THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALAŢI
FASCICLE V, TECHNOLOGIES IN MACHINEBUILDING,
ISSN 2668-4829 (PRINT) 2668-4888 (ONLINE)
https://doi.org/10.35219/tmb.2024.1.04

NUMERICAL REPRESENTATION OF OBJECTS – REPRESENTATION OF 3D BODIES

Răzvan-Sebastian CRĂCIUN[1], Virgil-Gabriel TEODOR[1]

[1]Department of Manufacturing Engineering, "Dunarea de Jos" University of Galati, Romania
email: razvan.craciun@ugal.ro

ABSTRACT

We are familiar with everything necessary to create the image of a point or a line in digital environment, and we can represent many points in both 2D and 3D. In order for these sets of points, projected in a design software, to be transformed into somewhat suggestive images of solid objects, it is necessary to know the order in which certain points must be connected by segments or curves, thus suggesting constructive surfaces to the operator. The information needed to build a body image is stored in a "database", having a very well-defined ordered structure of numbers and characters. The structure of the database strongly influences the speed of work, the required memory of the program, its flexibility, as well as the ease of writing the program. This paper presents some common ways of organizing data to generate different types of simple representations in ascending order of complexity.

KEYWORDS: point cloud, wireframe, polygon mesh, curves and curved surfaces.

1. INTRODUCTION

We are familiar with everything needed to create the image of a point or line in a digital environment, which allows us to represent a large number of points in both 2D and 3D.

In order for these sets of points projected in a design software to be transformed into the most suggestive possible images of solid objects, it is necessary to know the order in which certain points must be joined together by segments or curves, thus suggesting constructive surfaces to the operator.

The information required to build a body image is stored in a "database", having a very well-defined ordered structure of numbers and characters. Its structure strongly influences the speed of work, the memory required by the program, its flexibility, as well as the ease of writing the program [1].

This paper presents some common ways of organizing data to generate different types of simple representations, listed in increasing order of complexity.

2. VARIOUS NUMERICAL REPRESENTATIONS OF 3D OBJECTS

2.1. In the form of a cloud of points

The numerical model of a surface consists of the coordinates of a set of points chosen on that surface to describe it as accurately as possible.

If the object to be modeled has surfaces composed of flat facets, an exact model of it can be obtained using the coordinates of the vertices of these facets.

However, since many real-world bodies that need to be modeled have curved surfaces, the models are in most cases approximate. Of course, the approximation can be made as accurate as possible, which is why there are numerically controlled machining centers for 3D curved surfaces, which, based on the coordinates of a point cloud, physically create the projected surface with the desired precision.

For the set of points that model a body, typically its surface, we can set 2 types of conditions [1]:

1) The surface of the body must actually pass through the given points.
2) The distance between the real surface and the points defining the model must not exceed a user-imposed limit.

This type of representation was initially used in medicine and chemistry and was later adopted in other application areas of interactive graphics.

The basic element of many data structures used in computer graphics, and the only one used for point representation, is the vertex list. A vertex, or branching node, is a point on the surface of a model where several lines used in the representation of that body meet. The term has also been extended to point representation, where no lines appear as elements of the resulting drawing.

A vertex list begins by specifying the total number of points, and then, for each point, its 3D coordinates are provided. In the case of the tetrahedron in Figure 1, the vertex list has the form:

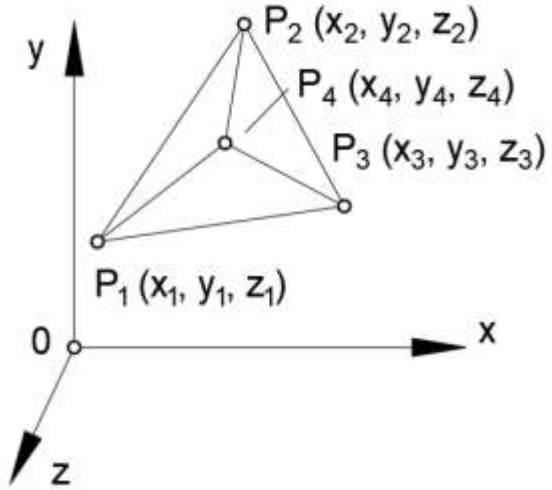

Fig. 1. *List of vertices of the tetrahedron [1]*

$$\begin{cases} [1] & x_1 & y_1 & z_1 \\ [2] & x_2 & y_2 & z_2 \\ [3] & x_3 & y_3 & z_3 \\ [4] & x_4 & y_4 & z_4 \end{cases}$$

The vertex list can be either complete or truncated:

- A complete list contains the coordinates of all points that describe the body.
- A truncated list includes information about only a subset of points, as well as other information needed to determine the coordinates of the remaining points using symmetries, rotations, translations, and scaling.

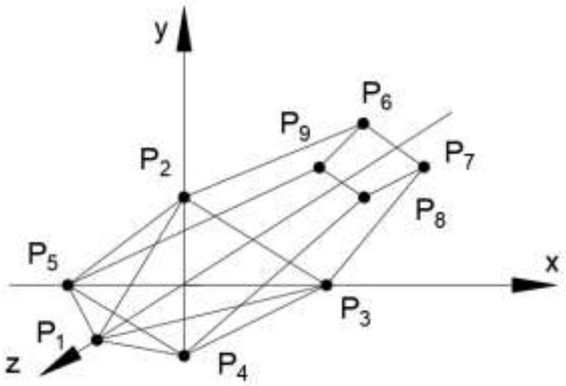

Fig. 2. *Body defined in a Cartesian system xOyz*

As an example, we can consider the body in Figure 2, defined in a Cartesian reference frame *xOyz* (defining reference frame). A truncated list of vertices, in the composition of which symmetry with respect to the *yOz* plane is taken into account, does not include the coordinates of points P5 and P9.

However, it is necessary to group the points into parallel cross-sections and specify that the number of points that make up each section.

2.2. Wireframe

The name "wireframe" comes from the similarity between this type of representation and a model of the object created in the form of a wireframe. In wireframe representation, the concepts of volume and surface are not used. A body is represented as a set of line segments or portions of curves.

For wireframe representations composed of line segments, we have the following forms of data storage for creating a wireframe representation of an object:

a) Explicit segments – In this form, an object is seen as a collection of segments, for which the total number of elements is known. For each segment, the following information is specified:

[i] – optional segment index,

$x_1, y_1, z_1, x_2, y_2, z_2$ – the coordinates of the segment's endpoints.

Since the coordinates of the endpoints are specified in the segment list, a vertex list is not used. The method requires a relatively large amount of memory, since 6 coordinates are specified for each segment. Additionally, when a curve is composed of several concatenated segments, many points may appear at least 2 times in the list.

For example, consider the body in Figure 3. The corresponding data structure will have the following form:

$Segment$ 1.	x_1	y_1	z_1	x_2	y_2	z_2
$Segment$ 2.	x_1	y_1	z_1	x_3	y_3	z_3
$Segment$ 3.	x_1	y_1	z_1	x_3	y_3	z_3
$Segment$ 4.	x_1	y_1	z_1	x_3	y_3	z_3
$Segment$ 5.	x_2	y_2	z_2	x_6	y_6	z_6
$Segment$ 6.	x_3	y_3	z_3	x_7	y_7	z_7
$Segment$ 7.	x_4	y_4	z_4	x_8	y_8	z_8
$Segment$ 8.	x_5	y_5	z_5	x_9	y_9	z_9
$Segment$ 9.	x_2	y_2	z_2	x_3	y_3	z_3
$Segment$ 10.	x_3	y_3	z_3	x_4	y_4	z_4
$Segment$ 11.	x_4	y_4	z_4	x_5	y_5	z_5
$Segment$ 12.	x_5	y_5	z_5	x_2	y_2	z_2
$Segment$ 13.	x_6	y_6	z_6	x_7	y_7	z_7
$Segment$ 14.	x_7	y_7	z_7	x_8	y_8	z_8
$Segment$ 15.	x_8	y_8	z_8	x_9	y_9	z_9
$Segment$ 16.	x_9	y_9	z_9	x_6	y_6	z_6

b). Implicit segments – In this form, each segment is specified by a pair of indices that identify its endpoints in a list of vertices. The amount of memory required is reduced compared to the previously presented form.

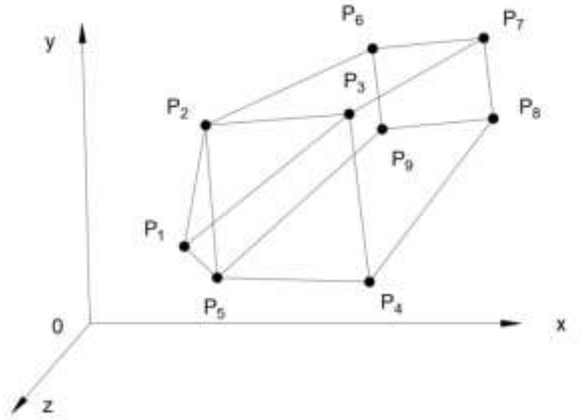

Fig. 3. *Model of the object made in the form of a wireframe*

The data structure for the object in Figure 3 has the following form:

$$List\ of\ vertices \begin{cases} 1 & x_1 & y_1 & z_1 \\ 2 & x_2 & y_2 & z_2 \\ 3 & x_3 & y_3 & z_3 \\ 4 & x_4 & y_4 & z_4 \\ 5 & x_5 & y_5 & z_5 \\ 6 & x_6 & y_6 & z_6 \\ 7 & x_7 & y_7 & z_7 \\ 8 & x_8 & y_8 & z_8 \\ 9 & x_9 & y_9 & z_9 \end{cases} \quad \begin{matrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 3 & 1 & 4 \\ 4 & 1 & 5 \\ 5 & 2 & 6 \\ 6 & 3 & 7 \\ 7 & 4 & 8 \\ 8 & 5 & 9 \\ 9 & 2 & 3 \\ 10 & 3 & 4 \\ 11 & 4 & 5 \\ 12 & 5 & 2 \\ 13 & 6 & 7 \\ 14 & 7 & 8 \\ 15 & 8 & 9 \\ 16 & 9 & 6 \end{matrix}$$

c). Lines given by indices – When a polygonal line can be described by concatenating a string of segments, it is more appropriate to use the following type of data structure for the object in Figure 3:

$$List\ of\ vertices \begin{cases} 1 & x_1 & y_1 & z_1 \\ 2 & x_2 & y_2 & z_2 \\ 3 & x_3 & y_3 & z_3 \\ 4 & x_4 & y_4 & z_4 \\ 5 & x_5 & y_5 & z_5 \\ 6 & x_6 & y_6 & z_6 \\ 7 & x_7 & y_7 & z_7 \\ 8 & x_8 & y_8 & z_8 \\ 9 & x_9 & y_9 & z_9 \end{cases}$$

$$\begin{matrix} 1. & 5 & 6 & 2 & 1 & 3 & 7 \\ 2. & 5 & 9 & 5 & 1 & 4 & 8 \\ 3. & 5 & 2 & 3 & 4 & 5 & 2 \\ 4. & 5 & 6 & 7 & 8 & 9 & 6 \end{matrix}$$

Here, each line is described as follows:

$[i]$ – optional line index.

np – number of points on the line.

$j_1, j_2, \ldots, j_{np}$ – indices that locate the ends of the segments that make up the line in the vertex list.

d). Cross-sections and longitudinal lines – The procedure is the same as for lines given by indices, but most of the cross-sections are those for which the database structure was previously presented.

The database for the body in Figure 3 is as follows:

$$List\ of\ vertices \begin{cases} 1 & x_1 & y_1 & z_1 \\ 2 & x_2 & y_2 & z_2 \\ 3 & x_3 & y_3 & z_3 \\ 4 & x_4 & y_4 & z_4 \\ 5 & x_5 & y_5 & z_5 \\ 6 & x_6 & y_6 & z_6 \\ 7 & x_7 & y_7 & z_7 \\ 8 & x_8 & y_8 & z_8 \\ 9 & x_9 & y_9 & z_9 \end{cases}$$

$$1, 4, 4\ number\ of\ points\ per\ section$$
$$5, 6, 2, 1, 3, 7 - curve\ 1$$
$$5, 9, 5, 1, 4, 8 - curve\ 2$$

If a set of points that constitute the successive ends of concatenated segments are collinear, obviously, it is sufficient to specify only the extremities of the polygonal line with this property.

Although wireframe representation is somewhat simplistic and does not provide complete information on the geometry of the body, due to its ease of use and the speed of displaying the representation, it is widely used today. By using the wireframe technique, high work speeds and even interesting animation effects can be achieved using less sophisticated materials.

2.3. Polygon network

The "wireframe" representation of a 3D object does not allow for the definition of surfaces and, therefore, the calculation of areas, volumes, masses, centers of gravity, or the display of the visible portion of the analyzed object on screen. Simple representations that allow for the recognition of surfaces and the performance of calculations related to these surfaces are obtained through 2 body modeling processes: surface modeling and solid modeling. In the first case, a body is modeled by specifying its boundary, effectively modeling a surface. The latter can be obtained as the surface of a polyhedron, composed of a network of flat polygonal facets, or as a curved surface in space, composed of portions or "patches" of curved surfaces. [3]

In the case of solid modeling, the body is "built" by joining elementary volumes- such as cubes, pyramids, tetrahedra, spheres, and cylinders - that approximate the desired shape as closely as possible. Both processes fall under the more general class of geometric modeling, also called shape modeling. [4]

Polyhedral or polygon mesh representation involves modeling 3D objects using one or more polyhedral surfaces. Each polyhedral surface is treated as a collection of adjacent planar polygonal facets. If the real object has curved surfaces, its polygonal model will, of course, be an approximation. This approximation can be made as good as possible by increasing the number of planar polygonal facets that model a curved surface. The disadvantage is an increased memory requirement, but it is worth remembering that algorithms for processing planar polygonal surfaces are much simpler than those for curved surfaces. [5]

For this reason, most applications that do not require the actual processing of the analyzed body rely on polyhedral modeling, making a compromise between precision on the one hand, required volume of memory, simplicity and speed of work, on the other hand.

The approximation of a curved profile by a polygonal line is represented in Figure 4 a), and the approximation of a curved surface by a polyhedral surface in Figure 4 b). [6]

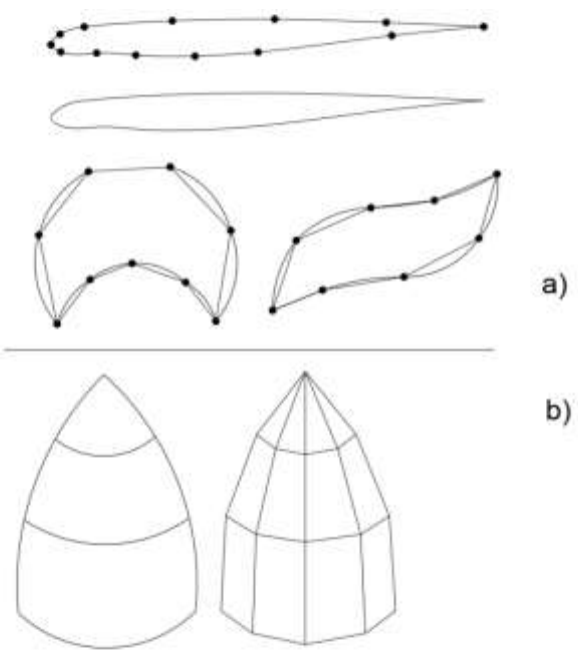

Fig. 4. *Approximation of a curved profile by a polygonal line (a) and Approximation of a curved surface by a polyhedral surface (b)*

As basic elements for polyhedral modeling, we encounter again the list of vertices. This time, the points whose coordinates are written in the list will represent the vertices of the polyhedron. These, together with the edges and faces of the polyhedron, constitute the defining elements of the polygonal mesh.

Depending on the requirements of the program we are developing, we can store the information necessary to represent polygonal vertices, edges, and facets in various ways.

The criteria by which we choose the form of data storage mainly concern 2 features of the program: the amount of memory required and the speed of operation. Since increasing speed typically requires more memory, we usually resort to compromises: we first determine the type of computer for which we are writing the application, and then, depending on the available memory and the maximum complexity that we expect for the bodies that we will have to model, we choose the form of data storage. To increase the speed of work, it is useful to be able to quickly and easily identify the following:

a) The edges of a given polygon.
b) The endpoints of a given edge.
c) Polygons that have a given side in common.
d) Edges that converge at a vertex.

It is also necessary to choose an order of representation of the facets so that the observer can better understand the depth relationships between them (dynamic construction).

When working with a polygonal network, it is essential to traverse all the facets of the network one by one, for example, in a loop. The vertices of a facet can be specified in 2 ways:

1. Explicitly, where the vertices that define a facet are read either as indices in the vertex list or by coordinates.

2. Implicitly, where the vertices that define a facet are determined as indices in the vertex list based on an algorithm for "traversing" this list.

The first method has the advantage of being faster, but the memory required is generally larger. We will refer to this as "read-through facet traversal". The second variant is much slower, but also has much lower memory consumption. We will refer to it as "generate facet traversal".

2.4. Curves and curved surfaces

Curved shapes are more difficult to represent, but are particularly useful in computer-aided processing, as well as in applications where precise calculations are required. Various models may require the representation of curved curves and/or surfaces in 3 dimensions. To simplify calculations, parametric representations of these geometric varieties are used. The curves or surfaces can then be described by traversing the domain of definition of the parameters usually used, namely the interval [0,1], with a conveniently chosen step. [4,7]

Cubic parametric curves

It can be shown that cubic parametric functions, in which the parameters appear to the 3rd power, have the minimum degree necessary to satisfy 2 conditions: the represented 3D curve must pass through 2 points and have given tangents at those points (Figure 5).

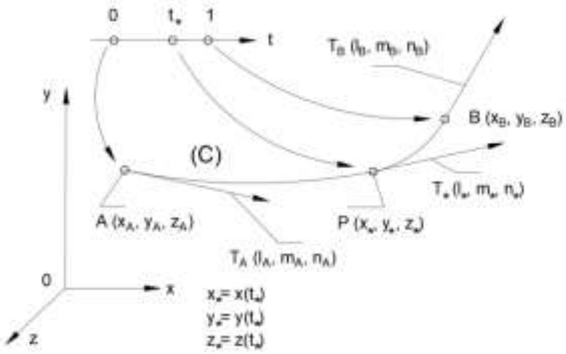

Fig. 5. *Cubic parametric curves*

Curve *C* is described by the relationships:

$$\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \end{cases} \quad (1)$$

Where t is a parameter between 0 and 1, $(t \in [0,1])$.

The tangent vector to the curve in $x(t_*), y(t_*), z(t_*)$ has the components:

$$l = \frac{dx}{dt}\Big|_{t_*}, m = \frac{dy}{dt}\Big|_{t_*}, n = \frac{dx}{dt}\Big|_{t_*} \quad (2)$$

That is:

$$\begin{cases} l_* = 3a_x t_*^2 + 2b_x t_* + c_x \\ m_* = 3a_y t_*^2 + 2b_y t_* + c_y \\ n_* = 3a_z T_*^2 + 2b_z t_* + c_z \end{cases} \quad (3)$$

We notice that the relationships have the same form for *x*, *y* and *z*. It will therefore suffice to analyze a function of the form:

$$K(t) = at^3 + bt^2 + ct + d \quad (4)$$

For which the derivative is:

$$K = \frac{dK}{dt} = 3at^2 + 2bt + c \quad (5)$$

For the purpose of finding the values of *a*, *b*, *c* and *d*.

$K_{(t)}$ can then be replaced by $x(t), y(t), z(t)$ and *a*, *b*, *c*, and *d* with the corresponding coefficients. There are a number of ways to define cubic parametric curves. Of these, 3 types are analyzed: the Hermite form, the Bezier form and the B-Spline form.

a. 3D curves in Hermite form

For the situation in Figure 5, in order to obtain the Hermite form, the following conditions are imposed:

- The ends of the curve should coincide with points *A* and *B*.
- The tangents to the curve at its extreme points should coincide with vectors T_A and T_B.

As *t* varies between 0 and 1, for point $A, t = 0$ and for points $B, t = 1$, these conditions will be written as follows:

$$1) \quad \begin{matrix} x(0) = x_A & respectively & x(1) = x_B \\ y(0) = y_A & & y(1) = y_B \\ z(0) = z_A & & z(1) = z_B \end{matrix}$$

$$2) \quad \begin{matrix} \frac{dx}{dt}\Big|_0 = l_A \;\; și \;\; \frac{dx}{dt}\Big|_1 = l_B \\ \frac{dy}{dt}\Big|_0 = m_A \frac{dy}{dt}\Big|_1 = m_B \\ \frac{dz}{dt}\Big|_0 = n_A \frac{dz}{dt}\Big|_1 = n_B \end{matrix} \quad (6)$$

Taking into account the form of cubic functions, we will have:

$$1) \quad \begin{matrix} d_x = x_A \;\; și \;\; a_x + b_x + c_x + d_x = x_B \\ d_y = y_A a_y + b_y + c_y + d_y = y_B \\ d_z = z_A a_z + b_z + c_z + d_z = z_B \end{matrix}$$

$$2) \quad \begin{matrix} c_x = l_A 3a_x + 2b_x + c_x = l_B \\ c_y = m_A 3a_y + 2b_y + c_y = m_B \\ c_z = n_A 3a_z + 2b_z + c_z = n_B \end{matrix} \quad (7)$$

Considering the general form $K_{(t)} = at^3 + bt^2 + ct + d$, we will obtain 3 systems of the type:

$$\begin{cases} d = P_A \\ a + b + c + d = P_B \\ c = T_A \\ 3a + 2b + c = T_B \end{cases} \quad (8)$$

Where P_A and P_B are position components relative to the ends *A* and *B*, respectively, and T_A and T_B are the components of the tangents to the curve at *A* and *B*.

Considering the above system with the unknowns *a*, *b*, *c* and *d*, it can be written in the form:

$$\begin{bmatrix} P_A \\ P_B \\ T_A \\ T_B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (9)$$

Relationship found in the literature in the following form:

$$[G_h] = [M_h]^{-1} \times [C] \quad (10)$$

By inverting the matrix $[M_h]^{-1}$ and multiplying the above relation by the calculated inverse, we obtain the relation for calculating the coefficients *a*, *b*, *c*, and *d*:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} P_A \\ P_B \\ T_A \\ T_B \end{bmatrix} \quad (11)$$

$$or$$

$$[C] = [M_h] \times [G_h]$$

$[M_h]$ is called the Hermite matrix and $[G_h]$ is a geometric component of Hermite form.

If we denote by $[T]$ the line vector $[t^3, t^2, t, 1]$, then:

$$K(t) = [t^3, t^2, t, 1]\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = [T] \times [C] = \quad (12)$$
$$= [T] \times [M_h] \times [G_h]$$

Replacing $K(t)$ with each of the coordinates, we will obtain the final relations:

$$x(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} X_A \\ X_B \\ l_A \\ l_B \end{bmatrix}$$
$$y(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} Y_A \\ Y_B \\ m_A \\ m_B \end{bmatrix} (13)$$
$$z(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} Z_A \\ Z_B \\ n_A \\ n_B \end{bmatrix}$$

The factors T_A and T_B must have the same order of magnitude as P_A and P_B.

b. 3D curves in the form of Bezier

Unlike the Hermite shape, the Bezier shape has a control interval corresponding to the variation of the parameter t from 0 to 1, in which there are 4 control points. The first and last points specify the ends of the interval, while the additional points (compared to the Hermite shape) determine, together with the ends, the direction of the tangents, as shown in Figure 6.

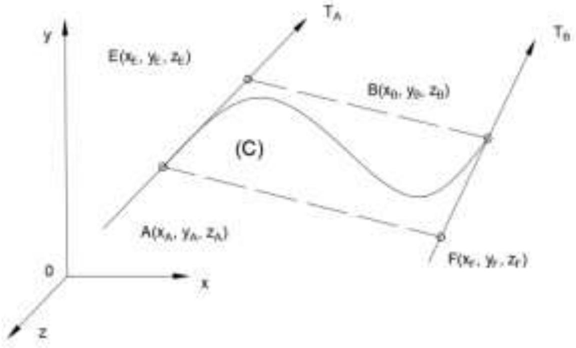

Fig. 6. *3D curves in the form of Bezier*

The conditions that the curve must meet will be written:

$$x(0) = x_A x(1) = x_B$$
$$1. y(0) = y_A and y(1) = y_B$$
$$z(0) = z_A z(1) = z_B$$
$$\dot{x}(0) = \frac{dx}{dt}\Big|_0 = (x_E - x_A) \times m \dot{x}(0) = \frac{dx}{dt}\Big|_1 = (x_E - x_A) \times m \quad (14)$$
$$2. \dot{x}(0) = \frac{dx}{dt}\Big|_0 = (x_E - x_A) \times m and \dot{x}(0) = \frac{dx}{dt}\Big|_1 = (x_E - x_A) \times m$$
$$\dot{x}(0) = \frac{dx}{dt}\Big|_0 = (x_E - x_A) \times m \dot{x}(0) = \frac{dx}{dt}\Big|_1 = (x_E - x_A) \times m$$

Where m is called the form factor.

Using the general form $K_{(t)} = at^3 + bt^2 + ct + d$, so $K(t) = \frac{dK}{dt} = 3at^2 + 2bt + c$, these conditions lead to writing systems of the form:

$$\begin{cases} K(0) = P_A \\ K(1) = P_B \\ K(0) = (P_E - P_A) \times m \\ K(1) = (P_B - P_F) \times m \end{cases} \rightarrow$$
$$\begin{cases} d = P_A \\ a + b + c + d = P_B \\ C = (P_E - P_A) \times m \\ 3a + 2b + c = (P_B - P_F) \times m \end{cases} \quad (15)$$

The system is analogous to the one obtained in the Hermite form, if we make the substitutions:

$$T_A = (P_E - P_A) \times m, T_B = (P_B - P_F) \times m \quad (16)$$

The transition from the geometric component of Bezier shape to the geometric component of Hermite shape is therefore done with the relationship:

$$\begin{cases} P_A = P_A \\ P_B = P_B \\ T_A = (P_E - P_A) \times m \\ T_B = (P_B - P_F) \times m \end{cases} or \begin{bmatrix} P_A \\ P_B \\ T_A \\ T_B \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -m & m & 0 & 0 \\ 0 & 0 & -m & m \end{bmatrix} \times \begin{bmatrix} P_A \\ P_E \\ P_F \\ P_B \end{bmatrix} \quad (17)$$

For the normal Bezier shape, we work with $m = 3$, so:

$$\begin{bmatrix} P_A \\ P_B \\ T_A \\ T_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \times \begin{bmatrix} P_A \\ P_E \\ P_F \\ P_B \end{bmatrix} \quad or \quad [G_h] = [M_{hb}] \times [G_b] \quad (18)$$

Where $[M_{hb}]$ is the transition matrix from Hermite form to Bezier form.

We can write:

$$[C] = [M_h] \times [G_h] = [M_h] \times [M_{hb}] \times [G_b] \quad (19)$$

Respectively:

$$K(t) = [T] \times [C] = [T] \times [M_h] \times [M_{hb}] \times [G_b] \quad (20)$$

We denote the product $[M_h] \times [M_{hb}]$ by $[M_b]$.

This matrix is called the Bezier matrix:

$$\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} =$$

$$= \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

$$So\ [M_b] = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The general form of the Bezier calculation relationship for the value of a coordinate is:

$$K(t) = [T] \times [M_B] \times [G_b] =$$

$$= [t^3 t^2\ t\ 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} P_A \\ P_E \\ P_F \\ P_B \end{bmatrix} \quad (22)$$

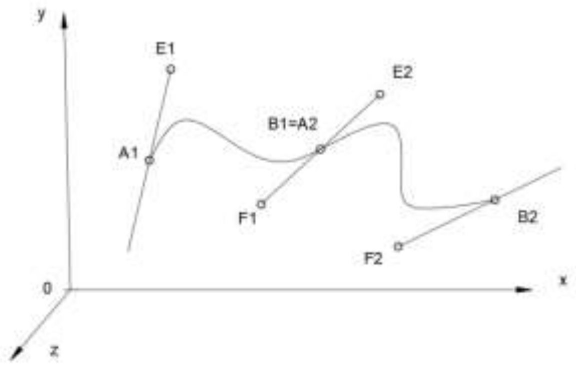

Fig. 7. *Bezier curve*

By substitution, we obtain the relations:

$$x(t) = [t^3 t^2\ t\ 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_A \\ x_E \\ x_F \\ x_B \end{bmatrix}$$

$$y(t) = [t^3 t^2\ t\ 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} y_A \\ y_E \\ y_F \\ y_B \end{bmatrix} \quad (23)$$

$$z(t) = [t^3 t^2\ t\ 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} z_A \\ z_E \\ z_F \\ z_B \end{bmatrix}$$

For several adjacent Bezier curve segments to form a continuous curve, it is necessary that at the junction point, the tangents be parallel. Therefore, the points F_1, B_1, and E_2 (in Figure 8) must be collinear.

For the representation on the screen, the procedure is the same as for Hermite curves, but the coordinates of 4 points are used for each control interval.

c. 3D curves in the form of B-Spline

B-Spline curves use a sequence of *n* control points $P_1, P_2, \ldots, P_n$ through which, in the general case, they do not pass. The calculations of the coordinates of the intermediate points are made using a formula of the well-known form:

$$K^{i,i+1}(t) = [T] \times [M_S] \times [G_S]^{i,i+1} \quad (24)$$

The indices $i, i+1$ show us that the formula is used for approximation between the control points P_i and P_{i+1} with $i \in [2, n-2]$.

$[M_S]$ is the Spline matrix and has the form:

$$[M_S] = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad (25)$$

$[G]^{i,\ i+1}$ is the geometric component of the B-Spline shape used between points P_i and P_{i+1}. To determine the shape of the curve between P_i and P_{i+1}, the B-Spline shape uses the coordinates of points P_{i-1}, P_i, P_{i+1} and P_{i+2}.

$$[G_S]^{i,i+1} = \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \quad (26)$$

The formulas for calculating the coordinates are:

$$\begin{cases} x(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} x_{i-1} \\ x_i \\ x_{i+1} \\ x_{i+2} \end{bmatrix} \times \frac{1}{6} \\ y(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} y_{i-1} \\ y_i \\ y_{i+1} \\ y_{i+2} \end{bmatrix} \times \frac{1}{6} \\ z(t) = [t^3, t^2, t, 1] \times \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} z_{i-1} \\ z_i \\ z_{i+1} \\ z_{i+2} \end{bmatrix} \times \frac{1}{6} \end{cases} \quad (27)$$

3D bicubic surfaces – Using 2 families of 3D cubic curves, we can define a curved surface in space, as seen in Figure 8. [7]

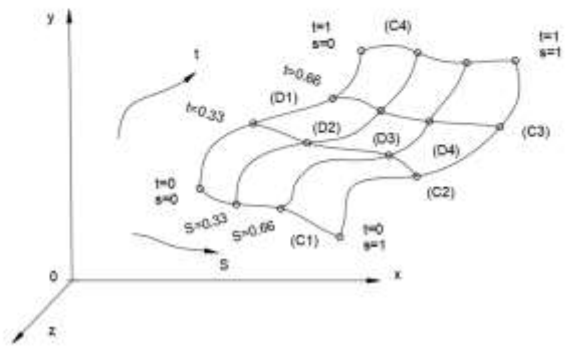

Fig. 8. *3D bicubic surfaces*

The family of 3D Cubic curves *C* can be obtained by introducing into the equation: $x = x(t), y = y(t),$ and $z = z(t)$, another parameter *s* that varies between 0 and 1.

The curves in the *C* family are obtained for various values of *s*.

Analogously, the curves in the *D* family are obtained by introducing into the equations: $x = x(s)$,

$y = y(s)$, and $z = z(s)$, another parameter t varying between 0 and 1.

In order to define a surface using the two families of curves, there must be a common form of their equations written with 2 parameters: $x = x(s,t), y = y(s,t)$, and $z = z(s,t)$. These relations represent the equations of a 3D curved surface, called a bicubic surface (bi = has 2 parameters; cubic = each of the surface parameters appears to the maximum power of 3).

The accuracy of the surface rendering depends on the choice of step sizes ps and pt. Usually, $ps = pt$ is used, and the general form of the expression of a coordinate function of the parameters s and t is:

$$\begin{aligned} K(s,t) &= a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + \\ &+a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + \\ &+a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + \\ &+a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44} \end{aligned} \quad (28)$$

If we note:

$$[C] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, [S] = [s^3 s^2\ s\ 1] \quad (29)$$

and $[T] = [t^3 \times t^2 \times t \times 1]$, the above relationship becomes: $K_{(s,\ t)} = [S] \times [C] \times [T]^T$. As with 3D curves, 3D curved surfaces can be defined in the Hermite, Bezier, and B-Spline forms.

a) Bicubic surfaces in Hermite form

A Hermite bicubic surface is defined by 4 points in space, usually denoted P_{00}, P_{01}, P_{10}, and P_{11}, corresponding to the extreme values (0 and 1) for *s* and *t*, as well as by 3 tangents to the surface at each of these points.

The situation is presented in Figure 9.

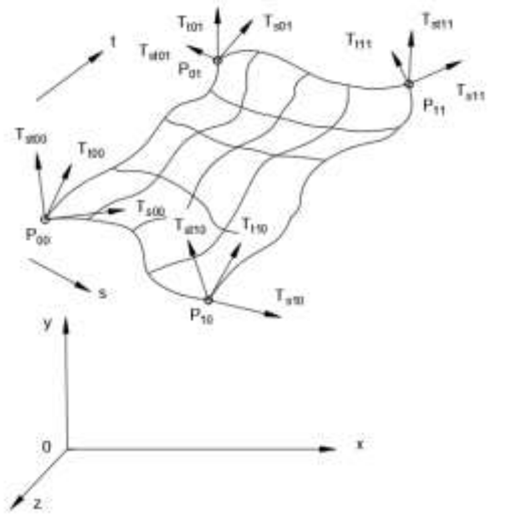

Fig. 9. *Bicubic surfaces in Hermite form*

Each tangent vector has components determined by the 1st and 2nd order derivatives of the functions $x = x(s,t),\ y = y(s,t),\ z = z(s,t)$.

$$\begin{aligned} &P_{00}\big(x(0,0), y(0,0), z(0,0)\big) \\ &P_{10}\big(x(1,0), y(1,0), z(1,0)\big) \\ &P_{01}\big(x(0,1), y(0,1), z(0,1)\big) \\ &P_{11}\big(x(1,1), y(1,1), z(1,1)\big) \\ &T_{sij} = \left(\frac{dx}{ds}\Big|_{i,j}, \frac{dy}{ds}\Big|_{i,j}, \frac{dz}{ds}\Big|_{i,j}\right) \\ &T_{tij} = \left(\frac{dx}{dt}\Big|_{i,j}, \frac{dy}{dt}\Big|_{i,j}, \frac{dz}{dt}\Big|_{i,j}\right) \\ &T_{stij} = \left(\frac{d^2x}{ds\,dt}\Big|_{i,j}, \frac{d^2y}{ds\,dt}\Big|_{i,j}, \frac{d^2z}{ds\,dt}\Big|_{i,j}\right) \end{aligned} \quad (30)$$

The conditions required for a Hermite surface are:

1) Let it pass through the 4 points.
2) Let it have 3 tangents given at each of the 4 points.

That is:

$$1. \begin{cases} x(0,0) = x_{00} \\ y(0,0) = y_{00} \\ z(0,0) = z_{00} \end{cases} \begin{cases} x(1,0) = x_{10} \\ y(1,0) = y_{10} \\ z(1,0) = z_{10} \end{cases} \begin{cases} x(0,1) = x_{01} \\ y(0,1) = y_{01} \\ z(0,1) = z_{01} \end{cases}$$

$$and \begin{cases} x(1,1) = x_{11} \\ y(1,1) = y_{11} \\ z(1,1) = z_{11} \end{cases}$$

$$2. \begin{cases} \frac{dx}{ds}\Big|_{i,j} = \left(\frac{dx}{ds}\right)_{i,j} \\ \frac{dy}{ds}\Big|_{i,j} = \left(\frac{dy}{ds}\right)_{i,j} \\ \frac{dz}{ds}\Big|_{i,j} = \left(\frac{dz}{ds}\right)_{i,j} \end{cases} with\ i,j \in \{0,1\},$$

$$\begin{cases} \frac{dx}{dt}\Big|_{i,j} = \left(\frac{dx}{dt}\right)_{i,j} \\ \frac{dy}{dt}\Big|_{i,j} = \left(\frac{dy}{dt}\right)_{i,j} \\ \frac{dz}{dt}\Big|_{i,j} = \left(\frac{dz}{dt}\right)_{i,j} \end{cases} with\ i,j \in \{0,1\}.$$

$$\begin{cases} \frac{d^2x}{dsdt}\Big|_{i,j} = \left(\frac{d^2x}{dsdt}\right)_{i,j} \\ \frac{d^2y}{dsdt}\Big|_{i,j} = \left(\frac{d^2y}{dsdt}\right)_{i,j} \\ \frac{d^2z}{dsdt}\Big|_{i,j} = \left(\frac{d^2z}{dsdt}\right)_{i,j} \end{cases} with\ i,j \in \{0,1\}.$$

(31)

Using the general form of the relations for determining the coordinates of points on a 3D bicubic surface, we obtain:

$$\begin{cases} K(0,0) = P_{00} \\ K(1,0) = P_{10} \\ K(0,1) = P_{01} \\ K(1,1) = P_{11} \end{cases}, \begin{cases} \frac{dK}{ds}\Big|_{0,0} = T_{s\,00} \\ \frac{dK}{ds}\Big|_{1,0} = T_{s\,10} \\ \frac{dK}{ds}\Big|_{0,1} = T_{s\,01} \\ \frac{dK}{ds}\Big|_{1,1} = T_{s\,11} \end{cases}, \begin{cases} \frac{dK}{dt}\Big|_{0,0} = T_{t\,00} \\ \frac{dK}{dt}\Big|_{1,0} = T_{t\,10} \\ \frac{dK}{dt}\Big|_{0,1} = T_{t\,01} \\ \frac{dK}{dt}\Big|_{1,1} = T_{t\,11} \end{cases}$$

$$and \begin{cases} \left.\frac{d^2K}{dsdt}\right|_{0,0} = T_{st\ 00} \\ \left.\frac{d^2K}{dsdt}\right|_{1,0} = T_{st\ 10} \\ \left.\frac{d^2K}{dsdt}\right|_{0,1} = T_{st\ 01} \\ \left.\frac{d^2K}{dsdt}\right|_{1,1} = T_{st\ 11} \end{cases} \quad (32)$$

Here $P_{00}, P_{10}, P_{01}, and\ P_{11}$ are position constants, and $T_{sij}, T_{tij}, and\ T_{stij}$ are constants determined by tangents.

Taking into account that:

$$K(s,t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + \\ +a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + \\ +a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + \\ +a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44} = [S] \times [C] \times [T]^T \quad (33)$$

We have:

$$\frac{dK(s,t)}{ds} = 3a_{11}s^2t^3 + 3a_{12}s^2t^2 + 3a_{13}s^2t + 3a_{14}s^2 + \\ +2a_{21}st^3 + 2a_{22}st^2 + 2a_{23}st + 2a_{24}s + \\ +a_{31}t^3 + a_{32}t^2 + a_{33}t + a_{34}$$

$$\frac{dK(s,t)}{dt} = 3a_{11}s^3t^2 + 2a_{12}s^3t + a_{13}s^3 + \\ +3a_{21}s^2t^2 + 2a_{22}s^2t + a_{23}s^2 + \\ +3a_{31}st^2 + 2a_{32}st + a_{33}s + 3a_{41}t^2 + 2a_{42}t + a_{43}$$

$$\frac{dK(s,t)}{dsdt} = 9a_{11}s^2t^2 + 6a_{12}s^2t + 3a_{13}s^2 + \\ +6a_{21}st^2 + 4a_{22}st + 2a_{23}s + \\ +3a_{31}t^2 + 2a_{32}t + a_{33} \quad (34)$$

Proceeding further, as for 3D curves in Hermite form, we will obtain a relationship of the form:

$$K(s,t) = [S] \times [M_H] \times [T]^T \quad (35)$$

It can be demonstrated that:

$$[M_H] = [M_h] \times [Q_h] \times [M_H]^T \quad (36)$$

Where $[M_h]$ is the Hermite matrix, and $[Q_h]$ is a Hermite geometry matrix that has the form:

$$[Q_h] = \begin{bmatrix} P_{00} & P_{01} & T_{t00} & T_{t01} \\ P_{10} & P_{11} & T_{t10} & T_{t11} \\ T_{s00} & T_{s01} & T_{st00} & T_{st01} \\ T_{s10} & T_{s11} & T_{st10} & T_{st11} \end{bmatrix} \quad (37)$$

The final calculation relationships will be:

$$x(s,t) = [S] \times [M_h] \times \\ \times \begin{bmatrix} x_{00} & x_{01} & \frac{dx}{dt_{00}} & \frac{dx}{dt_{01}} \\ x_{10} & x_{11} & \frac{dx}{dt_{10}} & \frac{dx}{dt_{11}} \\ \frac{dx}{ds_{00}} & \frac{dx}{ds_{01}} & \frac{d^2x}{ds\,dt_{00}} & \frac{d^2x}{ds\,dt_{01}} \\ \frac{dx}{ds_{10}} & \frac{dx}{ds_{11}} & \frac{d^2x}{ds\,dt_{10}} & \frac{d^2x}{ds\,dt_{11}} \end{bmatrix} \times [M_h]^T \times [T]^T$$

$$y(s,t) = [S] \times [M_h] \times \\ \times \begin{bmatrix} y_{00} & y_{01} & \frac{dy}{dt_{00}} & \frac{dy}{dt_{01}} \\ y_{10} & y_{11} & \frac{dy}{dt_{10}} & \frac{dy}{dt_{11}} \\ \frac{dy}{ds_{00}} & \frac{dy}{ds_{01}} & \frac{d^2y}{ds\,dt_{00}} & \frac{d^2y}{ds\,dt_{01}} \\ \frac{dy}{ds_{10}} & \frac{dy}{ds_{11}} & \frac{d^2y}{ds\,dt_{10}} & \frac{d^2y}{ds\,dt_{11}} \end{bmatrix} \times [M_h]^T \times [T]^T$$

$$z(s,t) = [S] \times [M_h] \times \\ \times \begin{bmatrix} z_{00} & z_{01} & \frac{dz}{dt_{00}} & \frac{dz}{dt_{01}} \\ z_{10} & z_{11} & \frac{dz}{dt_{10}} & \frac{dz}{dt_{11}} \\ \frac{dz}{ds_{00}} & \frac{dz}{ds_{01}} & \frac{d^2z}{ds\,dt_{00}} & \frac{d^2z}{ds\,dt_{01}} \\ \frac{dz}{ds_{10}} & \frac{dz}{ds_{11}} & \frac{d^2z}{ds\,dt_{10}} & \frac{d^2z}{ds\,dt_{11}} \end{bmatrix} \times [M_h]^T \times [T]^T \quad (38)$$

The continuity conditions at the junction of 2 Hermite surfaces are relatively simple: the ends of the boundary curve must coincide, and the tangents to the surfaces at these ends must be proportional.

b) Bicubic surfaces in Bezier form

As with 3D curves in the Bezier form, to define a Bezier surface, we use the 4 control points corresponding to the values $(0,0), (0,1), (1,0), \text{and } (1,1)$ of the s and t parameters. Additionally, 12 other control points are used through which the tangents to the surface are specified (Figure 10).

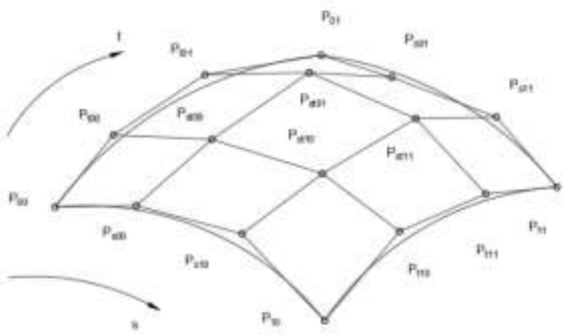

Fig. 10. *Bicubic surfaces in Bezier form*

The geometry of a Bezier surface is therefore characterized by the coordinates of 16 control points.

The Bezier geometry matrix (Q_b) will have the form:

$$[Q_b] = \begin{bmatrix} P_{00} & P_{s00} & P_{s10} & P_{10} \\ P_{t00} & P_{st00} & P_{st10} & P_{t10} \\ P_{t01} & P_{st01} & P_{st11} & P_{t11} \\ P_{01} & P_{s01} & P_{s11} & P_{11} \end{bmatrix} \quad (39)$$

The general form of the relationship used to determine the coordinates corresponding to the pair of parameters (s, t) in the case of Bezier surfaces is:

$$K(s,t) = [S] \times [M_b] \times [Q_b] \times [M_b]^T \times [T]^T \quad (40)$$

Substituting, we obtain the calculation relations:

$$x(s,t) = [S] \times [M_b] \times \begin{bmatrix} x_{00} & x_{s00} & x_{s10} & x_{10} \\ x_{t00} & x_{st00} & x_{st10} & x_{t10} \\ x_{t01} & x_{st01} & x_{st11} & x_{t11} \\ x_{01} & x_{s01} & x_{s11} & x_{11} \end{bmatrix} \times \times [M_b]^T \times [T]^T$$

$$y(s,t) = [S] \times [M_b] \times \begin{bmatrix} y_{00} & y_{s00} & y_{s10} & y_{10} \\ y_{t00} & y_{st00} & y_{st10} & y_{t10} \\ y_{t01} & y_{st01} & y_{st11} & y_{t11} \\ y_{01} & y_{s01} & y_{s11} & y_{11} \end{bmatrix} \times \times [M_b]^T \times [T]^T$$

$$z(s,t) = [S] \times [M_b] \times \begin{bmatrix} z_{00} & z_{s00} & z_{s10} & z_{10} \\ z_{t00} & z_{st00} & z_{st10} & z_{t10} \\ z_{t01} & z_{st01} & z_{st11} & z_{t11} \\ z_{01} & z_{s01} & z_{s11} & z_{11} \end{bmatrix} \times \times [M_b]^T \times [T]^T \quad (41)$$

Bezier surfaces pass through the four corner points $(P_{00}, P_{01}, P_{10}, \text{and } P_{11})$, and generally do not pass through the other control points.

The problem of continuity between 2 connected Bezier surfaces along an edge is solved by ensuring collinearity for 4 pairs of 3 points.

The situation is shown in Figure 11.

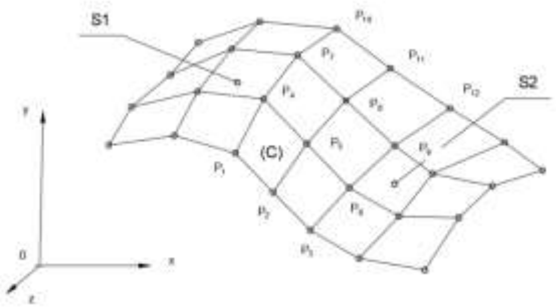

Fig. 11. *Continuity problem of 2 connected Bezier surfaces*

To ensure the continuity of the surfaces S_1 and S_2 along the curve (C), we must choose the control points so that the corresponding ones on the curve (C) (for both surfaces) coincide, and the triplets of points $P_1P_2P_3, P_4P_5P_6, P_7P_8P_9$, and $P_{10}P_{11}P_{12}$ are collinear.

c) Bicubic surfaces in B-Spline form

A bicubic surface in the form of a B-Spline is defined by 16 control points, by analogy with 3D B-Spline curves. The calculation relations for the coordinates corresponding to the parameter pair (s, t) have the general form:

$$K(s,t) = [S] \times [M_s] \times \left[Q_{s}{}_{i,j+1;i+1,j+1}^{i,i;i+1,j}\right] \times [M_s]^T \times [T^T] \quad (42)$$

Here, the indices specify that the intermediate points are calculated within the perimeter delimited by the control points:

$P_{ij}, P_{i+1,\ j}, P_{i,\ j+1}, P_{i+1,\ j+1}$, where $i, j \in [2, n-2]$. This assumes that we use an $n \times n$ point grid to describe the entire surface.

The situation is illustrated in Figure 12.

The 16 points that determine the matrix $\left[Q_{s}{}_{i,j+1\ \ i+1,j+1}^{i,j\ \ \ \ i+1,j}\right]$ are:

$$\begin{bmatrix} P_{i-1,j-1} & P_{i-1,j} & P_{i-1,j+1} & P_{i-1,j+2} \\ P_{i,j-1} & P_{i,j} & P_{i,j+1} & P_{i,j+2} \\ P_{i+1,j-1} & P_{i+1,j} & P_{i+1,j+1} & P_{i+1,j+2} \\ P_{i+2,j-1} & P_{i+2,j} & P_{i+2,j+1} & P_{i+2,j+2} \end{bmatrix} \quad (43)$$

A B-Spline surface does not generally pass through the given control points.

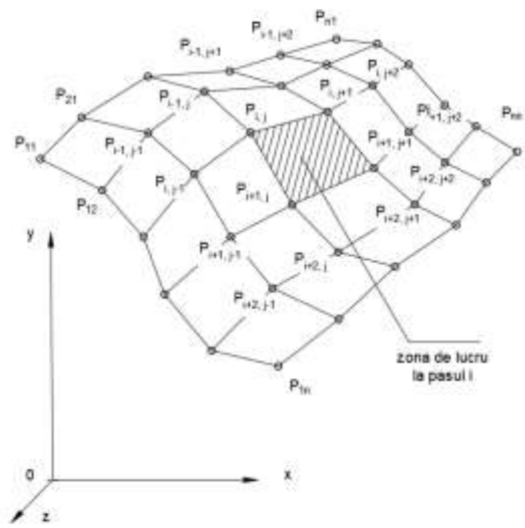

Fig. 12. *Bicubic surfaces in B-Spline form*

3. CONCLUSIONS

The numerical representation of objects, particularly that of 3D bodies, provides a powerful tool for modeling, analyzing, and visualizing complex structures. Through techniques such as discretization and computational geometry, 3D bodies can be accurately represented in digital form, enabling advancements in fields such as engineering [8-12], computer graphics, and scientific simulations. This approach facilitates precise design, optimization, and manipulation of objects, contributing to innovation across multiple disciplines.

To elaborate further, this process involves translating physical or conceptual structures into mathematical models that can be processed, analyzed, and visualized by computers. This process is central to various industries, including manufacturing, architecture, animation, and virtual reality.

The ability to numerically represent 3D bodies opens up significant possibilities. For instance, in engineering and product design, this representation allows for precise simulations of how objects will behave under different conditions, such as stress, heat, or motion, before physical prototypes are created. This reduces costs, accelerates the development process, and improves product quality.

Overall, the representation of 3D bodies through numerical methods not only enhances the accuracy and efficiency of design and analysis, but also drives innovation in both practical applications and creative fields.

REFERENCES

[1] Dogaru D., *Metode noi în proiectare, Elemente de grafică tridimensională (New methods in design, Three-dimensional graphic elements)*, Scientific and Encyclopedic Publishing House, Bucharest, 1988;

[2] Wolfgang Böhm, *Generating the Bézier points of B-spline curves and surfaces*, Computer-Aided Design, Volume 13, Issue 6, November 1981, Pages 365-366, https://doi.org/10.1016/0010-4485(81)90213-X;

[3] M.I.G. Bloor, M.J. Wilson, *Using partial differential equations to generate free-form surfaces*, Computer-Aided Design, Volume 22, Issue 4, May 1990, Pages 202-212, https://doi.org/10.1016/0010-4485(90)90049-I;

[4] Hoffmann, C., Hopcroft, J., *Automatic surface generation in computer aided design.* The Visual Computer 1, 92–100 (1985). https://doi.org/10.1007/BF01898351;

[5] T.S. Lan, S.H. Lo, *Finite element mesh generation over analytical curved surfaces*, Computers & Structures, Volume 59, Issue 2, 17 April 1996, Pages 301-309, https://doi.org/10.1016/0045-7949(95)00261-8;

[6] Uri Itai, Nira Dyn, *Generating surfaces by refinement of curves*, Journal of Mathematical Analysis and Applications, Volume 388, Issue 2, 15 April 2012, Pages 913-928, https://doi.org/10.1016/j.jmaa.2011.10.035;

[7] Márta Szilvási-Nagy, Teréz P. Vendel, *Generating curves and swept surfaces by blended circles*, Computer Aided Geometric Design, Volume 17, Issue 2, February 2000, Pages 197-206, https://doi.org/10.1016/S0167-8396(99)00045-X;

[8] Oancea N., *Surfaces Generation through Winding, Vol. I*, Galati University Press: Galati, Romania, 2004;

[9] Oancea N., *Surfaces Generation through Winding, Vol. II*, Galati University Press: Galati, Romania, 2004;

[10] Oancea N., *Surfaces Generation through Winding, Vol. III*, Galati University Press: Galati, Romania, 2004;

[11] V.G. Teodor, V. Păunoiu, N. Baroiu, F. Susac, *Optimization of the measurement path for the car body parts inspection*, Measurement, Vol. 146, pp. 15-23, ISSN 0263-2241, 2019;

[12] V.G. Teodor, N. Baroiu, F. Susac, *The synthesis of new algorithms for CAD profiling of cutting tools*, Lambert Academic Publishing, ISBN 978-613-7-08923-1, 2018