the npr set, to the linearized Bézier curve.

4. The minimization of the error function E is performed by defining nc equations $f_j = 0$ ($j = 1, \ldots, nc$)

$$f_j = \frac{\partial E}{\partial c_j} = \sum_{n=1}^{npr} \frac{\partial d_{k_n}}{\partial c_j} = 0 \tag{8}$$

These derivatives are computed numerically. This system of equations is solved by using the Newton-Raphson algorithm (i - iteration counter)

$$\mathbf{C}_{i+1} = \mathbf{C}_i - \mathbf{J}_i^{-1}\mathbf{F}_i \quad \Longleftrightarrow \quad \mathbf{J}_i \Delta\mathbf{C} = -\mathbf{F}_i \tag{9}$$

with $\Delta\mathbf{C} = \mathbf{C}_{i+1} - \mathbf{C}_i$,

$$\mathbf{C} = \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{nc} \end{Bmatrix} \quad , \qquad \mathbf{F} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{nc} \end{Bmatrix} \tag{10}$$

and

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial c_1} = \frac{\partial^2 E}{\partial c_1^2} & \frac{\partial f_1}{\partial c_2} = \frac{\partial^2 E}{\partial c_1 c_2} & \cdots & \frac{\partial f_1}{\partial c_{nc}} = \frac{\partial^2 E}{\partial c_1 c_{nc}} \\ \frac{\partial f_2}{\partial c_1} = \frac{\partial^2 E}{\partial c_2 c_1} & \frac{\partial f_2}{\partial c_2} = \frac{\partial^2 E}{\partial c_2^2} & \cdots & \frac{\partial f_2}{\partial c_{nc}} = \frac{\partial^2 E}{\partial c_2 c_{nc}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{nc}}{\partial c_1} = \frac{\partial^2 E}{\partial c_{nc} c_1} & \frac{\partial f_{nc}}{\partial c_2} = \frac{\partial^2 E}{\partial c_{nc} c_2} & \cdots & \frac{\partial f_{nc}}{\partial c_{nc}} = \frac{\partial^2 E}{\partial c_{nc}^2} \end{bmatrix} \tag{11}$$

These derivatives are also computed numerically, using high precision arithmetic.

5. Determination of the α parameter of the set of Bézier curves - Once the Bézier curve has been defined it has to be divided into several segments of the pre-defined length C_b so as to restart the whole mesh generation process for the following rings. The last segment of the curve naturally results in a length different from C_b. The curve is thus very slightly readjusted to change its length so that the last segment also has the prescribed length, C_b. The used approach consists of an iterative computation of the α parameter of the Bézier curves in such a way that the division of the final length of the set of Bézier curves into segments of length C_b gives a whole number. The iterative process for determining the α parameter can be summarized as follows.

   - Initial estimation of α: $\alpha = 0.35$.
   - Evaluation of the length of the last bar, $llb(\alpha)$, and definition of function $g(\alpha) = llb(\alpha) - C_b$.
   - Solution of equation $g(\alpha) = 0$ via Newton-Raphson iteration: $\alpha_{i+1} = \alpha_i - g/\left(\frac{\mathrm{d}g}{\mathrm{d}\alpha}\right)_i$; the derivative is numerically computed.

   Figure 11 shows the set of Bézier curves after convergence of both iterative procedures.

6. Connection bars between the previous ring and the new ring - The nodal points are numbered from i_{initial} to i_{final} in the previous ring and from j_{initial} to j_{final} to in the new ring, where j_{initial} is the point in the new ring that is closest to point i_{initial} of the previous ring. A triangulated connection between the two rings is generated as follows (Figure 12)

   (a) First bar connects i_{initial} to j_{initial}.

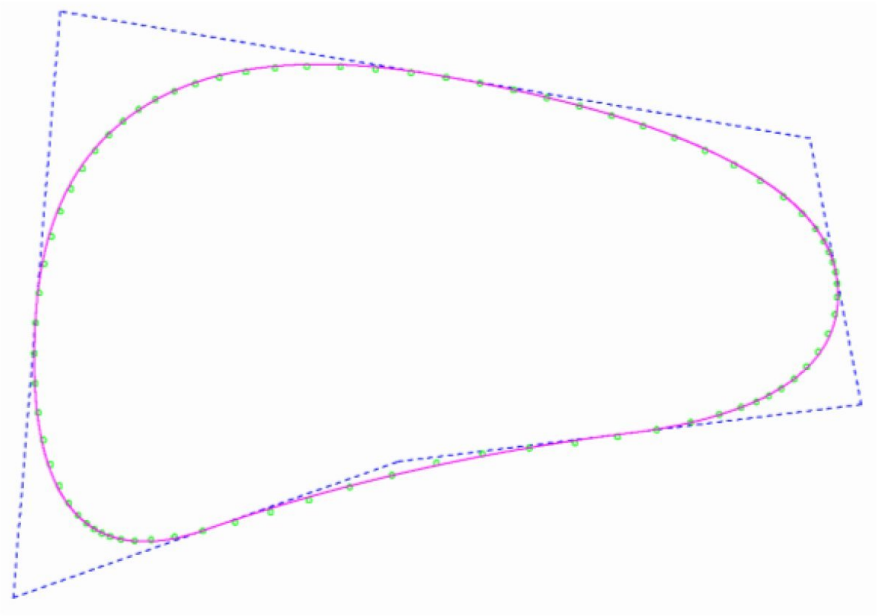


Figure 11: Set of Bézier curves after convergence of the iterative processes.

(b) For points i, from $i_{\text{initial}+1}$ to i_{final}:

- identify j_p as the closest nodal point to i in the new ring;
- connection bars between i and the points between the last connected point in the new ring, j_l and j_p; if $i = i_{\text{final}}$ and $j_p < j_{\text{final}}$, bars connecting i_{final} with j_{p+1} to j_{final} are added;
- the last bar connects j_{final} to i_{initial}

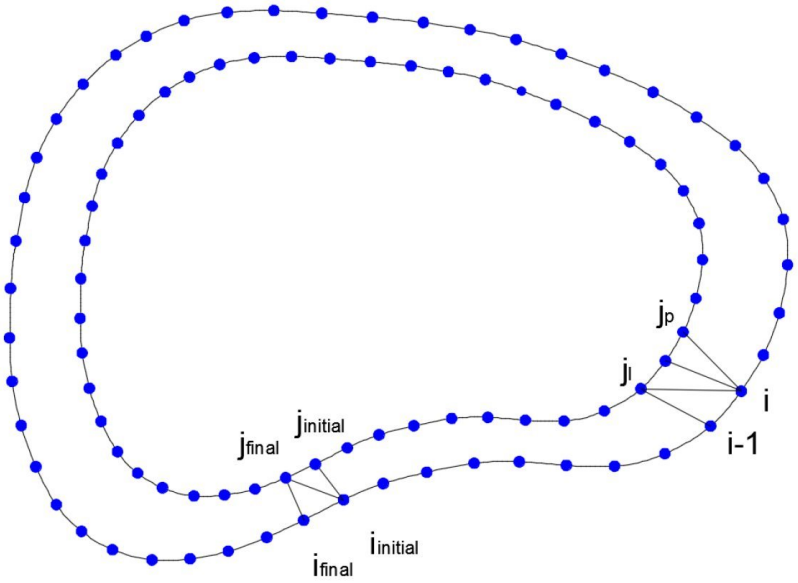


Figure 12: Connection bars between rings.

Figure 13 shows the new ring obtained using the above described procedure.

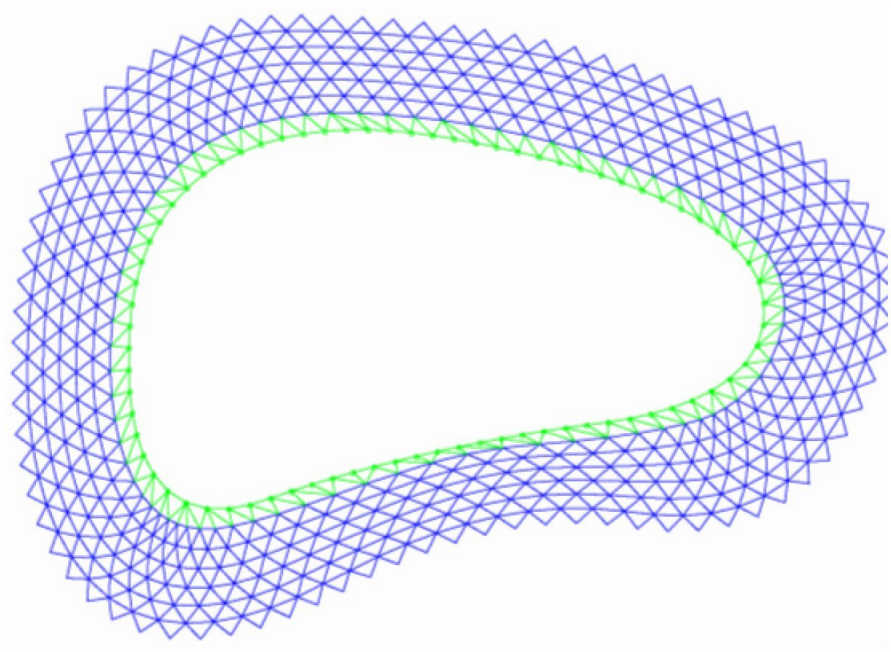


Figure 13: Final configuration of the nodes and bars in ring 7 (green).

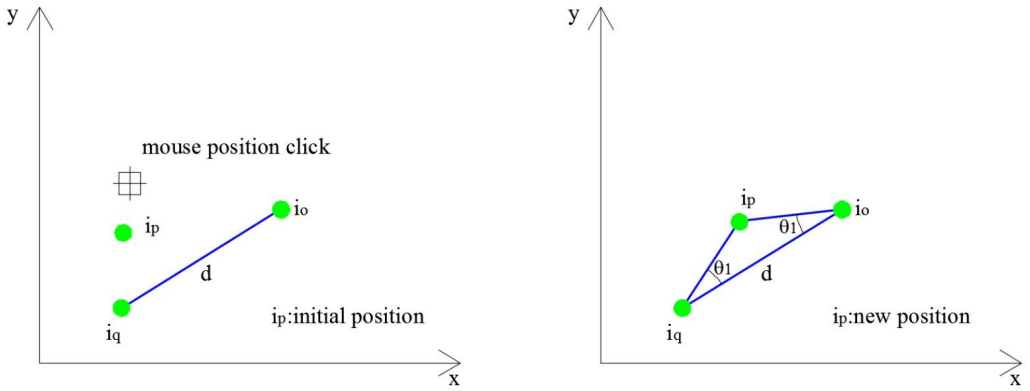
3.4 Algorithm C

As with algorithms A and B, this method, too, is based on the advancing front method, and it is therefore necessary to update the generation front after each ring is formed since this forms the basis for constructing the next i_a ring. The i_a ring is formed from i points of the i_{a-1} ring and gives k points that will be the generation front of the i_{a+1} ring. The number of k points is equal to i when the distance between two consecutive points in the new ring is greater than a given accepted value, according to the smoothing criterion established for the distance. In the absence of this condition, point C in Figure 3 is rejected and points A and B are connected to the last not-rejected point C. As a consequence, the number of k points is smaller than i. Besides this automatic smoothing process, smoothing is complemented by manually selecting the nodal points of the new ring that must be moved, using the algorithm described in Figure 14: the program automatically adjusts the point that is closest to the mouse pointer (Fig. 14-a) to a position that makes angles θ_1 with segment $i_q i_o$, which are the points before and after i_p in the new ring (Fig. 14-b).

The angle θ_1 in the examples was 5^o. Other values can be taken, depending on the type of adjustment intended. This is a simple smoothing method (no renumbering of the nodes or bars is performed) and it is quite effective at regularizing the lattice as the examples in the next chapter show.

4 EXAMPLE

This example represents an irregular boundary with distinct convexities and concavities. The maximum horizontal length is about $200\,m$ and the maximum vertical length is about $75\,m$. A value $C_b = 7\,m$ was considered. Table 1 gives the number of fixed and variable length bars



a) Manual choice of point to be moved i_p. b) Relocation of point i_p to new position.

Figure 14: Manual smoothing in Algorithm C.

obtained by the algorithms. Lattices generated with each algorithm are represented in Figures 15, 16 and 17.

	No. of fixed length bars	No. of variable length bars	Total no. of bars
Algorithm A	957 (62%)	590 (38%)	1547
Algorithm B	807 (51%)	774 (49%)	1581
Algorithm C	903 (61%)	589 (39%)	1492

Table 1: Bars of fixed and variable length for a mesh with an irregular boundary.

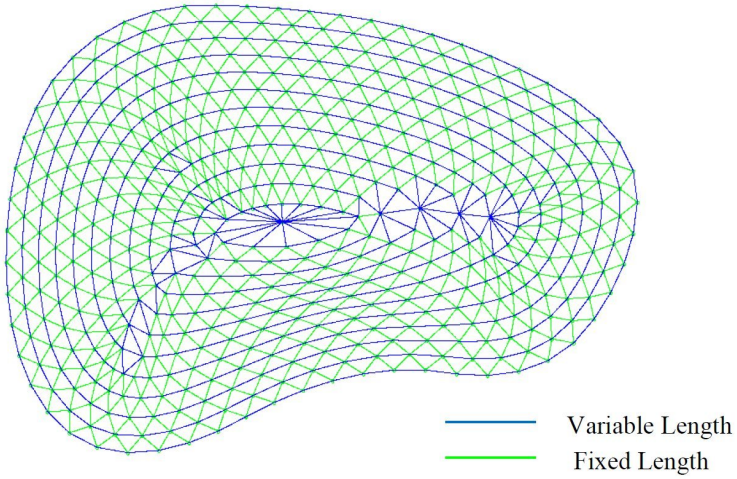


Figure 15: Lattice obtained with Algorithm A.

In this example, Algorithm B generated the mesh with the largest number of elements, the smallest number of bars of standard length and the most irregular boundary. Algorithms A and C give a larger number of bars of fixed length simultaneously with more regular meshes than Algorithm B does. Note the regularity of the Algorithm A mesh compared with that of C.

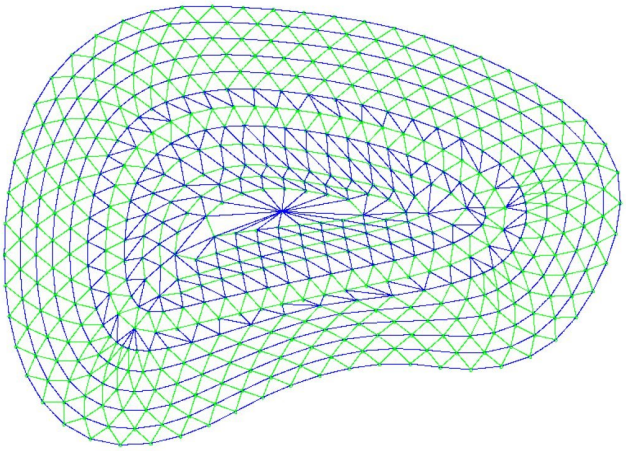


Figure 16: Lattice obtained with Algorithm B.

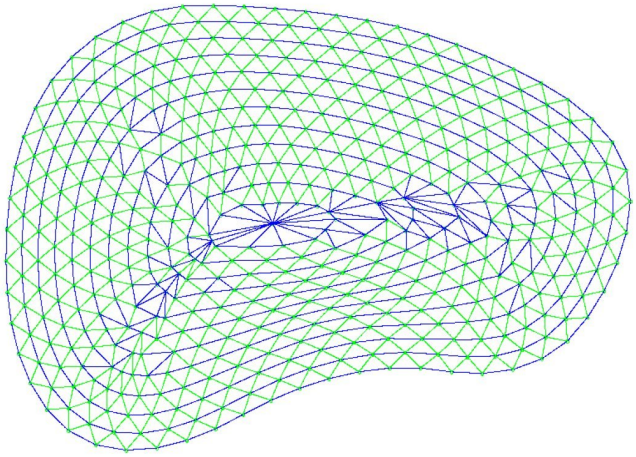


Figure 17: Lattice obtained with Algorithm c.

5 CONCLUSIONS

Three algorithms for generating triangulated meshes for curved boundaries were developed so that the meshes could be used to study the behaviour of double curvature shells, including analysis of construction processes, without shoring. The feasibility of simulating the construction stages of shells without shoring warranted the procedure used for generating triangulated meshes in successive rings. Another restriction imposed is the use of as many bars of the same length and equal to a predefined value as possible. The development of curved boundaries with successive rings of gradually smaller perimeters makes it impossible to use bars of just one length.

The three types of mesh generation can be classified according to their automation, ease of implementation and efficiency. Measures of efficiency are the number of elements, number and percentage of elements of the same length. The following conclusions can be drawn from applying the three algorithms:

Algorithm A - This is the only fully automatic algorithm. It always let to the highest percentage of bars of standard length and to the most regular meshes, that is, with triangles that were

less distorted compared with the other algorithms;

Algorithm B - Using Bézier curves for the smoothing process made this a semi-automatic algorithm: smoothing is introduced into the predefined rings by the user and the vertices of the support polygon can be used to improve the fit of the curve. However, the connection of the ring smoothed by the Bézier curve with the rings formed previously results in a less regular mesh and fewer bars of standard length. This algorithm led to the generation of meshes that were less regular and less efficient;

Algorithm C - This is a semi-automatic algorithm in which the user smoothes the mesh manually. It is easy to implement, with less supporting theory and provides efficient solutions with regular meshes. It was noted that the percentage of bars of the same length is similar to Algorithm A and that it generated the smallest number of elements. The main drawback is that relies wholly on the user and the solution cannot be replicated.

REFERENCES

[1] Sibson, R., Locally equiangular triangulations. *Computer Journal*, Vol.2, 243-245, 1979.

[2] Watson, D.F., Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *Computer Journal*, Vol.24, 167-172, 1981.

[3] S.H.Lo, A new mesh generation scheme for arbitrary planar domains. *Inter. J. N. Meth. Eng.*, Vol. 21, 1403-1426, 1985.

[4] Chew, L.Paul, Constrained Delaunay triangulations, *Algorithmica*, Vol.4, 97-108, 1989.

[5] S.H.Lo, Delaunay triangulation of non-convex planar domains, *Inter. J. N. Meth. Eng.*, Vol. 28, 2695-2707, 1989.

[6] Zienkieviewicz, O. Wu, J., Automatic directional refinement in adaptative analysis of compressible flows. *Inter. J. N. Meth. Eng.*, Vol. 37, 2189-2210, 1994.

[7] El-Hamalawi, A., A 2D combined advancing front-Delaunay mesh generation scheme, *Finite Elements in Analysis and Design.*, Vol. 4, 967-989, 2004.

[8] J. Argyris; I. St. Doltsinis; V. D. da Silva. Constitutive modelling and computation of non-linear viscoelastic solids. Part II: Application to orthotropic PVC-coated fabrics, *Comput. Meth. Appl. Engrg.*, Vol. 98, 159-226, 1992.