Approximation of Transitional Developable Surfaces between Plane Curve and Polygon

Ratko Obradović[1], Branislav Beljin[1], Branislav Popkonstantinović[2]

[1] University of Novi Sad, Faculty of Technical Sciences, Trg D. Obradovića 6, Novi Sad, Serbia, obrad_r@uns.ac.rs; beljin@uns.ac.rs

[2] University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, Belgrade, Serbia, bariton@afrodita.rcub.bg.ac.rs

Abstract: The paper describes and explains the computational geometry method that simulates the physical model of transitional pipeline parts whose cross sections are plane curve and polygon and are made of materials that cannot be wrinkled or stretched. This geometrical concept is taken from the classical line geometry and can be directly applied to the creation of computer algorithm that generates the transitional surfaces between plane curve and polygon lying in two planes. Mathematical concept of the solution for this problem is based on the approximation of the boundary surface triangulation and is presented in MATLAB program. The solutions are given in the form of three-dimensional surface models from which the necessary technical documentation can quickly be made using standard CAD/CAM tools thus shortening the time necessary to prepare the construction and production. In particular, the paper presents the basic computer graphics theory of ruled and developable surfaces, the concept of a geometrical solution for the problem, the algorithm for generation of transitional surface between plane curve and polygon, and MATLAB program which is useful for many engineering purposes.

Keywords: Transitional Developable Surfaces; Computational Geometry; CAD

1 Introduction

The zero Gaussian curvature is the basic feature of developable surfaces, which means that they can be flattened into a plane without distortion (stretching, compressing or tearing). It is evident and well known that these surfaces belong to the ruled surfaces group. Developable surfaces are applied in various branches of engineering and design wherever the construction of a form is based on thin-wall materials that cannot be wrinkled or stretched. However, the proper usage of these surfaces, especially the method for their correct flattening into a plain, can rarely be seen in CAD/CAM application since the user group of developable surfaces is rather diminutive and exclusive. Often, applications which approximate the surface development are designed conveniently for specific technical problems, needs or purposes. Examples of computer graphics problems that require approximate surfaces development can be found in shipbuilding (ship hull formed of surface parts that are flattened into a plain), textile and leather industry (production of shoes with folds at the top), production of sheet metal parts by stamping, fabrication of large transitional elements or thin wall materials and pipelines as well as in architecture (roof structure, special design of buildings), etc. The approximation of the developable surfaces is usually obtained by special computer algorithms based on certain mathematical and geometric techniques.

The problem of surface development represents important and venerable subject of research in various branches of computational geometry and computer graphics. Thus, Kuntal S. et al. [8] proposed a novel optimization methodology of material blending for heterogeneous object modeling. The presented method introduced the point-to-point correspondence represented by a set of connecting lines between two material directrices. By subdividing the connecting lines into an equal number of segments, a series of intermediate piecewise curves is generated to represent the material transformation between the governing material-features. Moreover, Kuntal S. et al. also presented in [8] an elucidative examples and computer interface implementation by which the non-self-intersecting and non-twisted connecting surfaces can be generated. In [5], Dahl H. E. I. disclosed that the continuity properties of a canal surface are inherited from the continuity properties of the associated curve. An algorithm for minimal bi-degree rational parametrizations of patches on canal surfaces is described, and explanation how this can be used to parameterize piecewise rational corner and edge blends is also presented. Li J. and Lu G. [9] proposed a novel approach that decouples the high coupling between 3D garment design and 2D pattern making. 3D new garment models are created on individual human models by compositing 3D parts directly from garment examples rather than 2D patterns. The locally parameterized 3D garment parts and the newly created garment models can be interactively edited by altering the parametric curves on the garment prototypes. The authors [9] emphasized that the main benefit of this approach is that it is easy to operate and capable to create elegant individualized garments with highly detailed geometries. Casale A. et al. [4] exposed a digital approach in design, parametric modeling and prototyping, with roots in descriptive geometry. They refocus attention on the study of geometry, in particular on ruled surfaces, their possible discrete forms and/or development in the plane. The critical analysis of the geometric shape, its virtual and parametric representation, as well as transformation into a physical object are exposed. Li C. at al. presented and explained in [10] a new method to construct a developable surface which possesses a given curve as the line of its curvature. The authors analyzed the necessary and sufficient conditions when the resulting developable surface is a cylinder, a cone or a tangent surface and illustrated the convenience and efficiency of this method using some characteristic examples. In [11], Liu Y. at al. proposed a dynamic programming method for the design task of interpolating 3D boundary curves with a discrete developable surface. This simple and effective method can be applied in direct industrial design using developable surfaces by interpolating descriptive curves. Rose K. et al. [19] simplified the modeling process by introducing an intuitive sketch-based approach for modeling developable surfaces. They developed an algorithm which generates a smooth discrete developable surface that interpolates a given arbitrary 3D polyline boundary, constructed by using a sketching interface. Moreover, the authors demonstrated the effectiveness of the exposed method through a series of examples, from architectural design to garments. Tang and Wang [21], describe the method for modeling developable folds on strip as a common operation in clothing and shoe design. Borrowing the classical boundary triangulation concept from descriptive geometry, they describe a computer-based method that automatically generates a specialized boundary triangulation approximation of a developable surface that interpolates a given strip. The development is accomplished by geometrical simulation of the sheet folding process as it would occur when the strip is rolled from one end to another. Frey W. H. [7] reviews the properties of buckled developable surfaces showing how to approximate buckled binder wrap surfaces by developable three-dimensional triangulations and discusses the insights gained from specific examples. Peternell M. [15] investigates the recognition and reconstruction of some geometrical objects from a set of points as measurements from a developable surface. In particular, he investigates the set of estimated tangent planes of the data points and shows that classical Laguerre geometry is a useful tool for recognition, classification and reconstruction of developable surfaces. In the conclusion he indicates that these surfaces can be generated as envelopes of a one-parameter tangent plane family. Frey W. H. [6] describes a computer-based method that generates a boundary triangulation by geometrical simulation of the sheet bending as it would occur during closing of the blank-holder in a sheet metal forming problem. Actually, this author exposes and explains the automated method for the proper boundary triangulation, which was previously the job of a draftsman. Pottmann and Wallner [16] analyze and present the approximation algorithm for developable surfaces. In principal, they introduce an appropriate metric in the dual space and derive linear approximation algorithms for developable NURBS surfaces, including multiscale approximation. Obradović and Beljin [12] describe geometrical models in three different cases of transitional developable surfaces: transition between two polygons, transition between a plane curve and a polygon and between two plane curves. Obradović, Popkonstantinović and Beljin [14] describe and explain the computational geometry method that simulates the physical model of transitional pipeline parts whose cross sections have polygonal form and are made of materials that cannot be wrinkled or stretched. The paper presents the basic computer graphics theory of ruled and developable surfaces, the concept of geometrical solution for the generation of transitional surface between two polygons, and the algorithm by which this problem can be implemented and thus automatically solved. Bo and Wang [2] used a geodesic curve to control the shape of developable surfaces. They presented an effective method for modeling a developable surface to simulate paper bending in interactive and animation applications. Xu G. [22] made an optimal spherical triangulation.

The present paper discloses and explains the geometrical method and algorithmic procedure that simulates the concrete model of transitional pipeline parts with polygonal and planed curve cross section, which are made of materials that cannot be contracted, elongated or distorted. More explicitly, the basic computer graphics theory of ruled and developable surfaces is explained, the concept of geometrical solution for the generation of transitional surface between a plane curve and a polygon is formed, and the algorithm by which this problem can be automatically solved is described. The solutions are given in the form of three-dimensional surface models from which the necessary technical documentation can be made by standard CAD/CAM tools thus shortening the time necessary to prepare construction and production.

2 Geometrical Concept of Developable Surfaces Forming between Plane Curve and Polygon

The transition between two pipes which have different cross sections is a common problem that can be encountered in mechanical engineering, civil engineering, architecture and design. The examples are low pressure pipes which are made from thin-wall material such as tin, plastics or glass (Fig. 1).

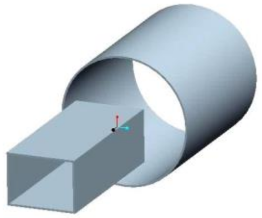


Figure 1
Cylindrical and polygonal pipe connecting problem

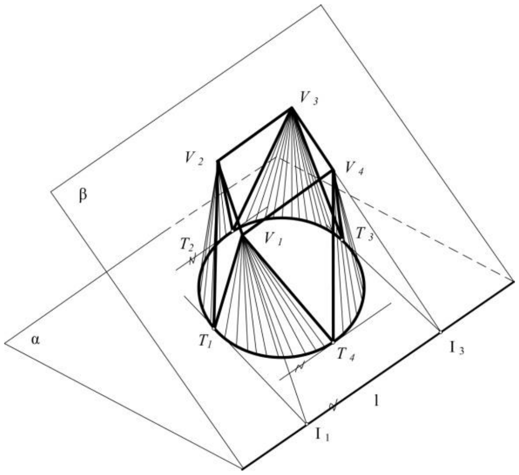


Figure 2
3D View of Developable Surfaces, transition from plane curve to polygon

where α and β are basis planes;V_1,V_2,V_3,V_4 – vertex of polygonal cross section; T_1,T_2,T_3,T_4 – tangent points to plane curve.

We would like the transition part between polygonal and cylindrical pipe to be a developable surface. In this case the developable surface is created from triangles and cones and Fig. 2 shows a 3D solution of geometrical concept of this problem.

The basic rule for connecting two pipes is that the used triangles must actually be tangent planes for contiguous cones. Triangle $V_1V_2T_1$ should be a part of tangent plane to cone $V_1T_1T_4$ (V_1 is a cone vertex) and cone $V_2T_1T_2$ (V_2 is a cone vertex, Fig. 2 and Fig. 3). In other words, if we choose point T_1 arbitrarily on a plane curve, in that case plane $V_1V_2T_1$ will intersect contiguous cones $V_1T_1T_4$ and $V_2T_1T_2$, it will intersect each cone by two generatrices. That is not a smooth transition from triangle to cones. What we want is to create just smooth transition, i.e. the used triangle $V_1V_2T_1$ must be tangent planes for contiguous cones.

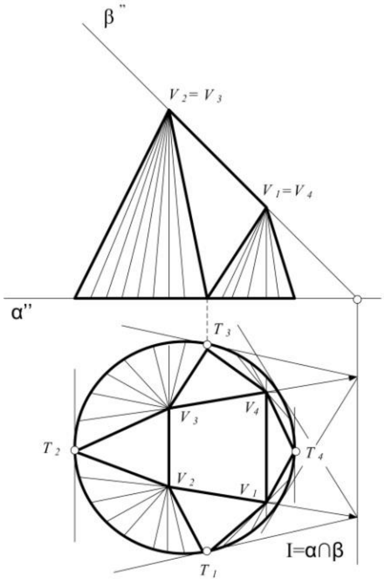
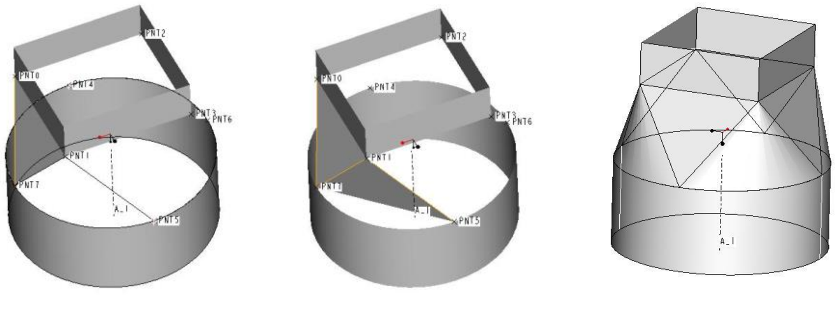


Figure 3
Developable surfaces between circle and polygon, two orthogonal projections

2.1 Creating of Developable Transitional Surfaces between Plane Curve and Polygon in Computer Graphics

Ruled surfaces are often used in present graphic software and in computer graphics theory. However, developable surfaces are not standard surfaces. Initial data for creating developable surfaces are cones patches (four patches in Figs. 3 and 4). Descriptive geometric approach offers an idea on how to solve this problem and how to create developable surfaces.

In Fig. 4a the first triangle patch is correctly created between the polygon edge and the tangent points on the basis circle. Fig. 4b shows that graphical software chooses again a triangle as the second patch. In that case there is a hole next to the second triangle (between the second triangle and the basis circle of the cylinder) and if we use this transition for fluid flow, there will be leak from the hole. Better solution is to use part of cone instead of the second triangle and to create developable surfaces as combination of triangles and cones as is shown in Fig. 4c. However, this solution is not created automatically by graphical software. In the next chapter we will present our algorithm for constructing transitional developable surfaces.



(a) (b) (c)

Figure 4

First triangle patch (a), automatic choosing of second triangle (b), developable surfaces as triangles and cones combination (c)

3 Algorithm for constructing Transitional Developable Surfaces

The algorithm is based on searching tangent points T_1,..., T_4 to the plane curve according to given polygon. There are three cases: the first one is when the basis planes α and β are parallel (Fig. 5), the second one is when the planes α and β are intersecting (Fig. 6) and the third is a special case when these intersecting planes are orthogonal (Fig. 7). Input data for calculation are vectors which represent point coordinates in global coordinate system. These vectors are in matrix form and the matrix dimension is 1×2. Output data is a display device of transitional developable surface.

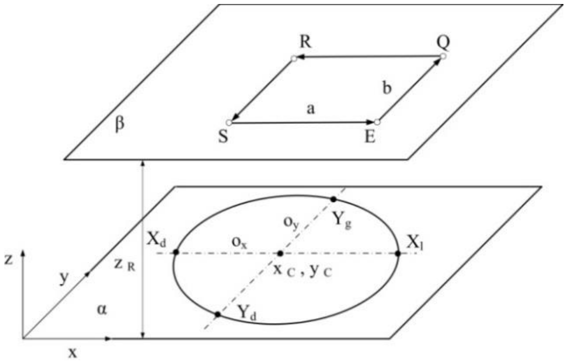


Figure 5
Basis planes are parallel

$\mathbf{x_C}$, $\mathbf{y_C}$ – coordinates of plane curve center (in this case a circle, or an ellipse);
r – circle radius; $\mathbf{r_x}$, $\mathbf{r_y}$ – semi-axis for ellipse;
$\mathbf{z_R}$ – distance between parallel planes (Fig. 5);
theta – angle between planes *α* and *β* (Fig. 6 and Fig. 7);
l – intersecting line between planes *α* and *β* (Fig. 6 and Fig. 7);
$\mathbf{I_a}$, $\mathbf{I_b}$, $\mathbf{I_c}$, $\mathbf{I_d}$ – points on intersecting line ***l*** (Fig. 6 and Fig. 7), between polygons edges *a*, *b*, *c*, *d* respectively and line *l*;
$\mathbf{x_S}$,$\mathbf{y_S}$ – polygons start point; $\mathbf{x_E}$,$\mathbf{y_E}$ – observed line end point;
$\mathbf{x_Q}$,$\mathbf{y_Q}$ – first control point; $\mathbf{x_R}$,$\mathbf{y_R}$ – second control point;
$\mathbf{z_S}$,$\mathbf{z_E}$,$\mathbf{z_Q}$,$\mathbf{z_R}$ – **z** coordinates for *theta=90°*(Fig. 7).

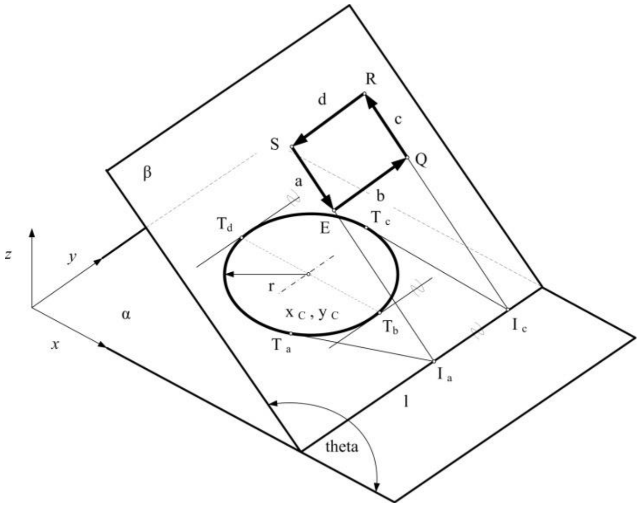


Figure 6
Basis planes are intersecting

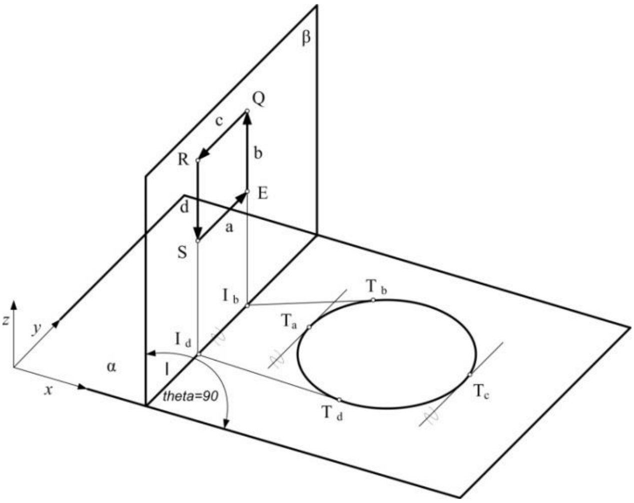


Figure 7
Basis planes are orthogonal

3.1 Tangent Points Determination in Case When Basis Planes are Parallel

First we choose a polygon edge. Planes *α* and *β* are parallel and now we need to compare the parallelism of edges *a* and *b* with axis *x* and *y* (Fig. 5).
Polygon edge could be parallel with one axis or these are skew lines.

Edge *a* is presented by parametric equations: $a = S + U_a \cdot (E - S)$, $S(x_S, y_S), E(x_E, y_E)$

axes are $o_x = X_d + U_x \cdot (X_l - X_d),\ o_y = Y_g + U_y \cdot (Y_d - Y_g)$ (1)

where: U_a, U_x and U_y are equations parameters, *(E-S)*, $(X_l\text{-}X_d)$, $(Y_d\text{-}Y_g)$ i.e. (dx_a, dy_a), (dx_{ox}, dy_{ox}), (dx_{oy}, dy_{oy}) are line vectors. If *a* is parallel to o_x than we have two equations with two variables:

$$x_S + U_a\, dx_a = x_d + U_x\, dx_{ox},\ y_S + U_a\, dy_a = y_d + U_x\, dy_{ox} \tag{2}$$

From these two equations we could find U_a and U_x:

$$U_a = \frac{dy_{ox}(x_d - x_S) - dx_{ox}(y_d - y_S)}{\mathrm{dx_a\, dy_{ox} - dy_a\, dx_{ox}}} \tag{3}$$

$$U_x = \frac{dy_a(x_d - x_S) - dx_a(y_d - y_S)}{\mathrm{dx_a\, dy_{ox} - dy_a\, dx_{ox}}} \tag{4}$$

The interesting part of equations (3) and (4) is the denominator, which is equal in both equations. If the value of the denominator is equal to zero that means that the observed polygon's edge is parallel to *x* axis. The same analysis can be applied to *y* axis.

In Fig. 8 edge *a* is parallel to *y* – axis, the corresponding tangent points of edge *a* could be T_{P2} or T_{P4}. In the case when edge *a* is parallel to *x* axis, tangent points could be T_{P3} or T_{P1}. Reference tangent point which the program will use for triangle forming must be a point which is external for the polygon. The reason for choosing the external point is that we do not want to close the transition from polygon to circle with the triangle. In Fig. 8 it is point T_{P2} for edge *a*. The program determines reference point by comparing coordinate values according to edge position and according to axis position, and in this case:

$a \| y$ => if $x_Q > x_E$ => then $T_P = T_{P4}$ else $T_P = T_{P2}$
$\boldsymbol{a} \| \boldsymbol{x}$ => if $y_Q > y_E$ => then $T_P = T_{P1}$ else $T_P = T_{P3}$ (5)

In the case when polygon edge is not parallel to any axis we will use the following procedure: in Fig. 9 we can see that we created a normal line to polygon edge *a* through the circle (ellipse) center. The normal line intersects curve (Fig. 9) at tangent lines points T_{1a} and T_{2a}. There are two tangent points for each normal line; one of them is a real point and that is the point which is external for the polygon (we do not want to close transition with the triangle). Polygon edge is given by parametric Equations as in (1): $a = S + U_a \cdot (E - S)$ (6)

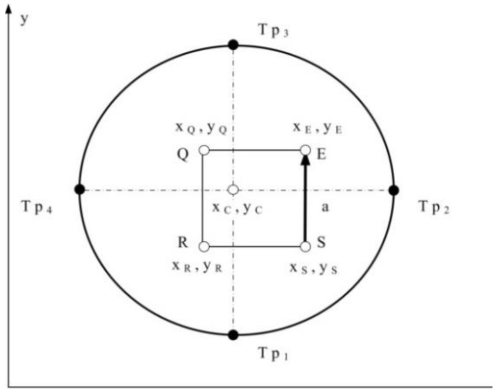


Figure 8
Polygon edge is parallel to y-axis
(tangent points detecting)

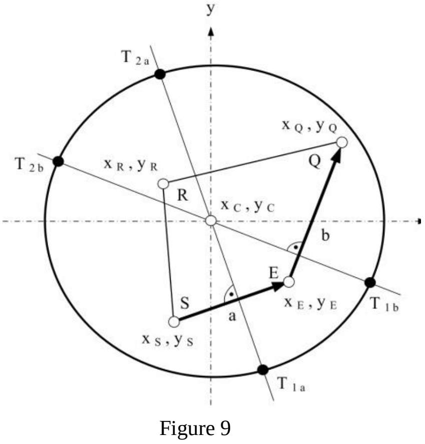


Figure 9
Tangent points determination

On the basis of the vector of edge $a(dx_a, dy_a)$ we can form a normal vector of polygon edges which is $n(dn_{xa}, dn_{ya})$, through the circle center $C(x_C, y_C)$. Parametric Equation of normal line is:

$$n_a = C + U_n \cdot dn \text{, i.e. } x = x_C + U_n\, dn_{xa}\text{, } y = y_C + U_n\, dn_{ya} \qquad (7)$$

and because *n* is orthogonal to *a* we will have $dn_{xa} = -\frac{1}{dx_a}; dn_{ya} = -\frac{1}{dy_a}$ (8)

Circle Equation with origin $C(x_C,y_C)$ has the following form

$$(x-x_C)^2+(y-y_C)^2=r^2 \text{ or in the case of ellipse } \frac{(x-x_C)^2}{r_x^2}+\frac{(y-y_C)^2}{r_y^2}=1 \quad (9)$$

By substitution of Equations of normal (7) and (8) into Equation of circle (9) the solution for square Equation is obtained:

$$U_{n1,2}=\pm\frac{r}{\sqrt{dn_{xa}^2+dn_{ya}^2}} \quad (10)$$

or for edge *a* in coordinates *x* and *y*:

$$xt_{1a}=x_C+U_{n1}\,dn_{xa}\,,\ yt_{1a}=y_C+U_{n1}\,dn_{ya}$$

$$xt_{2a}=x_C+U_{n2}\,dn_{xa}\,,yt_{2a}=y_C+U_{n2}\,dn_{ya} \quad (11)$$

Now, tangent points are:

$$T_{1a}=[xt_{1a} \quad yt_{1a}]\,,\ T_{2a}=[xt_{2a} \quad yt_{2a}] \quad (12)$$

In the case of ellipse for edge *a* are:

$$xt_{1,2a}=\pm\frac{r_x\,r_y}{\sqrt{r_x{}^2\,dn_{ya}{}^2+r_y{}^2\,dn_{xa}{}^2}}\,dn_{xa}+x_C$$

$$yt_{1,2a}=\pm\frac{r_x\,r_y}{\sqrt{r_x{}^2\,dn_{ya}{}^2+r_y{}^2\,dn_{xa}{}^2}}\,dn_{ya}+y_C \quad (13)$$

Tangent points of ellipse are: $T_{1a}=[xt_{1a} \quad yt_{1a}],\ T_{2a}=[xt_{2a} \quad yt_{2a}]$ (14)

In Fig. 9 we can see the determination of a real tangent point, between these two tangent points:

$y_Q>y_E$ and $yt_{1a}<y_C$ => then $T_a=T_{1a}$ else $T_a=T_{2a}$
$y_Q<y_E$ and $yt_{1a}>y_C$ => then $T_a=T_{1a}$ else $T_a=T_{2a}$ (15)

The same procedure can be applied for each edge.

3.2 Determination of Tangent Points for Observed Edge When Basis Planes are Intersecting

Basic circle is on plane α and basic polygon is on plane β. The intersecting line between these two planes is *l* (Figs. 10, 11 and 12).

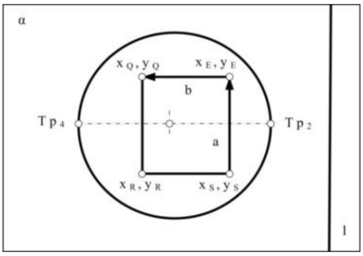


Figure 10
Analyzing of parallelism between polygon edges and intersecting line *l*

Intersecting line *l* is given by coordinate x_1 which is the distance from global coordinate system (Fig. 12). First, we should define two points, for example *E* and *Q* which represent edge *b* (Figs. 10, 11 and 12). Then we can analyze the mutual position between this edge *b* and intersecting line *l*. If they are parallel (Figs. 11 and 12), then tangent points are Tp_2 or Tp_4. Which of these two tangent points is real (Figs. 11 and 12) depends of mutual position of edge *b* instead basic circle. Actually, we should not close the hole between the circle and the polygon.

Second, if an edge, for example *c* intersects line *l* (Figs. 11 and 12), then we can find a point I_c as an intersecting point between lines *c* and *l*. From this point we can construct (calculate) tangent line to basic circle. There are two tangent lines from I_c to the circle, one of them is T_c which together with points *Q* and *R* (Figures 11 and 12) defines triangle QRT_c. Another solution will give us a triangle which closes the hole, and because of that this solution is not a real solution. Figures 11 and 12 present the same problem where angle *theta* is bigger or smaller than 90˚.
z coordinate of point S is given by next equation: $z_S = (x_S - x_1) \cdot \tan(theta)$ (16)

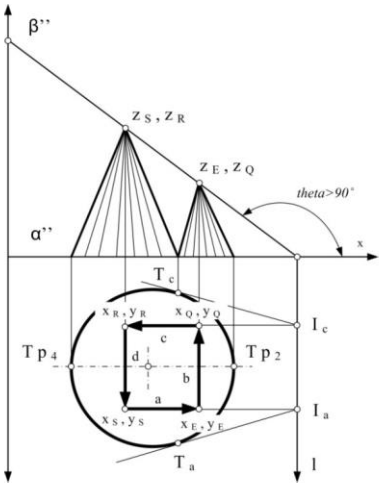


Figure 11
Calculation of z coordinates (theta>90˚)

For edge *b* we use the following procedure:

if *theta*<90°and $x_Q > x_S$ => then $T_P = T_{P4}$ else $T_P = T_{P2}$
if *theta*>90°and $x_Q > x_S$ => then $T_P = T_{P2}$ else $T_P = T_{P4}$ (17)

In the case when we should find an intersecting point between polygon edge and intersecting line *l* we should use the same Equation as in section 3.1 where:
Ua_a - parameter for Equation of edge *a*,
dx_a, dy_a–vectors of edge *a* in *x* and *y* directions,
dl_x, dl_y - vectors of line *l* in *x* and *y* directions,
x_1 – is *x* coordinate of line *l*, when *l*||*y* (Fig. 12)

$$Ua_a = \frac{dx_a \cdot (y_1 - y_S) - dy_a \cdot (x_1 - x_S)}{dy_a \cdot dl_x - dx_a \cdot dl_y} \tag{18}$$

Point I_a is an intersecting point between polygon edge *a* and line *l* (Figs. 11 and 12): $x_a = x_1 + Ua_a \cdot dlx$, $y_a = y_1 + Ua_a \cdot dly$ (19)

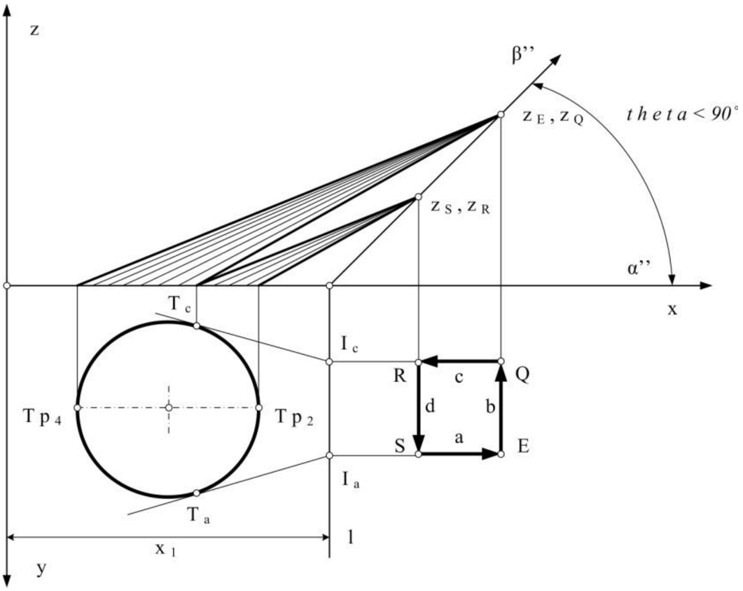


Figure 12
Calculation of z coordinates (theta<90°)

$$I_a = [x_a \quad y_a] \tag{20}$$

3.2.1 Polarity and Tangent Lines

In this section we will create a procedure for constructing (calculating) tangent lines by use of polarity. From point *P* (Fig. 13) we should find tangent points T_{1a} and T_{2a} to the circle. We can use polarity for determination of tangent points, i.e., four points *P, B, Q* and *A* are in harmonic relation:

$$(PQAB) = -1 \text{ or } \frac{[PB]}{[QB]} : \frac{[PA]}{[QA]} = -1 \tag{21}$$

$$P = P(x_P, y_P), C(x_C, y_C) \tag{22}$$

From this we can find length W_{QC} that is calculated as a function of circle radius and length $W_{CP} = \sqrt{(x_P - x_C)^2 + (y_P - y_C)^2 + (z_P - z_C)^2}$ (23)

Now (Obradović and Milojević, [13]): $f = W_{QC} = \dfrac{r^2}{W_{CP}}$ (24)

Cosines of angles of line *CP* are:

$$\begin{bmatrix} c_x & c_y & c_z \end{bmatrix} = \frac{\begin{bmatrix} x_P - x_C & y_P - y_C & z_P - z_C \end{bmatrix}}{W_{CP}} \tag{25}$$

Now: $x_P - x_Q + f\, c_x = x_P - x_C$ (26)

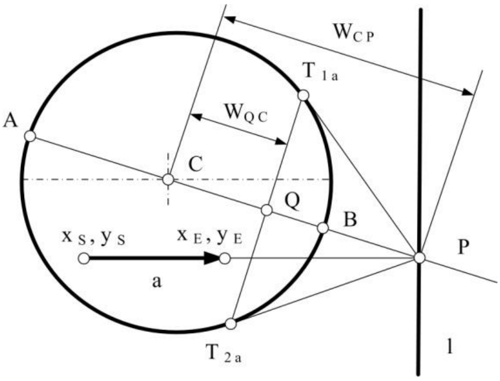


Figure 13

Polarity and tangent points determination

$$x_Q = x_C + f\, c_x = x_C + \frac{r^2}{W_{CP}} \frac{x_P - x_C}{W_{CP}} =$$

from this: (27)

$$x_C + \left(\frac{r}{W_{CP}} \right)^2 (x_P - x_C)$$

Coordinates of point *Q* are:

$$x_Q = x_C + \lambda (x_P - x_C), y_Q = y_C + \lambda (y_P - y_C) \tag{28}$$

$$\lambda = \left(\frac{r}{W_{CP}} \right)^2 \tag{29}$$

Polar line is given with *p*=*PA* where: $y = a_p x + b_p,\ a_p = \tan \alpha$ (30)

Normal line q=T_{1a} T_{2a} on *p* is: $y = a_q x + b_q; a_q = -\dfrac{1}{a_p}$ (31)

Now, because we know coordinates of point Q it is easy to determine b_q from Equation (31). Line q intersects circle $(x-x_C)^2+(y-y_C)^2=r^2$ at points T_{1a} and T_{2a}.

In the case when the basic curve is ellipse (Fig. 14), the procedure for determining tangent points has several steps: in coordinate system Oxy an ellipse is given, whose center coincides with the origin of coordinate system Cuv and the ellipse half-axis are r_x and r_y. It is necessary to find tangent lines from point $P(x_P,y_P)$ to ellipse (Fig. 14). The ellipse is defined by Equation

$$\frac{x^2}{r_x^2}+\frac{y^2}{r_y^2}=1 \tag{32}$$

The collection of lines through point P is given with expression $y=mx+n$ (33)

By substituting Equation (33) into (32) we create a square Equation

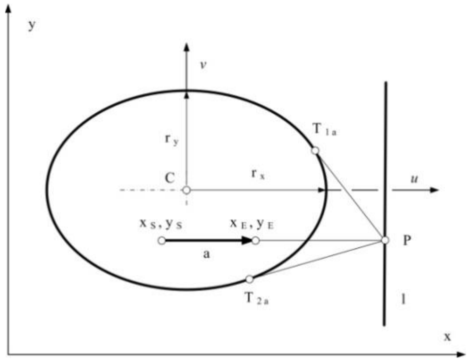


Figure 14
Tangent points to ellipse

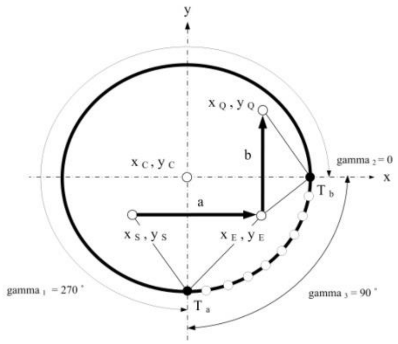


Figure 15
Patches approximation by points array

$$x^2\left(r_y^2+m^2r_x^2\right)+2mn\,r_x^2x+r_x^2\left(n^2-r_y^2\right)=0 \tag{34}$$

with solutions $x_{1,2}=\dfrac{-mn\,r_x^2\pm\sqrt{D_x}}{r_y^2+m^2r_x^2}$ (35)

where D_x is a discriminant in the form $D_x=r_x^2r_y^2\left(-n^2+r_y^2+m^2r_x^2\right)$ (36)

When solutions of (35) are real and different then line (33) intersects ellipse (32) in two points, and when solutions are real and equal then the line is a tangent line to the ellipse. In the case when solutions are real and equal, the discriminant (36) should be equal to zero and because r_x and r_y are different from zero the line (33) will be a tangent line when

$$-n^2+r_y^2+m^2r_x^2=0 \tag{37}$$

According to condition that lines contain point P we can find the value of n

$$n = y_P - mx_P \tag{38}$$

The substitution of (38) into (37) gives a square Equation where *m* is variable

$$m^2\left(r_x^2 - x_P^2\right) + 2x_P y_P m + r_y^2 - y_P^2 = 0 \tag{39}$$

The solutions of this Equation are

$$m_{1,2} = \frac{-x_P y_P \pm \sqrt{D_m}}{r_x^2 - x_P^2} \tag{40}$$

$$D_m = r_x^2 y_P^2 + r_y^2 x_P^2 - r_x^2 r_y^2 \tag{41}$$

When $D_m=0$ then we can put just one tangent line from the point and that is the case when point *P* is on ellipse, i.e.

$$P(x_P, y_P) = P\left(x_P, \frac{\pm\sqrt{r_x^2 r_y^2 - x_P^2 r_y^2}}{r_x}\right) \Rightarrow D_m = 0 \tag{42}$$

When $D_m<0$ then solutions of (40) are complex conjugate, and that is the case when point *P* is inside ellipse. For example, if *P* is coincident with origin *C* then

$$P(x_P, y_P) = P(0,0) \Rightarrow D_m = -r_x^2 r_y^2 < 0 \tag{43}$$

When point *P* is outside ellipse then $D_m>0$ and Equation (40) has two real and different solutions, i.e. there are two tangent lines.

Finally, the procedure for determining tangent points T_1 and T_2 has several steps:

- determining of discriminant D_m by Equation (41);
- calculating coefficients m_1 and m_2 according to (40);
- determining values n_1 and n_2 according to (38);
- calculating tangent points $T_1\left(x_{T_1}, y_{T_1}\right)$, $T_2\left(x_{T_2}, y_{T_2}\right)$ by using Equations

$$x_{T_1} = \frac{-m_1 n_1 r_x^2}{r_y^2 + m_1^2 r_x^2};\ y_{T_1} = m_1 x_{T_1} + n_1;\ x_{T_2} = \frac{-m_2 n_2 r_x^2}{r_y^2 + m_2^2 r_x^2};\ y_{T_2} = m_2 x_{T_2} + n_2 \tag{44}$$

In Fig. 14 ellipse is in global system *Oxy*, and connected to the ellipse is local system *Cuv*. Point *P* in system *Cuv* is given with

$$u_P = x_P - x_C; v_P = y_P - y_C \tag{45}$$

Coordinates of T_1 and T_2 are in coordinate system *Cuv* given by using Equations

$$D_m = r_x^2 v_P^2 + r_y^2 u_P^2 - r_x^2 r_y^2 \tag{46}$$

$$m_{1,2} = \frac{-u_P v_P \pm \sqrt{D_m}}{r_x^2 - u_P^2} \tag{47}$$

$$n_{1,2} = v_P - m_{1,2} u_P \tag{48}$$

$$T_1\left(u_{T_1}, v_{T_1}\right),\ T_2\left(u_{T_2}, v_{T_2}\right) \tag{49}$$

$$u_{T_{1,2}} = \frac{-m_{1,2}n_{1,2}r_x^2}{r_y^2 + m_{1,2}^2 r_x^2}; v_{T_{1,2}} = m_{1,2}u_{T_{1,2}} + n_{1,2} \tag{50}$$

In global system *Oxy* tangent points are defined with

$$x_{T_{1,2}} = x_C + u_{T_{1,2}}; y_{T_{1,2}} = y_C + v_{T_{1,2}} \tag{51}$$

3.3 Creating of Patches (Surface Triangulation)

Transitional developable surface is created by combination of two kinds of patches: triangles and cones. Triangles are composed by polygon edges and plane curve tangent points and this is given by matrix v_2 (Equation 57). Matrix f_2 (Eq. 58) represents an order of connecting points *S*, *E* from polygon edge and tangent point T_a (Figs. 15 and 17). Patch which is actually cone is created by cone vertex *E* (Figs. 15, 16 and 17) and points array from basis plane curve starting from T_a to T_b. Parametric equations of circle are:

$$x = x_C + r \cdot \cos(t);\; y = y_C + r \cdot \sin(t);\; z = zeros(size(t)) \tag{52}$$

where: *t* – equation parameter which represents array with constant step and in *Matlab* notation is represent as follows: $t = linspace\left(gamma_1, gamma_2, N\right)$, where $gamma_1$ and $gamma_2$ are the coordinates of start point and end point by using angular values.

Angular values are in radians and they are measured from horizontal axis in counterclockwise side (Fig. 15). *N* represents the number of points. In Fig. 15 there are 10 points and the angular distance between the two contiguous points is 10 degrees. Each point connected with vertex *E* gives ten cone generatrices. When the basic curve is ellipse we use parameter *q* which represents the relation between parametric equations of ellipse and angle in *Matlab* notation:

*q = mod(atan(rx/ry*tan(t)).*(cos(t)>0) + atan2(rx/ry*tan(2*pi-t), -1).*(cos(t)<0), 2*pi);*

Ellipse is given by parametric equations:

$$x = x_C + r_x \cdot \cos(q);\; y = y_C + r_y \cdot \sin(q);\; z = zeros(size(q)) \tag{53}$$

$$N = round\left(\frac{gamma3 \cdot \frac{360}{2 \cdot \pi}}{10}\right) + 1 \tag{54}$$

where number *N* is an array step, defined by condition that the angle between two consecutive generatrices is 10 degrees, i.e. if *gamma3*=π/2 (Fig. 15) then *N=10*. *Round* is a *Matlab*-s function which is used for number rounding.

$$v_1 = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \cdots & \cdots & \cdots \\ x_N & y_N & z_N \\ x_E & y_E & z_E \end{bmatrix} \qquad (55)$$

Matrix v_1 is used for cone approximation; it collects points (Fig. 15) from curve (52) or (53) and vertex from polygon. In Figures 15 and 16 it is the vertex E. Matrix f_1 (eq. 56) represents the order of connecting points which are in matrix $v_{1.}$

$$f_1 = \begin{bmatrix} 1 & 2 & N+1 \\ 2 & 3 & N+2 \\ \cdots & \cdots & \cdots \\ N-1 & N & 2N-1 \end{bmatrix} \qquad (56)$$

By matrix f_1 we connect two points from the curve with a point from polygon (E in Figures 15, 16 and 17). For points 1 and 2 (Fig. 16) from the curve, point E is (N+1) point in array; for points N-1 and N on the curve, point E is (2N-1) point in array, in *Matlab* notation. Matrix v_2 is used for approximation of triangle patches, and consists of polygon start point S, end point E (Fig. 17) and point T_a from the circle which is tangent point.

$$v_2 = \begin{bmatrix} x_S & y_S & y_S \\ x_E & y_E & z_E \\ xt_a & yt_a & zt_a \end{bmatrix} \qquad (57)$$

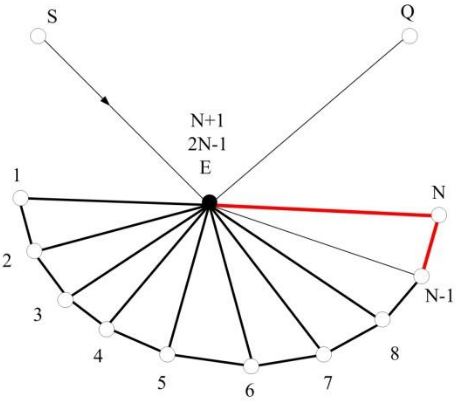


Figure 16
Cone patches creating

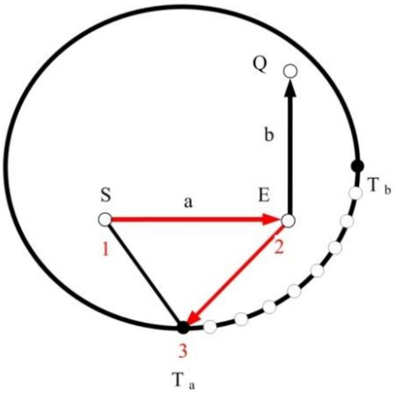


Figure 17
Triangle patches creating

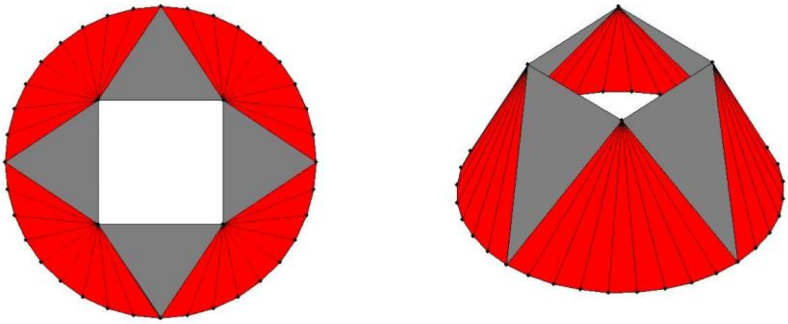
Matrix f_2 is used for connecting points which are in matrix v_2. Numbers in f_2 show row's number from matrix v_2 i.e. order of connecting of these points.

$$f_2 = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix} \qquad (58)$$

4 Results

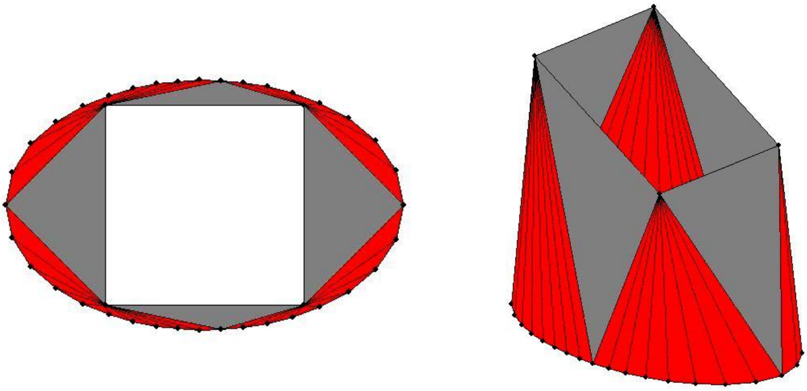
Fig. 18 shows transitional developable surfaces between square and circle. The circle and the square are in parallel plane, the developable surface is a combination of four triangles and four cones. In Fig. 19 transitional surfaces between rectangle and ellipse are created where basis planes are intersecting. In Fig. 20 basis planes are orthogonal and that is a special case. In Fig. 21 there are examples of several patches.

Immediately after the application has finished the calculation for every chosen polygon side, it displays the orthogonal projection of the solution. As is shown in Fig. 22, the creation of every patch can be monitored, checked and corrected during the coordinates input. After the last coordinate has been entered by the user, the application creates a 3D view of the transitional developable surface and its net (Fig. 23), which can be observed and examined by the rotation, panning and scaling of its display.



(a) (b)

Figure 18
Transition between square and circle which are in parallel planes, top view (a) and 3D view (b)



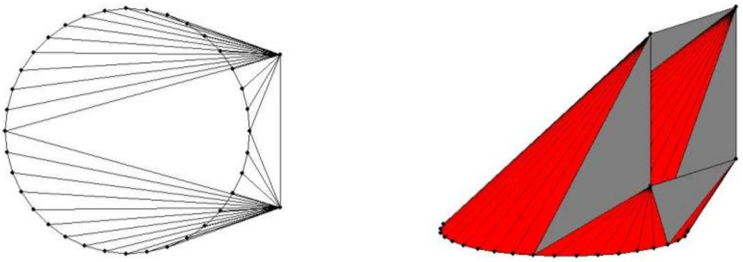
(a) (b)

Figure 19
Developable surfaces approximation when basis planes intersect, top view (a) and 3D view (b)

Conclusions

This paper describes and explains the computational geometry method that simulates the physical model of transitional pipeline parts whose cross sections are plane curve and polygon and are made of materials that cannot be wrinkled or stretched. In particular, the paper presents the basic computer graphics theory of ruled and developable surfaces, the concept of geometrical solution for the generation of transitional surface between plane curve and polygon, and the algorithm by which this problem can be implemented and thus automatically solved.

Geometrical and mathematical methods for transitional developable surfaces are created, in two different cases, when planes which consist of basis plane curve and basis polygon are parallel and when these planes are intersecting. The third case is a special case of the second one, i.e. when basis planes are orthogonal.



(a) (b)

Figure 20

Transitional surfaces in case when basis planes are orthogonal, polygon plane is vertical, top view (a) and 3D view (b)

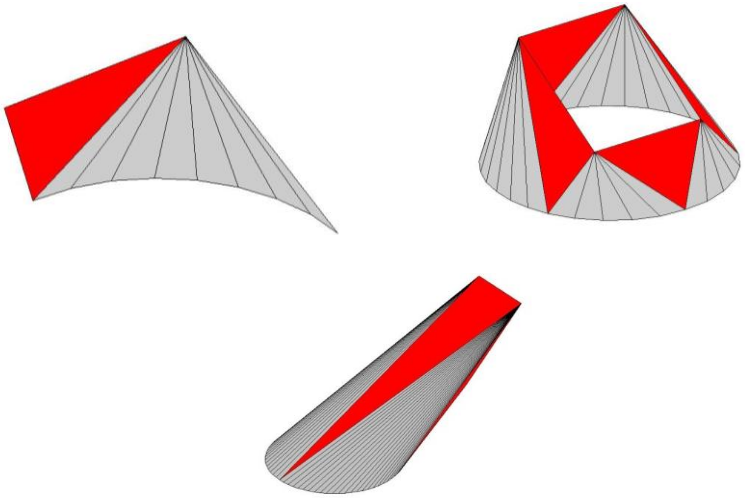


Figure 21

Patches examples

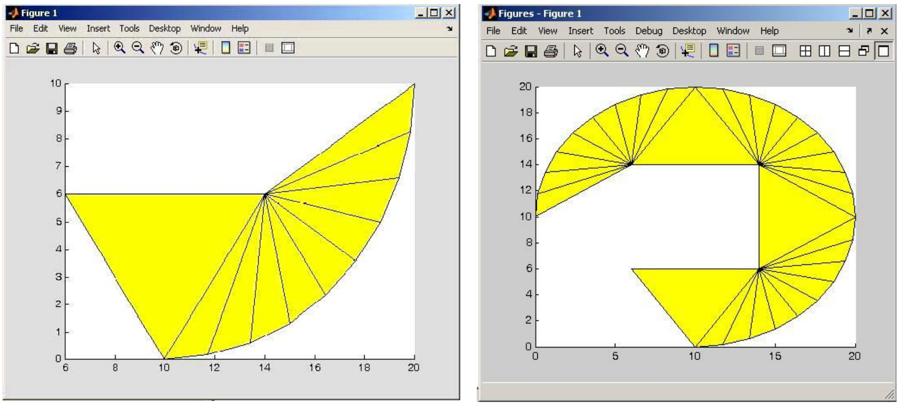


Figure 1
File Edit View Insert Tools Desktop Window Help
10
9
8
7
6
5
4
3
2
1
0
6
8
10
12
14
16
18
20
Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
20
18
16
14
12
10
8
6
4
2
0
0
5
10
15
20

Figure 22

Active solving of the problem in MATLAB program

According to these mathematical models, transitional developable surfaces are created with sets of triangles and cones. Each triangle lies in a plane which is a tangent plane to the contiguous cones at the developable surfaces.

Algorithms and mathematical apparatus for determining tangent points which are on plane curve and which together with polygon vertex create triangles are shown in detail. The procedure for creating cones by array of generatrices is explained.

The procedure is presented in this paper using MATLAB working environment. The commercial version of this procedure can be designed as the upgrade tool for some CAD/CAM applications (*Blending*, *Proengineer*, *Solid Works*), or as a part of new application with its own working environment.

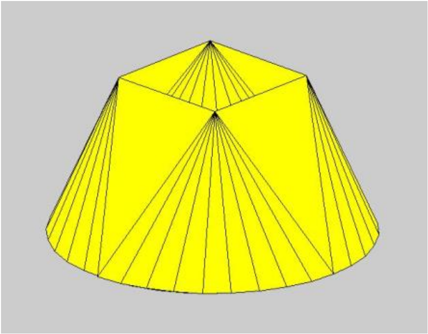


Figure 23

Solution of the problem in 3D

This procedure is created for modeling transitional developable surfaces in many engineering disciplines. It can be used for creating industrial pipeline in mechanical and civil engineering or for roof constructions in architectural design.

Also, the procedure can be used for ship hull design and for design in general. The procedure is useful for any engineering purpose where we need to create developable surfaces between plane curve and polygon.

Our approach is based on creating developable surfaces which is better solution than ruled surfaces for many engineering practical applications, because we can develop these surfaces in the plane without distortion which results in easier production of surfaces.

In our future research we will analyze a problem of creating transitional developable surfaces between two plane curves from different planes.

Acknowledgement

The authors appreciate very much the comments and valuable suggestions from the anonymous reviewers and the editor. Their advice helped us to improve the presentation of this paper.

References

[1] Aumann G.: *Interpolation with Developable Bezier Patches*, Computer Aided Geometric Design, Vol. 8, No. 5, 1991, pp. 409-420

[2] Bo P., Wang W.: *Geodesic-controlled Developable Surfaces for Modeling Paper Bending*, Computer Graphics Forum 3, Vol. 26, no. 3, 2007, pp. 365-374

[3] Bourke P.: *The Shortest Line between Two Lines in 3D*, from *http://www.paulbourke.net*, accessed *on 2013-02-02*

[4] Casale A., Valenti G. M., Calvano M., Romor J.: *Surfaces: Concept, Design, Parametric Modeling and Prototyping*, Nexus Network Journal, Volume 15, Issue 2, Springer Basel, 2013, pp. 271-283

[5] Dahl H. E. I.: *Piecewise Rational Parametrizations of Canal Surfaces,* Mathematical Methods for Curves and Surfaces, 8th International Conference, MMCS 2012, Oslo, Norway, 2012, Revised Selected Papers, Springer-Verlag Berlin Heidelberg, 2014, pp. 88-111

[6] Frey W. H.: *Boundary Triangulations Approximating Developable Surfaces that interpolate a closed space curve*, International Journal Foundation of Computer Science, Vol. 13, 2002, pp. 285-302

[7] Frey W. H.: *Modeling Buckled Developable Surfaces by Triangulation*, Computer Aided Design, Vol. 36, No. 4, 2004, pp. 299-313

[8] Kuntal S., Ozbolat I. T., Koc B.: *Optimized Normal and Distance Matching for Heterogeneous Object Modeling*, Computers & Industrial Engineering, Volume 69, March 2014, Elsevier, 2014, pp. 1–11

[9] Li J., Lu G.: *Modeling 3D Garments by Examples*, Computer-aided Design, Volume 49, April 2014, Elsevier, 2014, pp. 28–41

[10] Li C., Wang R., Zhu C.: *An Approach for Designing a Developable Surface through a Given Line of Curvature,* Computer-aided Design, Volume 45, Issue 3, Elsevier, 2013, pp. 621-627

[11] Liu Y., Tang K., Gong W., Wu T.: *Industrial Design using Interpolatory Discrete Developable Surfaces,* Computer-aided Design, Volume 43, Issue 9, Elsevier, 2011, pp. 1089-1098

[12] Obradović R., Beljin B.: *Modeling Transitional Developable Surfaces in Computer Graphics - Transition from Polygon to Polygon,* Proceedings of 1st International Scientific Conference moNGeometrija, Niš, Serbia, 2008, pp. 233-242

[13] Obradović R., Milojević Z.: *Plane Section of Cone and Cylinder in Computer Geometry*, Facta Universitatis, Series Architecture and Civil Engineering, University of Niš, Vol. 3, No. 2, 2005, pp. 195-207

[14] Obradović R., Popkonstantinović B., Beljin B.: *Algorithm for Approximation Transitional Developable Surfaces Between two Polygons,* TTEM – Technics Technologies Education Management, Vol. 7, No. 4, 2012, pp. 1907-1913

[15] Peternell M.: *Developable surfaces Fitting to Point Clouds,* Computer-aided Geometric Design, Vol. 21, No. 8, 2004, pp. 785-803

[16] Pottmann H., Wallner J.: *Approximation Algorithms for Developable Surfaces,* Computer-aided Geometric Design, Vol. 16, No. 6, 1999, pp. 539-556

[17] PTC, ProEngineer 2001, *Designing with Surfaces,* Tutorial, Release 2000i2

[18] Rogers D. F., Adams J. A.: *Mathematical Elements for Computer Graphics,* McGraw-Hill Publishing Company, 1990

[19] Rose K., Sheffer A, Wither J., Cani M., Thibert B.: *Developable Surfaces from Arbitrary Sketched Boundaries*, Proceedings of the 5th Eurographics Symposium on Geometry Processing, The Eurographics Association, 2007, pp. 163-172

[20] Sun M., Fiume E.: *A Technique for Constructing Developable Surfaces,* Department of Computer Science, University of Toronto, from *http://www.dgp.toronto.edu/~elf/.misc/GI96.pdf*, accessed *on 2013-02-02*

[21] Tang K., Wang C.: *Modeling Developable Folds on a Strip,* Journal of Computing and Information Science in Engineering, Vol. 5, 2005, pp. 35-47

[22] Xu G.: *Discrete Laplace Beltrami Operator on Sphere and Optimal Spherical Triangulations*, International Journal of Computational Geometry and Applications, Vol. 16, No. 1, 2006, pp. 75-93