Blucher Mechanical Engineering Proceedings
May 2014, vol. 1 , num. 1
www.proceedings.blucher.com.br/evento/10wccm

10th World Congress on Computational Mechanics
8-13 July 2012 • São Paulo • Brazil

COMPUTATIONAL GENERATION OF SHELLS WITH FREE FORM AND REGULAR POLYGONAL PLANTS

E. M. P. Sabino[1], I. Vizotto[2], A.M. Fidalgo[3]

[1]Department of Structural Engineering, University of Campinas - UNICAMP
(e085823@dac.unicamp.br)

[2] Department of Structural Engineering, University of Campinas - UNICAMP

[3] Department of Structural Engineering, University of Campinas - UNICAMP

Abstract. *The shells are structures which resist mainly by the inherent properties of its forms when they are requested by several actions, enabling constructions extremely slender and different architecture. The property of resistance is even more evident when the shapes of the shells are optimized to obtain the structural behavior in membrane regime with bending stresses no or negligible However, design of free-form shells are complex due to the computational difficulties of its generation, since the three-dimensional structures with shapes optimized of shell cannot be represented by mathematical equations. This paper aims to present a computational approach to facilitate the generation of optimized shapes of shells through the development of a software generator of membranes with different polygonal plants. After the initial modeling of a flat mesh formed by triangular elements and automatically generated, the computational generation of the three-dimensional shapes is performed by means of Finite Element Method considering the membranes subjected to large displacements and large deformations. The software for meshing initial plan through a standard set was developed in C programming language, with capacity to generate information relating to Finite Element Method such as the coordinates and nodal incidence of triangular elements with six degrees of freedom per node. The structures in shell generated through flexible membranes are polygonal projection with a minimum of three sides, and its boundary conditions are the regions of the vertices that remain fixed after the application of the loadings. The various models of optimized shapes depend of the numbers of fixed supports, and are obtained by varying the intensity of the load actions on the membrane. Several models of shapes obtained are presented to corroborate the results obtained.*

Keywords: *Generation, Computational, Shell*

1. INTRODUCTION

The generation of optimized shapes of shell structures involves a very complicated mathematical analysis. A computational process to generate free-form shells through simulation of membranes by means of mathematical programming combined with the finite element technique is presented in the paper VIZOTTO [Automation in Construction, 2010]. The computational model simulates automatically a flexible membrane, initially in the horizontal plane surface, with any shape and boundary conditions.

In this paper, is presented a computational approach to facilitate the generation of free-form shells with different regular polygonal plants through the development of a software generator of finite element meshes. After the initial modeling of a flat mesh formed by triangular elements and automatically generated, the shell structures obtained from flexible membranes are polygonal projection with a minimum of three sides, and its boundary conditions are the regions of the vertices that remain fixed after the application of the loadings in the structures.

2. DEVELOPMENT OF SOFTWARE TO GENERATION OF PLANE MESH OF FINITE ELEMENTS TO REGULAR POLYGONS

The idea of software developed in this paper to generate plan membrane is basically determine the mesh projection of any shell in the plane inscribed in a circle of radius pre-defined. The input of the software is the external radius, the internal radius, the number of divisions of the mesh and the number of fixed nodes for support. The external radius (R) is only one auxiliary parameter that must be greater than the internal radius that in fact is the effective radius of the structure, resulting in the rounded shape of the shell support. The structure span is determined by multiplying the effective radius (Ref) by two. The division of the mesh is established between supports and radially, the latter is limited by the difference between the external and effective radii. The software allows varying the width of the support with a fixed number of nodes which must always be odd to keep the structure symmetric. The work's challenge is create a mesh with the nodes' coordinates and the nodal incidence of elements to any type of polygonal projection in the plane through the same software. For this it is required finding a standard in the mesh to facilitate the implementation of the algorithm for software development.

2.1. Identifying the standard for generating of regular polygon

The search for a standard present in any polygon led to exploration of symmetry around a central point. As shown in Figure 1, fixed to the origin of the coordinate system, P1 (0,0) on the center. A next point is obtained at the end of the line segment R, P2 (R, 0), where R is the external radius of the polygon, i.e. the radius of the circle that circumscribing the polygon. The edge formed by the segment connecting the origin to one of the vertices of the polygon contained in X axis, $\overline{P1P2}$ is subjected to rotation through an angle θ in a counterclockwise direction from the origin to determine another point P3 (Rcosθ, Rsinθ). The angle θ can be expressed by the number of sides with:

$$\theta = \frac{360°}{N}. \tag{1}$$

where N is the number of polygon sides which can vary from 3 without the upper limit. When N is large, θ approaches zero and thus can be obtained circular plant to generate domes.

With three known points can define a triangle which is the standard basis for the construction of the remaining of polygon through rotations of the first triangle in a counterclockwise direction. Following this pattern, it is not necessary directly determine the coordinates of all mesh nodes, optimizing the computational process. Figure 1 illustrates the standard found in three polygons represented with three, four and five sides.

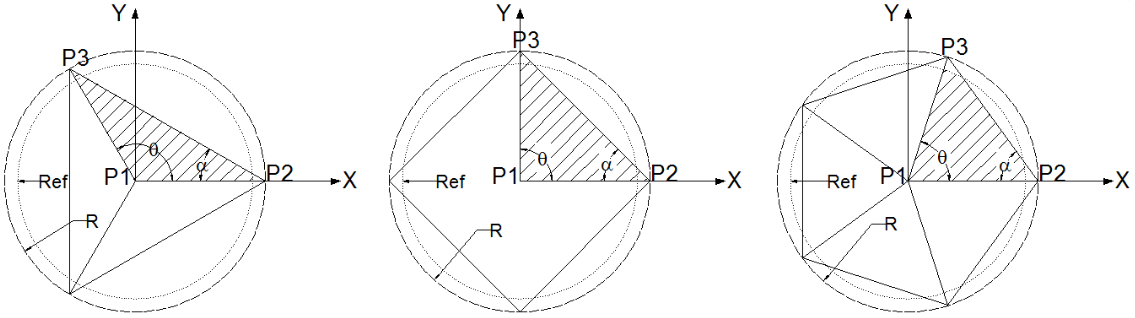

Figure 1: Outline of the pattern found in software development.

2.2. Coordinates of nodal points

For determining the coordinates of the nodal points of the finite element mesh membrane in the xy plane it is necessary to know the discretization mesh pattern. Keeping the standard described above, divide the angle θ in smaller equals angles φ_o. This operation is an angular division, and angle φ_o can be determined by:

$$\varphi_0 = \frac{\theta}{N_a}, \tag{2}$$

where N_a is a number of angular divisions.

Another division of mesh composition is the radial division represented by Nr, which divide the external radius, R in smaller parts of length, r_o as following:

$$r_0 = \frac{R}{N_r}. \tag{3}$$

Other important information to determine the coordinates is the angle, α formed between the segment $(\overline{P1P2})$ and the segment$(\overline{P2P3})$. Another way α is the complement of θ and is written as:

$$\alpha = \frac{180-\theta}{2}. \tag{4}$$

From the information obtained in the equations 1-4 can be determined coordinates (x, y) of each nodal point of the mesh to the first triangle by intersections of lines. These lines are defined in two groups: the lines through the origin with slopes obtained with the φ_o increments, and the lines parallel to the segment$\overline{P2P3}$, i.e. with slope α and linear coefficients defined from increments $r_{o.}$ The equations of these lines are expressed by:

$$\boldsymbol{y_i} = \boldsymbol{r_j} tg\ \alpha - tg\ \alpha\ \boldsymbol{x_i}. \tag{5}$$

$$\boldsymbol{y_i} = tg\boldsymbol{\varphi_k}\boldsymbol{x_i}. \tag{6}$$

where,

$$\boldsymbol{r_j}\ = j \times r_0, \text{com } 1 < j \leq N_r, \tag{7}$$

$$\boldsymbol{\varphi_k} = k \times \varphi_0, \text{com } 0 < k < N_\alpha, \tag{8}$$

with j e k represented by positive integers.

Equating equations (5) and (6), has:

$$\boldsymbol{x_i} = \frac{\boldsymbol{r_j} tg\ \alpha}{(\ tg\boldsymbol{\varphi_k} + \boldsymbol{r_j} tg\ \alpha)}\ . \tag{9}$$

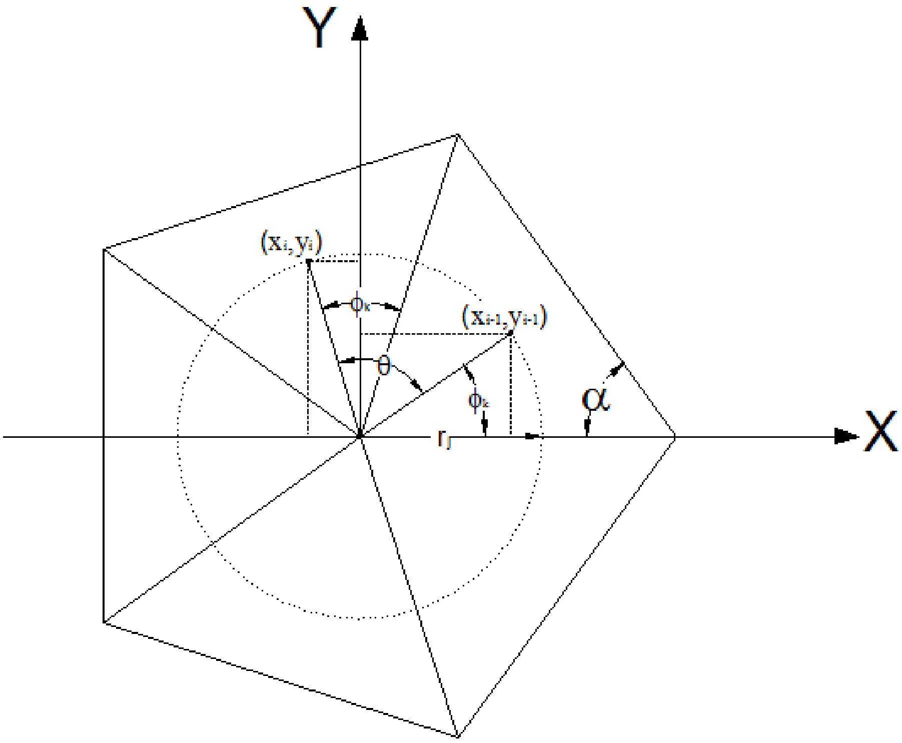

Figure 2: General scheme of rotation of the coordinates.

As θ is always different from zero, then its complement is always different from 90 ° (ninety degrees), so that the above equation (9) is satisfied. But the angleφ_k, can be equal to

90° depending on the number of angular divisions, for this case, x_i should receive the value 0 (zero). However this occurs only for the three-sided polygon, because it is the only case that the internal angle is greater than 90 °.

Substituting the value obtained from (9) in (5) can be obtained values y_i. The others ordered pairs (x_i, y_i) for the first triangle are determined by varying r_j e $\varphi_k$within the respective intervals. For all other points of the polygon, the coordinates are calculated by rotating the point with the same angle φ_k by θ as shown in Figure 2.

Generically,

$$x_i = \left(\sqrt{{x_{i-1}}^2 + {y_{i-1}}^2}\right) \sin(\theta + \varphi_k) = r_j \sin(\theta + \varphi_k). \qquad (10)$$

$$y_i = \left(\sqrt{{x_{i-1}}^2 + {y_{i-1}}^2}\right) cos(\theta + \varphi_k) = r_j cos(\theta + \varphi_k). \qquad (11)$$

Repeating this procedure for the number of faces of the polygon, for all points of the first triangle to determine coordinates of all the nodal points of the mesh of the polygon in the plane. Figures 3 and 4 illustrate the coordinates obtained for some polygons.

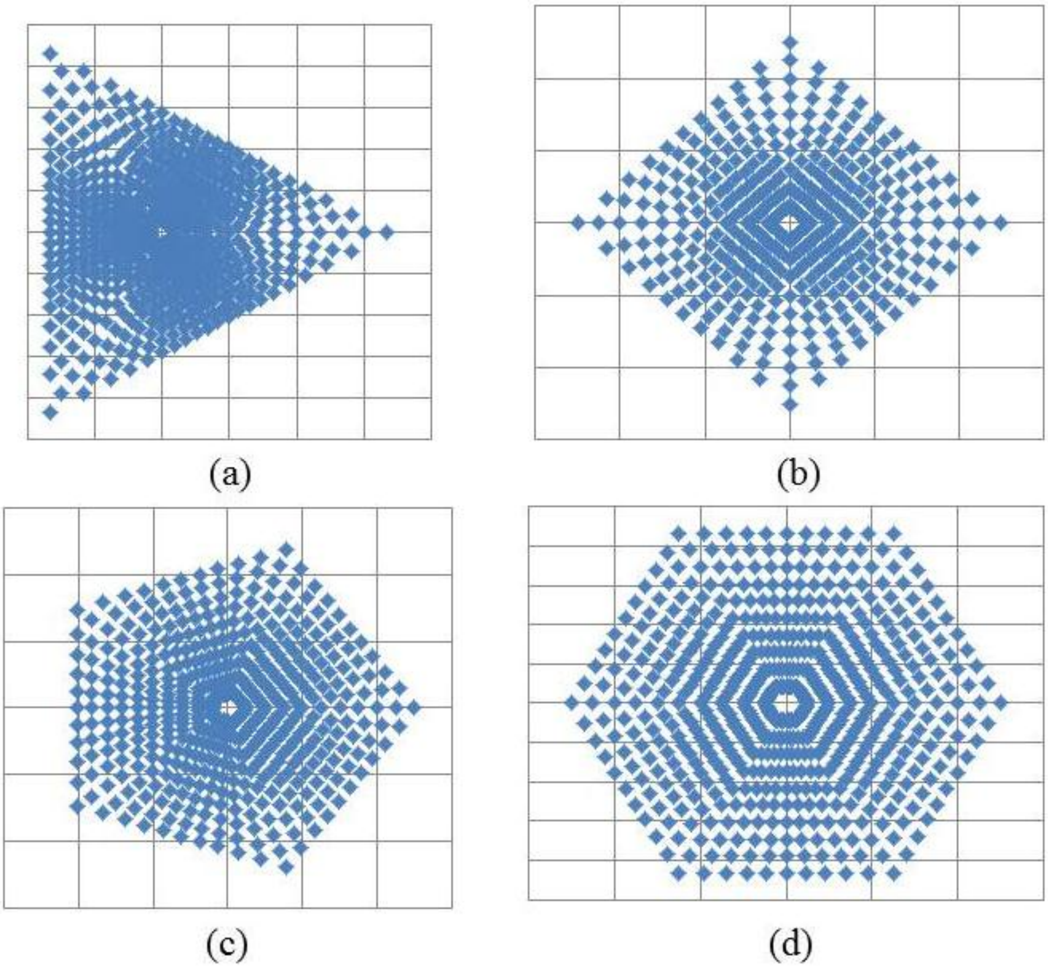

Figure 3: Representation of the coordinates to plants
(a) triangular; (b) quadrilateral; (c) pentagonal and (d) hexagon.

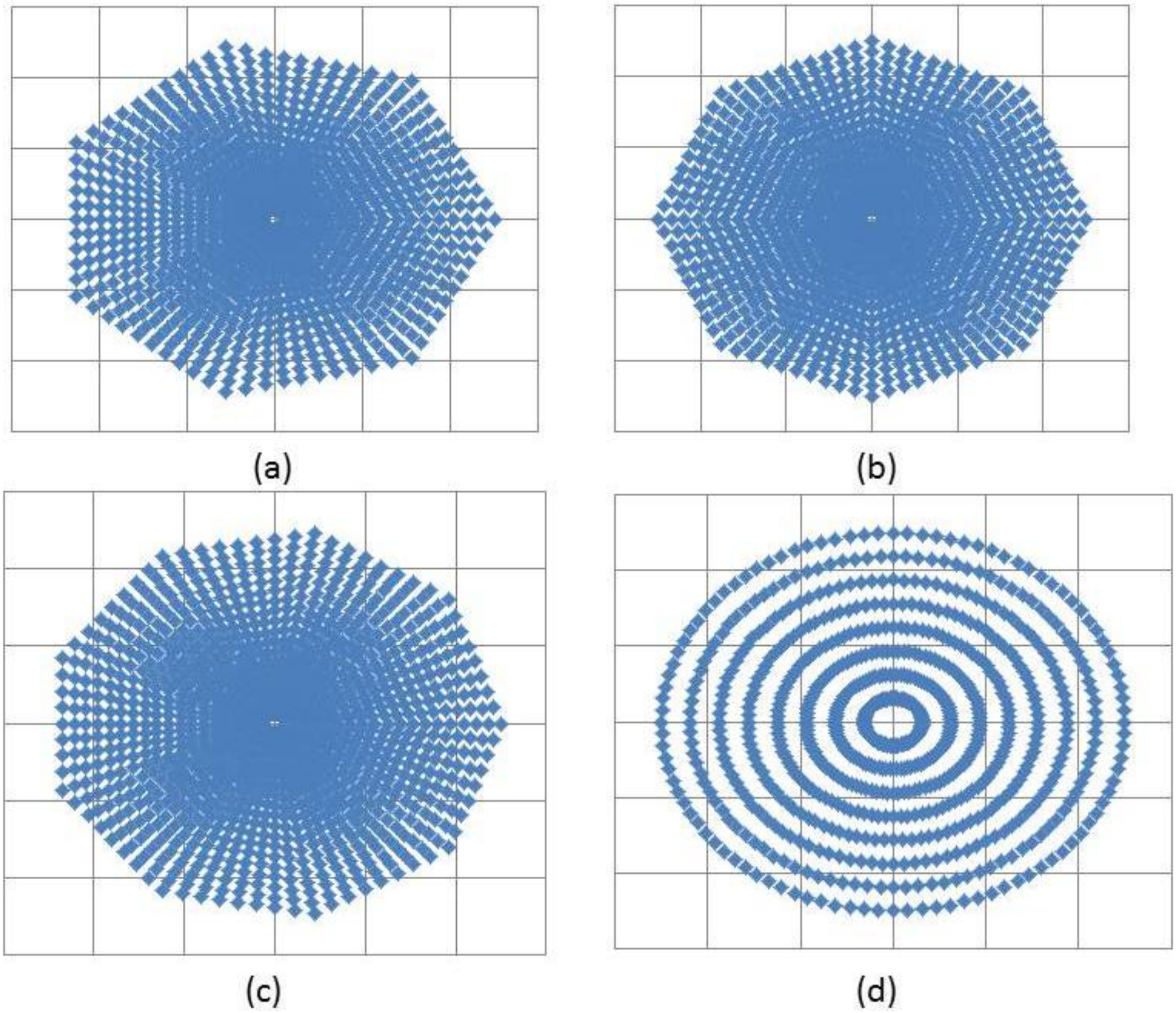

Figure 4: Representation of the coordinates to plants (a) heptagonal; (b) octagonal; (c) enneagonal and (d) hectagonal.

2.3. Automatic numbering of nodal points

The numbering of nodal points starts from the center to the edges in a counterclockwise direction starting at the edge coincident positive x-axis. The exception is the central nodal point will be the last in the numbering to facilitate the algorithm. The identification numbers of each point are saved in a matrixNO_{ij}. The number of columns of the matrix is equal to the number of radial divisions and the number of rows is determined by the product of the angular division by the number of polygon sides, i.e. the matrix stores in the same line the numbers for all nodal points belonging the line segment with slope φ_k, with this, all the nodal points of the first triangle defined as standard base of the polygon are numbered and this process is repeated N times, until all points are numbered. Thus, the same sequence established for calculation of the coordinates is used for numbering of the nodal points so as to simplify the algorithm in software. Maintaining this sequence, determining the nodal incidence of the elements is facilitated, since the matrix of nodal numeration always present the same pattern independently of the type of polygon. The nodal numbering for a small polygon example is shown in Figure 5.

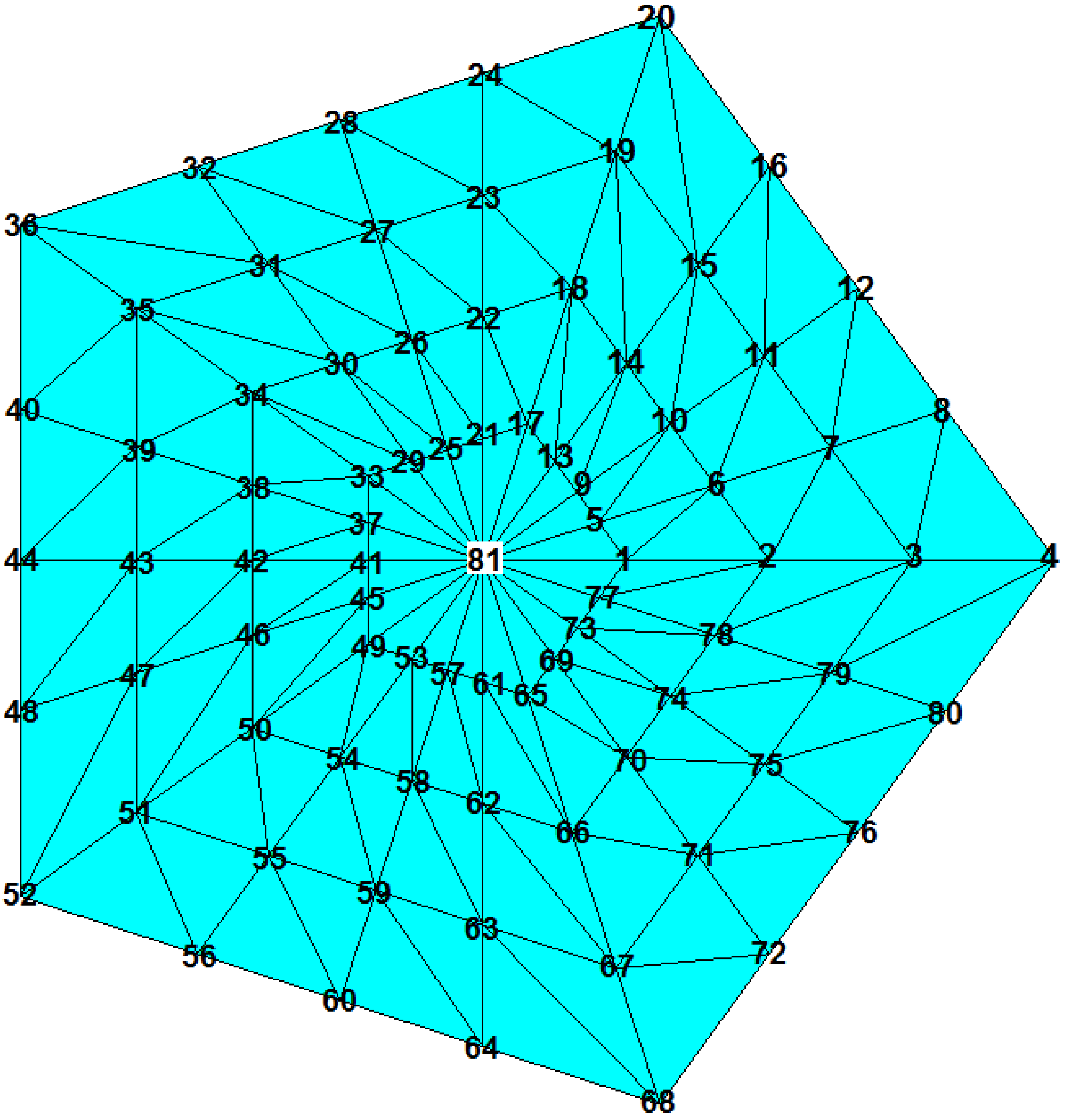

Figure 5: Numbering scheme for the nodal points to pentagonal plant.

The total number of nodal points of the mesh is calculated by:

$$NNOS = N_r \times (N_a \times N) + 1 = col \times lin + 1, \qquad (12)$$

where, *col* and *lin* are respectively the numbers of columns and rows of the matrix of nodal numbering. Any nodal point of matrix NO_{ij} can be determined from:

$$N0_{ij} = i \times N_r + j + 1, \qquad (13)$$

with i and j represented by positive integers, such that:

$$0 \leq i < (Na \times N) e \; 0 \leq j \leq Nr. \tag{14}$$

2.4. Numbering of elements and nodal incidence

Triangular finite elements were adopted for the composition of the mesh. The numbering is done in pairs with the same sequence as the numbering of the nodal points, with the exception of the central elements that are numbered latter. The total number of mesh elements and the number of nodal points is determined by the number of surfaces and divisions radial and between supports and can be calculated by:

$$NEL = 2N_r \times (N \times N_a) - (N \times N_a) = 2col \times lin - lin. \tag{15}$$

The incidence is determined for each element in a counterclockwise direction, as well as the numbering of elements and in pairs. Using the matrix of the nodal pointsNO_{ij}, to determine the incidence of each element, because, as explained above, the matrix will always have the same pattern independently of the polygon and this turns possible to develop only one algorithm capable of determining the incidence of the nodal elements of any configuration. These data are stored in an incidence matrix$INCD_{ij}$. The general form of $INCD_{ij}$ is presented below:

$$\begin{gathered} INCD\;[2i(N_r - 1) + 2j, 1] = NO[i, j]; \\ INCD\;[2i(N_r - 1) + 2j, 2] = NO[i, (j + 1)]; \\ INCD\;[2i(N_r - 1) + 2j, 3] = NO[(i + 1), (j + 1)]; \end{gathered} \tag{16}$$

$$\begin{gathered} INCD\;[2i(N_r - 1) + 2j, 1] = NO[i, j]; \\ INCD\;[2i(N_r - 1) + 2j, 2] = NO[(i + 1), (j + 1)]; \\ INCD\;[2i(N_r - 1) + 2j, 3] = NO[(i + 1), j]. \end{gathered} \tag{17}$$

with $0 < j(N_r - 1) \; e \; 0 < i < N \times N_a$.

The expressions 16 and 17 determine the three nodal points of each element in pairs with the exception of the central elements that have in common the last numbered point (NNOS) and are determined last.

2.5. Support constraints

Since the ultimate objective is to determine the deformed membrane in which are known the coordinates of nodal points and the incidence nodal of the elements, it is necessary to determine the nodal points of support, for not moving while load is applied on the membrane.

The supports are located at the vertices of the polygon; however it is not recommended that only one nodal point is set for support, because the region of support is related to stress concentration. To minimize this effect, the software allows to extend support, considering more than one nodal point fixed per support. This information is another input for the software. To maintain symmetry, it is recommended that the number of fixed points of support is an odd number so that it is regarded the vertex, and the ends immediately preceding and immediately following the edges in counterclockwise. The software finds the wanted nodal points through the matrix of the nodal pointsNO_{ij} and applying restrictions of displacement in all three directions and leaving them free to rotate.

2.6. Mitigation of the vertices of the polygon

The polygons with less side, i.e., with the internal angles smaller, membranes exhibit unwanted shaped support in order to avoid this problem it is defined an effective radius (Ref) that turns rounded the vertex of the polygon.

$$Ref < R. \tag{18}$$

A care must be taken in the radial division, for the program does not generate coordinates outside the effective radius. This care is the smallest division of R cannot be less than or equal to the difference between the rays, because then the software would generate an element outside the projection of the effective radius or with the null area. Then:

$$R - Ref < \frac{R}{N_r}. \tag{19}$$

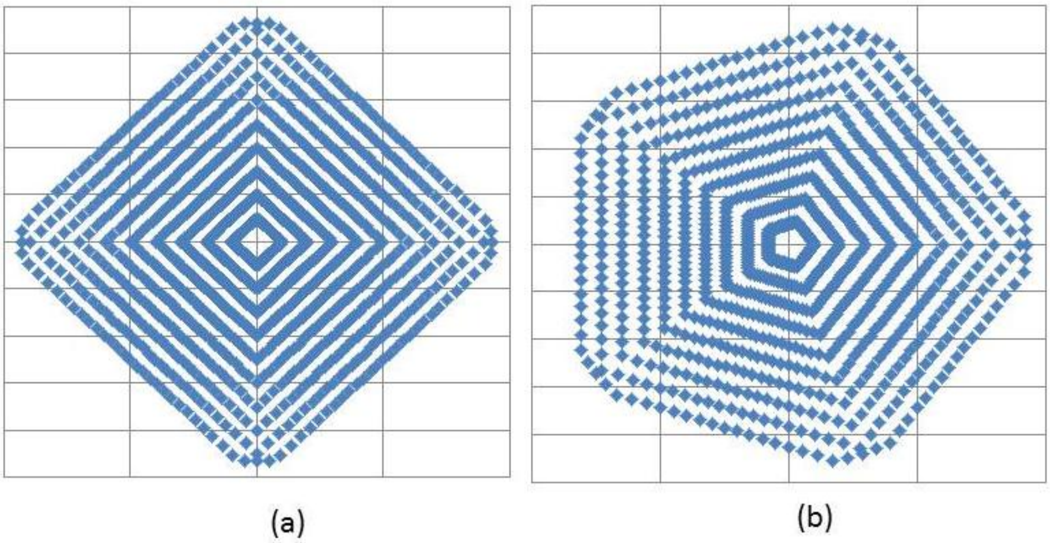

Figure 6: Scheme of mitigated nodal coordinates of vertices (a) quadrangle and (b) pentagonal.

The coordinates near the vertices are recalculated through the intersection of straight lines with slope φ_k with the circular curve of effective radius. Thus, if do not want to consider the effective radius, simply enter the software with external radius equals to effective. As the coordinates calculated on the basis in Ref are smaller than the calculated normally only for the vertices, the software compares the coordinates of Ref x_i to the edge of the polygon with coordinates previously calculated as a function of R, and adopting the smaller in absolute value. Figure 6 shows the smoothing of vertices to plant a quadrangle and a pentagon respectively.

3. COMPUTATIONAL MODEL DEVELOPED BY VIZOTTO

After generation of the membrane polygonal shape in plan, is used in an integrated way the software developed by Vizotto [2010] for generating free-form shell.

The hypothesis adopted for the material is a linear stress-strain relationship and, with Young's modulus E and Poisson's ratio ν, the linear constitutive relationship is:

$$\boldsymbol{\sigma} = \mathbf{D}\,\boldsymbol{\varepsilon}. \qquad (20)$$

with **σ** and **ε** vectors and **D** matrix for the plane state of stress given by:

$$\sigma^t = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{xy} \end{bmatrix}. \qquad (21)$$

$$\varepsilon^t = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{xy} \end{bmatrix}. \qquad (22)$$

$$D = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu) \end{bmatrix}. \qquad (23)$$

The finite element adopted was the CST (Constant Strain Triangle), in which the hypothesis of strain and the stress constant are considered. The element was modified to enable displacements orthogonal to its plane, resulting in an element with nine degrees of freedom, three displacements by nodal point at the vertices of the triangle. The initial configuration of the element is represented in the *xy* plane of the tri-orthogonal system of reference adopted, which, after deformation, occupies a final equilibrium position in the *Oxyz* system. The nodal variables are the displacements of translation u_k (k=1, …, 9) in *x*, *y*, and *z* directions, as shown in Figure.7.

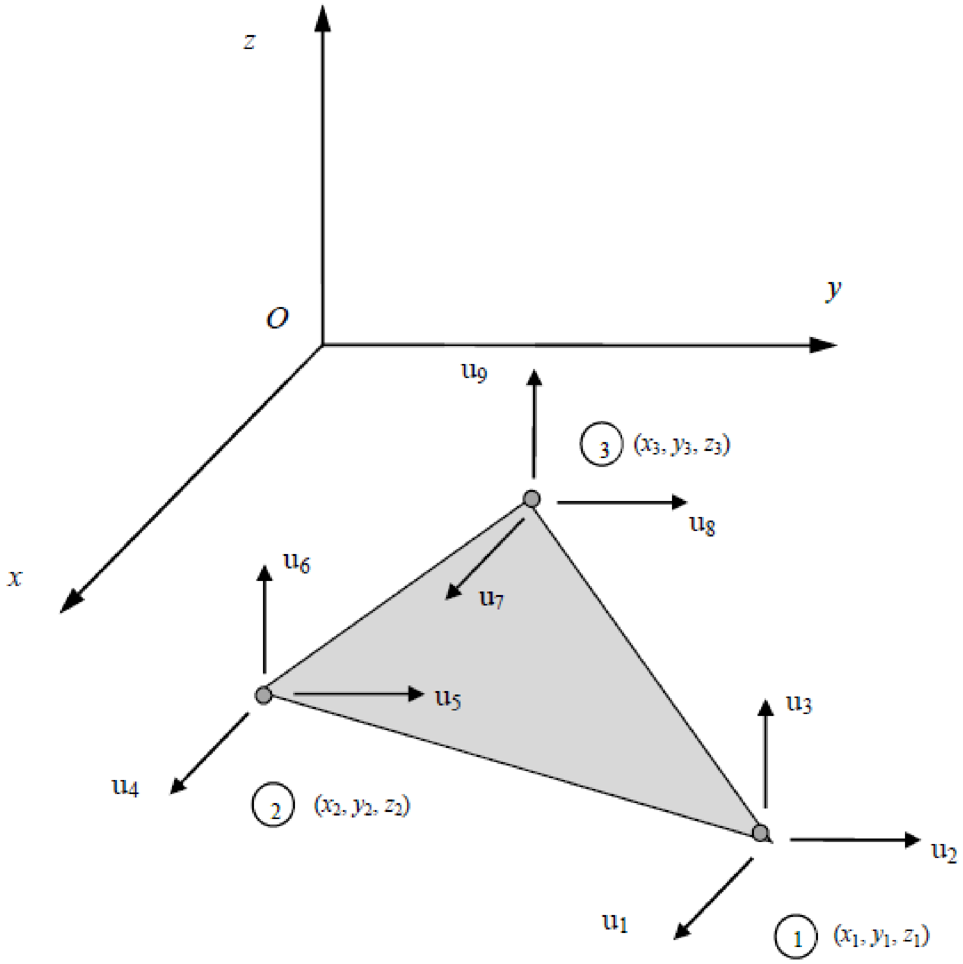

Figure 7: Modified finite element CST (Constant Strain Triangle).

The interpolating functions from using the coordinates *x* and *y* for the displacement field on the domain of the element are given by:

$$\begin{aligned} u_x(x,y) &= \alpha_1 x + \beta_1 y + \gamma_{1;} \\ u_y(x,y) &= \alpha_2 x + \beta_2 y + \gamma_{2;} \\ u_z(x,y) &= \alpha_3 x + \beta_3 y + \gamma_{3.} \end{aligned} \tag{24}$$

with expressions α_k, β_k and γ_k (k=1, 2, 3) written as functions of the initial coordinates and displacements of the nodal points of the element.

Considering the nodal displacements in respective directions *x*, *y*, and *z* for element nodal points, the expressions for α_k, β_k, and γ_k (k=1, 2, 3) can be written as:

$$\begin{aligned} \alpha_k &= c_{11}\, u_k + c_{12}\, u_{k+3} + c_{13}\, u_{k+6,} \\ \beta_k &= c_{21}\, u_k + c_{22}\, u_{k+3} + c_{23}\, u_{k+6,} \\ \gamma_k &= c_{31}\, u_k + c_{32}\, u_{k+3} + c_{33}\, u_{k+6,} \end{aligned} \tag{25}$$

with c_{ij} constants (i=1, 2, 3; j=1, 2, 3) presented in Table 1.

Table 1: Constants cij (i=1, 2, 3; j=1, 2, 3) for the expressions αk , βk and γk (k=1, 2, 3).

$c_{11} = (y_2 - y_3)\ \delta$	$c_{12} = (y_3 - y_1)\ \delta$	$c_{13} = (y_1 - y_2)\ \delta$
$c_{21} = (x_3 - x_2)\ \delta$	$c_{22} = (x_1 - x_3)\ \delta$	$c_{23} = (x_2 - x_1)\ \delta$
$c_{13} = (x_2\, y_3 - x_3\, y_2)\ \delta$	$c_{23} = (x_3\, y_1 - x_1\, y_3)\ \delta$	$c_{33} = (x_1\, y_2 - x_2\, y_1)\ \delta$
$\delta = 1 / [(x_1\, y_2 + x_2\, y_3 + x_3\, y_1) - (x_1\, y_3 + x_2\, y_1 + x_3\, y_2)]$		

With these equations, it is possible to write $u_x(x,y)$, $u_y(x,y)$ and $u_z(x,y)$ as functions of the initial coordinates and displacements at nodal points, as in the interpolating equations adopted initially.

In this computational model, can be considered the actions of self-weight, concentrated forces, uniformly and linearly distributed forces, and uniform pressure. It is also possible to impose known displacements to the membrane to obtain geometric shapes simply by distorting the initial configuration.

The concentrated forces can be applied on the nodal points in *x*, *y* and *z* directions. The simulation effect due to self-weight and forces uniformly distributed can be applied only on the element in a vertical line (direction of *z*-axis). The pressure forces always act perpendicularly to the element surface from initial configuration, in which loading begins to act, through to final equilibrium configuration.

As boundary conditions it is possible to impose any displacements (zero or nonzero) to the finite element nodal points. Thus, for a given nodal point, can be simulated from the full freedom of movement to a complete fixed support, and also impose prescribed displacements (nonzero) to provide various configurations for the equilibrium of structures.

Considering the plane stress state for homogeneous and isotropic materials with quadratic strain definition of Green (in which large displacements and large strains in exact approach are assumed), the strain components can be written as:

$$\varepsilon_{xx} = \left(\frac{\partial u_x}{\partial x}\right) + \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial x}\right)^2 + \left(\frac{\partial u_y}{\partial x}\right)^2 + \left(\frac{\partial u_z}{\partial x}\right)^2\right];$$
$$\varepsilon_{yy} = \left(\frac{\partial u_y}{\partial y}\right) + \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial y}\right)^2 + \left(\frac{\partial u_y}{\partial y}\right)^2 + \left(\frac{\partial u_z}{\partial y}\right)^2\right]; \qquad (26)$$
$$\varepsilon_{xy} = \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial y}\right) + \left(\frac{\partial u_y}{\partial x}\right) + \left(\frac{\partial u_x}{\partial x}\right)\left(\frac{\partial u_x}{\partial y}\right) + \left(\frac{\partial u_y}{\partial x}\right)\left(\frac{\partial u_y}{\partial y}\right) + \left(\frac{\partial u_z}{\partial x}\right)\left(\frac{\partial u_z}{\partial y}\right)\right],$$

which can be written as:

$$
\begin{aligned}
\varepsilon_{xx} &= (\alpha_1) + \frac{1}{2}\left[(\alpha_1)^2 + (\alpha_2)^2 + (\alpha_3)^2\right]; \\
\varepsilon_{yy} &= (\beta_2) + \frac{1}{2}\left[(\beta_1)^2 + (\beta_2)^2 + (\beta_3)^2\right]; \\
\varepsilon_{xy} &= \frac{1}{2}\left(\alpha_2 + \beta_1 + \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3\right).
\end{aligned}
\tag{27}
$$

resulting in:

$$
\begin{aligned}
\varepsilon_{xx} &= (c_{11}\, u_1 + c_{12}\, u_4 + c_{13}\, u_7) + [\, (c_{11}\, u_1 + c_{12}\, u_4 + c_{13}\, u_7)^2 + \\
&+ (c_{11}\, u_2 + c_{12}\, u_5 + c_{13}\, u_8)^2 + (c_{11}\, u_3 + c_{12}\, u_6 + c_{13}\, u_9)^2\,] / 2; \\
\varepsilon_{yy} &= (c_{21}\, u_1 + c_{22}\, u_5 + c_{23}\, u_8) + [\, (c_{21}\, u_1 + c_{22}\, u_4 + c_{23}\, u_7)^2 + \\
&+ (c_{21}\, u_2 + c_{22}\, u_5 + c_{23}\, u_8)^2 + (c_{21}\, u_3 + c_{22}\, u_6 + c_{23}\, u_9)^2\,] / 2; \\
\varepsilon_{xy} &= [\, (c_{11}\, u_2 + c_{12}\, u_5 + c_{13}\, u_8) + (c_{21}\, u_1 + c_{22}\, u_4 + c_{23}\, u_7) + \\
&+ (c_{11}\, u_1 + c_{12}\, u_4 + c_{13}\, u_7)\,(c_{21}\, u_1 + c_{22}\, u_4 + c_{23}\, u_7) + \\
&+ (c_{11}\, u_2 + c_{12}\, u_5 + c_{13}\, u_8)\,(c_{21}\, u_2 + c_{22}\, u_5 + c_{23}\, u_8) + \\
&+ (c_{11}\, u_3 + c_{12}\, u_6 + c_{13}\, u_9)\,(c_{21}\, u_3 + c_{22}\, u_6 + c_{23}\, u_9)\,] / 2.
\end{aligned}
\tag{28}
$$

Defining V as the element undeformed volume and considering the element to have constant thickness, the potential strain energy $\pi(u)$ can be written as:

$$
\pi(u) = \frac{1}{2}\int_V \left(\sigma^t \varepsilon\right) dV = \frac{1}{2}\int_V \, (\varepsilon^t D \varepsilon)\; dV = \frac{1}{2} V\left(\varepsilon^t D \varepsilon\right). \tag{29}
$$

The Gradient of potential strain energy, known as the internal forces vector, and the Hessian of potential strain energy, known as the stiffness matrix connected to the internal forces vector, can be computed with vector component derivatives helping in the calculation of the nodal displacements. Thus:

$$
G(\pi) = \frac{\partial \pi}{\partial u_i} = V\sigma^t\left(\frac{\partial \varepsilon}{\partial u_i}\right). \tag{30}
$$

$$
H(\pi) = \frac{\partial^2 \pi}{\partial u_i \partial u_j} = V\left[\left(\frac{\partial \varepsilon^t}{\partial u_i}\right) D \left(\frac{\partial \varepsilon}{\partial u_j}\right) + \sigma^t\left(\frac{\partial^2 \varepsilon}{\partial u_i \partial u_j}\right)\right]. \tag{31}
$$

Under the principle of minimum total potential energy, steady equilibrium positions correspond to the local minimum points of the total potential energy function. By applying mathematical techniques to optimize nonlinear functions, it is possible to determine such local minimum points.

The model considered here presents two distinct situations. The first relates to structures with total potential energy known. The second relates to structures without total potential energy.

When the total potential energy function exists, the stable equilibrium configurations correspond to local minimum points of this function. It is advisable to use a quasi-Newton-type method to find these local minima. When the total potential energy function does not exist, it is advisable to use an incremental Newton-Raphson-type method to find nonlinear system solutions for equations given by the equilibrium equations:

This process of optimized form finding results directly in numerical data derived from the generated shape, which can be further processed in a structural analysis.

4. RESULTS

The results obtained with the software developed are the finite element meshes of flat membranes for regular polygons. The software generator of membrane appears efficient to be able to generate any polygonal, however the description of the mesh should be improved so that elements close to the origin of the coordinate system are not adopted with flattened shapes.

Integration with the software developed by Vizotto for generating free-form shells enables the analysis of the membranes to generate the three-dimensional shapes. The proof of these results can be verified through the images of the membranes in the plan and its respective spatial form. Were generated membranes with plants triangular, quadrangle, pentagonal, hexagonal, heptagonal, octagonal, enneagonal, decagonal, hectagonal which are presented respectively in Figures 8-16.

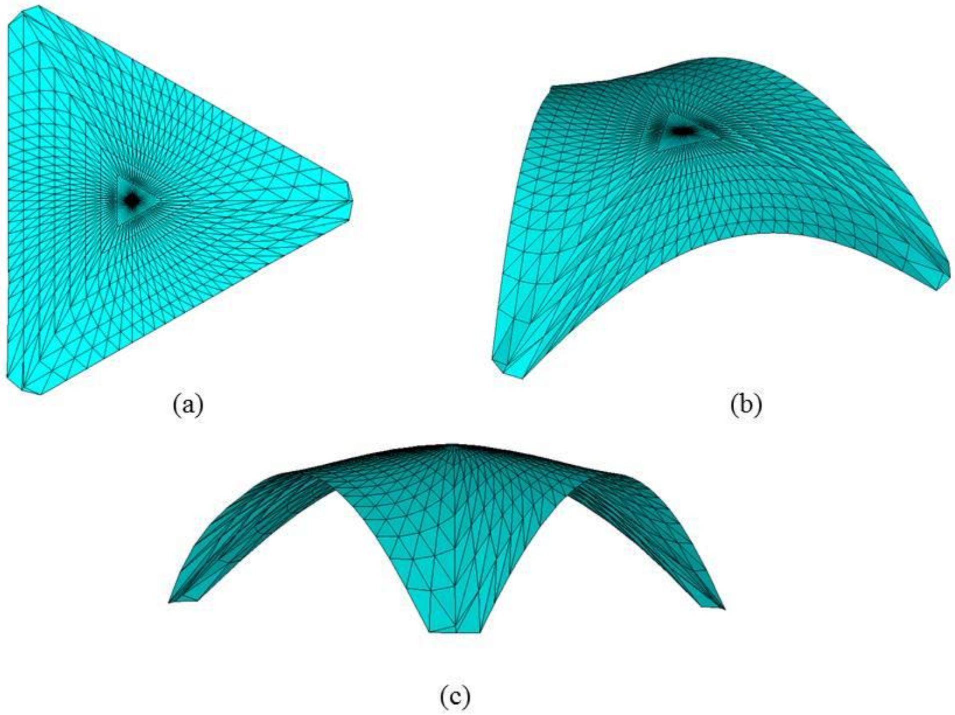

Figure 8: (a) Representation in the plane of the triangular membrane; (b) three dimensional view of the shell; (c) Front view of shell with triangular plant.

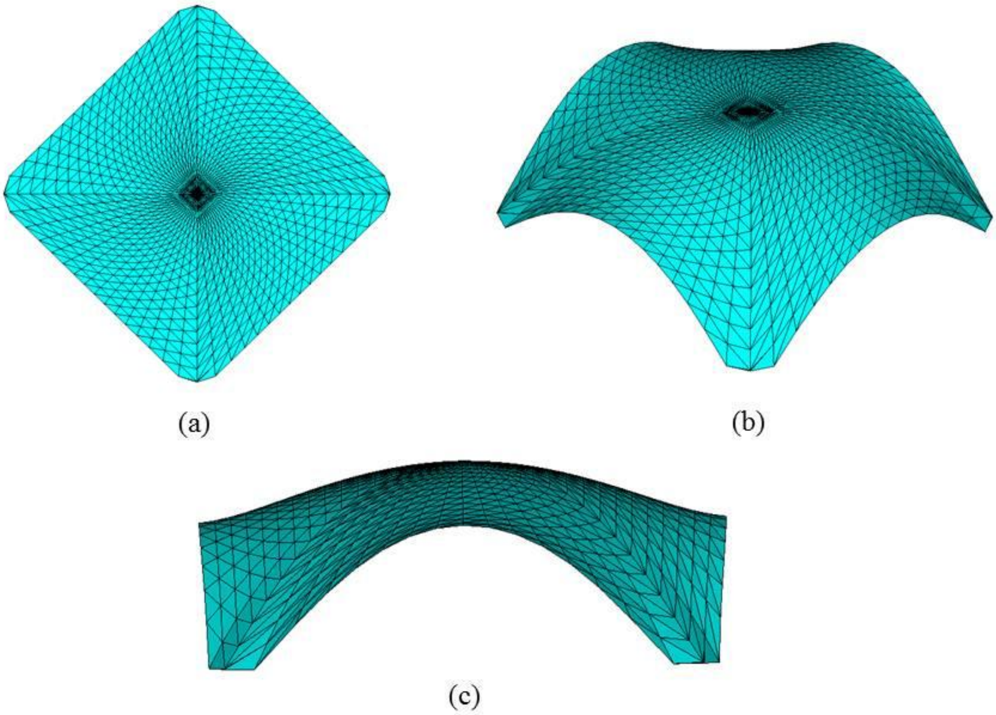

Figure 9: (a) Representation in the plane of the quadrangle membrane; (b) three dimensional view of the shell; (c) Front view of shell with quadrangle plant.

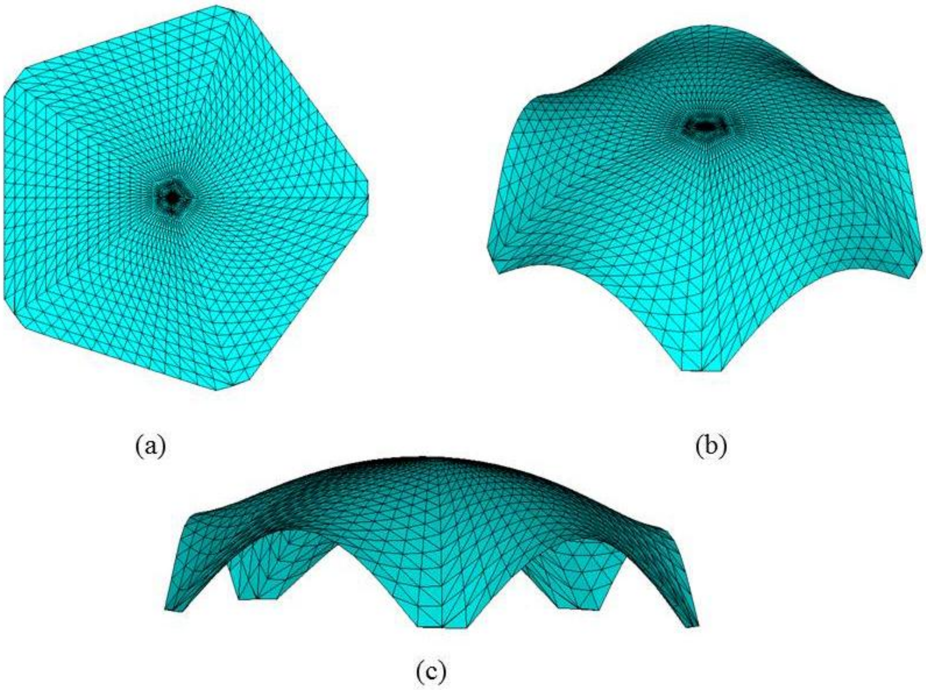

Figure 10: (a) Representation in the plane of the pentagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with pentagonal plant.

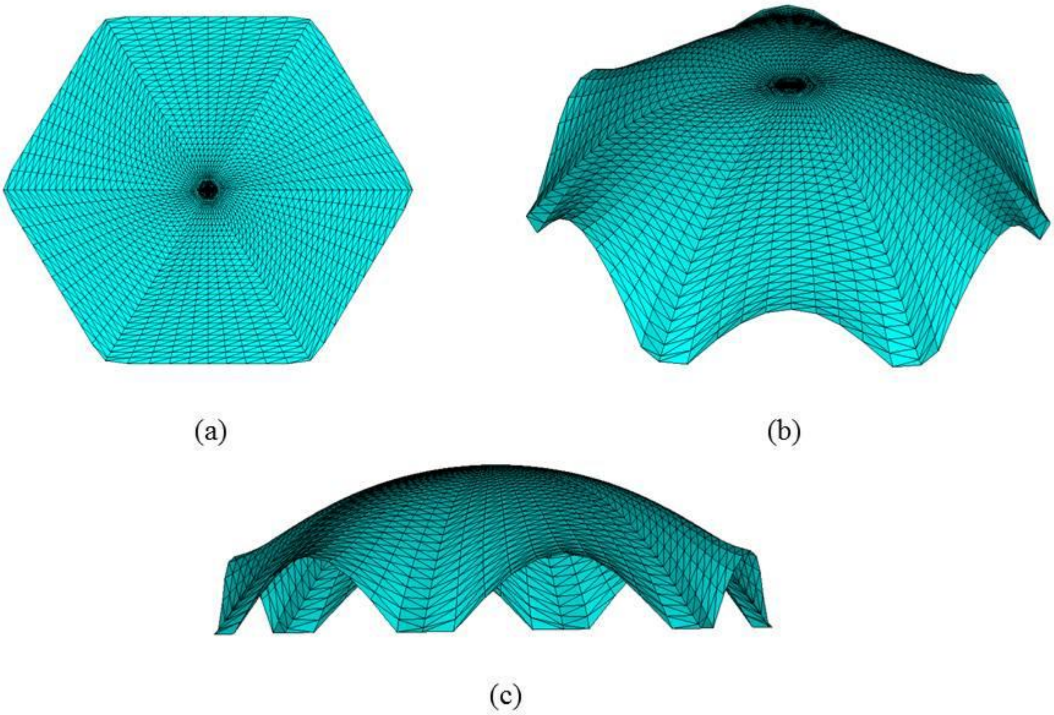

Figure 11: (a) Representation in the plane of the hexagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with hexagonal plant.

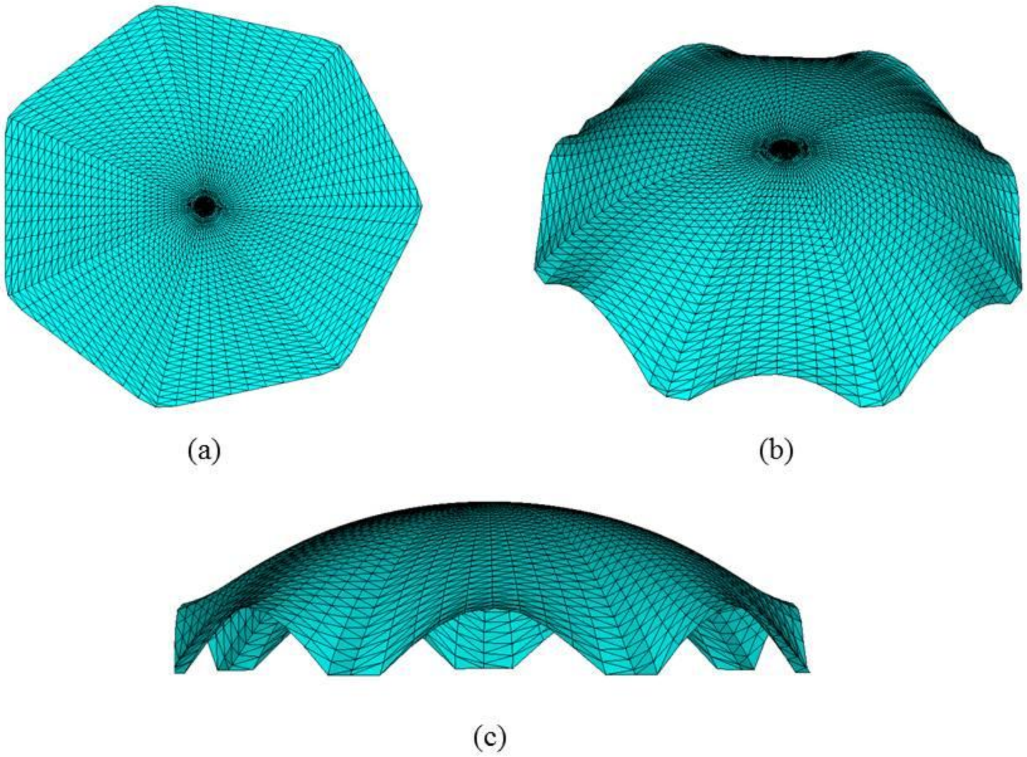

Figure 12: (a) Representation in the plane of the heptagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with heptagonal plant.

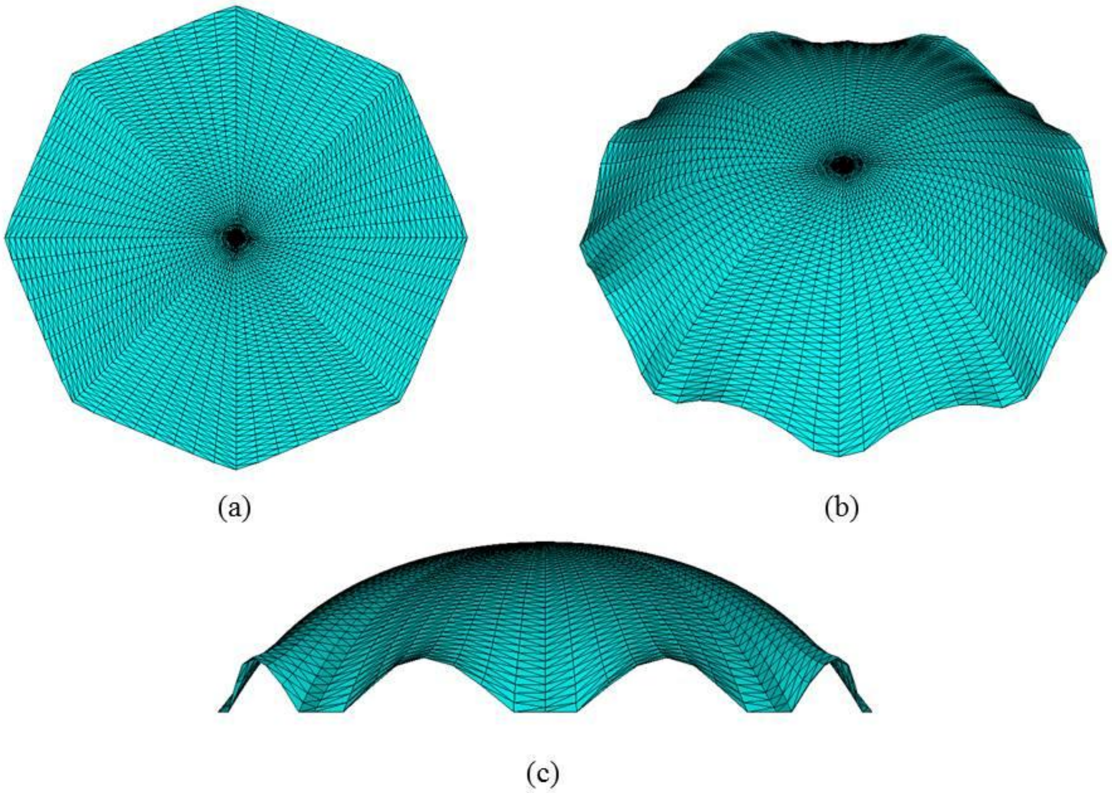

Figure 13: (a) Representation in the plane of the octagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with octagonal plant.

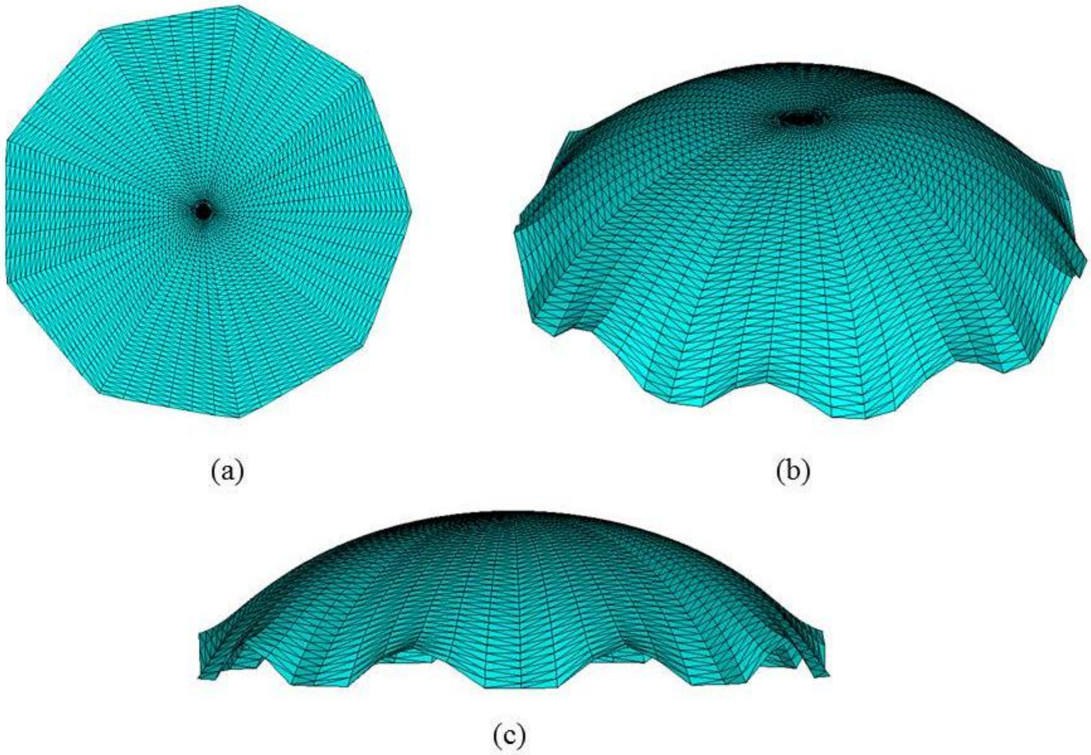

Figure 14: (a) Representation in the plane of the enneagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with enneagonal plant.

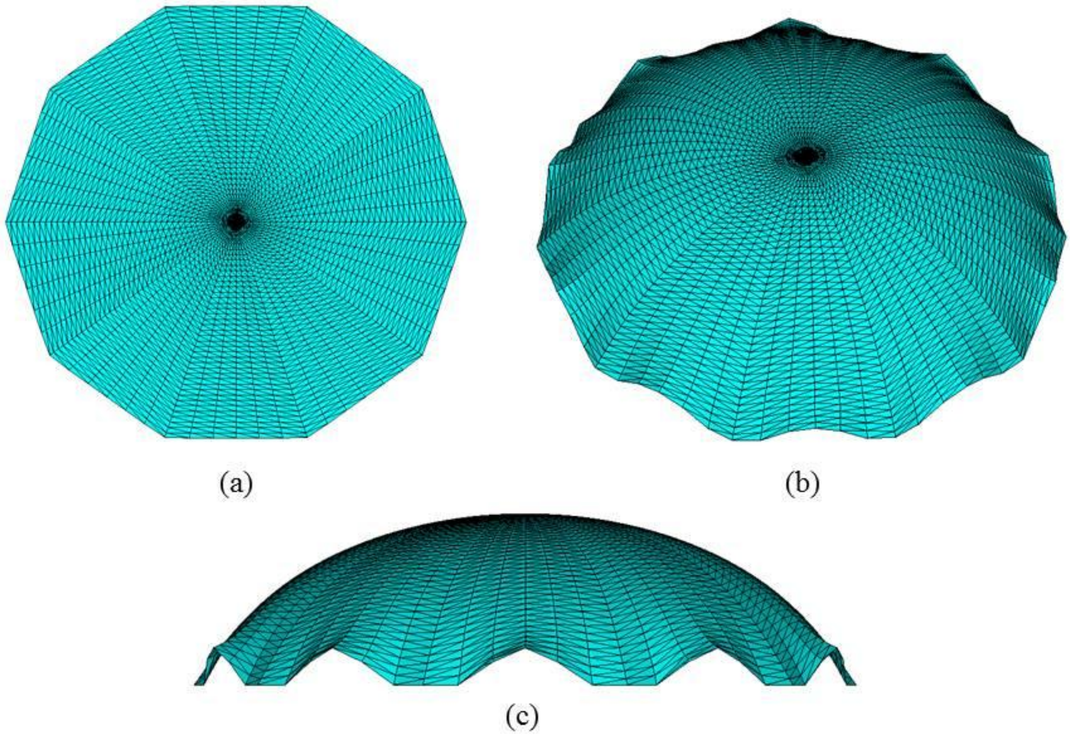

Figure 15: (a) Representation in the plane of the decagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with decagonal plant.

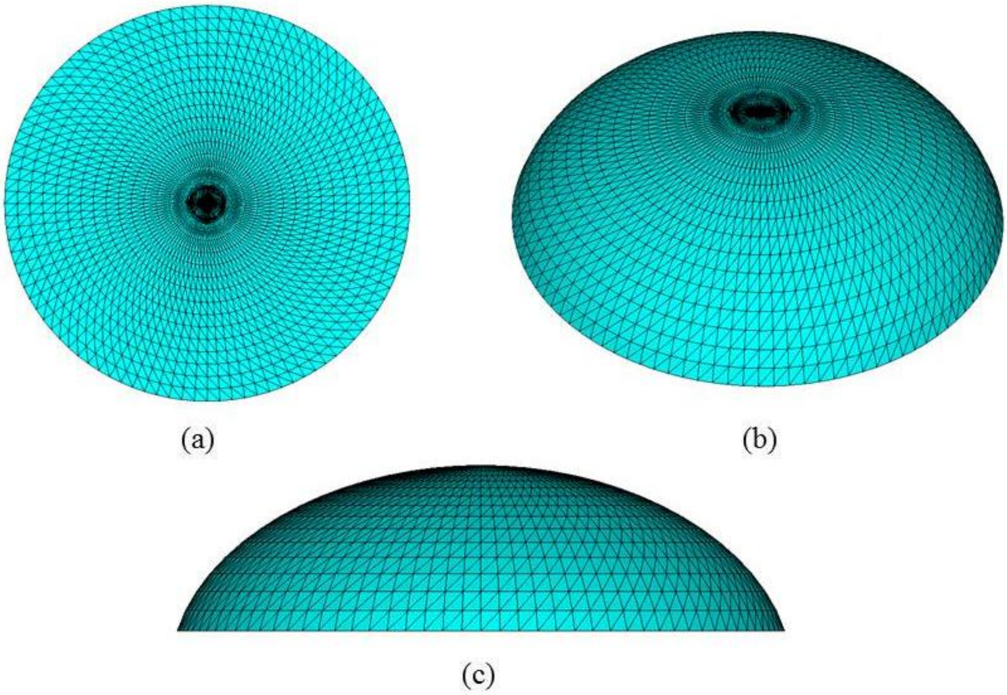

Figure 16: (a) Representation in the plane of the hectagonal membrane; (b) three dimensional view of the shell; (c) Front view of shell with hectagonal plant.

5. CONCLUSIONS

With developed software it is possible the automatically generation of meshes of finite elements with polygonal plants and integrated with software developed by Vizotto to generation of free form shells. The shells structures with polygonal plants can be used as coverage for many architectural projects with different purposes.

The standard to calculate coordinates and numbering the nodes and elements developed in this paper was comproved with presented results. Although, it is proposed for future work, studies for improving the mesh for generating best elements in the center and near the supports.

6. REFERENCES

[1] Vizotto I., “Computational generation of free-form shells in architectural design and civil engineering”. *Automat Constr*. 19(8):1087 –1105, 2010.