Mangalayatan Journal of Engineering and Technology, Vol. 1 (I), 2025
Article ID 251002

Line Graphics vs Polygon Mesh in Representing Curves on Surfaces

Vesna I. Velickovic
Faculty of science, University of Nis, Nis, Serbia
vesna@pmf.ni.ac.rs

Abstract

This paper investigates methods of representing curves on surfaces in 3D computer graphics, in particular, in the polygon mesh and line graphics approach. The basic characteristics of both approaches are analyzed. The specifics of the author's own software for visualization of mathematics, MV-Graphics, are also discussed. Special attention is paid to the problem of visibility in these two approaches and the possibility of controlling the visibility functions in MV-Graphics. The paper also discusses the existence of the contour line (silhouette) and its significance. The problem of accurately displaying curves on surfaces is analyzed, emphasizing their importance for mathematical accuracy and visual interpretation. Finally, concrete examples of intersection of two surfaces, visualization of special curves on a surface and obtaining the cross-section of solid bodies using only manipulation of visibility in MV-Graphics are given.

Keywords: Computer graphics, Visibility of 3D surfaces, Object design by controlling visibility, Contour line (silhouette), Application software.

1. Introduction

Visualization plays a very important role in education, science and everyday life. In education, graphical models help students to understand abstract mathematical concepts in fields such as geometry or differential geometry, through intuitive and interactive representations. In science, visualisation enables a better understanding of complex structures and processes, from molecular models and medical 3D images to astrophysical simulations. In technology, visualization is extremely important for design, engineering analyses, simulations and optimization of production processes, and enables precise planning and rapid evaluation of complex systems. In everyday life, visualization is present in digital media, video games, animation and interactive applications, facilitating users to intuitively understanding complex objects and processes. Representation of curves on surfaces is of particular importance, since curves are the basic elements for defining the shape, contours and structure of three-dimensional objects. They serve as surface boundaries, leading elements in parameterization and the basis for many algorithms in modeling and rendering. Accurate representation of curves is crucial for both mathematical accuracy and visual quality.

Received 19 August 2025; Accepted 15 September 2025

In the context of representing curves, there are two dominant approaches: line graphics and polygon mesh. Line graphics allow for a more accurate representation of curves, since they rely on mathematical models that define the exact paths of lines and contours. This approach is simple, intuitive and effective for visual communication, but is limited in photo-realistic rendering of complex surfaces and textures.

Conversely, the polygon mesh is the standard in modern computer graphics. It describes surfaces through a network of vertices, edges and faces, which enables easy manipulation of complex geometries. However, the mesh provides only an approximation of the shape of the surface, which is the main obstacle to the accurate representation of curves on surfaces. Polygon mesh brings advantages in modeling and rendering surfaces, but at the cost of reduced accuracy in representing curves on them.

2. POLYGON MESH

Currently, the most widely used technology for representing surfaces in computer graphics is the polygon mesh. In this approach, surfaces are represented by a set of vertices, edges, and faces. A vertex is a point on the surface and is located at a precise position on the surface. An edge is a straight line segment that connects two adjacent vertices in the mesh. Since surfaces are generally not flat, an edge represents an approximation of the path between corresponding vertices. Several edges that connect to each other and thus form a closed polygonal line without self-intersection form a face. A face is an approximation of that part of the surface. The simplest and most commonly used case is when a face is formed by three edges. In practice, four-sided faces are also common, and less often multi-sided faces. If all faces of a mesh are formed by three edges, then the polygon mesh represents a triangulation of that surface, and if all faces are formed by four edges, then it is a rectangulation of the surface. In practice, mixed meshes also occur, when the faces of a mesh are not formed by an equal number of edges. We are considering the entire polygonal mesh which is a polyhedral approximation of the surface.

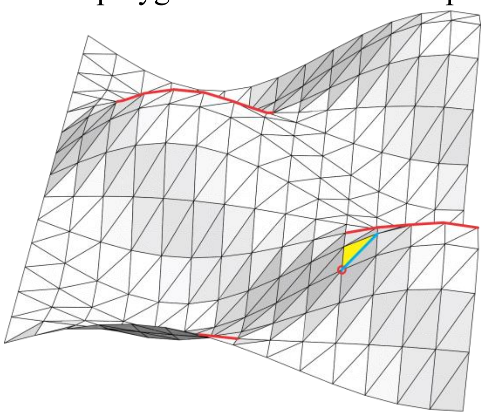

Figure 1. Triangulation of a surface, vertex, edge, face and contour lines in a polygonal mesh.

Polygon mesh is the state-of-the-art and the most common data structures used for representing surfaces in modern 3D computer graphics, computational geometry, visualization, engineering analysis and other fields. Its simplicity in data structure and efficiency in rendering and algorithmic processing make it suitable for practical and scientific applications. Due to its flexibility, simple data structure, and efficiency in displaying and processing complex shapes, polygon mesh has become the de facto standard in many technical disciplines, including medical visualization, CAD/CAM systems, 3D printing, simulation of physical systems, and video games [Botsch et al., 2010].

In addition to geometry, each element of a polygonal mesh can have normals, texture coordinates, colors, materials, deformation attributes, physical properties, and more. Normals are used in shading and lighting, which greatly contributes to the understanding of surface shapes, see Figure 1.

A large number of papers dealing with polygonal mesh processing have been published. For example, the method of *simplifying meshes* using Quadric Error Metrics [Garland & Heckbert, 1997] has become a standard in reducing the complexity of 3D models, and is still used today in real-time applications. Efficient methods for *smoothing and aligning meshes* are discussed in [Taubin, 1995] and [Fleishman et al., 2003]. A systematic review and comparison of different 3D *segmentation* methods is given in the paper [Attene et al., 2006]. The *parameterization of meshes*, as well as their conversion into other forms for texturing and simulation purposes, are discussed in [Hormann & Greiner, 2000]. *Surface parameterization* using conformal and isometric approaches that minimize distortion is given in [Lévy et al., 2002]. *Remeshing* techniques, which generate new meshes with better quality properties (eg., uniformity, element validity), have become the basis for simulation and adaptive methods [Alliez et al., 2008] and [Valette et al., 2008]. Also, *repairing faulty models*, such as meshes with holes or self-intersections, is an increasingly common topic in applications involving scanned data, and automated reconstruction methods are discussed in [Attene et al., 2013] and [Ju, 2009]. Contemporary research directions include the integration of machine learning for classification, segmentation, and reconstruction of 3D models, using a polygon mesh as the primary representation [Hanocka et al., 2019].

2.1 Line Graphics

Line graphics is a type of computer graphics that represents objects using lines and curves only, without filling the surfaces with color, texture, or shadows. It is the simplest way to display geometric structures in 2D or 3D space. A relatively small number of lines are required to represent surfaces, so figures created in this way are simple and without unnecessary details. Only the information about the object that is relevant in the desired context is displayed. All mathematical sketches and technical drawings are created in this way.

The surface model in line graphics is the simplest possible. Only the values of the vertices on the surface are calculated. Adjacent vertices in the model are connected by lines, typically straight line segments. In a more complicated model, interpolation or approximation can be used to obtain a curved line based on several accurately calculated vertices, as in the Bezier or B-spline model [Piegl & Tiller, 1997]. These models are used in CAD design when it is necessary to accurately describe smooth, controlled curves and surfaces, such as in the automotive and aeronautical industries, or in computer typography to describe TrueType and PostScript fonts. However, by simply increasing the number of vertices per line in a simple model, we obtain a line that the human eye has the impression of being curved. The number of vertices per line depends on the resolution of the display device on which the graphics are displayed.

The advantages of line graphics are fast visualization, lower memory consumption, and clarity in structure, which is useful for analysis of topology of surfaces, especially in education, science and engineering applications.

The author is developing the *MV-Graphics* software package for visualization of mathematical objects and scenes using line graphics [Veličković & Dolićanin, 2023]. We connect two adjacent vertices with a straight line segment, and the curved appearance of the line is achieved by a strategy of a large number of vertices per line. This approach is sufficient for the line to look smooth even on printed media with graphics 1.5 meters wide. Analytical evaluation of values whenever possible, extended type of variable values, and floating point operations ensure the accuracy of the position of the points.

The beginnings of this software date back to 1993 [Malkowsky & Nickel, 1993]. It was originally intended for student education, but the flexibility of developing our own software made it suitable for scientific visualization. It has since been used to visualize various mathematical, natural, and technical fields, from geometry and differential geometry [Veličković, 2006] and [Malkowsky & Veličković, 2004], through functional analysis [Veličković et al., 2022], [Malkowsky et al., 2017] and [Malkowsky & Veličković, 2013] and topology [Malkowsky & Veličković, 2011], to chemistry and crystallography [Malkowsky et al., 2013] and [Malkowsky & Veličković, 2012].

2.2 Visibility

At this point, we should consider the problem of visibility of points, lines and parts of surfaces. If we do not take into account that some parts of the surface can be hidden by other parts of the same or some other surface, we get a completely transparent surface. Such a representation can be ambiguous and confusing, as it is shown in Figure 2 on the left. Therefore, a strategy of removing invisible parts of the surface should be developed, like in Figure 2 on the right. There are several algorithms that deal with this problem [Agoston, 2005].

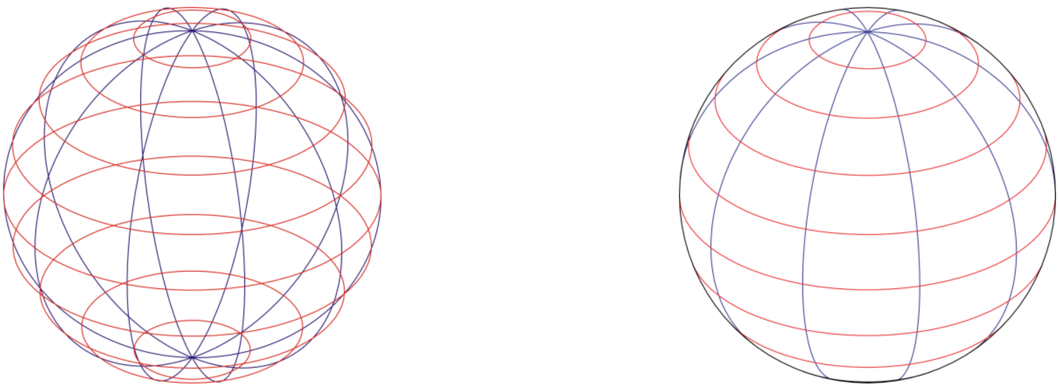

Figure 2. A sphere without and with visibility and contour

In polygon mesh approach, visibility is determined by the visibility of the face. A face is visible if it is not hidden by another face in the same mesh or by another surface in the scene. Vertices and edges are visible if they lie on a face that is visible. The normal vector of a face is taken into account, so the face may be visible only if it is facing in direction of the camera. This strategy can speed up the process of determining visibility. Also, in these models it is common to define the transparency of objects so the hidden parts of the surface are semi-visible, as well as is the surfaces behind it, Figure 3.

Figure 3. Transparency of objects

In *MV-Graphics*, the visibility of each point is calculated analytically, immediately after determining its position. A point on a surface is invisible if there exists another point of the same or another surface in the scene closer to the view point on the same projection ray. An edge, which is a straight line segment, is visible if both of its endpoints (vertices) are visible [Malkowsky & Nickel, 1993] and [Malkowsky & Veličković, 2008]. In line graphics, no other visibility strategy is needed.

In *MV-Graphics* there are two independent procedures, *Visibility* and *NotHidden*, for checking the visibility of a point according to the surface on which it lies and according to other surfaces in the scene. This approach, together with the fact that visibility is checked independently of the drawing, allows for great flexibility in determining what is displayed on a scene. To the best of the author's knowledge, all software packages for visualizing mathematical objects, except our, have built-in realistic object visibility that cannot be changed.

Figure 4 shows the realistic visibility and concept of obtaining Dandelin's spheres. If the intersection of a cone and a plane is an ellipse or a hyperbola, then there are exactly two spheres that are tangent to both the plane and the cone along a line, and if the intersection is a parabola, then there is one and only one sphere with this property. Moreover, spheres are tangent to the intersecting plane at the foci of the conic sections. In the realistic visibility (Figure 4, left) this concept cannot be shown, and the second Dandelin's sphere is completely hidden by the cone. Using the same program, but with changed visibility functions, we show the concept of Dandelin's spheres (Figure 4, right). The cone is transparent from the front, the intersecting plane is visible only inside the cone but with full visibility of the ellipse and with realistic visibility according to the spheres, the foci of the ellipse are highlighted, the larger sphere is cut open so that the interior is visible (this is achieved by the interval of definition of the sphere), the tangent circles have realistic visibility and the contour lines of the spheres are fully visible.

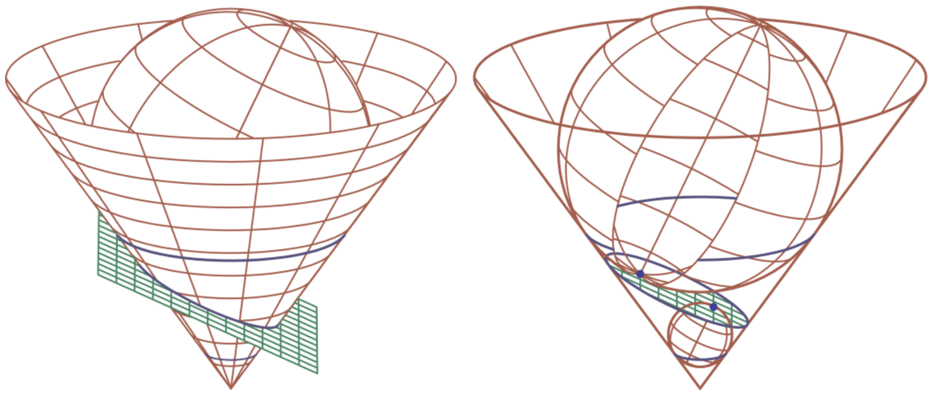

Figure 4. Dandelin spheres, reality and concept

2.3 Contour line

A special feature of line graphics implemented in *MV-Graphics* is the existence of contour lines (silhouettes) of surfaces. It contributes greatly to a good understanding of the shape of surfaces, as we see in Figure 5. No matter how precisely the lines on the surface are drawn, at some point they pass from the visible part to the invisible. Without the contour, the surface looks unfinished.

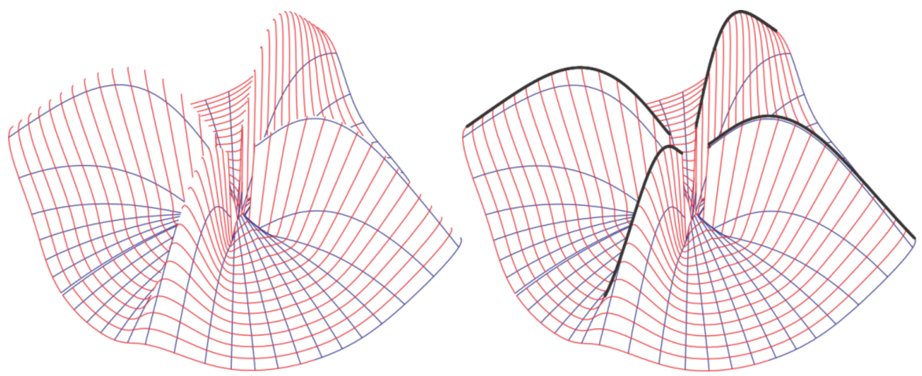

Figure 5. A surface without and with contour line

Roughly speaking the contour of a surface *S* consists of lines on *S* which separate its visible points from its invisible ones. More precisely, if *C* is the centre of projection then a point *P* on a surface *S* is a *contour point* if and only if

(i) The projection ray to P is orthogonal to the surface normal vector $\overrightarrow{N(P)}$ of S at P.
(ii) There exists a neighbourhood *U(P)* such that the line of intersection *IS* of S with the plane through *P*, spanned by the vectors $\overrightarrow{CP}$ and $\overrightarrow{N(P)}$, is on one side of the projection ray thoughout *U(P)* and does not intersect the projection ray in $U(P)\backslash\{P\}$. This means the scalar product $\overrightarrow{N(P')} \cdot \overrightarrow{CP}$ has the same sign for all points $P' \in IS \cap U(P)$.

Since it is very time consuming to check condition (ii) which only excludes some rare pathological cases, we drop it from our definition of contour points and use condition (i) only.

The *contour* of a surface is the set of its contour points. Determining the contour of a surface involves finding the zeros of a real valued function of two variables in a domain $D \subset \mathbb{R}^2$. The numerical method for this can be found in [Failing & Malkowsky, 1996].
Furthermore, one should pay attention to the fact that the contour line can also be hidden by part of a surface, so its visibility should also be considered.
In the polygon mesh approach, the contour line is not highlighted, but is intuitively seen as the boundary of the last visible faces on transition to the invisible part of the surface, as shown in Figures 1 and 7.

2.4 Curves on surfaces

Curves on surfaces play a very important role both in theory, for example in differential geometry, and in practical applications, for example in CAD systems, in various visualizations and animations. They allow the description of paths, edges, contours and functional boundaries on complex 3D objects. Of particular importance are parametric lines that reflect the surface parametrization, geodesic lines that represent the shortest path between two points on a surface, but it should also be mentioned curved interpolations, level lines, lines of constant slope, asymptotic lines and curved features of a surface such as lines of maximum curvature, [do Carmo, 2016], [O'Neill, 2006], [Kühnel, 2015] and [Malkowsky et al., 2023].
In digital representations of surfaces using polygonal meshes, the construction of smooth and accurate curves is not trivial due to the approximation of surfaces by polyhedra. A curve can be represented quite accurately using a large number of points for its interpolation, but then it does not lie on the surface, see the Figure 6 on the left. Therefore, approximation methods such as piecewise linear curves that lie on the faces of the polygon mesh and break at the edges of the mesh are used, as shown in Figure 6 on the right and in Figure 7.

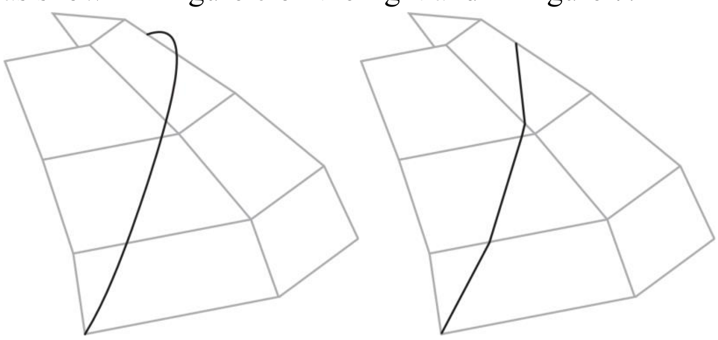

Figure 6. Left: Curve is not on the surface. Right: Curve is not smooth

On the other hand, since in line graphics we do not approximate a surface, but rather represent the surface by families of lines on it, no additional strategy is needed to display any line on the surface. But in this approach it is necessary to determine the parameterization of the desired line, which is not a trivial problem, as we will see in the next section.

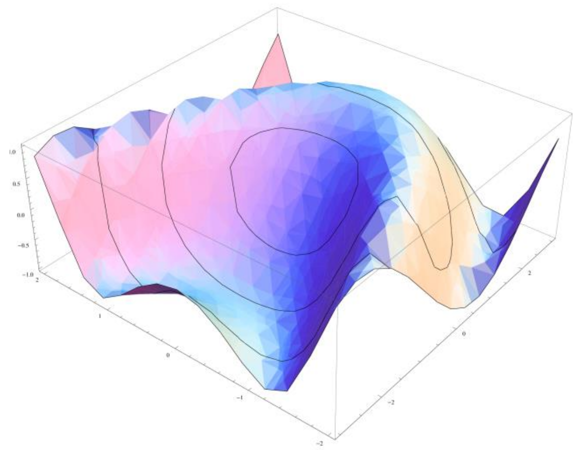

Figure 7. Curves on a surface in a polygon mesh

3. APPLICATIONS

As an application, we will discuss the intersection of two surfaces. Although we are used to polygonal approximations for the representation of a single surface, we already have the problem of representing the intersection line of two surfaces. Since the faces in a polygon mesh are flat, two faces intersect along a straight line segment. Then the intersection line of two polygon meshes consists of a broken polygonal line where the length of one segment is large enough to clearly see its non-smooth structure, see Figure 8 on the left.

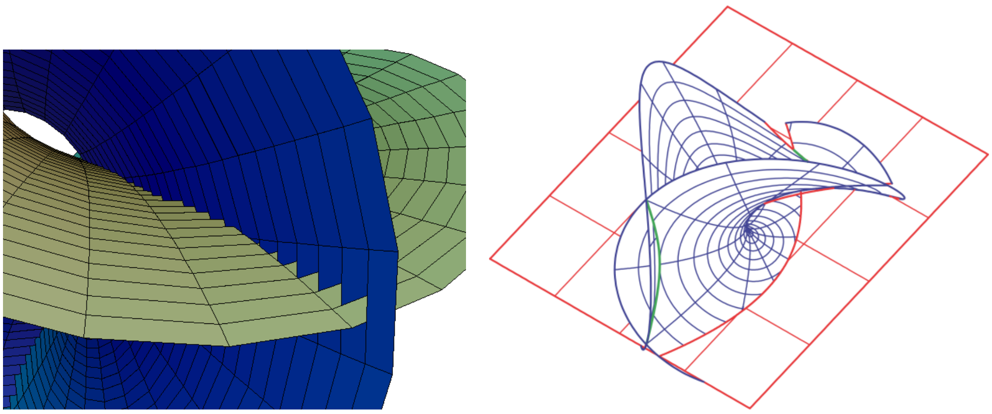

Figure 8. Intersection of surfaces in polygon mesh and line graphics approach

On the other hand, in line graphics, all lines on a surface can be displayed correctly, including the lines of intersection of two surfaces, Figure 8 on the right [Veličković, 2022]. However, this is not obtained automatically as in the case of a polygon mesh, but it is necessary to determine the line of intersection, either analytically or numerically. In *MV-Graphics* we prefer to calculate the parametric representation of desired lines, but if that is not possible, we use numerical methods, and as a last resort we program a geometric procedure to obtain the points. Of course, the parametric representation gives the fastest and most accurate results, while the geometric procedures are the slowest and are used only if it is not possible to obtain the results in any other way.

For example the intersection line of an *Enneper's surface* with parametric representation in polar coordinates [Veličković, 2017]

$$\vec{x}\,(\rho,\phi) = \left\{\rho\cos\phi - \frac{\rho^3}{3}\cos 3\phi\,,\ \rho\sin\phi - \frac{\rho^3}{3}\sin 3\phi\,,\ \rho^2\cos 2\phi\right\}$$

with a sphere with center at point $X_0 = (x_0^1,\ x_0^2,\ x_0^3)$ of radius $r > 0$ satisfies the equation

$$\left(\rho\cos\phi - \frac{\rho^3}{3}\cos 3\phi - x_0^1\right)^2 + \left(\rho\sin\phi - \frac{\rho^3}{3}\sin 3\phi - x_0^2\right)^2 + (\rho^2\cos 2\phi - x_0^3)^2 = r^2.$$

Figure 9 on the left shows an Ennneper's surface with its lines of self-intersection, a sphere, and the intersection line of these two surfaces. Figure 9 on the right shows an Enneper's surface with the intersection lines of a family of concentric spheres of different radii.

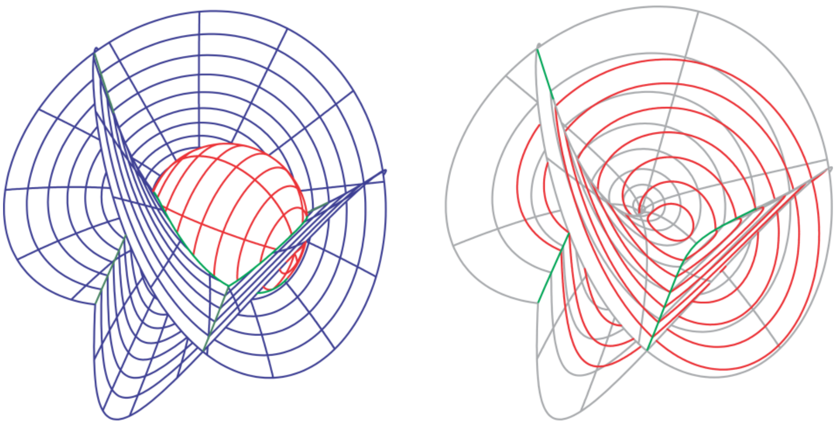

Figure 9. Intersections of Enneper's surface with spheres

Now we want to visualize some **special curves on a surface**. The *lines of curvature* on an Enneper's surface are given by

$$\rho^{(1)}(\phi) = \frac{c^{(1)}}{|\cos\phi|} \quad \text{and} \quad \rho^{(1)}(\phi) = \frac{c^{(1)}}{|\cos\phi|} \quad \text{for } \phi \neq \frac{\pi}{2}, \pi, \frac{3\pi}{2}$$

where $c^{(1)}$ and $c^{(2)}$ are positive constants, and ρ-lines that correspond to $\phi = \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ (Figure 10, left).

Furthermore, let $(\rho_0, \phi_0) \in I_\rho \times I_\phi$ and $\Theta_0 \in (-\pi/2\,,\,\pi/2)$ be given. We put $c = \rho_0\,(1+\rho_0^2)\cos\Theta_0$ and

$$\rho_1 = 2\sqrt{\frac{1}{3}}\sinh\left(\frac{1}{3}\log\left(\frac{3\sqrt{3}\,|c|}{2} + \sqrt{\frac{27c^2}{4} + 1}\right)\right)$$

Then the *geodesic line* through (ρ_0, ϕ_0) with an angle Θ_0 to the ϕ –line through ρ_0 is given by (Figure 10, right)

$$\phi(\rho) = c \int_{\rho_0}^{\rho} \frac{dt}{t\sqrt{t^2(1+t^2)^2 - c^2}} + \phi_0 \quad \text{for } \rho > \rho_1.$$

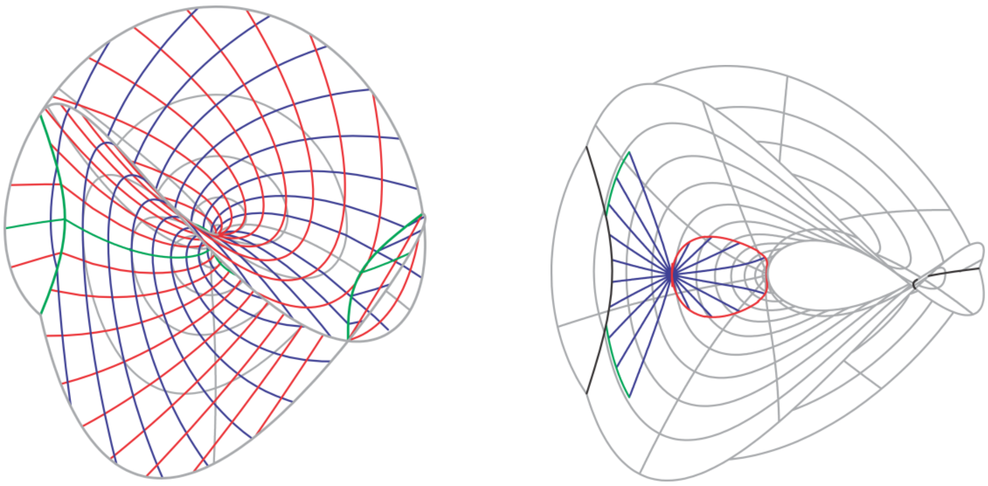

Figure 10. Lines of curvature (left) and geodesic lines (right) on Enneper's surface

Finally, we demonstrate how our independent visibility check may be applied to represent surfaces which are the boundary of the intersection of several solid bodies in the case of the circular cylinders C_1, C_2 and C_3 of radius 1 with the x^1-, x^2- and x^3-axes as their axes. Then C_1 and C_2 are given by parametric representations

$$\vec{x}^{(1)}(u^i) = (u^1, \cos u^2, \sin u^2), \quad \left((u^1, u^2) \in I_1 \times [0, 2\pi]\right)$$

$$\vec{x}^{(2)}(v^i) = (\sin v^2, v^1, \cos v^2), \quad \left((v^1, v^2) \in I_2 \times [0, 2\pi]\right)$$

where the intervals I_1 and I_2 both contain the interval [-1,1]. For their line of intersection I_{12} we must have

$$(u^1)^2 + 1 = (v^1)^2 + 1, \quad \text{that is} \quad v^1 = \pm u^1 = \pm \sin v^2.$$

We write $v^2 = t$. Then the two branches of I_{12} are given by the parametric representations

$$\vec{x}^{(12)}(t) = (\sin t, \pm \sin t, \cos t), \quad (t \in [0, 2\pi]).$$

Similarly we find parametric representations for the other lines of intersection. Figure 11 on the left shows a realistic scene of a cross-section of three cylinders with their lines of intersection.

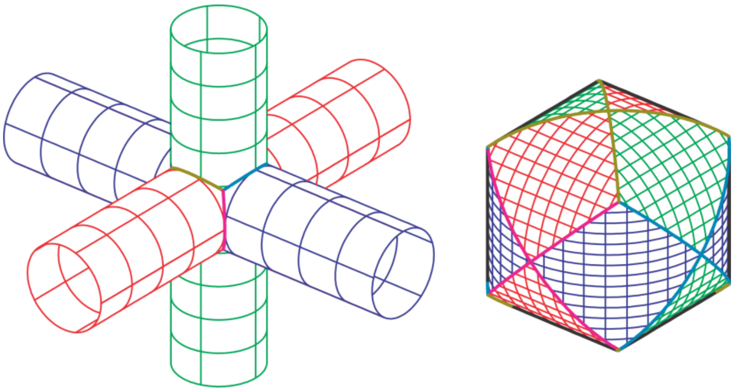

Figure 11. Left: Cross-section of three cylinders. Right: Body in cross-section of three cylinders

Now we describe how to obtain the cross-section of these three cylinders using only visibility manipulation. First we determine if a point $P = \{P_x,P_y,P_z\}$ is inside one of the cylinders C_1, C_2 or C_3. Point P is inside the cylinder C_1 if $P_y^2 + P_z^2 \leq 1$, it is inside the cylinder C_2 if $P_x^2 + P_z^2 \leq 1$, and it inside the cylinder C_3 if $P_x^2 + P_y^2 \leq 1$. We will define that a point on the cylinder C_i , $i \in \{1,2,3\}$ is visible if it is visible with respect to the cylinder C_i and is located inside the other two cylinders. The visibility of the contour line is defined in a similar way. The line of intersection of two cylinders is visible if it is visible according to both cylinders on which it is located and is located inside the third cylinder. Thus, using the same program, just by manipulating of functions for visibility of the objects on the scene, we obtain the cross-sectional body shown in Figure 11 on the right.

4. Conclusion

The representation of curves on surfaces remains a central challenge in computer graphics and geometric modeling. In this paper, two approaches for performing this task, line graphics and polygon mesh, are compared. In addition to the properties of the basic model of line graphics, the properties of the software package for the visualization of mathematics, *MV-Graphics*, developed by the author, were also considered.

The problem of visibility is considered separately in this paper. If it is not included in the model, we get a picture that can be ambiguous and confusing. To solve this problem, the polygon mesh can include transparency in the model of an object. Line graphics considers the visibility of each evaluated point in relation to the rest of the scene. The particularity of *MV-Graphic* is the existence of two functions, visibility and not hidden, which can be independently programmed to obtain unrealistic but desired results.

The existence of a contour line (silhouette) in the graphic representation of three-dimensional surfaces in *MV-Graphics* is emphasized. In the polygon mesh, this line does not exist, but the observer can only intuitively recognize it as a transition from the visible to the invisible part of the surface. However, this line in the polygon mesh is not smooth.

Curves on surfaces were especially investigated with an emphasis on their role in theory and practical examples. The problem with polygon meshes is that the curve that is accurately represented does not lie on the mesh or, if it is forced to lie on the mesh, then it is a very rough approximation of the desired curve. In line graphics, the curves are precisely placed on the surfaces, but their parametric representation or the equation that describes them must be calculated.

Finally, three applications are given. First, the intersections of the two surfaces were considered. It is explained why, in the polygon mesh approach, the intersecting line cannot be smooth. For the line graphics approach, the equation that must be satisfied by the intersection line of a sphere and an Enneper's surface is given. This example is visualized, together with the representation of the intersection lines of the family of concentric spheres of different radii on a Enneper's surface. For the second application, equations and graphics of lines of curvature and geodesic lines on an Enneper's surface are given. At the end, the parametric equations of the lines of intersection of three circular cylinders with axes along the coordinate axes were calculated. A realistic representation of this cross-section is shown, but it is also explained how, by manipulating the visibility functions *Visibility* and not hidden of these cylinders, a figure of the cross-sectional body can be obtained.

5. References

[1] [Agoston, 2005] Agoston, M. K. (2005). Computer graphics and geometric modeling:implementation & algorithms, Springer.

[2] [Alliez et al., 2008] Alliez, P., Ucelli, G., Gotsman, C., & Attene, M. (2008). Recent advances in remeshing of surfaces. In: De Floriani, L., Spagnuolo, M. (eds) Shape Analysis and Structuring. Mathematics and Visualization. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-33265-7_2

[3] [Attene et al., 2006] Attene, M., Katz, S., Mortara, M., Patane, G., Spagnuolo, M., & Tal, A. (2006). "Mesh Segmentation - A Comparative Study," IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06), Matsushima, Japan, pp. 7-7, doi: 10.1109/SMI.2006.24.

[4] [Attene et al., 2013] Attene, M., Campen, M., & Kobbelt., L. (2013). Polygon mesh repairing: An application perspective. ACM Computing Surveys 45(2), Article 15, 33 pages. doi: 10.1145/2431211.2431214

[5] [Botsch et al., 2010] Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., & Lévy, B. (2010). Polygon Mesh Processing. A K Peters / CRC Press.

[6] [do Carmo, 2016] do Carmo, M. P. (2016). Differential Geometry of Curves and Surfaces. Revised and Updated Second Edition. Dover Publications.

[7] [Failing & Malkowsky, 1996] Failing M., & Malkowsky E. (1996). Ein effizienter Nullstellenalgorithmus zur computergrafischen Darstellung spezieller Kurven und Flächen, Mitt. Math. Sem. Giessen, 229, 11-25

[8] [Fleishman et al., 2003] Fleishman, S., Drori, I., & Cohen-Or, D. (2003). Bilateral mesh denoising. ACM Transactions on Graphics, 22(3), 950–953.

[9] [Garland & Heckbert, 1997] Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error metrics. Proceedings of the 24th

annual conference on Computer graphics and interactive techniques (SIGGRAPH), ACM.
[10] [Hanocka et al., 2019] Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., & Cohen-Or, D. (2019). MeshCNN: A network with an edge. ACM Transactions on Graphics, 38(4), Article No. 90.
[11] [Hormann & Greiner, 2000] Hormann, K., & Greiner, G. (2000). MIPS: An efficient global parametrization method. Curve and Surface Design, 153–162.
[12] [Ju, 2009] Ju, T. (2009). Fixing Geometric Errors on Polygonal Models: A Survey. Journal of Computer Science and Technology. 24, 19–29. doi: 10.1007/s11390-009-9206-7
[13] [Kühnel, 2015] Kühnel, W. (2015). Differential Geometry: Curves–Surfaces–Manifolds (3rd ed.). American Mathematical Society.
[14] [Lévy et al., 2002] Lévy, B., Petitjean, S., Ray, N., & Maillot, J. (2002). Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics (TOG), 21(3), 362–371.
[15] [Malkowsky & Nickel, 1993] Malkowsky E., & Nickel W. (1993). Computergrafik in der Differentialgeometrie, Vieweg-Verlag Wiesbaden, Braunschweig.
[16] [Malkowsky & Veličković, 2004] Malkowsky, E., & Veličković, V. (2004). Analytic transformations between surfaces with animations, Proceedings of the Institute of Mathematics of NAS of Ukraine, (50)(3), p. 1496-1501
[17] [Malkowsky & Veličković, 2008] Malkowsky, E., & Veličković, V. (2008). Solutions of some visibility and contour problems in the visualisation of surfaces. APPS. Applied Sciences, 10, 125–142.
[18] [Malkowsky & Veličković, 2011] Malkowsky, E., & Veličković, V. (2011). Topologies of some new sequence spaces, their duals, and the graphical representations of neighborhoods, Topology and its Applications, Volume 158, Issue 12, 1 August 2011, p. 1369-1380
[19] [Malkowsky & Veličković, 2012] Malkowsky, E., & Veličković, V. (2012). Some New Sequence Spaces, Their Duals and a Connection with Wulff's Crystal, MATCH Commun. Math. Comput. Chem. 67, p. 589-607
[20] [Malkowsky et al., 2013] Malkowsky, E., Özger, F., & Veličković, V. (2013). Some Spaces Related to Cesàro Sequence Spaces and an Application to Crystallography, MATCH Commun. Math. Comput. Chem., Vol. 70, No. 3, 867-884
[21] [Malkowsky & Veličković, 2013] Malkowsky, E., & Veličković, V. (2013). The duals of certain matrix domains of factorable triangles and some related visualisations, Filomat, vol.27:5, 821 – 829
[22] [Malkowsky et al., 2017] Malkowsky, E., Özger, F., & Veličković, V.I. (2017). Some Mixed Paranorm Spaces, Filomat, 31:4, 1079–1098
[23] [Malkowsky et al., 2023] Malkowsky, E., Dolićanin, Ć., & Veličković, V. (2023). Differential Geometry and Its Visualization, CRC Press, Taylor&Francics Group, ISBN 9781032436661

[24] [O'Neill, 2006] O'Neill, B. (2006). Elementary Differential Geometry (2nd ed.). Academic Press.
[25] [Piegl & Tiller, 1997] Piegl, L., & Tiller, W. (1997). The NURBS Book (2nd ed.). Springer.
[26] [Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH).
[27] [Valette et al., 2008] Valette, S., Chassery, J. M., & Prost, R. (2008). Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. IEEE Transactions on Visualization and Computer Graphics, 14(2), 369–381.
[28] [Veličković, 2006] Veličković, V.I. (2006). On Surfaces of Rotation of a Given Constant Gaussian Curvature and Their Visualisations, Proceedings of the Conference Contemplorary Geometry and Related Topics, p. 523-534
[29] [Veličković, 2017] Veličković, V. (2017). Visualization of Enneper's Surface by Line Graphics, Filomat, vol.31:2, p. 387–405
[30] [Veličković, 2022] Veličković, V.I. (2022). Intersections of Surfaces of Revolution, Facta Universitatis. Series: Mathematics and Infromatics, vol.37(1)
[31] [Veličković et al., 2022] Veličković, V.I., Malkowsky, E., & Dolićanin, E. (2022). Modeling Spheres in Some Paranormed Sequence Spaces, Mathematics 10(6), 917.
[32] [Veličković & Dolićanin, 2023] Veličković, V.I., & Dolićanin, E. (2023). Line Graphics for Visualization of Surfaces and Curves on Them, (2023) Filomat 37(4): 1017–1027.