
Czech Technical University in Prague
Faculty of Mechanical Engineering
Department of Technical Mathematics

**CURVES AND SURFACES
FOR COMPUTER GRAPHICS**



Ivana Linkeová

Prague 2022

Contents

Introduction	6
1 Curves and surfaces – fundamental properties	9
1.1 Definition and properties of curves	9
1.2 Curves joining	19
1.3 Analysis of curves in Rhinoceros	21
1.4 Definition and properties of surfaces	26
1.5 Patching	33
1.6 Analysis of surfaces in Rhinoceros	34
2 Curves modelling	39
2.1 Ferguson cubic curve	40
2.1.1 Properties of Ferguson cubic curve	40
2.1.2 Hermite polynomials derivation	42
2.1.3 Continuity at common point of Ferguson cubic curves	43
2.1.4 Ferguson cubic curve in Rhinoceros	45
2.2 Bézier curve	45
2.2.1 Properties of Bézier curve	49
Bézier and Ferguson cubic curves interrelation	49
2.2.2 Bernstein polynomials derivation	51
2.2.3 De Casteljaun algorithm for Bézier curve	53
2.2.4 Continuity at common point of Bézier curves	54
2.2.5 Bézier curve in Rhinoceros	61
2.3 Coons cubic curve	64
2.3.1 Properties of Coons cubic curve	66
Coons, Bézier and Ferguson cubic curves interrelation	67
2.3.2 Coons polynomials derivation	69
2.3.3 Construction of Coons cubic curve endpoints	69
2.3.4 Coons cubic curve in Rhinoceros	70
2.4 Coons cubic B-spline	70
2.4.1 Properties of Coons cubic B-spline	72
2.4.2 Coons cubic B-spline in Rhinoceros	76
2.5 Uniform clamped B-spline curve of third degree	76
2.5.1 Properties of clamped curve	78
2.5.2 Construction of knots of clamped curve	79
2.5.3 Clamped curve in Rhinoceros	83

3 Surfaces modelling	85
3.1 Vector equation of a surface	85
3.2 Ruled surface	87
3.2.1 Properties of ruled surface	91
3.2.2 Patching – patches from ruled surface	91
3.2.3 Ruled surface in Rhinoceros	92
3.3 Surface of hyperbolic paraboloid	94
3.3.1 Properties of surface of hyperbolic paraboloid	96
3.3.2 Patching – patches from surface of hyperbolic paraboloid	96
3.3.3 Surface of hyperbolic paraboloid in Rhinoceros	96
3.4 Coons bilinear surface	97
3.4.1 Properties of Coons bilinear surface	100
3.4.2 Vector equation of Coons bilinear surface derivation	100
3.4.3 Patching – patches from Coons bilinear surface	102
3.4.4 Coons bilinear surface in Rhinoceros	103
3.5 Bézier surface	104
3.5.1 Bézier surface with control mesh above unit squared grid	105
3.5.2 Properties of Bézier surface	110
Bézier bicubic surface and Coons bilinear surface interrelation	112
3.5.3 Vector equation of Bézier surface derivation	116
Repeated linear interpolation of four points	116
Bézier curve with variable control points	117
3.5.4 De Casteljaou algorithm for Bézier surface	117
De Casteljaou algorithm based on a surface character	118
De Casteljaou algorithm based on a curve character	119
3.5.5 Patching – patches from Bézier surface	122
3.5.6 Bézier surface in Rhinoceros	125
3.6 Uniform clamped bicubic B-spline surface	130
3.6.1 Properties of clamped curve	130
3.6.2 Clamped surface in Rhinoceros	131
3.7 Clamped surface application	136
A Rhinoceros – list of commands used	139
Literatura	141

Introduction

This book has been written as a textbook for Computer Graphics course (second semester, Faculty of Mechanical Engineering, Czech Technical University in Prague), but it can be also used for self-study.

Computer modelling of free-form curves and surfaces represents a highly-developed part of computer graphics widely used in many applications such as numerically controlled manufacturing, computer aided simulations and visualizations of scientific experiments, film animations, computer games, etc.

Nowadays, the most of CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) systems are based on NURBS (NonUniform Rational B-Spline) representation. In NURBS representation, curves and surfaces are parameterized by piecewise nonuniform rational functions. This relatively complicated mathematical description does not allow to study the NURBS representation directly because of the minimal teaching hours of Computer Graphics course. However, understanding of NURBS theory is considered as a necessary prerequisite for effective usage of CAD/CAM systems in mechanical engineering.

To penetrate the rules of NURBS curves and surfaces modelling, the following elementary special cases of NURBS curves and surfaces are primarily described in this book: Ferguson cubic curve, Bézier curve, Coons cubic curve, ruled surface, surface of hyperboloid paraboloid and Bézier surface. All these curves and surfaces are created by only one segment, i.e. it is possible to mathematically express them by only one equation. Thereafter, special attention is focused on a problem of continuity caused by joining elementary models of curves and surfaces to design more complicated shapes. Thus, segment-wise curves and surfaces are created with mathematical expression including a set of individual equations of all elementary parts. In this textbook, the following segment-wise curves and surfaces are described: Coons cubic B-spline, uniform clamped B-spline cubic curve and uniform clamped B-spline bicubic surface.

The theoretical part to each individual theme is presented in this textbook together with detailed solved examples demonstrating all important properties of curves and surfaces and their interrelations. At the end of each section, a number of exercises is provided to practise the material studied.

Practical design of curves and surfaces in software Rhinoceros – NURBS modelling for Windows (www.rhino3d.com) is considered to be an inseparable part of this textbook. Rhinoceros is based on NURBS representation and can create, edit, analyze, document, render and animate NURBS curves, surfaces, and solids with no limits on complexity, degree, or size.

At the end of each chapter, a list of commands, tools and procedures of constructions in Rhinoceros is given to guide how to practically create the curves and surfaces and demonstrate their properties. The order of these commands, tools and procedures corresponds to the order of topics described in the theoretical part of a chapter. If construction of a certain figure or visualization of a certain geometric property is not included in Rhinoceros directly, an alternative procedure of construction is introduced in the list. These indirect alternative procedures do not present the only method of solution. Certainly, the reader of this textbook and Rhinoceros user

will find out his or her own ways to achieve the same aim.

In Appendix [A](#), alphabetic list of Rhinoceros commands used in this textbook is given together with their toolbar buttons and menu paths.

I would like to thank all the people who contributed to this textbook by providing useful advices, by reading chapters and suggesting changes, and by finding and correcting errors. Especially, I would like to thank the many students whose curious questions often inspired me to prepare many examples given in this text.

Prague, December 2022

Ivana Linkeová

Chapter 1

Curves and surfaces – fundamental properties

In this chapter, the most important terms, definitions and fundamental properties of curves and surfaces are given. This information will be used in all of the following chapters.

1.1 Definition and properties of curves

From physical point of view, a *curve* is considered to be a trajectory of a point moving in a plane or space in dependence on time. From geometric point of view, a curve is characterized by one-parametric sequence of *points* in the space of n -th dimension. These points represent function values of *univariate point function*. A *position vector* (also known as location vector or radius vector) of a curve point starts at the origin of Cartesian coordinate system and terminates at the curve point. The position vectors represent function values of a *univariate vector function*.

■ **Definition 1.1 – Univariate point and vector function.** Let $I \subset R$ be an interval of real numbers. Then continuous mapping F of interval I into the Euclidean space R^n , $n > 1$ is a *univariate point function*

$$F(t) = [x_1(t), x_2(t), \dots, x_n(t)], \quad (1.1)$$

and continuous mapping \mathbf{F} of interval I into the Euclidean vector space V^n , $n > 1$ is a *univariate vector function*

$$\mathbf{F}(t) = (x_1(t), x_2(t), \dots, x_n(t)). \quad (1.2)$$

Interval I is referred to as a *parametrization domain* of point or vector function, argument t is called a *parameter*.

Real functions

$$x_1 = x_1(t), x_2 = x_2(t), \dots, x_n = x_n(t), t \in I, \quad (1.3)$$

are called *coordinate functions* of univariate point or vector function. \square

Coordinate functions of univariate point or vector function are real univariate functions. Univariate point or vector function is a generalization of a univariate real function. It is easy to prove the following statements as well as related properties of univariate point or vector function.

- Point function $F(t)$, $t \in I$ and a vector function $\mathbf{F}(t)$, $t \in I$ has a limit at point $\alpha \in I$ equal to the point

$$F_\alpha = [x_1(\alpha), x_2(\alpha), \dots, x_n(\alpha)]$$

and the vector

$$\mathbf{F}_\alpha = (x_1(\alpha), x_2(\alpha), \dots, x_n(\alpha)),$$

respectively, if the limits of their coordinate functions are as follows

$$\lim_{t \rightarrow \alpha} x_1(t) = x_1(\alpha), \lim_{t \rightarrow \alpha} x_2(t) = x_2(\alpha), \dots, \lim_{t \rightarrow \alpha} x_n(t) = x_n(\alpha). \quad (1.4)$$

- Point function $F(t)$, $t \in I$ and vector function $\mathbf{F}(t)$, $t \in I$ is continuous at point $\alpha \in I$ if its coordinate functions are continuous at point α , i.e. the limits (1.4) at point $\alpha \in I$ exist and are equal to the function values of coordinate functions at point α .
- Point function $F(t)$, $t \in I$ and vector function $\mathbf{F}(t)$, $t \in I$ is continuous on I if its coordinate functions are continuous on I .
- Point function $F(t)$, $t \in I$ and vector function $\mathbf{F}(t)$, $t \in I$ has at point $\alpha \in I$ derivative of k -th order (the function is k -times differentiable) if derivatives of k -th order $x_1^{(k)}(\alpha)$, $x_2^{(k)}(\alpha), \dots, x_n^{(k)}(\alpha)$ of its coordinate functions exist. Then, the derivative $F^{(k)}(\alpha)$ of point function F at point $\alpha \in I$ is a vector

$$F^{(k)}(\alpha) = \left(x_1^{(k)}(\alpha), x_2^{(k)}(\alpha), \dots, x_n^{(k)}(\alpha) \right), \quad (1.5)$$

and the derivative $\mathbf{F}^{(k)}(\alpha)$ of vector function \mathbf{F} at point $\alpha \in I$ is a vector

$$\mathbf{F}^{(k)}(\alpha) = \left(x_1^{(k)}(\alpha), x_2^{(k)}(\alpha), \dots, x_n^{(k)}(\alpha) \right). \quad (1.6)$$

- Point function $F(t)$, $t \in I$ and vector function $\mathbf{F}(t)$, $t \in I$ is C^k continuous at point $\alpha \in I$ if the derivatives of its coordinate functions are continuous at point $\alpha \in I$ to k -th order.
- Point function $F(t)$, $t \in I$ and vector function $\mathbf{F}(t)$, $t \in I$, is C^k continuous on I if the derivatives of its coordinate functions are continuous on I to k -th order.

■ **Definition 1.2 – Curve.** A *curve* is any connected non-empty subset k in R^n , which is a continuous mapping of the real interval $I \subset R$. If $n = 2$, the curve is referred to as a *planar*. If $n = 3$, the curve is referred to as a *spatial*¹.

If planar and spatial curve is analytically represented by point function (1.1) which is defined, continuous and at least once differentiable on I , the planar and spatial curve is given by *point equation*

$$P(t) = [x(t), y(t)], \quad t \in I, \quad \text{and} \quad P(t) = [x(t), y(t), z(t)], \quad t \in I, \quad (1.7)$$

respectively.

If planar and spatial curve is analytically represented by vector function (1.2) which is defined, continuous and at least once differentiable on I , the planar and spatial curve is given by *vector equation*

$$\mathbf{P}(t) = (x(t), y(t)), \quad t \in I, \quad \text{and} \quad \mathbf{P}(t) = (x(t), y(t), z(t)), \quad t \in I, \quad (1.8)$$

respectively.

Parametric equations of a planar and spatial curve are obtained by itemizing the coordinate functions of the curve given by point equation (1.7) or vector equation (1.8)

¹In this textbook, the curves in higher dimensional spaces are not considered.

$$\begin{aligned}x &= x(t), \\y &= y(t), \quad t \in I\end{aligned}\tag{1.9}$$

and

$$\begin{aligned}x &= x(t), \\y &= y(t), \\z &= z(t), \quad t \in I,\end{aligned}\tag{1.10}$$

respectively.

The curve defined by parametric equations (1.9) or (1.10) is referred to as a *curve defined parametrically*. Equations (1.9) or (1.10) are referred to as a *parametric expression* or *parametrization* of a curve.

If interval $I = [a, b]$ is closed, the curve is referred to as a *regular element of a curve*. \square

Remark: With respect to the common terminology of computer graphics, the term *curve* means a regular element of a curve in the following text of this textbook.

A spatial curve is only considered in the following definitions given in this section. The necessary modifications to get the definitions related to a planar curve are obvious.

■ **Definition 1.3 – Curve point.** The function value of point function (1.1) for $\alpha \in [a, b]$

$$P(\alpha) = [x(\alpha), y(\alpha), z(\alpha)],\tag{1.11}$$

and the terminal point of position vector – function value of vector function (1.2) for $\alpha \in [a, b]$

$$\mathbf{P}(\alpha) = (x(\alpha), y(\alpha), z(\alpha)),\tag{1.12}$$

respectively, is referred to as a *curve point*. The parameter value $t = \alpha$ that unambiguously determines the position of a point on the curve is called a *parametric (curvilinear) coordinate* of the curve point.

The orientation of the curve is defined by orientation of its point or vector function. The curvilinear coordinate of the initial (start) curve point is equal to a , the curvilinear coordinate of the terminal curve point is equal to b . The start and terminal curve points are referred to as *endpoints* of the curve. \square

■ **Example 1.1 – Point, vector and parametric equations of an ellipse.** The terms given in Definitions 1.1 to 1.3 will be explained on a planar curve – ellipse k given by coordinate functions

$$\begin{aligned}x &= x(t) = 4 \cos t, \\y &= y(t) = 3 \sin t, \quad t \in [0, 2\pi].\end{aligned}\tag{1.13}$$

Point function $F(t)$, $t \in [0, 2\pi]$ that maps interval $[0, 2\pi]$ into Euclidean two-dimensional space is as follows

$$F(t) = [4 \cos t, 3 \sin t], \quad t \in [0, 2\pi].\tag{1.14}$$

Ellipse k given by point equation

$$P(t) = [4 \cos t, 3 \sin t], \quad t \in [0, 2\pi]$$

is shown in Fig. 1.1 a). Here, two parameter values $t = 0$ and $t = 2\pi$ are considered and the corresponding point function values $F(0)$ and $F(2\pi)$ are marked. These point function values are equal to the endpoints of the ellipse

$$P(0) = [4 \cos(0), 3 \sin(0)] = [4, 0]$$

and

$$P(2\pi) = [4 \cos(2\pi), 3 \sin(2\pi)] = [4, 0].$$

The endpoints of the ellipse coincide. Additionally, for parameter value $t = \frac{2}{3}\pi$, the point function value $F\left(\frac{2}{3}\pi\right)$ corresponding to the point of the ellipse

$$P\left(\frac{2}{3}\pi\right) = \left[4 \cos\left(\frac{2}{3}\pi\right), 3 \sin\left(\frac{2}{3}\pi\right)\right] = \left[-2, \frac{3}{2}\sqrt{3}\right]$$

is marked in Fig. 1.1 a).

Vector function $\mathbf{F}(t)$, $t \in [0, 2\pi]$ that maps interval $[0, 2\pi]$ into Euclidean two-dimensional vector space is as follows

$$\mathbf{F}(t) = (4 \cos t, 3 \sin t), \quad t \in [0, 2\pi]. \quad (1.15)$$

Ellipse k given by vector equation

$$\mathbf{P}(t) = (4 \cos t, 3 \sin t), \quad t \in [0, 2\pi]$$

is drawn in Fig. 1.1 b). Here, two parameter values $t = 0$ and $t = 2\pi$ are considered. The corresponding vector function values $\mathbf{F}(0)$ and $\mathbf{F}(2\pi)$, i.e. position vectors

$$\mathbf{P}(0) = (4 \cos(0), 3 \sin(0)) = (4, 0)$$

and

$$\mathbf{P}(2\pi) = (4 \cos(2\pi), 3 \sin(2\pi)) = (4, 0)$$

of endpoints of the ellipse are drawn. The position vectors of endpoints of the ellipse coincide. Additionally, for parameter value $t = \frac{2}{3}\pi$, the vector function value $\mathbf{F}\left(\frac{2}{3}\pi\right)$ corresponding to the position vector

$$\mathbf{P}\left(\frac{2}{3}\pi\right) = \left(4 \cos\left(\frac{2}{3}\pi\right), 3 \sin\left(\frac{2}{3}\pi\right)\right) = \left(-2, \frac{3}{2}\sqrt{3}\right)$$

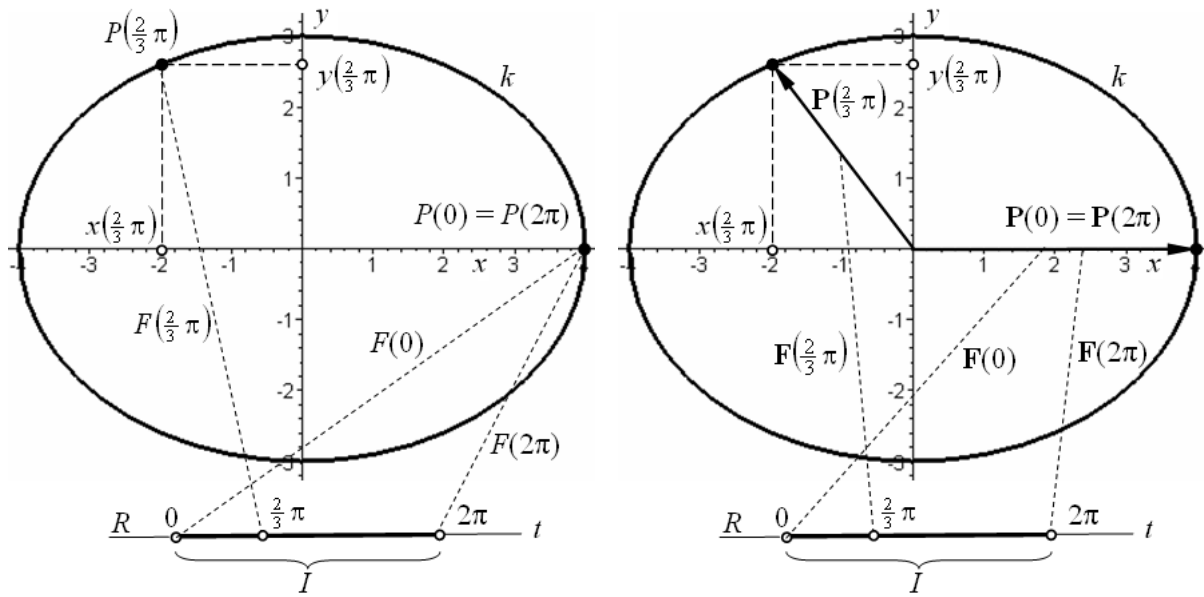
of point $P\left(\frac{2}{3}\pi\right) = \left[-2, \frac{3}{2}\sqrt{3}\right]$ of the ellipse is marked in Fig. 1.1 b).

Parametric equations of the ellipse k are obtained when itemizing coordinate functions of point function (1.14) or vector function (1.15)

$$\begin{aligned} x(t) &= 4 \cos t, \\ y(t) &= 3 \sin t, \quad t \in [0, 2\pi]. \end{aligned} \quad (1.16)$$

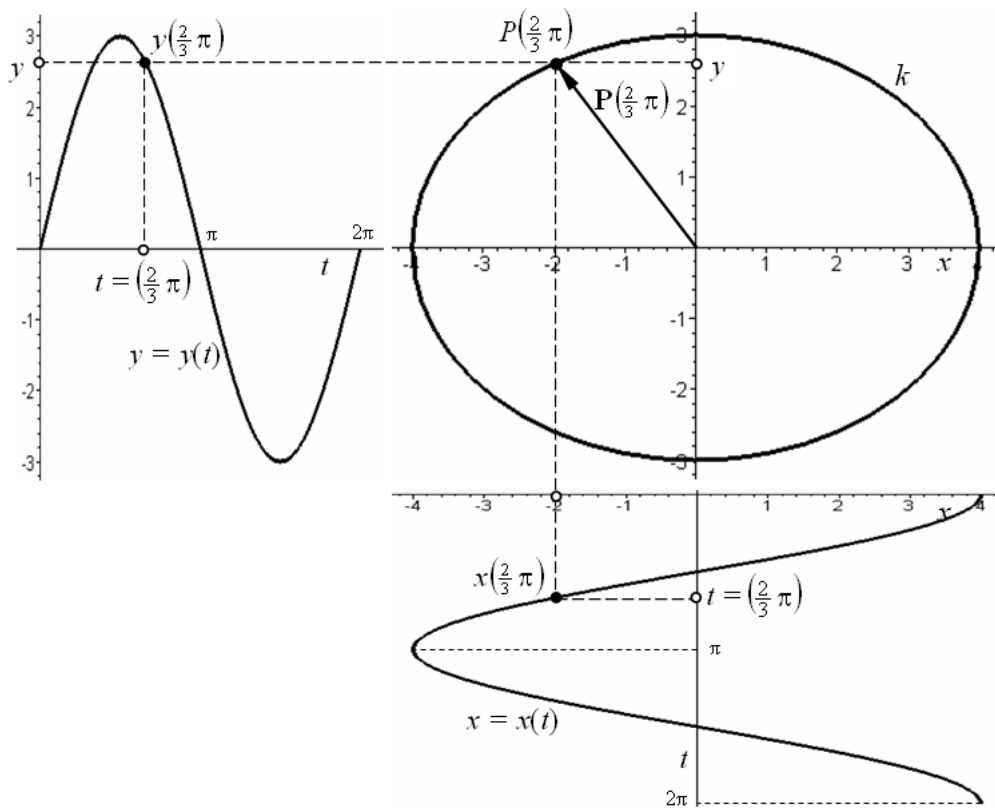
In Fig. 1.1 c), the graph of ellipse k given by parametric equations (1.16) is shown, including suitable oriented graphs of coordinate functions (1.13).

After substitution a chosen parameter value t in coordinate functions, we obtain Cartesian coordinates of a point of ellipse k and, simultaneously, the coordinates of its position vector. The point $P\left(\frac{2}{3}\pi\right) = \left[-2, \frac{3}{2}\sqrt{3}\right]$ and its position vector $\mathbf{P}\left(\frac{2}{3}\pi\right) = \left(-2, \frac{3}{2}\sqrt{3}\right)$ are depicted in Fig. 1.1 c).



a) Ellipse k given by point equation (1.14)

b) Ellipse k given by vector equation (1.15)



c) Ellipse k given by parametric equations (1.16)

Figure 1.1: Point, vector and parametric equations of ellipse k

□

It is obvious that point, vector and parametric equations of a curve are mutually equivalent. Usually, the vector equation is chosen in computer graphics. Accordingly, in the following of this textbook, a curve is designated by $\mathbf{P}(t)$ and the term *curve point* for $t = \alpha$, $\alpha \in [a, b]$ means terminal point of its position vectors $\mathbf{P}(\alpha)$. Therefore, denotation of a curve point is the same as denotation of its position vector, i.e. $\mathbf{P}(\alpha) = (x(\alpha), y(\alpha), z(\alpha))$.

■ **Definition 1.4 – Regular and singular curve point.** Curve point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ is called a *regular curve point* if the vector $\mathbf{P}'(\alpha) = (x'(\alpha), y'(\alpha), z'(\alpha))$ is non-zero vector and only one value of parameter $t = \alpha$ from (a, b) corresponds to this point. Each other curve point is called a *singular curve point*. The coinciding endpoints of the curve are not considered singular points. \square

■ **Definition 1.5 – Tangent vector, binormal vector and principal normal vector at curve point.** Derivation of vector function

$$\mathbf{P}'(t) = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt} \right) = (x'(t), y'(t), z'(t)), \quad t \in [a, b] \quad (1.17)$$

is a vector function that expresses for $\alpha \in [a, b]$ a *tangent vector of a curve* $\mathbf{P}'(t)$ at its regular point $\mathbf{P}(\alpha)$

$$\mathbf{P}'(\alpha) = (x'(\alpha), y'(\alpha), z'(\alpha)). \quad (1.18)$$

Orientation of tangent vector $\mathbf{P}'(t)$ is identical to the orientation of curve $\mathbf{P}(t)$. The unit tangent vector $\mathbf{t}(\alpha)$ at regular point $\mathbf{P}(\alpha)$ of curve $\mathbf{P}(t)$ is given by

$$\mathbf{t}(\alpha) = \frac{\mathbf{P}'(\alpha)}{|\mathbf{P}'(\alpha)|}. \quad (1.19)$$

The straight line given by point $\mathbf{P}(\alpha)$ and direction vector $\mathbf{t}(\alpha)$ is the *tangent line* t_α of curve $\mathbf{P}(t)$ at its point $\mathbf{P}(\alpha)$.

The *binormal vector* is obtained as a cross product of the first and second derivatives of vector function of curve $\mathbf{P}(t)$, $t \in [a, b]$ at its regular and non-inflection (see Definition 1.15) point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$. The unit binormal vector $\mathbf{b}(\alpha)$ is given by

$$\mathbf{b}(\alpha) = \frac{\mathbf{P}'(\alpha) \times \mathbf{P}''(\alpha)}{|\mathbf{P}'(\alpha) \times \mathbf{P}''(\alpha)|}. \quad (1.20)$$

The straight line given by point $\mathbf{P}(\alpha)$ and direction vector $\mathbf{b}(\alpha)$ is the *binormal line* b_α of curve $\mathbf{P}(t)$ at its non-inflection point $\mathbf{P}(\alpha)$.

Cross product of binormal vector and tangent vector of curve $\mathbf{P}(t)$, $t \in [a, b]$ at point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ is called a *principal normal vector*. The unit principal normal vector $\mathbf{n}(\alpha)$ is given by

$$\mathbf{n}(\alpha) = \mathbf{b}(\alpha) \times \mathbf{t}(\alpha). \quad (1.21)$$

The straight line given by point $\mathbf{P}(\alpha)$ and direction vector $\mathbf{n}(\alpha)$ is the *principal normal line* n_α of curve $\mathbf{P}(t)$ at its point $\mathbf{P}(\alpha)$. \square

■ **Definition 1.6 – Double (node) and multiple curve point.** If there exist real numbers

$$\alpha_1, \alpha_2 \in (a, b), \quad \alpha_1 \neq \alpha_2,$$

for which $\mathbf{P}(\alpha_1) = \mathbf{P}(\alpha_2)$, i.e. the curve point given by curvilinear coordinate α_1 and curve point given by curvilinear coordinate α_2 coincide, the point $\mathbf{P}(\alpha_1) = \mathbf{P}(\alpha_2)$ is called the *double curve point* or *curve node*. If there exist $k > 2$ such numbers from the interval I , the point is called the *k-multiple curve point*. \square

The double (multiple) curve point is a point where a curve intersects itself so that two (k) branches of the curve have distinct tangent lines, see Fig. 1.2 b).

■ **Definition 1.7 – Cuspidal curve point.** The curve point, in which the orientation of tangent vector turns up is called the *cuspidal curve point* or *cuspidal point*. □

The curve has two coinciding tangent lines at its cuspidal point, see Fig. 1.2 c).

■ **Definition 1.8 – Angular curve point.** The curve point, in which the direction of tangent line changes by angle $\gamma < 180^\circ$, is called the *angular curve point*. □

At angular point, the curve has two intersecting tangent lines forming the angle γ , see Fig. 1.2 d).

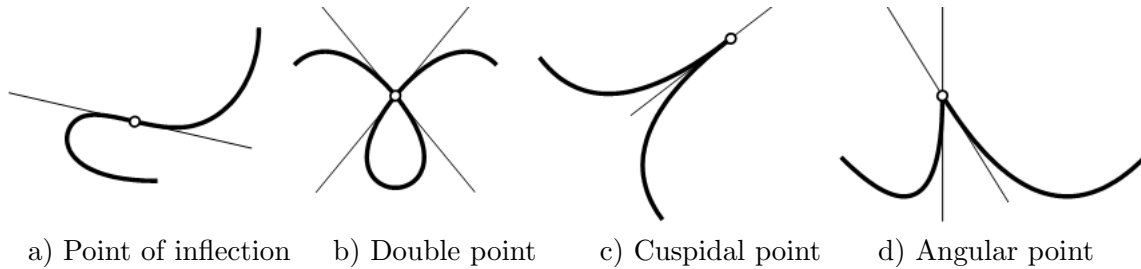


Figure 1.2: Curve points classification

In computer graphics, the derivatives of vector function of a curve and their function values are very important with respect to the shape modelling and curves joining. According to the terminology in computer graphics, in the following in this textbook, the term *k-th curve derivative* means k -th derivative of vector function analytically representing the curve $\mathbf{P}(t)$, $t \in [a, b]$. The first three curve derivatives are denoted with $\mathbf{P}'(t)$, $\mathbf{P}''(t)$ and $\mathbf{P}'''(t)$. The function value of k -th curve derivative for parameter value $t = \alpha$, $\alpha \in [a, b]$ is referred to as the *vector of k-th curve derivative*.

■ **Example 1.2 – Ellipse derivatives.** The first three derivatives of ellipse k given by vector equation (1.15) from Example 1.1 are expressed in this example. Additionally, function values of these derivatives for several parameter values are calculated and their vectors are drawn.

The first three ellipse derivatives are given by

$$\begin{aligned}\mathbf{P}'(t) &= (-4 \sin t, 3 \cos t), \quad t \in [0, 2\pi], \\ \mathbf{P}''(t) &= (-4 \cos t, -3 \sin t), \quad t \in [0, 2\pi], \\ \mathbf{P}'''(t) &= (4 \sin t, -3 \cos t), \quad t \in [0, 2\pi].\end{aligned}$$

In the following table, there are stated the vectors of these derivatives for parameter values $t = \frac{i}{3}\pi$, $i = 0, 1, \dots, 6$. Vector coordinates are rounded to two decimal numbers.

	$t = 0$	$t = \frac{\pi}{3}$	$t = \frac{2}{3}\pi$	$t = \pi$	$t = \frac{4}{3}\pi$	$t = \frac{5}{3}\pi$	$t = 2\pi$
$\mathbf{P}(t)$	(4.00, 0.00)	(2.00, 2.60)	(-2.00, 2.60)	(-4.00, 0.00)	(-2.00, -2.60)	(2.00, -2.60)	(4.00, 0.00)
$\mathbf{P}'(t)$	(0.00, 3.00)	(-3.46, 1.50)	(-3.46, -1.50)	(0.00, -3.00)	(3.46, -1.50)	(3.46, 1.50)	(0.00, 3.00)
$\mathbf{P}''(t)$	(-4.00, 0.00)	(-2.00, -2.60)	(2.00, -2.60)	(4.00, 0.00)	(2.00, 2.60)	(-2.00, 2.60)	(-4.00, 0.00)
$\mathbf{P}'''(t)$	(0.00, -3.00)	(3.46, -1.50)	(3.46, 1.50)	(0.00, 3.00)	(-3.46, 1.50)	(-3.46, -1.50)	(0.00, -3.00)

Ellipse k , points $\mathbf{P}(\frac{i}{3}\pi)$ and three vectors $\mathbf{P}'(\frac{i}{3}\pi)$, $\mathbf{P}''(\frac{i}{3}\pi)$ and $\mathbf{P}'''(\frac{i}{3}\pi)$, $i = 0, 1, \dots, 6$ at each point are depicted in Fig. 1.3. Vectors of derivatives are distinguished by arrow shapes.

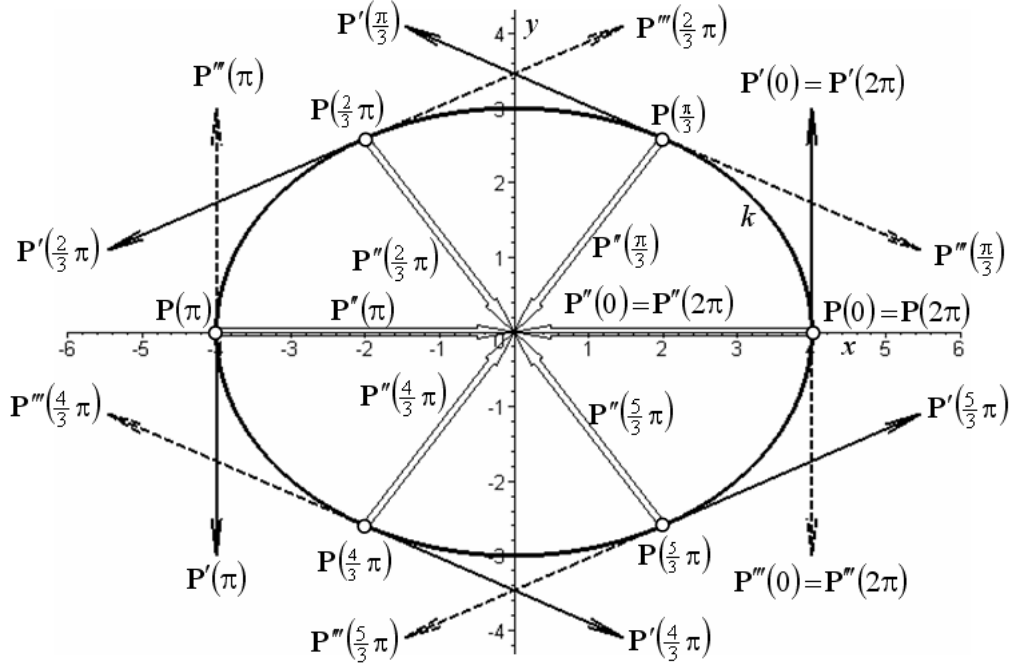


Figure 1.3: Vectors of the first three derivatives of ellipse k

□

Frenet moving trihedron of a curve is used to describe intrinsic (geometric) properties of the curve in the neighborhood of its regular point.

■ **Definition 1.9 – Frenet moving trihedron of a curve.** The normalized orthogonal right-handed (with positive orientation) trihedron created at a regular point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ by unit tangent vector $\mathbf{t}(\alpha)$, unit principal normal vector $\mathbf{n}(\alpha)$ and unit binormal vector $\mathbf{b}(\alpha)$ is called *Frenet moving trihedron of a curve*. □

■ **Definition 1.10 – Normal, rectification and osculation plane.** The plane given at a regular point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ by principal normal line and binormal line is the *normal plane* ν_α . The plane given at a regular point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ by binormal line and tangent line is the *rectification plane* ρ_α . The plane given at a regular point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ by principal normal line and tangent line is the *osculation plane* ω_α . □

The normal plane is perpendicular to the tangent line, the rectification plane is perpendicular to the principal normal line and the osculation plane is perpendicular to the binormal line.

■ **Example 1.3 – Frenet moving trihedron of ellipse.** Considering ellipse k from Example 1.1, the vector equations of the ellipse and its first two derivatives are given by

$$\begin{aligned}\mathbf{P}(t) &= (x(t), y(t), z(t)) = (4 \cos t, 3 \sin t, 0), \quad t \in [0, 2\pi], \\ \mathbf{P}'(t) &= (-4 \sin t, 3 \cos t, 0), \quad t \in [0, 2\pi], \\ \mathbf{P}''(t) &= (-4 \cos t, -3 \sin t, 0), \quad t \in [0, 2\pi].\end{aligned}$$

The vector equations of unit tangent vector $\mathbf{t}(t)$, unit principal normal vector $\mathbf{n}(t)$ and unit binormal vector $\mathbf{b}(t)$ are as follows

$$\begin{aligned}\mathbf{t}(t) &= \left(\frac{-4 \sin t}{\sqrt{-7 \cos^2 t + 16}}, \frac{3 \cos t}{\sqrt{-7 \cos^2 t + 16}}, 0 \right), \quad t \in [0, 2\pi], \\ \mathbf{b}(t) &= (0, 0, 1), \quad t \in [0, 2\pi], \\ \mathbf{n}(t) &= \left(\frac{-3 \cos t}{\sqrt{-7 \cos^2 t + 16}}, \frac{-4 \sin t}{\sqrt{-7 \cos^2 t + 16}}, 0 \right), \quad t \in [0, 2\pi].\end{aligned}$$

Ellipse k and its Frenet moving trihedron at several points for $t = \frac{i}{3}\pi$, $i = 0, 1, \dots, 6$ are depicted in Fig. 1.4. To preserve readability of the picture, the trihedron vectors are labeled only for parameter value $t = \frac{2}{3}\pi$, i.e. at point $\mathbf{P}(\frac{2}{3}\pi)$. For this parameter value, the tangent line $t_{\frac{2}{3}\pi}$, principal normal line $n_{\frac{2}{3}\pi}$, binormal line $b_{\frac{2}{3}\pi}$, normal plane $\nu_{\frac{2}{3}\pi}$, rectification plane $\rho_{\frac{2}{3}\pi}$ and osculation plane $\omega_{\frac{2}{3}\pi}$ are drawn, too.

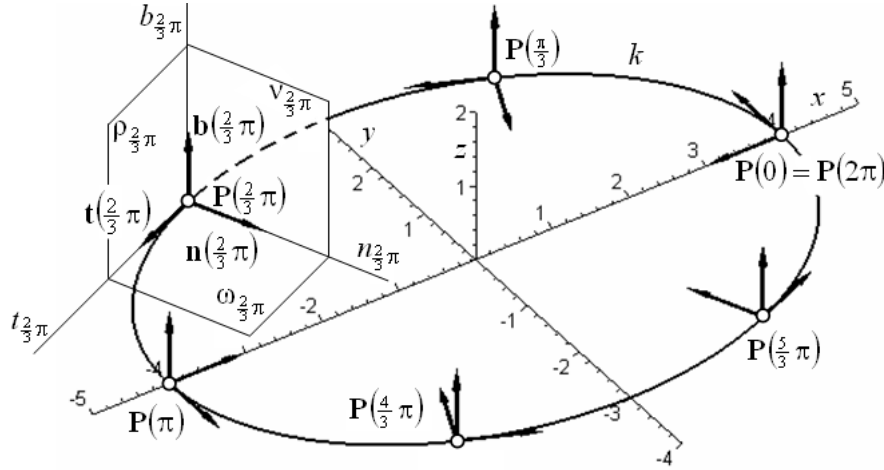


Figure 1.4: Frenet moving trihedron of ellipse k

□

- **Definition 1.11 – The first curvature – flection.** The first curvature – flection ${}^1k(\alpha)$ at a regular point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ is a non-negative number expressed by

$${}^1k(\alpha) = \frac{|\mathbf{P}'(\alpha) \times \mathbf{P}''(\alpha)|}{|\mathbf{P}'(\alpha)|^3}, \quad (1.22)$$

where $|\mathbf{P}'(\alpha) \times \mathbf{P}''(\alpha)|$ is the magnitude of cross product of the first and second derivatives of the curve and $|\mathbf{P}'(\alpha)|$ is the magnitude of tangent vector of the curve at point $\mathbf{P}(\alpha)$.

If ${}^1k(t) = 0$, $t \in [a, b]$, the curve $\mathbf{P}(t)$ is a straight line. □

The first curvature of the curve corresponds to elevation of the curve from its tangent line.

- **Definition 1.12 – Radius of the first curvature.** The number

$$r(\alpha) = \frac{1}{{}^1k(\alpha)} \quad (1.23)$$

is called the *radius of the first curvature* at point $\mathbf{P}(\alpha)$. □

■ **Definition 1.13 – Center of the first curvature.** The point

$$\mathbf{S}(\alpha) = \mathbf{P}(\alpha) + r(\alpha)\mathbf{n}(\alpha) \quad (1.24)$$

lying in osculation plane ω_α on the halfline given by point $\mathbf{P}(\alpha)$ and direction vector $\mathbf{n}(\alpha)$ at the distance $r(\alpha)$ from $\mathbf{P}(\alpha)$ is called the *centre of the first curvature* at point $\mathbf{P}(\alpha)$. \square

■ **Definition 1.14 – Osculation circle.** The circle with the centre at point $\mathbf{S}(\alpha)$ and radius $r(\alpha)$ is called the *osculation circle* at point $\mathbf{P}(\alpha)$. \square

■ **Definition 1.15 – Inflection curve point.** Curve point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ is called an *inflection curve point* or *point of inflection* if the curvature $k^1(\alpha)$ at this point changes sign from plus to minus or vice versa. \square

The tangent line crosses the curve at its inflection point (see Definition 1.5), as is obvious in Fig. 1.2 a).

■ **Definition 1.16 – The second curvature – torsion.** The *second curvature – torsion* ${}^2k(\alpha)$ at a non-inflection point $\mathbf{P}(\alpha)$, $\alpha \in [a, b]$ of curve $\mathbf{P}(t)$, $t \in [a, b]$ is a real number expressed by

$${}^2k(\alpha) = \frac{[\mathbf{P}'(\alpha)\mathbf{P}''(\alpha)\mathbf{P}'''(\alpha)]}{|\mathbf{P}'(\alpha) \times \mathbf{P}''(\alpha)|}, \quad (1.25)$$

where $[\mathbf{P}'(\alpha)\mathbf{P}''(\alpha)\mathbf{P}'''(\alpha)]$ is mixed product of vectors of the first, second and third curve derivatives at point $\mathbf{P}(\alpha)$. If ${}^2k(t) = 0$, $t \in [a, b]$, the curve $\mathbf{P}(t)$ is a planar curve. \square

The second curvature corresponds to elevation of the curve from its osculation plane.

■ **Example 1.4 – The first curvature of ellipse.** Considering ellipse k from Example 1.3, the first curvature is given by

$$\begin{aligned} {}^1k(t) &= \frac{|(-4 \sin t, 3 \cos t, 0) \times (-4 \cos t, -3 \sin t, 0)|}{|(-4 \sin t, 3 \cos t, 0)|^3} = \\ &= \frac{12}{\left(\sqrt{16 - 7 \cos^2 t}\right)^3}, \quad t \in [0, 2\pi] \end{aligned} \quad (1.26)$$

and radius of the first curvature is given by

$$r(t) = \frac{\left(\sqrt{16 - 7 \cos^2 t}\right)^3}{12}, \quad t \in [0, 2\pi]. \quad (1.27)$$

Graphs of functions (1.26) and (1.27) are depicted in Fig. 1.5.

After substitution $t = 0$ and $t = \frac{\pi}{2}$ in (1.27), we obtain radius $r(0) = \frac{9}{4}$ of osculation circle at the major vertex $\mathbf{P}(0)$ of the ellipse with centre at point $\mathbf{S}(0) = \left(\frac{7}{4}, 0\right)$ and radius $r\left(\frac{\pi}{2}\right) = \frac{16}{3}$ of osculation circle at the minor vertex $\mathbf{P}\left(\frac{\pi}{2}\right)$ of the ellipse with centre at point $\mathbf{S}\left(\frac{\pi}{2}\right) = \left(0, -\frac{7}{3}\right)$, respectively, see Fig. 1.6.

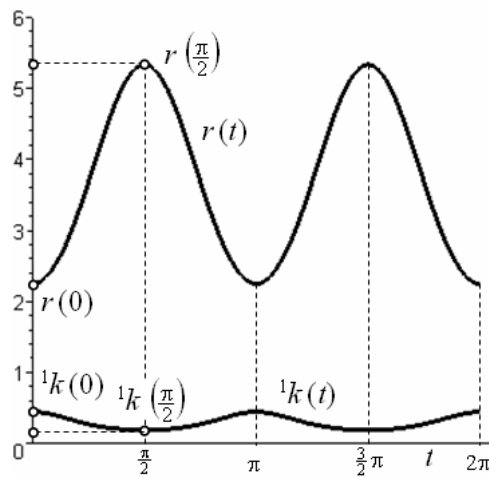


Figure 1.5: Graph of the first curvature and radius of the first curvature of ellipse k

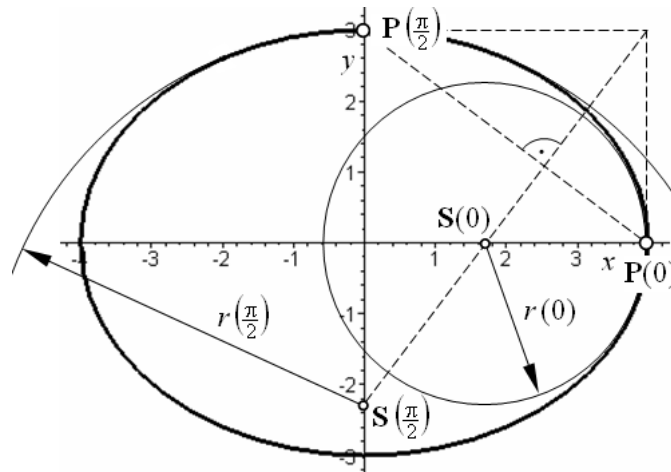


Figure 1.6: Osculation circles of ellipse k

□

1.2 Curves joining

The only one curve given by only one vector equation is insufficient when modelling curves of free-form shape. More complicated curve shapes are composed of several individual curves that are joined at their endpoints. The quality of this joining has to fit the technical application requirements in which the resulting curve is used.

First of all, the curve continuity definition following from the continuity of its vector function will be defined. After that, a *parametric continuity* and *geometric continuity* of two curves will be introduced.

■ **Definition 1.17 – Curve continuity.** Curve $\mathbf{P}(t)$, $t \in [a, b]$ is C^k continuous at point $\alpha \in [a, b]$, if its vector function is C^k continuous at this point.

Curve $\mathbf{P}(t)$, $t \in [a, b]$ is C^k continuous on $[a, b]$, if its vector function is C^k continuous on $[a, b]$. □

- **Definition 1.18 – Parametric continuity of two curves.** Let $\mathbf{P}(t)$ be a curve given by vector equation $\mathbf{P}(t) = (x(t), y(t), z(t))$, $t \in [a, b]$ with coordinate functions C^k continuous on $[a, b]$. Let $\mathbf{R}(s)$ be a curve given by vector equation $\mathbf{R}(s) = (\tilde{x}(s), \tilde{y}(s), \tilde{z}(s))$, $s \in [c, d]$ with coordinate functions C^k continuous on $[c, d]$. Then, the curves are joined at their common point (the initial point of $\mathbf{R}(s)$ and the terminal point of $\mathbf{P}(t)$) with C^k continuity (the curves are C^k continuously joined at their common point), if the following relation is valid

$$\mathbf{P}^{(i)}(b) = \mathbf{R}^{(i)}(c), \quad i = 0, 1, \dots, k. \quad (1.28)$$

C^k continuity of two curves is referred to as the *parametric continuity of k -th order*. \square

Parametric continuity C^k means that vectors of the first k derivatives of both curves at the common point are identical, i.e. the origins (the common point), directions, orientations and magnitudes of these vectors are identical.

- **Definition 1.19 – Geometric continuity of the first order of two curves.** Let $\mathbf{P}(t)$ be a curve given by vector equation $\mathbf{P}(t) = (x(t), y(t), z(t))$, $t \in [a, b]$ with coordinate functions at least C^1 continuous on $[a, b]$. Let $\mathbf{R}(s)$ be a curve given by vector equation $\mathbf{R}(s) = (\tilde{x}(s), \tilde{y}(s), \tilde{z}(s))$, $s \in [c, d]$ with coordinate functions at least C^1 continuous on $[c, d]$. Then, the curves are joined at their common point (the initial point of $\mathbf{R}(s)$ and the terminal point of $\mathbf{P}(t)$) with G^1 continuity (the curves are G^1 continuously joined) if the following relation is valid

$$\mathbf{P}'(b) = \lambda \cdot \mathbf{R}'(c), \quad \lambda \in R, \quad (1.29)$$

where λ is a real number, ${}^1k_{\mathbf{P}}(b)$ is the first curvature of $\mathbf{P}(t)$ and ${}^1k_{\mathbf{R}}(c)$ is the first curvature of $\mathbf{R}(s)$ at common point.

G^1 continuity of two curves is referred to as the *geometric continuity of the first order*. \square

- **Definition 1.20 – Geometric continuity of the second order of two curves.** Let $\mathbf{P}(t)$ be a curve given by vector equation $\mathbf{P}(t) = (x(t), y(t), z(t))$, $t \in [a, b]$ with coordinate functions at least C^2 continuous on $[a, b]$. Let $\mathbf{R}(s)$ be a curve given by vector equation $\mathbf{R}(s) = (\tilde{x}(s), \tilde{y}(s), \tilde{z}(s))$, $s \in [c, d]$ with coordinate functions at least C^2 continuous on $[c, d]$. Then, the curves are joined at their common point (the initial point of $\mathbf{R}(s)$ and the terminal point of $\mathbf{P}(t)$) with G^2 continuity (the curves are G^2 continuously joined) if the following relation is valid

$${}^1k_{\mathbf{P}}(b) = {}^1k_{\mathbf{R}}(c), \quad (1.30)$$

where ${}^1k_{\mathbf{P}}(b)$ is the first derivative of $\mathbf{P}(t)$ and ${}^1k_{\mathbf{R}}(c)$ is the first derivative of $\mathbf{R}(s)$ at common point.

G^2 continuity of two curves is referred to as the *geometric continuity of the second order*. \square

The tangent lines of two curves joined with G^1 continuity are identical at the common point because tangent vector $\mathbf{R}'(c)$ is equal to λ -multiple of tangent vector $\mathbf{P}'(b)$. The osculation circles of two curves joined with G^2 continuity are identical at the common point because the first curvatures of both curves are the same.

The geometric continuity of higher orders is not introduced, here.

Many examples to better understand the curves joining are given in Chapter 2.

1.3 Analysis of curves in Rhinoceros

Curve – In Rhinoceros, there are many commands for free-form curve drawing. These commands are described in detail in Sections 2.2.5, 2.3.4, 2.4.2 and 2.5.3. Here, only the command necessary to draw the ellipse from examples in this section is only given.

Command: *Ellipse: From Center* → Ellipse Centre: type 0,0,0 in command prompt → press Enter → End of first axis: type w4,0,0 in command prompt² → press Enter → End of second axis: type w0,3,0 in command prompt. The ellipse is drawn.

Curve point – Command: *Point* → Location of point object: activate *Object Snap: On Curve* → Select curve: click on the curve (the marker moves only along the selected curve) → Location of point object: click to place the point at the required position on the curve. The point (as an individual entity) is drawn on the curve in the position of the last click.

Initial curve point – Command: *Mark Curve Start* → Select curves for marking curve start: click on the curve → press Enter. The point is drawn at the initial point of the selected curve.

Terminal curve point – Command: *Mark Curve End* → Select curves for marking curve end: click on the curve → press Enter. The point is drawn at the terminal point of the selected curve.

Curve orientation – Command: *Direction* → Select object for direction display: click on the curve → press Enter. Three fixed arrows are displayed: at the initial point of the curve, at the middle point of the curve and at the terminal point of the curve. The orientation of the arrows is identical to the orientation of the curve, see Fig. 1.7. The fourth arrow moves along the curve accordingly to the movement of the marker.

Remark: In Rhinoceros, the orientation of a curve is often given by parametrization used in the system. This orientation does not have to be identical to the orientation of the vector equation considered. However, it is possible to change it by choosing the option *Flip* in command prompt.

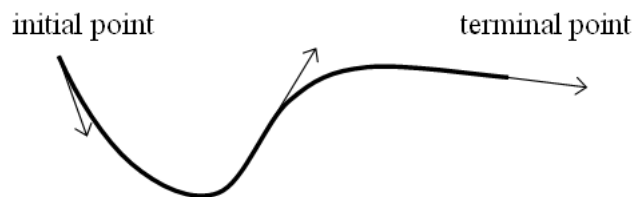


Figure 1.7: Curve orientation

Curve point $P(\alpha)$ – In Rhinoceros, a command for curve point drawing is not available if the precise position given by known curvilinear coordinate is required. However, it is possible to overcome this disadvantage by using the command *Point from UV coordinate* intended for surfaces.

²The world coordinate system is used when typing the prefix "w" in front of Cartesian coordinates of a point.

1. Create an auxiliary cylindrical surface. The directrix of this cylindrical surface is given by the curve on which the point $\mathbf{P}(\alpha)$, $\alpha \in [0, 1]$ has to be drawn. Command: *Extrude Straight* → Select curve to Extrude: click on the curve → press Enter → Extrusion distance: type a suitable altitude of cylindrical surface in command prompt or click to place the point at a suitable altitude. The cylindrical surface is drawn. Its orientation is as follows: u direction is oriented along the directrix, v direction is oriented along the extrusion straight lines, see Fig. 1.8 a).
2. Draw point $\mathbf{P}(\alpha)$. Command: *Point from UV coordinates* → Select surface to evaluate: choose in command prompt: *CreatePoint=Yes, Normalized=Yes* → click on the cylindrical surface → Enter U value between 0.0 and 1.0: type the chosen value α in command prompt → Enter V value between 0.0 and 1.0: type 0 in command prompt → press Enter. The point (as an individual entity) is drawn at the curve point $\mathbf{P}(\alpha)$, see Fig. 1.8 b).
3. Remove the auxiliary cylindrical surface (select the surface and press Delete), see Fig. 1.8 c).

Remark: It is necessary to preserve the orientation of the curve. It is possible to use the command *Direction* to recognize it, see above.

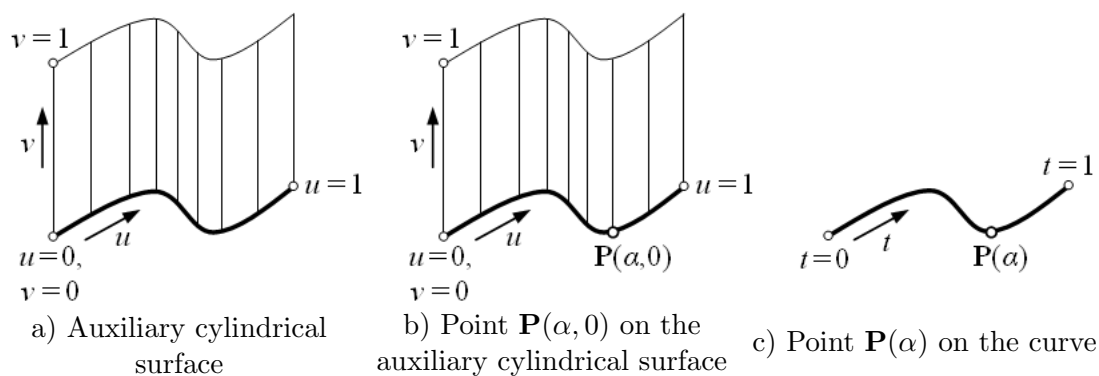


Figure 1.8: Construction of point $\mathbf{P}(\alpha)$ on the curve $\mathbf{P}(t)$, $t \in [0, 1]$

Cartesian coordinates of curve point – Command: *Evaluate Point* → activate *Point object snap* → Point to evaluate: click at the curve point of which Cartesian coordinates have to be determined. The Cartesian coordinates are displayed in the command prompt.

Tangent line of a curve – Command: *Line: Tangent from Curve* → Start of line: choose in command prompt: *BothSides* → click on the curve at the future point of contact between the curve and tangent line → press Enter → End of line: click to place a suitable endpoint of tangent line. The straight line segment representing the tangent line with the centre at the point of contact is drawn.

Principal normal line of planar curve – Command: *Line: Perpendicular from Curve* → Start of line: choose in command prompt *BothSides* → click on the curve at the required intersection of the curve and principal normal line → press Enter → End of line: click to place a suitable endpoint of principal normal line. The straight line segment representing the principal normal line with the centre at the intersection is drawn.

Principal normal line of spatial curve – The principal normal line of spatial curve is constructed indirectly.

1. Draw the osculation circle (see below) with point of contact at the required intersection of the curve and the future principal normal line.
2. The principal normal line is constructed by command *Line: from Midpoint* → Middle of line: click at the point of contact of osculation circle and the curve → End of line: activate *Center object snap* → click on the osculation circle. The straight line segment with the centre at point of contact representing the principal normal line is drawn.

Remark: When using the command *Line: Perpendicular from Curve* in the case of spatial curve, a normal line is drawn, i.e. the straight line perpendicular to the tangent line of the curve and lying in the normal plane of the curve. However, it is not generally the principal normal line.

Binormal line of spatial curve – The binormal line of spatial curve is constructed indirectly as an intersecting line between the normal plane (see below) and rectification plane (see below).

Command: *Object Intersection*: Select objects to intersect: click on the normal plane and rectification plane → press Enter. The straight line segment of intersection representing the binormal line is drawn.

Normal plane of curve – The normal plane of a curve is constructed indirectly.

1. Create the boundary of normal plane by command: *Circle: Around Curve*: Select curve → click on the curve → Center of circle → click on the curve at the required intersection of the curve and normal plane → Radius: click at a suitable terminal point of radius. The circle lying in the normal plane of the curve with the centre at the intersection between normal plane and the curve is drawn.
2. Create the normal plane by command: *Surface from Planar Curves* → Select planar curves to build surface: click on the boundary circle → press Enter. The disc representing the normal plane is drawn.

Osculation plane of a curve – The osculation plane of a curve is constructed indirectly.

1. Create the osculation circle (see below) at the required point. The osculation circle represents the boundary of future osculation plane.
2. Create the osculation plane by command: *Surface from Planar Curves* → Select planar curves to build surface: click on the boundary circle → press Enter. The disc representing the osculation plane is drawn.

Rectification plane of a curve – The rectification plane of a curve is constructed indirectly.

1. Construct the principal normal line (see above).
2. Draw an auxiliary circle by command: *Circle: Around Curve* → Select curve → click on the normal line → Center of circle: activate *Endpoint object snap* → click at an arbitrary endpoint of principal normal line → Radius: click at a suitable endpoint of radius.
3. Move this circle by command *Move* → select objects to move: click on the circle → press Enter → Point to move from: click at the center of the circle lying at the endpoint of principal normal line → Point to move to: click at the intersection of principal normal line and the curve. Now, the circle represents the boundary of the rectification plane.

4. Create the rectification plane by command: *Surface from Planar Curves* → Select planar curves to build surface: click on the boundary circle → press Enter. The disc representing the rectification plane is drawn.

Graph of the first curvature – Command: *Curvature Graph On* → Select objects for curvature graph display → click on the curve → press Enter → in the dialog box *Curvature Graph* adjust the *Display scale* and *Density* of the graph. The graph of the first curvature appears along the curve in the form of curvature indicators – straight line segments perpendicular to the curve (perpendicular segments) connected at their endpoints. The length of these perpendicular segments is proportional to the first curvature of the curve at the intersection of perpendicular segment with the curve.

If the curvature graph is displayed, it is possible to recognize the following special points on the curve: an inflection point, a cusp or an angular point.

- In the case of inflection point, the curvature graph intersects the curve (the first curvature changes its sign at inflection point), see Fig. 1.9 a).
- In the case of cusp, the perpendicular segments lie on the same straight line (the direction of perpendicular segments is given by common normal line at the cuspidal point) and have opposite orientation (the orientation of tangent vector is changed at cuspidal point), see Fig. 1.9 b).
- In the case of angular point, the perpendicular segments do not lie on the same straight line (normal lines at angular point are intersecting), see Fig. 1.9 c).

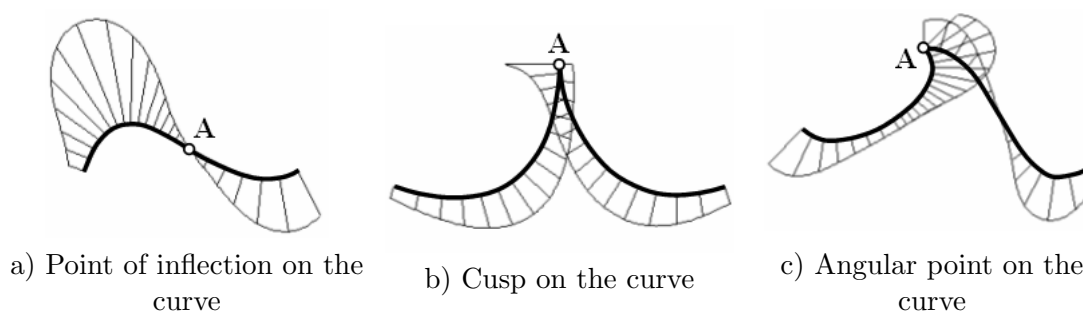


Figure 1.9: Classification of curve points via curvature graph

Radius of the first curvature – Command: *Radius* → Select point on curve for radius measurement: click on the curve. The radius of the first curvature of the curve at the clicked point is displayed in the command prompt.

Osculation circle – Command: *Curvature* → Select curve or surface for curvature measurement: click on the curve → Select point on curve for curvature measurement: choose in the command prompt *MarkCurvature=Yes* → click on the curve at the required point of contact between the curve and osculation circle → press Enter. The osculation circle of the curve at the point of contact is drawn.

In the command prompt, the following information is displayed: parameter value α of point of contact, Cartesian coordinates of point of contact, coordinates of tangent vector at point of contact, Cartesian coordinates of the osculation circle center and radius of the first curvature of the curve at point of contact.

Remark: If the radius of osculation circle is too large, only a part of osculation circle (an arc) is drawn. It is necessary to close this arc when using this circle as a boundary of osculation plane (see above) by command *Extend by Arc, keep radius* → Select curve to extend: click on the osculation circle near its endpoint → End of extension or enter extension length: activate *Endpoint object snap* → click on the osculation circle near its second endpoint. The osculation circle is closed.

Continuity of two curves – Command: *Geometric Continuity* → First curve – select near end: click on the first curve → Second curve – select near end: click on the second curve. The geometric continuity of these curves is displayed in the command prompt.

Remark: Since the parametric continuity depends on parametrization of curves, it is impossible to determine the parametric continuity of two curves in Rhinoceros.

If the curvature graph is displayed, it is possible to recognize C^0 , (at least) G^1 and (at least) G^2 continuity of two curves, see Fig. 1.10.

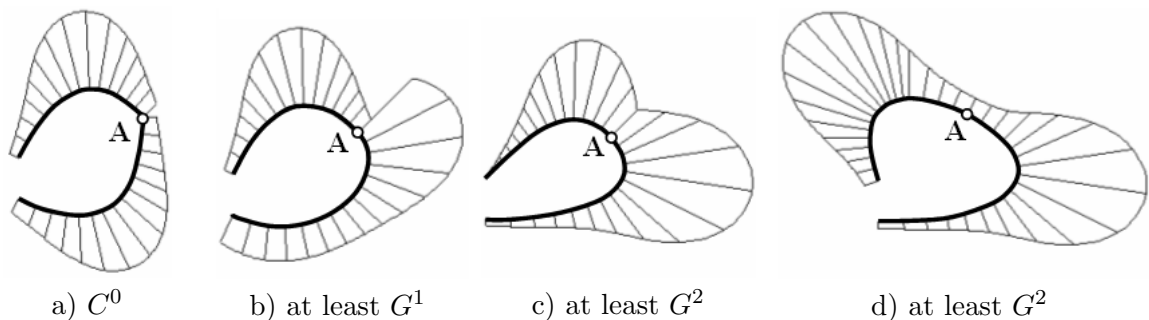


Figure 1.10: Continuity of two curves determination via curvature graph

In the case of C^0 continuity of two curves, the perpendicular segments are intersecting, see Fig. 1.10 a). In the case G^1 continuity, the perpendicular segments lie on the same straight line, however, there is a step change in their length, see Fig. 1.10 b). In the case of G^2 continuity, the curvature graphs are continuous, see Fig. 1.10 c) and d).

Remark: Depending on parametrization, the curves in Fig. 1.10 b) can be C^1 continuously joined, as well as the curves in Fig. 1.10 c) and d) can be C^2 continuously joined. However, the curvature graphs look the same way. Therefore, the words *at least* are used.

- **Exercise 1.1** In Rhinoceros, draw the ellipse from Example 1.1. Draw the points for $t = \frac{i}{6}$, $i = 0, 1, \dots, 5$. Find Cartesian coordinates of these points and check them with the values given in the table on page 15. Accordingly to Fig. 1.4, create the Frenet moving trihedron at each of these points. Next, construct osculation circles at major and minor vertices of the ellipse according to Example 1.4. Finally, display the curvature graphs of the ellipse and osculation circles.

Compare the length of perpendicular segments of both the ellipse and the osculation circles at the ellipse vertices.

1.4 Definition and properties of surfaces

A *surface* is a bivariate set of *points* in space R^3 . These points represent the function values of *bivariate point function*. Similarly to the curves, the function values of *bivariate vector function* (the terminal points of their *position vectors*) are considered here. Therefore, the bivariate point function is omitted in the following definitions.

■ **Definition 1.21 – Bivariate vector function.** Let $I \subset R^2$ be a non-empty connected area. Then continuous mapping \mathbf{F} of area I into the Euclidean vector space V^3 is a *bivariate vector function*

$$\mathbf{F}(u, v) = (x(u, v), y(u, v), z(u, v)). \quad (1.31)$$

Area I is referred to as a *parametrization domain* of vector function. The arguments u and v are called *surface parameters*. A vector is assigned to each pair $(u, v) \in I$ by a vector function. Bivariate real functions

$$x = x(u, v), y = y(u, v), z = z(u, v), (u, v) \in I \quad (1.32)$$

are called *coordinate functions* of bivariate vector functions. \square

Bivariate vector function is a generalization of a bivariate real function. Therefore, it is easy to prove the following statements and related properties of vector function of two variables.

- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ has a limit at point $(\alpha, \beta) \in I$ equal to the vector

$$\mathbf{F}_{\alpha, \beta} = (x(\alpha, \beta), y(\alpha, \beta), z(\alpha, \beta)),$$

if the limits of its coordinate functions are as follows

$$\begin{aligned} \lim_{u \rightarrow \alpha, v \rightarrow \beta} x(u, v) &= x(\alpha, \beta), \\ \lim_{u \rightarrow \alpha, v \rightarrow \beta} y(u, v) &= y(\alpha, \beta), \\ \lim_{u \rightarrow \alpha, v \rightarrow \beta} z(u, v) &= z(\alpha, \beta). \end{aligned} \quad (1.33)$$

- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ is continuous at point $(\alpha, \beta) \in I$ if its coordinate functions are at point (α, β) continuous, i.e. the limits (1.33) at point $(\alpha, \beta) \in I$ exist and are equal to the function values of coordinate functions at point (α, β) .
- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ is continuous on I if its coordinate functions are continuous on I .
- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ has the first partial derivative with respect to u and v at point $(\alpha, \beta) \in I$, if the first partial derivatives of its coordinate functions exist

$$x^u(\alpha, \beta) = \left. \frac{\partial x(u, v)}{\partial u} \right|_{u=\alpha, v=\beta}, \quad y^u(\alpha, \beta) = \left. \frac{\partial y(u, v)}{\partial u} \right|_{u=\alpha, v=\beta}, \quad z^u(\alpha, \beta) = \left. \frac{\partial z(u, v)}{\partial u} \right|_{u=\alpha, v=\beta}, \quad (1.34)$$

and

$$x^v(\alpha, \beta) = \left. \frac{\partial x(u, v)}{\partial v} \right|_{u=\alpha, v=\beta}, \quad y^v(\alpha, \beta) = \left. \frac{\partial y(u, v)}{\partial v} \right|_{u=\alpha, v=\beta}, \quad z^v(\alpha, \beta) = \left. \frac{\partial z(u, v)}{\partial v} \right|_{u=\alpha, v=\beta}, \quad (1.35)$$

respectively. The first partial derivative $\mathbf{F}^u(\alpha, \beta)$ and $\mathbf{F}^v(\alpha, \beta)$ of vector function $\mathbf{F}(u, v)$ with respect to u and v is a vector

$$\mathbf{F}^u(\alpha, \beta) = (x^u(\alpha, \beta), y^u(\alpha, \beta), z^u(\alpha, \beta)), \quad (1.36)$$

and

$$\mathbf{F}^v(\alpha, \beta) = (x^v(\alpha, \beta), y^v(\alpha, \beta), z^v(\alpha, \beta)), \quad (1.37)$$

respectively.

- In the similar way, the partial derivatives of vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ of higher order are defined at point $(\alpha, \beta) \in I$. We will use the following denotation

$$\begin{aligned} \mathbf{F}^{uu}(\alpha, \beta) &= (x^{uu}(\alpha, \beta), y^{uu}(\alpha, \beta), z^{uu}(\alpha, \beta)), \\ \mathbf{F}^{uv}(\alpha, \beta) &= (x^{uv}(\alpha, \beta), y^{uv}(\alpha, \beta), z^{uv}(\alpha, \beta)), \\ &\dots, \end{aligned}$$

where the order of the superscripts corresponds to the order of partial derivatives.

- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ is C^k continuous at point $(\alpha, \beta) \in I$ if the partial derivatives of its coordinate functions are at point $(\alpha, \beta) \in I$ continuous to k -th order.
- Vector function $\mathbf{F}(u, v)$, $(u, v) \in I$ is C^k continuous on I if the partial derivatives of its coordinate functions are on I continuous to k -th order.

■ **Definition 1.22 – Surface.** A *surface* is any connected non-empty subset κ in space R^3 , which is a continuous mapping of region $I \subset R^2$.

If the analytical representation of the surface is vector function (1.31) which is defined, continuous and at least once differentiable on I , the surface is given by *vector equation*

$$\mathbf{P}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in I. \quad (1.38)$$

Parametric equations of the surface are obtained by itemizing the coordinate functions of the surface given by vector equation (1.31)

$$\begin{aligned} x &= x(u, v), \\ y &= y(u, v), \\ z &= z(u, v), \quad (u, v) \in I. \end{aligned} \quad (1.39)$$

The surface defined by parametric equation (1.39) is referred to as a *surface defined parametrically*. The equations 1.39 are referred to as a *parametric expression* or *parametrization* of a surface.

If the region I is rectangular and closed, i.e. $I = [a, b] \times [c, d]$, the surface is referred to as a *patch*. □

The squared normalized region $I = [0, 1] \times [0, 1]$ is widely used. We will denote this region for short with $[0, 1]^2$.

The development of mathematical theory for surfaces with more complicated shapes led to the term *patch* introduction. It is impossible to express the analytical representation of the surface of a more complicated shape by only one vector equation. Therefore, the resulting surface is created by *patching*, i.e. by joining of suitable surface elements (patches). The individual patches (usually) defined on squared parametrization domain $I = [0, 1]^2$, are described by one

vector equation. See Section 1.5, where the conditions of individual patches joining are discussed. Here, we will focus on the fundamental surface properties from the point of view of differential geometry. Unless specified otherwise, all the following definitions and surface properties are valid for a patch, too.

■ **Definition 1.23 – Surface point.** The terminal point of position vector – the function value of vector function (1.31) for $(\alpha, \beta) \in I$

$$\mathbf{P}(\alpha, \beta) = (x(\alpha, \beta), y(\alpha, \beta), z(\alpha, \beta)) \quad (1.40)$$

is referred to as a *surface point*. The parameter values $u = \alpha$ and $v = \beta$ that unambiguously define the position of a point on the surface are called *parametric (curvilinear) coordinates* of the surface point.

The orientation of patch $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ is determined by orientation of its vector function. Points $\mathbf{P}(0, 0)$, $\mathbf{P}(0, 1)$, $\mathbf{P}(1, 0)$ and $\mathbf{P}(1, 1)$ are called *corners of the patch*. \square

■ **Example 1.5 – Torus vector equation.** Consider the following coordinate functions

$$\begin{aligned} x(u, v) &= (r \cos u + R) \cos v, \\ y(u, v) &= (r \cos u + R) \sin v, \\ z(u, v) &= r \sin u + 4, \quad (u, v) \in [0, 2\pi]^2. \end{aligned}$$

In Fig. 1.11, graphs of these coordinate functions are depicted for $R = 2$ and $r = 1$ together with the function values $x(\frac{1}{4}\pi, \frac{3}{4}\pi)$, $y(\frac{1}{4}\pi, \frac{3}{4}\pi)$ and $z(\frac{1}{4}\pi, \frac{3}{4}\pi)$ for $(u, v) = (\frac{1}{4}\pi, \frac{3}{4}\pi)$.

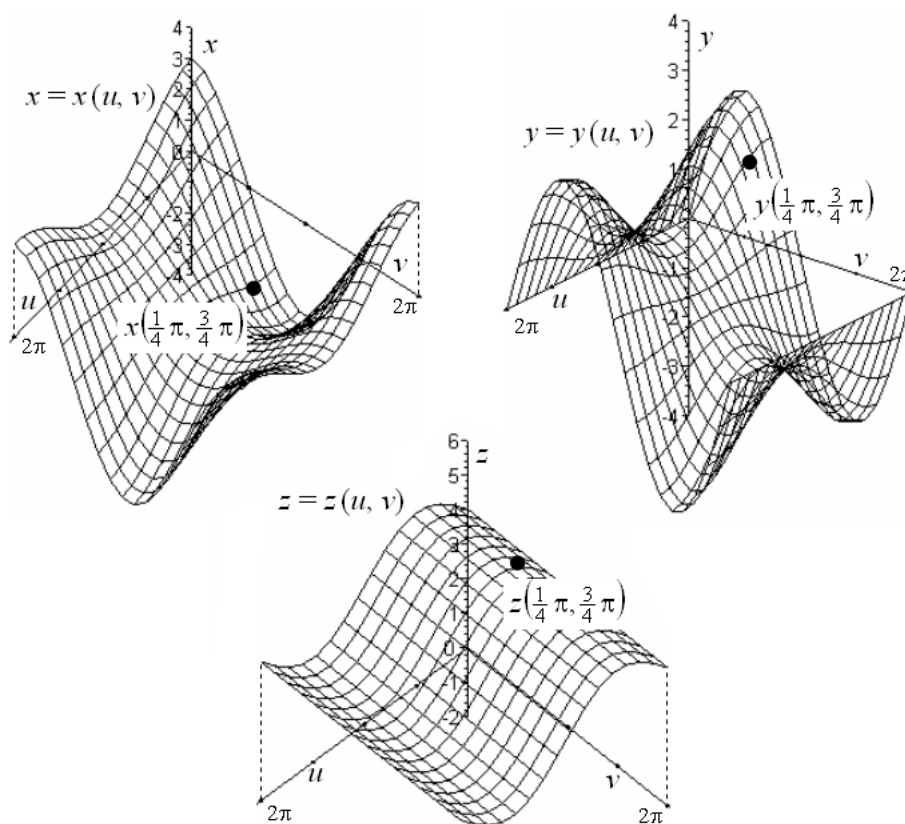


Figure 1.11: Coordinate functions of torus κ

The surface given by vector equation

$$\mathbf{P}(u, v) = ((r \cos u + R) \cos v, (r \cos u + R) \sin v, r \sin u + 4), \quad (u, v) \in [0, 2\pi]^2 \quad (1.41)$$

is torus κ , where r is the radius of the generating circle (lying in the plane passing through z -axis) which rotates around z axis and R is the radius of trajectory of the generating circle centre. Torus κ is shown in Fig. 1.12 together with the function value of vector function $\mathbf{F}(\frac{1}{4}\pi, \frac{3}{4}\pi)$, i.e. position vector $\mathbf{P}(\frac{1}{4}\pi, \frac{3}{4}\pi)$.

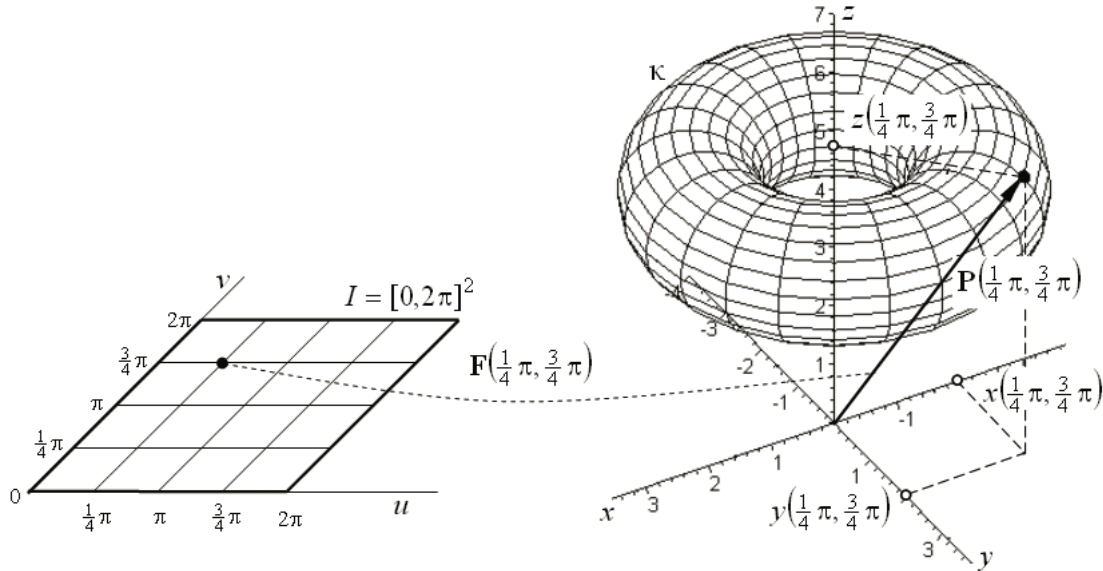


Figure 1.12: Torus κ given by vector equation (1.41) □

For constant value of one variable in a bivariate vector function, we obtain a univariate vector function representing a curve located on the surface. This curve is called a *parametric curve of the surface*.

■ **Definition 1.24 – Parametric curves of a surface.** Let $\mathbf{P}(u, v)$, $(u, v) \in I$ be a vector equation of a surface and α and β are parameter values $(\alpha, \beta) \in I$. Then the curve

$$\mathbf{P}(u, \beta) = (x(u, \beta), y(u, \beta), z(u, \beta)), \quad (1.42)$$

is called a *parametric u -curve of the surface* and the curve

$$\mathbf{P}(\alpha, v) = (x(\alpha, v), y(\alpha, v), z(\alpha, v)), \quad (1.43)$$

is called a *parametric v -curve of the surface*.

Parametric curves $\mathbf{P}(0, v)$, $\mathbf{P}(1, v)$, $\mathbf{P}(u, 0)$ and $\mathbf{P}(u, 1)$ are called *boundaries of a patch* $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. □

On a surface, parametric curves form two systems of curves, where each curve from one system intersects all curves from the other system. Two parametric curves from different systems intersect at a common point located on a surface. Curvilinear coordinates of this point correspond to constant values of parameters u and v . The boundaries of a patch intersect at corners of the patch.

In computer graphics, systems of parametric curves are widely used for wireframe display mode, see example in Fig. 1.12. In the case of torus, parametric u -curves are generating circles, parametric v -curves are parallel circles.

■ **Definition 1.25 – Tangent vectors of parametric curves.** The first partial derivative

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (x^u(u, v), y^u(u, v), z^u(u, v)), \quad (u, v) \in I \quad (1.44)$$

is a vector function. For $(\alpha, \beta) \in I$, this vector function determines a *tangent vector of parametric u -curve* $\mathbf{P}^u(u, v)$ at point $\mathbf{P}(\alpha, \beta)$

$$\mathbf{P}^u(\alpha, \beta) = \left. \frac{\partial \mathbf{P}(u, v)}{\partial u} \right|_{u=\alpha, v=\beta} = (x^u(\alpha, \beta), y^u(\alpha, \beta), z^u(\alpha, \beta)). \quad (1.45)$$

The first partial derivative

$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (x^v(u, v), y^v(u, v), z^v(u, v)), \quad (u, v) \in I \quad (1.46)$$

is a vector function. For $(\alpha, \beta) \in I$, this vector function determines a *tangent vector of parametric v -curve* $\mathbf{P}^v(u, v)$ at point $\mathbf{P}(\alpha, \beta)$

$$\mathbf{P}^v(\alpha, \beta) = \left. \frac{\partial \mathbf{P}(u, v)}{\partial v} \right|_{u=\alpha, v=\beta} = (x^v(\alpha, \beta), y^v(\alpha, \beta), z^v(\alpha, \beta)). \quad (1.47)$$

Orientation of tangent vectors of parametric curves is identical to orientation of a corresponding parametric curve. The straight line given by point $\mathbf{P}(\alpha, \beta)$ and tangent vector $\mathbf{P}^u(\alpha, \beta)$ is *tangent line of parametric u -curve* at point $\mathbf{P}(\alpha, \beta)$. The straight line given by point $\mathbf{P}(\alpha, \beta)$ and tangent vector $\mathbf{P}^v(\alpha, \beta)$ is *tangent line of parametric v -curve* at point $\mathbf{P}(\alpha, \beta)$. \square

■ **Definition 1.26 – Regular and singular surface point.** Surface point $\mathbf{P}(\alpha, \beta)$, $(\alpha, \beta) \in I$ is called a *regular surface point*, if the vectors $\mathbf{P}^u(\alpha, \beta)$ and $\mathbf{P}^v(\alpha, \beta)$ are non-zero, not parallel, and only one pair of parameter values $(u, v) = (\alpha, \beta)$ from I corresponds to this point. Every other surface point is called a *singular surface point*.

■ **Definition 1.27 – Tangent plane at surface point.** The plane $\tau_{\alpha, \beta}$ given by regular point $\mathbf{P}(\alpha, \beta)$, $(\alpha, \beta) \in I$ of the surface and tangent vectors of parametric curves $\mathbf{P}^u(\alpha, \beta)$ and $\mathbf{P}^v(\alpha, \beta)$ is called a *tangent plane* of surface $\mathbf{P}(u, v)$ at point $\mathbf{P}(\alpha, \beta)$. The vector

$$\mathbf{n}(\alpha, \beta) = \mathbf{P}^u(\alpha, \beta) \times \mathbf{P}^v(\alpha, \beta) \quad (1.48)$$

at regular point $\mathbf{P}(\alpha, \beta)$ of the surface is called a *normal vector* of a surface at point $\mathbf{P}(\alpha, \beta)$. The straight line given by point $\mathbf{P}(\alpha, \beta)$ and normal vector $\mathbf{n}(\alpha, \beta)$ is called a *normal line* of a surface at point $\mathbf{P}(\alpha, \beta)$. \square

■ **Definition 1.28 – Twist vector.** The second mixed partial derivative of vector function

$$\begin{aligned} \mathbf{P}^{uv}(u, v) &= \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = \frac{\partial^2 \mathbf{P}(u, v)}{\partial v \partial u} = \\ &= (x^{uv}(u, v), y^{uv}(u, v), z^{uv}(u, v)) = (x^{vu}(u, v), y^{vu}(u, v), z^{vu}(u, v)), \\ &\quad (u, v) \in I \end{aligned} \quad (1.49)$$

is a vector function. For $(\alpha, \beta) \in I$, this vector function determines a *twist vector* of the surface at point $\mathbf{P}(\alpha, \beta)$. \square

Twist vectors correspond to elevation of the surface from its tangent plane.

- **Example 1.6 – Tangent, twist and normal vectors of torus.** Consider the torus from Example 1.5. Vector equation of tangent vector of its parametric u -curves, vector equation of tangent vector of its parametric v -curves, vector equation of its twist vector and vector equation of its normal vector is determined by the following formula in the given order

$$\begin{aligned}\mathbf{P}^u(u, v) &= (-r \sin u \cos v, -r \sin u \sin v, r \cos u), (u, v) \in [0, 2\pi]^2, \\ \mathbf{P}^v(u, v) &= (-(r \cos u + R) \sin v, (r \cos u + R) \cos v, 0), (u, v) \in [0, 2\pi]^2, \\ \mathbf{P}^{uv}(u, v) &= (r \sin u \sin v, -r \sin u \cos v, 0), (u, v) \in [0, 2\pi]^2, \\ \mathbf{n}(u, v) &= (-r \cos u(r \cos u + R) \cos v, -r \cos u(r \cos u + R) \sin v, -r \sin u(r \cos u + R)), \\ &\quad (u, v) \in [0, 2\pi]^2.\end{aligned}$$

In Fig. 1.13, there are depicted the corresponding vectors at several point of torus for $R = 2$ and $r = 1$. The normal line of torus are depicted in Fig. 1.13 d) (the normal vectors are oriented inward the torus).

The boundaries of the patch and the tangent and twist vectors along the boundaries of the patch are very important when patching, see Section 1.5. To describe intrinsic properties of a surface, it is necessary to investigate *principal curvature*, *Gaussian curvature* and *mean curvature* of the surface.

Consider a regular surface point \mathbf{A} and tangent plane $\tau_{\mathbf{A}}$ and normal line $n_{\mathbf{A}}$ at this point. Infinitely many curves located on the surface pass through point \mathbf{A} . Thus, infinitely many tangent lines of these curves are lying in tangent plane $\tau_{\mathbf{A}}$. Each tangent line together with normal line $n_{\mathbf{A}}$ form a *plane of normal section of the surface*. Intersection of this plane of normal section and the surface is a so called *curve of normal section*. At point \mathbf{A} , each curve of normal section has the first curvature equal to a certain real number, see (1.22), called *normal curvature*. Directions in which the normal curvature reaches its minimum or maximum are so called *principal directions of the surface*. The corresponding values of the normal curvature are so called *principal curvatures of the surface*. We denote the principal curvatures with $k_{1\mathbf{A}}$ and $k_{2\mathbf{A}}$ and the corresponding osculation circles by $c_{1\mathbf{A}}$ and $c_{2\mathbf{A}}$.

We will use the following convention for the sign of the principal curvature: the principal curvature is positive if vector $\overrightarrow{\mathbf{A}\mathbf{S}}$, where \mathbf{S} is the centre of osculation circle, and normal vector $n_{\mathbf{A}}$ are identically oriented; the principal curvature is negative if vector $\overrightarrow{\mathbf{A}\mathbf{S}}$ and normal vector $n_{\mathbf{A}}$ have opposite orientation.

- **Definition 1.29 – Gaussian curvature.** *Gaussian curvature* at regular point \mathbf{A} of the surface κ is given by

$$G = k_{1\mathbf{A}} \cdot k_{2\mathbf{A}}, \quad (1.50)$$

where $k_{1\mathbf{A}}$ is minimal principal curvature and $k_{2\mathbf{A}}$ is maximal principal curvature at regular surface point \mathbf{A} .

A regular surface point is referred to as *elliptical*, if the Gaussian curvature at this point is positive.

A regular surface point is referred to as *parabolic*, if the Gaussian curvature at this point is equal to zero.

A regular surface point is referred to as *hyperbolic*, if the Gaussian surface at this point is negative.

Surfaces with zero Gaussian curvature are developable into a plane. \square

- **Example 1.7 – Classification of torus points.** The torus contains all three types of points, as is obvious from Fig. 1.14.

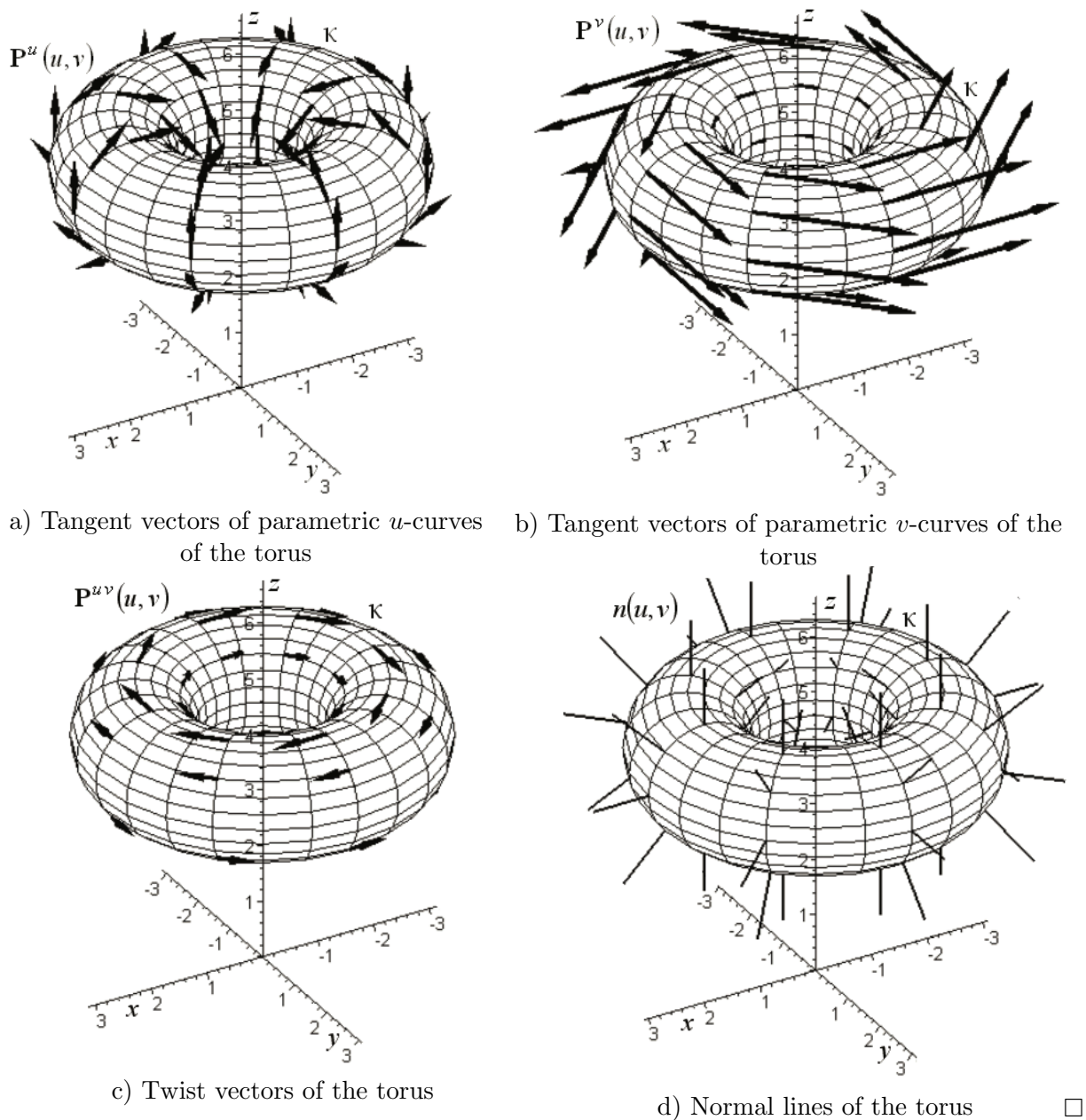


Figure 1.13: Tangent and twist vectors and normal lines of the torus

From Definition 1.29, the following geometric properties are obvious. Firstly, at an elliptical point, the centers of both osculation circles lie on the positive part of corresponding normal line. The corresponding tangent plane touches the surface at only one point of contact. Secondly, at a parabolic point, at least one of principal curvatures has to be zero. Therefore, at least one of the osculation circles is degenerated into a straight line. In this case, the tangent line touches the surface along a curve of contact. Thirdly, at a hyperbolic point, the signs of principal curvatures are different. Thus, the centers of osculation circles lie on opposite parts of corresponding normal line and tangent plane intersects the surface.

The points located between crater circles on the part of torus including the equator are elliptical points (point **A** in Fig. 1.14). The points located on the crater circles are parabolic

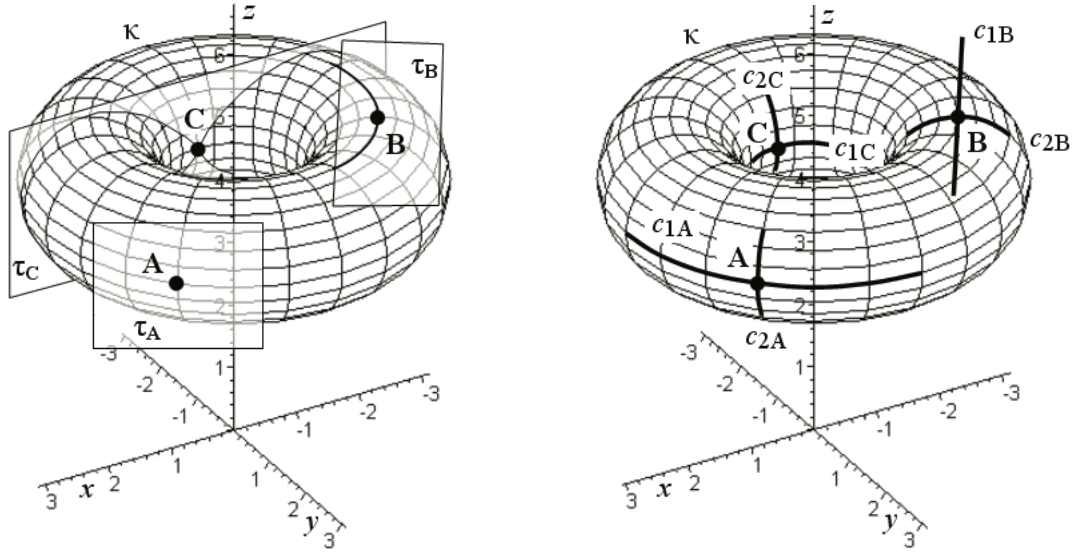


Figure 1.14: Tangent planes and osculation circles at selected points of torus κ

□

points (point **B** in Fig. 1.14). The points located between crater circles on the part of torus including the throat are hyperbolic points (point **C** in Fig. 1.14).

- **Definition 1.30 – Mean curvature.** Mean curvature at regular point **A** of surface κ is given by

$$H = \frac{k_{1A} + k_{2A}}{2}. \quad (1.51)$$

Surfaces with zero mean curvature are so called *minimal surfaces*.

□

1.5 Patching

It is necessary to use *patching* to create a resulting surface of a more complicated shape as a composition of individual patches all given by one vector equation. The quality of individual patches joining is evaluated according to the achieved order of parametric or geometric continuity along the common boundaries.

In this section, we will first formulate the definition of surface continuity. This definition follows from continuity of vector equation of the surface. After that, we will focus on continuity along the common boundary of two joined patches.

- **Definition 1.31 – Surface continuity.** Surface $\mathbf{P}(u, v)$, $(u, v) \in I$ is C^k continuous at point $(\alpha, \beta) \in I$, if its vector function is C^k continuous at this point.
The surface is C^k continuous on I , if its vector function is C^k continuous on I . □

- **Definition 1.32 – Parametric continuity of two joined patches.** Let $\mathbf{P}(u, v)$ be a patch given by vector equation

$$\mathbf{P}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in [0, 1]^2 \quad (1.52)$$

with coordinate functions C^k continuous on $[0, 1]^2$. Let $\mathbf{R}(s, t)$ be a patch given by vector equation

$$\mathbf{R}(s, t) = (\tilde{x}(s, t), \tilde{y}(s, t), \tilde{z}(s, t)), \quad (s, t) \in [0, 1]^2 \quad (1.53)$$

with coordinate functions C^k continuous on $[0, 1]^2$. Assume that patch $\mathbf{R}(s, t)$ is joined by its boundary $\mathbf{R}(0, t)$ to the boundary $\mathbf{P}(1, v)$ of patch $\mathbf{P}(u, v)$.

Then, the patches are joined along their common boundary with C^k *parametric continuity* (the patches are C^k *continuously joined*), if the vectors of the first k partial derivatives of corresponding vector functions along boundaries $\mathbf{P}(1, v)$ and $\mathbf{R}(0, t)$ are identical. \square

Parametric continuity of two patches joined along other boundaries can be defined similarly.

- **Definition 1.33 – Geometric continuity G^1 of two joined patches.** Let $\mathbf{P}(u, v)$ be a patch given by vector equation (1.52) with coordinate functions at least C^1 continuous on $[0, 1]^2$. Let $\mathbf{R}(s, t)$ is a patch given by vector equation (1.53) with coordinate functions at least C^1 continuous on $[0, 1]^2$. Assume that patch $\mathbf{R}(s, t)$ is joined by its boundary $\mathbf{R}(0, t)$ to the boundary $\mathbf{P}(1, v)$ of patch $\mathbf{P}(u, v)$.

Then, the patches are joined with G^1 *geometric continuity* (the patches are G^1 *continuously joined*), if the tangent planes of both patches along the common boundary are identical.

- **Definition 1.34 – Geometric continuity G^2 of two joined patches.** Let $\mathbf{P}(u, v)$ be a patch given by vector equation (1.52) with coordinate functions at least C^2 continuous on $[0, 1]^2$. Let $\mathbf{R}(s, t)$ is a patch given by vector equation (1.53) with coordinate functions at least C^2 continuous on $[0, 1]^2$. Assume that patch $\mathbf{R}(s, t)$ is joined by its boundary $\mathbf{R}(0, t)$ to the boundary $\mathbf{P}(1, v)$ of patch $\mathbf{P}(u, v)$.

Then, the patches are joined with G^2 *geometric continuity* (the patches are G^2 *continuously joined*), if the first curvatures of parametric curves of both patches along the common boundary are identical.

Geometric continuity of two patches joined along other boundaries can be defined similarly.

An example of continuity of two patches joining is not given here. The problem is discussed in detail in Chapter 3.

1.6 Analysis of surfaces in Rhinoceros

Surface – In Rhinoceros, there are many commands for free-form surfaces drawing. These commands are described in detail in Sections 3.2.3, 3.3.3, 3.4.4, 3.5.6 and 3.6.2. Here, only the command necessary to draw the torus from examples in this section is only given.

Command: *Torus* → activate *Top view* → Center of torus: type 0,0,4 in command prompt → press Enter → Radius: type 2 in command prompt → press Enter → Second radius: type 1 in command prompt → press Enter.

Surface point – Command: *Point* → Location of point object: activate *Object snap: On Surface* → Select surface: click on the surface (the marker moves only on the selected surface) → Location of point object: click to place the point at the required position on the surface. The point (as an individual entity) is drawn on the surface in the position of the last click.

Surface point $\mathbf{P}(\alpha, \beta)$ – Command: *Point from UV Coordinates* → Select surface to evaluate: choose in command prompt *CreatePoint=Yes, Normalized=Yes* → click on the surface → Enter U value between 0.0 and 1.0: type the chosen value α in command prompt → Enter V value between 0.0 and 1.0: type the chosen value β in command prompt → press Enter. The point (as an individual entity) is drawn at the surface point $\mathbf{P}(\alpha, \beta)$.

Remarks:

1. It is necessary to know the orientation of parameters u and v on the surface. To recognize the direction of the surface, use the command *Direction*, see below.
2. The option *Normalized=Yes* ensures the normalization of the original parametric domain $I = [a, b] \times [c, d]$ on the normalized one $I = [0, 1]^2$. The option *Normalized=No* preserves the original parametric domain $I = [a, b] \times [c, d]$.

Surface orientation – Command: *Direction* → Select object for direction display: click on the surface → press Enter → move marker along the surface. The fixed normal vectors (arrows) on the selected surface are displayed. Additionally, three vectors move along the surface according to the marker moving: the red vector indicating the orientation of u parameter, the green vector indicating the orientation of v parameter and the black normal vector of the surface, see Fig. 1.15 (the colours are visible in Rhinoceros).

Remark: In Rhinoceros, the orientation of a surface follows from the parametrization used in the system. This orientation does not have to be identical with the orientation given by vector equation considered. However, it is possible to select a suitable option in command prompt (*UReverse*, *VReverse*, *SwapUV*, *Flip*) to modify the preset properties.

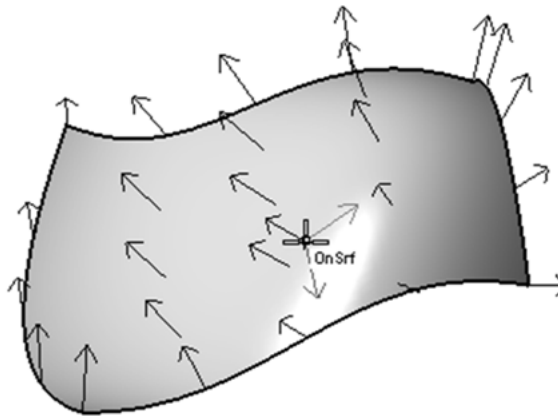


Figure 1.15: Surface orientation

Curvilinear coordinates of surface point – Command: *UV Coordinates of a Point* → Select surface to get UV values from: choose in command prompt *CreatePoint=Yes*, *Normalized=Yes* → click on the surface → Point to evaluate: click on the surface at the point of which curvilinear coordinates have to be determined → press Enter. In the command prompt, parameter values u and v corresponding to the selected point are displayed.

Remark: It is necessary to preserve orientation of parameters u and v on the surface. If necessary, use the command *Direction* to modify it, see above.

Cartesian coordinates of surface point – Command: *Evaluate Point* → activate *Point object snap* → Point to evaluate: click on the surface at the point of which Cartesian coordinates have to be determined. The Cartesian coordinates of point are displayed in the command prompt.

Patch corners – Command: *Point* → Location of point object: activate *Endpoint object snap* → click near the patch corner. A point (as an individual entity) is drawn at the patch corner.

Parametric curves of surface – Command: *Extract Isocurve* → Select surface for isocurve extraction: click on the surface; the marker moves along the surface only and the parametric curve corresponding to its actual position is dynamically displayed → Select isocurve to extract: choose in the command prompt *Direction=U*, *Toggle Direction=V* or *Direction=Both* → click to place the required parametric curve or the pair of parametric curves → press Enter. Depending on the option chosen in the command prompt, the parametric *u*-curve, parametric *v*-curve or both parametric curves are drawn as individual entities.

Remarks:

1. It is possible to create a wireframe model of the surface by using command *Extract wireframe*.
2. If the direction of parameters given by system does not correspond to the vector equation considered, it is possible to modify this direction by using the command *Direction*, see above.

Patch boundary – Command: *Duplicate Edge*: Select edges to duplicate → click on the required patch boundary → press Enter. The boundary curve is drawn as an individual entity.

Patch border – Command: *Duplicate border*: Select surfaces for duplicate border → click on the patch → press Enter. The border curve is drawn as a group of individual boundaries. Individual boundaries can be obtained by command *Explode* → Select objects to explode: click on the border → press Enter.

Normal line of a surface – Command: *Line: Surface Normal* → Select surface for normal line: click on the surface → Start of line: click at the required intersection of the surface and its normal line → End of line: click at a suitable terminal point of normal line. The straight line segment with the initial point on the surface representing the surface normal line is drawn.

Remark: If the option *BothSides* in the command prompt is chosen, the normal line is drawn as a straight line segment with its center on the surface.

Tangent plane of a surface – Tangent plane of a surface is constructed indirectly.

1. Construct normal line (see above) at a required point of contact between tangent plane and the surface.
2. Create a boundary of the tangent plane by command: *Circle: Around Curve*: Select curve: click on the normal line → Center of circle: activate *Endpoint object snap* → click on the initial point of the normal line → Radius: click at a suitable terminal point of radius.
3. Create the tangent plane by command: *Surface from Planar Curves* → Select planar curves to build surface: click on the boundary circle → press Enter. The disc representing the tangent plane is drawn.

The first curvature of curves located on a surface – Command: *Curvature Graph On* → Select object for curvature graph display: click on the surface or on the drawn parametric curve of surface (see below) → press Enter → adjust *Display scale* and *Density* of perpendicular segments in dialog box *Curvature Graph*. The curvature graph is displayed along a wireframe model of the patch as well as along the previously drawn parametric curves of the surface. Length of perpendicular segments is proportional to the first curvature of the curve at the intersection of the perpendicular segment and the curve.

Radius of curves located on the surface – Command: *Radius* → Select point on curve for radius measurement: click on the curve located on the surface → radius value is displayed in the command prompt.

Principal curvatures – Command: *Curvature* → Select curve of surface for curvature measurement: click on the surface → Select point on surface for curvature measurement → choose in the command prompt *MarkCurvature=Yes* → click on the surface at the point in which the principal curvatures have to be determined. Two osculation circles of the curves of principal normal section are drawn. In the command prompt, the following information is displayed: parameter values (α, β) of the selected point, Cartesian coordinates $\mathbf{P}(\alpha, \beta)$, coordinates of unit normal vector, maximal principal curvature, minimal principal curvature, Gaussian curvature and mean curvature.

If the osculation circles of principal normal section of the surface are drawn, it is possible to classify the elliptical, hyperbolic or parabolic point located on the surface. Consider that the three-dimensional space is divided by the tangent plane of the surface into two half-spaces.

- In the case of elliptical point, both osculation circles are located in the same half-space.
- In the case of hyperbolic point, each osculation circle is located in the different half-space.
- In the case of parabolic point, one osculation circle degenerates into a straight line.

Gaussian curvature, mean curvature – Command: *Curvature Analysis* → Select object for curvature analysis: click on the surface → press Enter. In the command prompt, the following information is displayed: maximum principal curvature together with its unit direction vector, minimum principal curvature together with its unit direction vector, Gaussian curvature and mean curvature. Simultaneously, the curvature is visualized by colour map. Dialog box *Curvature* is displayed. Here, the detailed options of the visualized curvature can be chosen.

Continuity of two patches – Command: *Zebra analysis* → Select objects for zebra stripe analysis: click on the patches their continuity has to be analyzed → press Enter. In the dialog box *Zebra Options* adjust the direction, size and colour of zebra stripes. In the shaded mode, the stripes are mapped on the surface.

The information about continuity of patches is given by continuity of the zebra stripes in the following way.

- The patches are C^0 continuously joined, if their zebra stripes have kinks or they jump sideways as they cross the common boundary, see Fig. 1.16 a).
- The patches are at least G^1 continuously joined, if their zebra stripes match and continue but turn sharply as they cross the common boundary, see Fig. 1.16 b).
- The patches are at least G^2 continuously joined, if their zebra stripes continue smoothly as they cross the common boundary, see Fig. 1.16 c).

It is possible to analyze the continuity of two patches by means of curvature graph (see above) of parametric curves of the surface (see above) as is shown in Fig. 1.17.

■ **Exercise 1.2** In Rhinoceros, create the torus from example 1.5. Choose the radii values R and r in a suitable way.

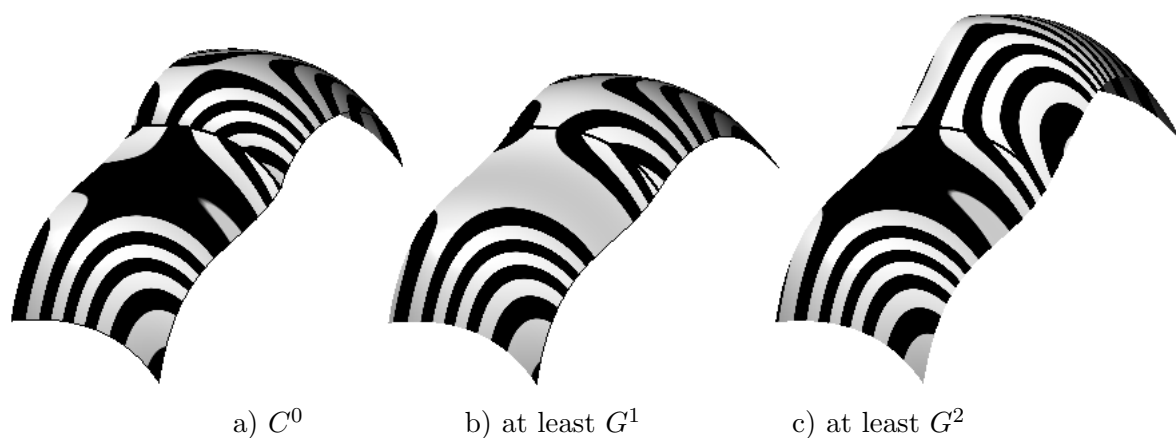


Figure 1.16: Analysis of continuity of two patches by means of zebra stripes

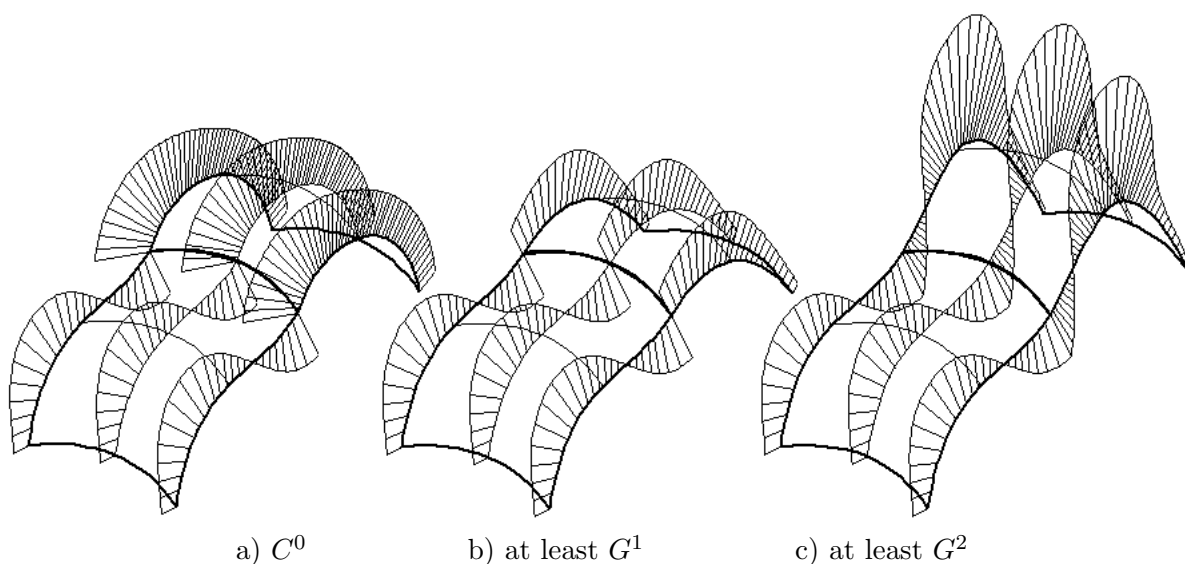


Figure 1.17: Analysis of continuity of two patches by means of curvature graph

At the chosen elliptic, hyperbolic and parabolic points of the torus, construct the tangent plane, normal line and osculation circles of parametric curves. Then, determine the radius of the first curvature, minimal curvature, maximal curvature, Gaussian curvature and mean curvature at these points, too.

To what value is the minimal radius of the first curvature of parametric curves of the torus at an arbitrary point of the torus equal?

Chapter 2

Curves modelling

This chapter is focused on mathematical modelling of free-form curves given by a sequence of points in a plane or space. It is impossible to create more complicated shapes by one curve segment analytically represented by only one vector equation. In CAD/CAM systems, more complicated free-form shapes are usually represented by NURBS (NonUniform Rational B-Spline) representation, where the curves are parameterized by piecewise nonuniform rational functions, see [2] to study NURBS representation. Here, a detailed explanation of mathematical models representing only one curve segment is given. Additionally, the conditions required to create an arbitrary shape by joining curve segments are discussed. Thus, a very effective but still simple tool for mathematical modelling of free-form curves is introduced.

The term *curve* will be generally used. If necessary, the terms *curve* and *curve segment* will be distinguished.

In computer graphics, vector equation with polynomial coordinate functions is widely used as analytical representation of curves. In the case third degree polynomials, the resulted curve is a so called *cubic curve*. The parametrization domain of polynomial curves is closed interval $[0, 1]$, so called a *normalized parametrization domain*¹. Orientation of the curve is given by parametrization domain. At the initial point of the curve, the parameter value equals zero. At the terminal point of the curve, the parameter value equals one.

Considering the input data of a curve in a form of a sequence of points, the vector equation is represented by a linear combination of *basic functions* (polynomials), where the coefficients of individual basis functions are the input data. Two basic approaches can be distinguished when interpreting the input data: *interpolation* and *approximation*. In the case of interpolation, the input data is referred to as *definition points* and creates a *definition polygon*. Interpolation curve has to pass through all definition points. The shape of interpolation curve between definition points has to meet all requirements of technical application used. In the case of approximation, the input data is referred to as *control points* creating a *control polygon*. Approximation curve does not have to pass through the control points (some of the control points can be interpolated). The shape of an approximation curve follows the shape of its control polygon.

Remark: Position vectors of definition points as well as position vectors of control points are always considered in this textbook. Therefore, vector denotation (Cartesian coordinates in round brackets) will be used.

¹No restriction follows from using the normalized parametrization domain. Reparametrization $t = \frac{s-a}{b-a}$ of general domain of parametrization can be done for each parameter value $s \in [a, b]$ to normalize the original parametrization domain $[a, b]$.

2.1 Ferguson cubic curve

Ferguson cubic curve is created by one segment that passes through two given definition points (the curve interpolates these points).

■ **Definition 2.1 – Ferguson cubic curve.** Let \mathbf{A} and \mathbf{B} be definition points and \mathbf{a} and \mathbf{b} be tangent vectors at these points. Then, the vector equation of *Ferguson cubic curve* is

$$\mathbf{P}(t) = F_0(t)\mathbf{A} + F_1(t)\mathbf{B} + F_2(t)\mathbf{a} + F_3(t)\mathbf{b}, \quad t \in [0, 1], \quad (2.1)$$

where the basis functions

$$\begin{aligned} F_0(t) &= 2t^3 - 3t^2 + 1, \\ F_1(t) &= -2t^3 + 3t^2, \\ F_2(t) &= t^3 - 2t^2 + t, \\ F_3(t) &= t^3 - t^2, \quad t \in [0, 1] \end{aligned} \quad (2.2)$$

are third degree *Hermit polynomials*. □

Graphical representation of Hermit polynomials is shown in Fig. 2.1 a). In Fig. 2.1 b), an example of planar Ferguson cubic curve is drawn.

A spatial Ferguson cubic curve arises if z -coordinate of definition point \mathbf{A} or \mathbf{B} is non-zero.

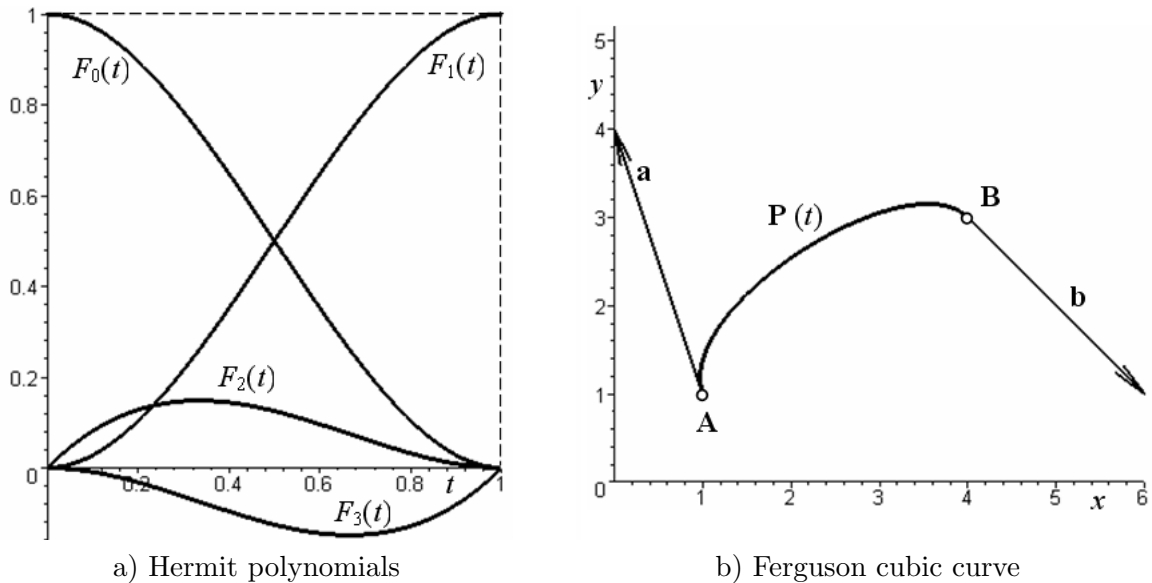


Figure 2.1: Hermit polynomials and properties of Ferguson cubic curve

2.1.1 Properties of Ferguson cubic curve

The properties of Ferguson cubic curve follow from the definition.

- The initial point $\mathbf{P}(0)$ of Ferguson cubic curve (2.1) is the given point \mathbf{A} .
- The terminal point $\mathbf{P}(1)$ of Ferguson cubic curve (2.1) is the given point \mathbf{B} .
- Tangent vector $\mathbf{P}'(0)$ of Ferguson cubic curve (2.1) is the given tangent vector \mathbf{a} .
- Tangent vector $\mathbf{P}'(1)$ of Ferguson cubic curve (2.1) is the given tangent vector \mathbf{b} .

- **Example 2.1 – Ferguson cubic curve.** Ferguson cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by definition points $\mathbf{A} = (1, 0)$ and $\mathbf{B} = (4, 2)$ and tangent vectors $\mathbf{a} = (-1, 3)$ and $\mathbf{b} = (2, -2)$ at these points.

Find the parametric expression and vector equation of Ferguson cubic curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1, calculate the coordinates of points on Ferguson cubic curve and tangent vectors at these points. Draw all calculated points and construct tangent vectors at these points. Sketch Ferguson cubic curve.

Solution. After substituting x -coordinates of the definition points and x -coordinates of tangent vectors in (2.1), we obtain x -coordinate function of Ferguson cubic curve

$$x(t) = (2t^3 - 3t^2 + 1) + 4(-2t^3 + 3t^2) - 1(t^3 - 2t^2 + t) + 2(t^3 - t^2) = -5t^3 + 9t^2 - t + 1.$$

Similarly, after substituting y -coordinates of the given definition points and y -coordinates of tangent vector in (2.1), we obtain y -coordinate function of Ferguson cubic curve

$$y(t) = 2(-2t^3 + 3t^2) + 3(t^3 - 2t^2 + t) - 2(t^3 - t^2) = -3t^3 + 2t^2 + 3t.$$

Then, the parametric expression of Ferguson cubic curve is

$$\begin{aligned} x(t) &= -5t^3 + 9t^2 - t + 1, \\ y(t) &= -3t^3 + 2t^2 + 3t, \quad t \in [0, 1]. \end{aligned} \quad (2.3)$$

The vector equation of Ferguson cubic curve is

$$\mathbf{P}(t) = (-5t^3 + 9t^2 - t + 1, -3t^3 + 2t^2 + 3t), \quad t \in [0, 1]. \quad (2.4)$$

The first derivative of x -coordinate function and y -coordinate function (2.3) is x -coordinate function and y -coordinate function of tangent vector of Ferguson cubic curve, respectively. The parametric expression of tangent vector of Ferguson cubic curve is

$$\begin{aligned} x'(t) &= (-5t^3 + 9t^2 - t + 1)' = -15t^2 + 18t - 1, \\ y'(t) &= (-3t^3 + 2t^2 + 3t)' = -9t^2 + 4t + 3, \quad t \in [0, 1]. \end{aligned} \quad (2.5)$$

The vector equation of tangent vector of Ferguson cubic curve is

$$\mathbf{P}'(t) = (-15t^2 + 18t - 1, -9t^2 + 4t + 3), \quad t \in [0, 1]. \quad (2.6)$$

After substituting $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1 in (2.4), we obtain the points on Ferguson cubic curve. After substituting $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1 in (2.6), we obtain tangent vectors at these points. All calculated values are summarized in the following table.

The shape of Ferguson cubic curve (2.4) and length and direction of tangent vector (2.6) along this curve is obvious from Fig. 2.2.

	$t = 0$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
$\mathbf{P}(t)$	(1, 0)	$(\frac{79}{64}, \frac{53}{64})$	$(\frac{17}{8}, \frac{13}{8})$	$(\frac{205}{64}, \frac{135}{64})$	(4, 2)
$\mathbf{P}'(t)$	(-1, 3)	$(\frac{41}{16}, \frac{55}{16})$	$(\frac{17}{4}, \frac{11}{4})$	$(\frac{65}{16}, \frac{15}{16})$	(2, -2)

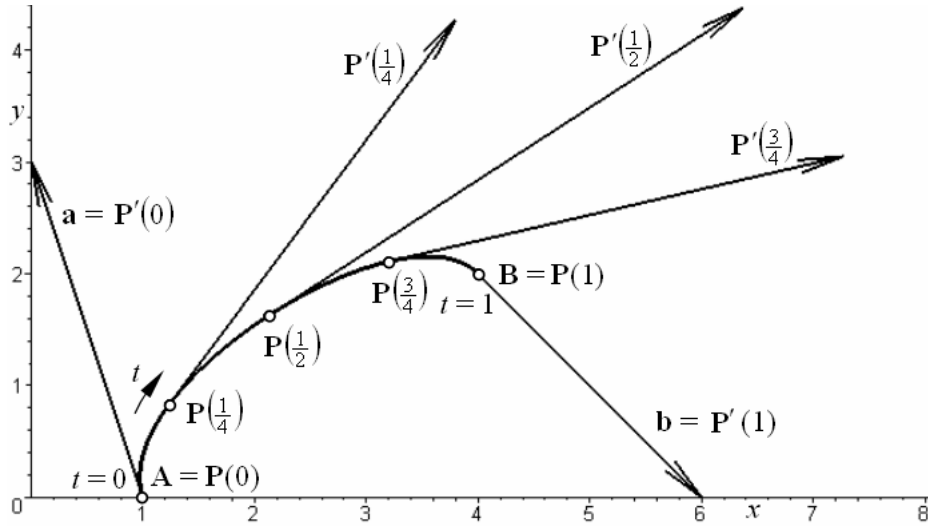


Figure 2.2: Ferguson cubic curve

□

2.1.2 Hermite polynomials derivation

Assume the input data \mathbf{A} , \mathbf{B} , \mathbf{a} and \mathbf{b} of Ferguson cubic curve is given according to the Definition 2.1. To derive Hermite polynomials, we write a general vector equation of a cubic polynomial curve

$$\mathbf{P}(t) = mt^3 + nt^2 + pt + q, \quad t \in [0, 1] \quad (2.7)$$

and its first derivative

$$\mathbf{P}'(t) = 3mt^2 + 2nt + p, \quad t \in [0, 1], \quad (2.8)$$

where the coefficients m , n , p and q are unknown. If the curve (2.7) is a Ferguson cubic curve, the following conditions have to be satisfied:

$$\mathbf{P}(0) = \mathbf{A}, \quad \mathbf{P}(1) = \mathbf{B}, \quad \mathbf{P}'(0) = \mathbf{a} \quad \text{and} \quad \mathbf{P}'(1) = \mathbf{b}.$$

After substituting $t = 0$ and $t = 1$ in (2.7) and (2.8), the following set of four equations is obtained

$$\begin{aligned} \mathbf{P}(0) &= q = \mathbf{A} \\ \mathbf{P}(1) &= m + n + p + q = \mathbf{B} \\ \mathbf{P}'(0) &= p = \mathbf{a} \\ \mathbf{P}'(1) &= 3m + 2n + p = \mathbf{b}. \end{aligned} \quad (2.9)$$

Solving for unknown coefficients m , n , p and q , we get

$$\begin{aligned} m &= 2\mathbf{A} - 2\mathbf{B} + \mathbf{a} + \mathbf{b}, \\ n &= -3\mathbf{A} + 3\mathbf{B} - 2\mathbf{a} - \mathbf{b}, \\ p &= \mathbf{a}, \\ q &= \mathbf{A}. \end{aligned} \quad (2.10)$$

After substituting 2.10 in (2.7), we obtain

$$\mathbf{P}(t) = (2\mathbf{A} - 2\mathbf{B} + \mathbf{a} + \mathbf{b})t^3 + (-3\mathbf{A} + 3\mathbf{B} - 2\mathbf{a} - \mathbf{b})t^2 + \mathbf{a}t + \mathbf{A}. \quad (2.11)$$

After simplifying, we get the equation of Ferguson cubic curve

$$\mathbf{P}(t) = (2t^3 - 3t^2 + 1)\mathbf{A} + (-2t^3 + 3t^2)\mathbf{B} + (t^3 - 2t^2 + t)\mathbf{a} + (t^3 - t^2)\mathbf{b}, \quad (2.12)$$

where the polynomials multiplying the individual input data are the derived Hermit polynomials (2.2).

2.1.3 Continuity at common point of Ferguson cubic curves

Consider a sequence of definition points $\mathbf{Q}_0, \mathbf{Q}_1, \dots, \mathbf{Q}_n$ and a sequence of tangent vectors $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_n$ at these points. Each two consecutive definition points \mathbf{Q}_i and \mathbf{Q}_{i+1} are considered to be the endpoints of i -th Ferguson cubic curve k_i given by vector equation

$$\mathbf{P}_i(t) = F_0(t)\mathbf{Q}_i + F_1(t)\mathbf{Q}_{i+1} + F_2(t)\mathbf{q}_i + F_3(t)\mathbf{q}_{i+1}, \quad t \in [0, 1], \quad i = 0, 1, \dots, n-1. \quad (2.13)$$

Therefore, it is possible to interpolate the given sequence of definition points by piecewise curve k consisting of n curve segments – Ferguson cubic curves k_i , $i = 0, 1, \dots, n-1$. Each two consecutive curve segments are automatically C^1 continuously joined, because their tangent vectors at the common point are identical. Parametrization domain of each individual Ferguson cubic curve is normalized, i.e. $t \in [0, 1]$.

The general procedure to ensure C^2 continuity along a piecewise interpolation curve consisting of Ferguson cubic curves is not given in this textbook.

■ **Example 2.2 – “@” sign created by Ferguson cubic curves.** Two piecewise curves k^* and k consisting of individual Ferguson cubic curves are given by the following sequences of definition points and tangent vectors at these points:

$$k^* : \quad \mathbf{Q}_0^* = (47, 46), \quad \mathbf{Q}_1^* = (22, 34), \quad \mathbf{Q}_2^* = (31, 20), \quad \mathbf{Q}_3^* = \mathbf{Q}_0^* = (47, 46), \\ \mathbf{q}_0^* = (-54, 21), \quad \mathbf{q}_1^* = (-3, -24), \quad \mathbf{q}_2^* = (18, -3), \quad \mathbf{q}_3^* = (6, 72),$$

$$k : \quad \mathbf{Q}_0 = (47, 46), \quad \mathbf{Q}_1 = (58, 20), \quad \mathbf{Q}_2 = (64, 35), \quad \mathbf{Q}_3 = (55, 56), \\ \mathbf{Q}_4 = (26, 61), \quad \mathbf{Q}_5 = (9, 28), \quad \mathbf{Q}_6 = (47, 7), \\ \mathbf{q}_0 = (3, -78), \quad \mathbf{q}_1 = (12, 6), \quad \mathbf{q}_2 = (0, 21), \quad \mathbf{q}_3 = (-21, 18), \\ \mathbf{q}_4 = (-30, -15), \quad \mathbf{q}_5 = (3, -42), \quad \mathbf{q}_6 = (81, 21).$$

Curves k^* and k form “@” sign.

Draw the definition points, construct the tangent vectors and sketch all Ferguson cubic curves. What is the continuity along piecewise curves k^* and k ?

Solution. In Fig. 2.3, the sign “@” created from Ferguson cubic curves interpolating all the given definition points is drawn.

All individual Ferguson cubic curves of both interpolation curves are C^1 continuously joined with the only exception at point $\mathbf{Q}_0^* = \mathbf{Q}_3^*$. Here, curve k^* is closed with C^0 continuity.

Remark: Moreover, as we will see in Example 2.6, the definition points as well as tangent vectors to create the “@” sign from Example 2.2 have been chosen in order to ensure C^2 continuity along curves k and k^* .

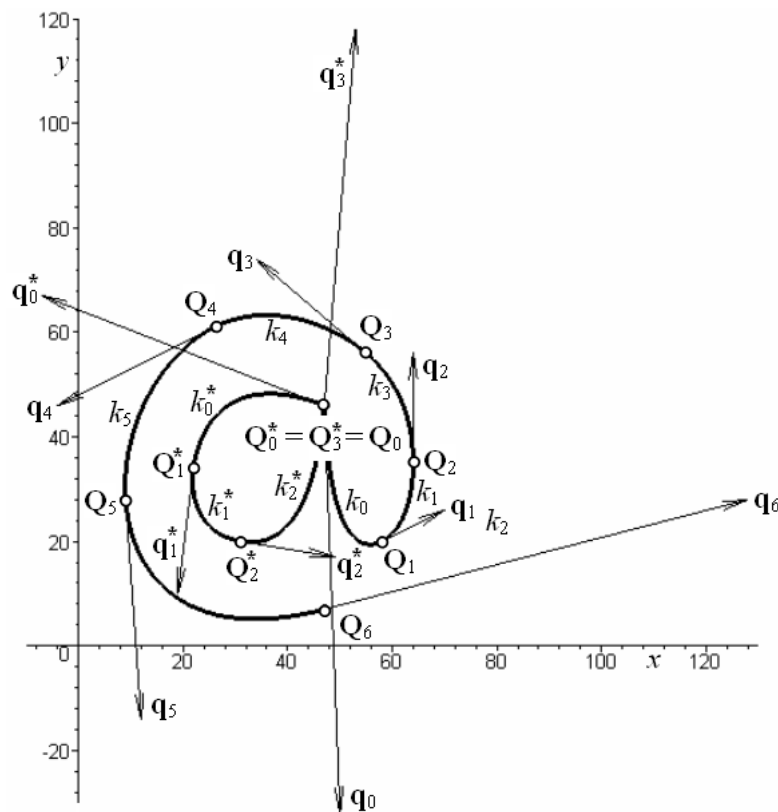


Figure 2.3: “@” sign created by Ferguson cubic curves

□

- **Exercise 2.1** Ferguson cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by definition points $\mathbf{A} = (0, 0)$ and $\mathbf{B} = (2, 0)$ and tangent vectors \mathbf{a} and \mathbf{b} at these points.

Find the parametric expression and vector equation of Ferguson cubic curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0$, $t = \frac{1}{2}$ and $t = 1$, calculate the coordinates of points on Ferguson cubic curve and tangent vectors at these points. Draw all calculated points and construct tangent vectors at these points. Sketch Ferguson cubic curve. Consider the following input data:

- a) $\mathbf{a} = (0, 1)$, $\mathbf{b} = (0, -1)$, b) $\mathbf{a} = (-1, 1)$, $\mathbf{b} = (1, -1)$,
 c) $\mathbf{a} = (1, 1)$, $\mathbf{b} = (1, 1)$, d) $\mathbf{a} = (-1, -1)$, $\mathbf{b} = (-1, -1)$.

- **Exercise 2.2** Piecewise interpolation curve k consisting of individual Ferguson cubic curves is given by the following sequence of definition points and tangent vectors at these points:

$$\mathbf{Q}_0 = (0, 0), \quad \mathbf{Q}_1 = (6, 0), \quad \mathbf{Q}_2 = (0, 0), \quad \mathbf{Q}_3 = (-6, 0), \quad \mathbf{Q}_4 = (0, 0),$$

$$\mathbf{q}_0 = (9, 9), \quad \mathbf{q}_1 = (0, -9), \quad \mathbf{q}_2 = (-9, 9), \quad \mathbf{q}_3 = (0, -9), \quad \mathbf{q}_4 = (9, 9).$$

Draw the given definition points, construct the given tangent vectors and sketch curve k . How many individual Ferguson cubic curves create curve k ? Determine the continuity along curve k without calculating it. Find vector equations of all Ferguson cubic curves and verify the assumed continuity.

- **Exercise 2.3** Find vector equations of individual Ferguson cubic curves creating the “@” sign from Example 2.2 and verify continuity along curves k and k^* .

2.1.4 Ferguson cubic curve in Rhinoceros

In Rhinoceros, it is impossible to create Ferguson cubic curve directly. However, the interrelation between Ferguson cubic curve and Bézier cubic curve (see Section 2.2.5) can be used to draw Ferguson cubic curve.

2.2 Bézier curve

Bézier curve of degree n is created by one segment given by $n + 1$ control points ordered in a control polygon. Even if Bézier curve passes through the first and the last control point (interpolates these points), it belongs to approximation curves.

- **Definition 2.2 – Bézier curve.** Let $\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_n$ be control points in space R^2 . Then, the vector equation of *Bézier curve* is

$$\mathbf{P}(t) = \sum_{i=0}^n B_{i,n}(t) \mathbf{V}_i = B_{0,n}(t) \mathbf{V}_0 + B_{1,n}(t) \mathbf{V}_1 + \dots + B_{n,n}(t) \mathbf{V}_n, \quad t \in [0, 1], \quad (2.14)$$

where the coefficients

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \quad t \in [0, 1], \quad i = 0, \dots, n, \quad (2.15)$$

are *Bernstein polynomials* of degree n . □

The first subscript of Bernstein polynomial correspond to the subscript of the associated control point. The second subscript denotes degree of Bernstein polynomial. Note that the domain of t is $[0, 1]$. As a result, all Bernstein polynomials are non-negative. Since t and i can both be zero and so can $1 - t$ and $n - i$, we adopt the convention that 0^0 equals 1.

After substituting $n = 0, 1, 2$ and 3 in (2.15), we obtain Bernstein polynomials of degrees 0, 1, 2 and 3. Analytical expressions of these Bernstein polynomials are written in Tab. 2.1 and graphical representations are shown in Fig. 2.4. Note that the sum of all Bernstein polynomials of the same degree is always equal to 1.

Table 2.1: Bernstein polynomials of degrees 0, 1, 2 and 3

	$n = 0$	$n = 1$	$n = 2$	$n = 3$
$i = 0$	$B_{0,0}(t) = 1$	$B_{0,1}(t) = 1 - t$	$B_{0,2}(t) = (1 - t)^2$	$B_{0,3}(t) = (1 - t)^3$
$i = 1$		$B_{1,1}(t) = t$	$B_{1,2}(t) = 2t(1 - t)$	$B_{1,3}(t) = 3t(1 - t)^2$
$i = 2$			$B_{2,2}(t) = t^2$	$B_{2,3}(t) = 3t^2(1 - t)$
$i = 3$				$B_{3,3}(t) = t^3$

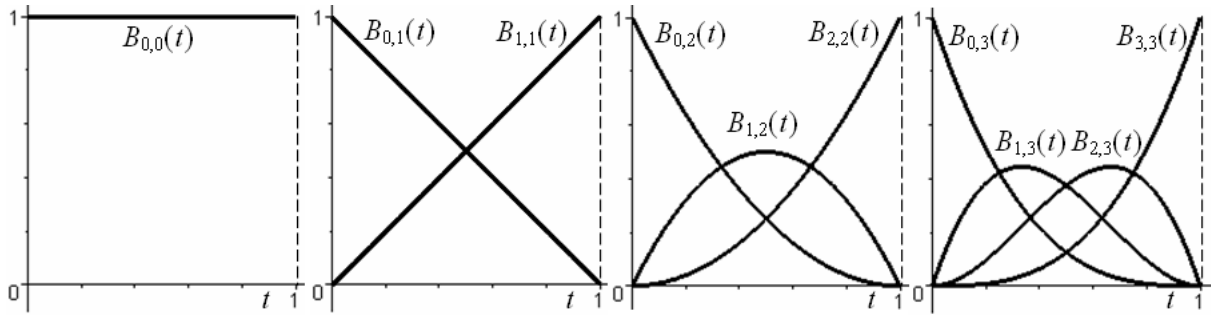


Figure 2.4: Bernstein polynomials of degrees 0,1,2 and 3

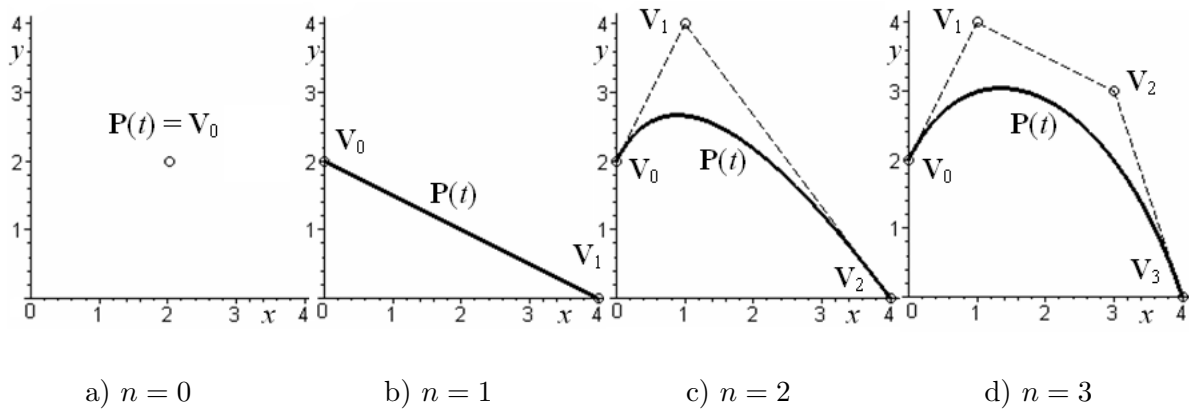


Figure 2.5: Bézier curves of degrees 0, 1, 2 and 3

If the control polygon has only one control point \mathbf{V}_0 , Bézier curve is exactly this control point \mathbf{V}_0 , because the vector equation of zeroth degree Bézier curve is

$$\mathbf{P}(t) = B_{0,0}(t)\mathbf{V}_0 = \mathbf{V}_0, \quad t \in [0, 1]. \quad (2.16)$$

In the case of two control points \mathbf{V}_0 and \mathbf{V}_1 , the vector equation of first degree Bézier curve is given by

$$\mathbf{P}(t) = B_{0,1}(t)\mathbf{V}_0 + B_{1,1}(t)\mathbf{V}_1 = (1-t)\mathbf{V}_0 + t\mathbf{V}_1, \quad t \in [0, 1]. \quad (2.17)$$

Thus, a straight line segment connecting control points \mathbf{V}_0 and \mathbf{V}_1 is obtained.

In practice, second degree Bézier curves (Bézier quadratic curves) and third degree Bézier curves (Bézier cubic curves) are widely used. The vector equation of Bézier quadratic curve given by control points \mathbf{V}_0 , \mathbf{V}_1 and \mathbf{V}_2 is

$$\begin{aligned} \mathbf{P}(t) &= B_{0,2}(t)\mathbf{V}_0 + B_{1,2}(t)\mathbf{V}_1 + B_{2,2}(t)\mathbf{V}_2 = \\ &= (1-t)^2\mathbf{V}_0 + 2t(1-t)\mathbf{V}_1 + t^2\mathbf{V}_2, \quad t \in [0, 1], \end{aligned} \quad (2.18)$$

and the vector equation of Bézier cubic curve given by control points \mathbf{V}_0 , \mathbf{V}_1 , \mathbf{V}_2 and \mathbf{V}_3 is

$$\begin{aligned} \mathbf{P}(t) &= B_{0,3}(t)\mathbf{V}_0 + B_{1,3}(t)\mathbf{V}_1 + B_{2,3}(t)\mathbf{V}_2 + B_{3,3}(t)\mathbf{V}_3 = \\ &= (1-t)^3\mathbf{V}_0 + 3t(1-t)^2\mathbf{V}_1 + 3t^2(1-t)\mathbf{V}_2 + t^3\mathbf{V}_3, \quad t \in [0, 1]. \end{aligned} \quad (2.19)$$

Examples of Bézier curves of degrees 0, 1, 2 and 3 are drawn in Fig. 2.5.

■ **Example 2.3 – Bézier quadratic curve.** Bézier curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_0 = (2, 0)$, $\mathbf{V}_1 = (4, 1)$, $\mathbf{V}_2 = (3, 2)$.

Find the parametric expression and vector equation of Bézier curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1, calculate the coordinates of points on Bézier curve and tangent vectors at these points. Draw control points, construct control polygon, draw calculated points on Bézier curve and tangent vectors at these points. Sketch Bézier curve.

Solution. Three control points are given. Therefore, we obtain parametric expression and vector equation of Bézier curve when substituting coordinates of given control points in vector equation of Bézier quadratic curve (2.18). The parametric expression of Bézier curve and its tangent vector are then (the shorter form is already used)

$$\begin{aligned}x(t) &= 2B_{0,2}(t) + 4B_{1,2}(t) + 3B_{2,2}(t) = 2(1-t)^2 + 4 \cdot 2t(1-t) + 3t^2 = -3t^2 + 4t + 2, \\y(t) &= B_{1,2}(t) + 2B_{2,2}(t) = 2t(1-t) + 2t^2 = 2t, \quad t \in [0, 1], \\x'(t) &= (-3t^2 + 4t + 2)' = -6t + 4, \\y'(t) &= (2t)' = 2, \quad t \in [0, 1],\end{aligned}$$

The vector equations of Bézier curve and its tangent vector are

$$\mathbf{P}(t) = (-3t^2 + 4t + 2, 2t), \quad t \in [0, 1], \quad (2.20)$$

$$\mathbf{P}'(t) = (-6t + 4, 2), \quad t \in [0, 1]. \quad (2.21)$$

After substituting $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1 in (2.20) and (2.21), we obtain the points on Bézier curve and tangent vectors at these points, respectively (see the following table). The shape of Bézier curve (2.20) and length and direction of tangent vector (2.21) along this curve are obvious from Fig. 2.6.

	$t = 0$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
$\mathbf{P}(t)$	$(2, 0)$	$(\frac{45}{16}, \frac{1}{2})$	$(\frac{13}{4}, 1)$	$(\frac{53}{16}, \frac{3}{2})$	$(3, 2)$
$\mathbf{P}'(t)$	$(4, 2)$	$(\frac{5}{2}, 2)$	$(1, 2)$	$(-\frac{1}{2}, 2)$	$(-2, 2)$

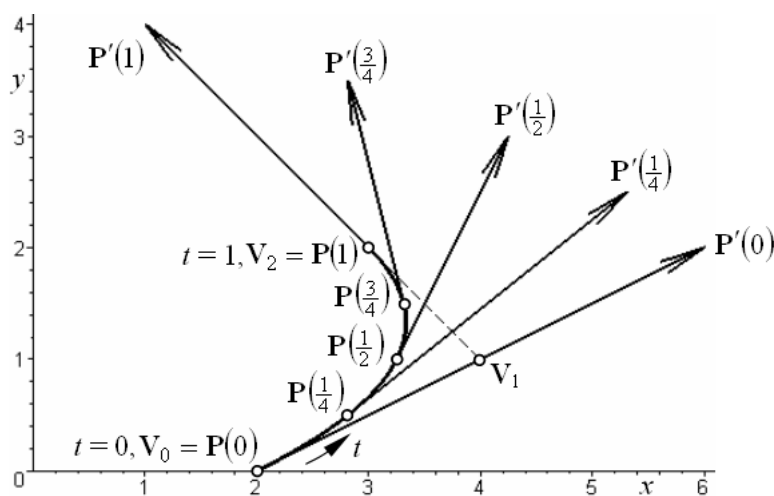


Figure 2.6: Bézier quadratic curve

□

■ **Example 2.4 – Bézier cubic curve.** Bézier curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_0 = (2, 3)$, $\mathbf{V}_1 = (4, 2)$, $\mathbf{V}_2 = (0, 0)$, $\mathbf{V}_3 = (-1, 1)$.

Find the parametric expression and vector equation of Bézier curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1, calculate the coordinates of points on Bézier curve and tangent vectors at these points. Draw control points, construct control polygon, draw calculated points on Bézier curve and tangent vectors at these points. Sketch Bézier curve.

Solution. Four control points are given. Therefore, vector equation of Bézier cubic curve (2.19) has to be considered. The parametric expression of Bézier cubic curve and its tangent vector are

$$x(t) = 2B_{0,3}(t) + 4B_{1,3}(t) - B_{3,3}(t) = 9t^3 - 18t^2 + 6t + 2,$$

$$y(t) = 3B_{0,3}(t) + 2B_{1,3}(t) + B_{3,3}(t) = 4t^3 - 3t^2 - 3t + 3, \quad t \in [0, 1],$$

$$x'(t) = 27t^2 - 36t + 6,$$

$$y'(t) = 12t^2 - 6t - 3, \quad t \in [0, 1].$$

The vector equations of Bézier cubic curve and its tangent vector are

$$\mathbf{P}(t) = (9t^3 - 18t^2 + 6t + 2, 4t^3 - 3t^2 - 3t + 3), \quad t \in [0, 1], \quad (2.22)$$

$$\mathbf{P}'(t) = (27t^2 - 36t + 6, 12t^2 - 6t - 3), \quad t \in [0, 1]. \quad (2.23)$$

Coordinates of points on Bézier cubic curve 2.22 and tangent vectors 2.23 at $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1 are summarized in the following table. The shape of Bézier cubic curve (2.22) and length and direction of tangent vector (2.23) along this curve are obvious from Fig. 2.7.

	$t = 0$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
$\mathbf{P}(t)$	(2, 3)	$(\frac{161}{64}, \frac{17}{8})$	$(\frac{13}{8}, \frac{5}{4})$	$(\frac{11}{64}, \frac{3}{4})$	(-1, 1)
$\mathbf{P}'(t)$	(6, -3)	$(-\frac{21}{16}, -\frac{15}{4})$	$(-\frac{21}{4}, -3)$	$(-\frac{93}{16}, -\frac{3}{4})$	(-3, 3)

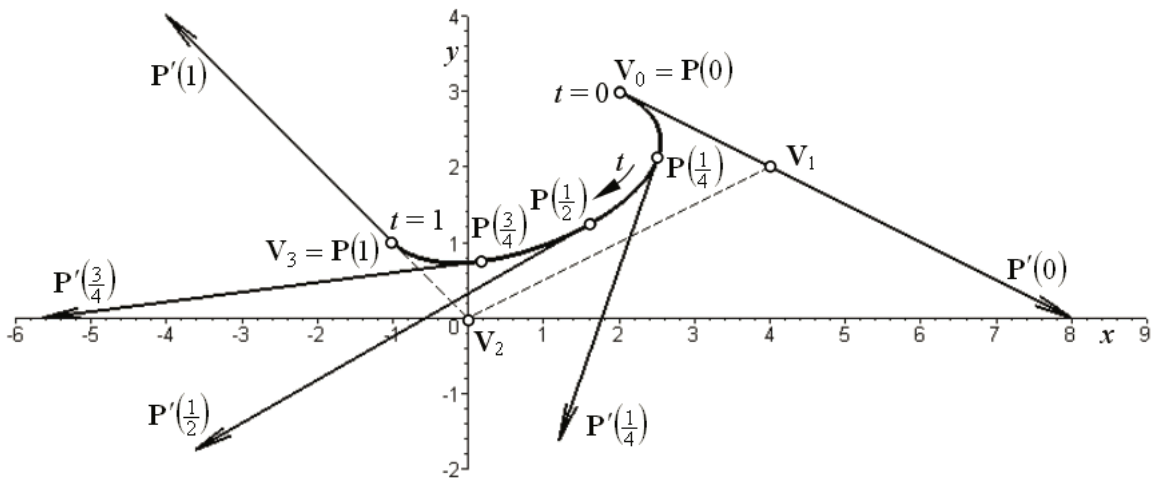


Figure 2.7: Bézier cubic curve

□

2.2.1 Properties of Bézier curve

Bézier curve has some interesting geometric properties:

- Bézier curve (2.14) always interpolates the endpoints of its control polygon, i.e. $\mathbf{P}(0) = \mathbf{V}_0$, $\mathbf{P}(1) = \mathbf{V}_n$.
- Tangent vector $\mathbf{P}'(0)$ at the initial point of Bézier curve (2.14) equals n -multiple of the vector given by the first leg of control polygon: $\mathbf{P}'(0) = n\overrightarrow{\mathbf{V}_0\mathbf{V}_1} = n(\mathbf{V}_1 - \mathbf{V}_0)$.
- Tangent vector $\mathbf{P}'(1)$ at the terminal point of Bézier curve (2.14) equals n -multiple of the vector given by the last leg of control polygon: $\mathbf{P}'(1) = n\overrightarrow{\mathbf{V}_{n-1}\mathbf{V}_n} = n(\mathbf{V}_n - \mathbf{V}_{n-1})$.
- If all control points of Bézier curve are collinear, the Bézier curve is a straight line segment. This property is a so called *linear accuracy*.

Bézier and Ferguson cubic curves interrelation

Considering the previously formulated properties of Bézier curve, it is obvious that some interrelation between Bézier cubic curve and Ferguson cubic curve will exist. To derive this interrelation, assume the Bézier cubic curve is given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$. After substitution the following input data

$$\mathbf{A} = \mathbf{V}_0, \mathbf{B} = \mathbf{V}_3, \mathbf{a} = 3\overrightarrow{\mathbf{V}_0\mathbf{V}_1} = 3(\mathbf{V}_1 - \mathbf{V}_0), \mathbf{b} = 3\overrightarrow{\mathbf{V}_2\mathbf{V}_3} = 3(\mathbf{V}_3 - \mathbf{V}_2), \quad (2.24)$$

in vector equation of Ferguson cubic curve (2.1), we get

$$\begin{aligned} \mathbf{P}(t) &= F_0(t)\mathbf{V}_0 + F_1(t)\mathbf{V}_3 + 3F_2(t)(\mathbf{V}_1 - \mathbf{V}_0) + 3F_3(t)(\mathbf{V}_3 - \mathbf{V}_2) = \\ &= [F_0(t) - 3F_2(t)]\mathbf{V}_0 + 3F_2(t)\mathbf{V}_1 - 3F_3(t)\mathbf{V}_2 + [F_1(t) + 3F_3(t)]\mathbf{V}_3 = \\ &= (2t^3 - 3t^2 + 1 - 3t^3 + 6t^2 - 3t)\mathbf{V}_0 + (3t^3 - 6t^2 + 3t)\mathbf{V}_1 - \\ &\quad - (3t^3 - 3t^2)\mathbf{V}_2 + (-2t^3 + 3t^2 + 3t^3 - 3t^2)\mathbf{V}_3 = \\ &= (1-t)^3\mathbf{V}_0 + 3t(1-t)^2\mathbf{V}_1 + 3t^2(1-t)\mathbf{V}_2 + t^3\mathbf{V}_3, \end{aligned} \quad (2.25)$$

i.e. the equation of Bézier cubic curve given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$.

Therefore, Ferguson cubic curve is special case of Bézier cubic curve its initial control point is equal to the initial point of Ferguson cubic curve, its terminal control point is equal to the terminal point of Ferguson cubic curve, the triplication of the initial leg of control polygon is equal to the tangent vector at the initial point of Ferguson cubic curve and the triplication of the terminal leg of control polygon is equal to the tangent vector at the terminal point of Ferguson cubic curve.

When solving the equation (2.25) with respect to the control points \mathbf{V}_0 , \mathbf{V}_1 , \mathbf{V}_2 and \mathbf{V}_3 , we get the following transformation between the control points of Bézier cubic curve and input data of Ferguson cubic curve:

$$\mathbf{V}_0 = \mathbf{A}, \mathbf{V}_1 = \mathbf{A} + \frac{1}{3}\mathbf{a}, \mathbf{V}_2 = \mathbf{B} - \frac{1}{3}\mathbf{b}, \mathbf{V}_3 = \mathbf{B}. \quad (2.26)$$

■ **Example 2.5 – Bézier and Ferguson cubic curves interrelation.** The following control points are given $\mathbf{V}_0 = (-2, 1)$, $\mathbf{V}_1 = (0, 0)$, $\mathbf{V}_2 = (1, 3)$ and $\mathbf{V}_3 = (2, 2)$. Find parametric expression of Bézier cubic curve $\mathbf{P}_B(t)$, $t \in [0, 1]$. Using (2.24), determine the input data of Ferguson cubic curve $\mathbf{P}_F(t) = (x_F(t), y_F(t))$ and find its parametric expression. Compare the results. Sketch the situation.

Solution. Parametric expression of Bézier cubic curve $\mathbf{P}_B(t) = (x_B(t), y_B(t))$ is

$$\begin{aligned} x_B(t) &= -2B_{0,3}(t) + B_{2,3}(t) + 2B_{3,3}(t) = t^3 - 3t^2 + 6t - 2, \\ y_B(t) &= B_{0,3}(t) + 3B_{2,3}(t) + 2B_{3,3}(t) = -8t^3 + 12t^2 - 3t + 1, \quad t \in [0, 1]. \end{aligned}$$

The input data of Ferguson cubic curve $\mathbf{P}_F(t)$ is according to (2.24)

$$\begin{aligned} \mathbf{A} &= \mathbf{V}_0 = (-2, 1), \\ \mathbf{B} &= \mathbf{V}_3 = (2, 2), \\ \mathbf{a} &= 3(\mathbf{V}_1 - \mathbf{V}_0) = (6, -3), \\ \mathbf{b} &= 3(\mathbf{V}_3 - \mathbf{V}_2) = (3, -3). \end{aligned}$$

Parametric expression of Ferguson cubic curve $\mathbf{P}_F(t) = (x_F(t), y_F(t))$ is

$$\begin{aligned} x_F(t) &= -2F_0(t) + 2F_1(t) + 6F_2(t) + 3F_3(t) = t^3 - 3t^2 + 6t - 2, \\ y_F(t) &= F_0(t) + 2F_1(t) - 3F_2(t) - 3F_3(t) = -8t^3 + 12t^2 - 3t + 1, \quad t \in [0, 1]. \end{aligned}$$

Since $x_B(t) = x_F(t)$ and $y_B(t) = y_F(t)$, both cubic curves are identical. This situation is drawn in Fig. 2.8.

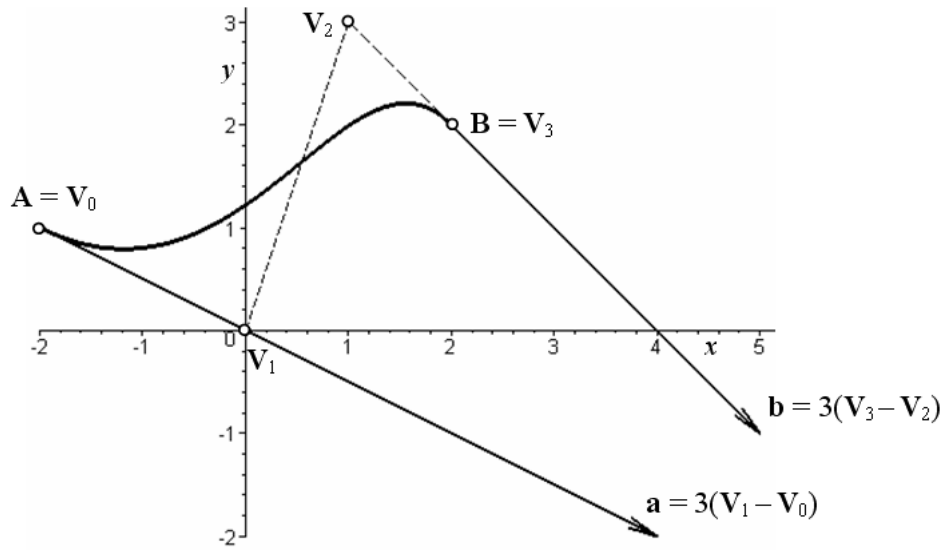


Figure 2.8: Bézier and Ferguson cubic curve interrelation

□

- **Example 2.6 – “@” sign created by Bézier cubic curves.** Assume the individual segments of interpolation curves k and k^* creating the “@” sign in Example 2.2 are Bézier cubic curves.

Determine the control points of individual Bézier cubic curves, draw these control points, construct control polygons and sketch all Bézier cubic curves.

Solution. Curve k^* consists of three Bézier cubic curves k_0^* , k_1^* and k_2^* , each of them given by four control points. Each two consecutive Bézier cubic curves are mutually joined. Therefore, $3 \times 4 - 3 = 9$ different control points have to be determined. Designate these unknown control points by $\mathbf{V}_0^*, \dots, \mathbf{V}_8^*$. According (2.26), we obtain

$$\begin{aligned}
\mathbf{V}_0^* &= \mathbf{A}_0 = (47, 46), \\
\mathbf{V}_1^* &= \mathbf{A}_0 + \frac{1}{3}\mathbf{a}_0 = (47, 46) + \frac{1}{3}(-54, 21) = (29, 53), \\
\mathbf{V}_2^* &= \mathbf{A}_1 - \frac{1}{3}\mathbf{a}_1 = (22, 34) - \frac{1}{3}(-3, -24) = (23, 42), \\
\mathbf{V}_3^* &= \mathbf{A}_1 = (22, 34), \\
\mathbf{V}_4^* &= \mathbf{A}_1 + \frac{1}{3}\mathbf{a}_1 = (22, 34) + \frac{1}{3}(-3, -24) = (21, 26), \\
\mathbf{V}_5^* &= \mathbf{A}_2 - \frac{1}{3}\mathbf{a}_2 = (31, 20) - \frac{1}{3}(18, -3) = (25, 21), \\
\mathbf{V}_6^* &= \mathbf{A}_2 = (31, 20), \\
\mathbf{V}_7^* &= \mathbf{A}_2 + \frac{1}{3}\mathbf{a}_2 = (31, 20) + \frac{1}{3}(18, -3) = (37, 19), \\
\mathbf{V}_8^* &= \mathbf{A}_3 - \frac{1}{3}\mathbf{a}_3 = (47, 46) - \frac{1}{3}(6, 72) = (45, 22).
\end{aligned}$$

The individual Bézier cubic curves of curve k^* are determined by the following control polygons: $k_0^* : \mathbf{V}_0^* \mathbf{V}_1^* \mathbf{V}_2^* \mathbf{V}_3^*$, $k_1^* : \mathbf{V}_3^* \mathbf{V}_4^* \mathbf{V}_5^* \mathbf{V}_6^*$, $k_2^* : \mathbf{V}_6^* \mathbf{V}_7^* \mathbf{V}_8^* \mathbf{V}_0^*$, see Fig. 2.9.

Similarly, curve k consists of six Bézier cubic curves. Thus, 19 different control points \mathbf{V}_i , $i = 0, \dots, 18$ have to be determined. According to (2.26), we get the following control points: $\mathbf{V}_0 = (47, 46)$, $\mathbf{V}_1 = (48, 20)$, $\mathbf{V}_2 = (54, 18)$, $\mathbf{V}_3 = (58, 20)$, $\mathbf{V}_4 = (62, 22)$, $\mathbf{V}_5 = (64, 28)$, $\mathbf{V}_6 = (64, 35)$, $\mathbf{V}_7 = (64, 42)$, $\mathbf{V}_8 = (62, 50)$, $\mathbf{V}_9 = (55, 56)$, $\mathbf{V}_{10} = (48, 62)$, $\mathbf{V}_{11} = (36, 66)$, $\mathbf{V}_{12} = (26, 61)$, $\mathbf{V}_{13} = (16, 56)$, $\mathbf{V}_{14} = (8, 42)$, $\mathbf{V}_{15} = (9, 28)$, $\mathbf{V}_{16} = (10, 14)$, $\mathbf{V}_{17} = (20, 0)$, $\mathbf{V}_{18} = (47, 7)$.

The control polygons of individual Bézier cubic curves k_i , $i = 0, \dots, 5$ are obvious from Fig. 2.9.

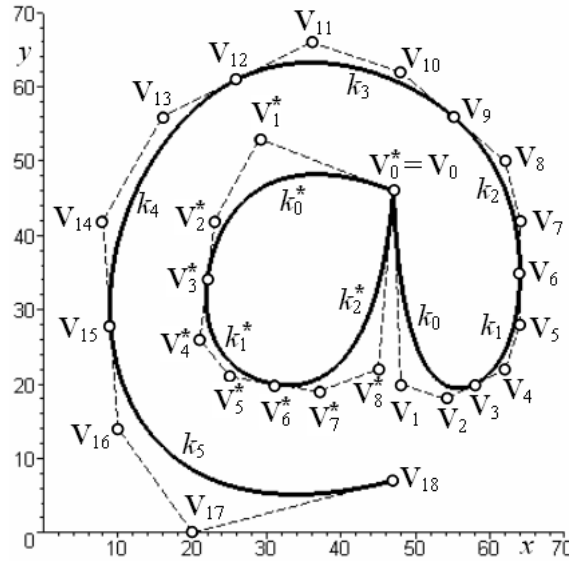


Figure 2.9: “@” sign created by Bézier cubic curves

□

2.2.2 Bernstein polynomials derivation

If the control polygon of Bézier curve has only one control point \mathbf{V}_0 , the zeroth degree Bernstein polynomial has to be $B_{0,0}(t) = 1$. It is the consequence of interpolation property – Bézier curve interpolates endpoints of control polygon.

In the case of two control points \mathbf{V}_0 and \mathbf{V}_1 , the vector equation of Bézier curve can be

written as an equation of a straight line segment given by two points

$$\mathbf{P}(t) = \mathbf{V}_0 + t(\mathbf{V}_1 - \mathbf{V}_0) = \mathbf{V}_0(1 - t) + t\mathbf{V}_1, \quad t \in [0, 1], \quad (2.27)$$

where linear coefficients $(1 - t)$ and t are first degree Bernstein polynomials. At any fixed value of parameter $\alpha \in [0, 1]$, we obtain a point

$$\mathbf{A}_0 = (1 - \alpha)\mathbf{V}_0 + \alpha\mathbf{V}_1 \quad (2.28)$$

on straight line segment $\mathbf{V}_0\mathbf{V}_1$ which is the $\mathbf{P}(\alpha)$ on Bézier curve of first degree given by control points \mathbf{V}_0 and \mathbf{V}_1 .

It is possible to separate the control polygon with three control points \mathbf{V}_0 , \mathbf{V}_1 and \mathbf{V}_2 into two individual control polygons – straight line segment $\mathbf{V}_0\mathbf{V}_1$ and straight line segment $\mathbf{V}_1\mathbf{V}_2$, each of which is considered a first degree Bézier curve. The point on straight line segment $\mathbf{V}_0\mathbf{V}_1$ at any fixed value of parameter $\alpha \in [0, 1]$ is given by (2.28). The point on straight line segment $\mathbf{V}_1\mathbf{V}_2$ can be expressed in the similar way

$$\mathbf{A}_1 = (1 - \alpha)\mathbf{V}_1 + \alpha\mathbf{V}_2, \quad (2.29)$$

where the fixed value α of parameter t is the same as in (2.28).

Next, points \mathbf{A}_0 and \mathbf{A}_1 can be considered the endpoints of a new straight line segment $\mathbf{A}_0\mathbf{A}_1$. Again, we use (2.28) to express point \mathbf{B}_0 lying on straight line segment $\mathbf{A}_0\mathbf{A}_1$

$$\mathbf{B}_0 = (1 - \alpha)\mathbf{A}_0 + \alpha\mathbf{A}_1. \quad (2.30)$$

After substituting (2.28) and (2.29) in (2.30), we obtain

$$\mathbf{B}_0 = (1 - \alpha)^2\mathbf{V}_0 + 2\alpha(1 - \alpha)\mathbf{V}_1 + \alpha^2\mathbf{V}_2. \quad (2.31)$$

The last equation represents point $\mathbf{P}(\alpha)$ on Bézier quadratic curve given by control points \mathbf{V}_0 , \mathbf{V}_1 and \mathbf{V}_2 at $t = \alpha$. If the value of parameter t in the whole domain $[0, 1]$ is considered, the vector equation of Bézier quadratic curve (2.18) is obtained. The coefficients of control points in (2.31) are changed in second degree Bernstein polynomials, see Tab. 2.1.

A control polygon with four control points \mathbf{V}_0 , \mathbf{V}_1 , \mathbf{V}_2 and \mathbf{V}_3 can be considered three control polygons – straight line segments $\mathbf{V}_0\mathbf{V}_1$, $\mathbf{V}_1\mathbf{V}_2$ and $\mathbf{V}_2\mathbf{V}_3$. Firstly, we express \mathbf{A}_0 according to (2.28), \mathbf{A}_1 according to (2.29) and

$$\mathbf{A}_2 = (1 - \alpha)\mathbf{V}_2 + \alpha\mathbf{V}_3 \quad (2.32)$$

at any fixed value of parameter $\alpha \in [0, 1]$. Next, we express points \mathbf{B}_0 on straight line segment $\mathbf{A}_0\mathbf{A}_1$ according to (2.30) and \mathbf{B}_1 on straight line segment $\mathbf{A}_1\mathbf{A}_2$

$$\mathbf{B}_1 = (1 - \alpha)\mathbf{A}_1 + \alpha\mathbf{A}_2. \quad (2.33)$$

Finally, point \mathbf{C}_0 on straight line segment $\mathbf{B}_0\mathbf{B}_1$ is given by

$$\mathbf{C}_0 = (1 - \alpha)\mathbf{B}_0 + \alpha\mathbf{B}_1. \quad (2.34)$$

After substituting and modification we get

$$\mathbf{C}_0 = (1 - \alpha)^3\mathbf{V}_0 + 3\alpha(1 - \alpha)^2\mathbf{V}_1 + 3\alpha^2(1 - \alpha)\mathbf{V}_2 + \alpha^3\mathbf{V}_3. \quad (2.35)$$

The last equation represents point $\mathbf{P}(\alpha)$ on Bézier cubic curve at $t = \alpha$. It is obvious that for $t \in [0, 1]$, the coefficients of control points in (2.35) are third degree Bernstein polynomials, see Tab. 2.1.

Bernstein polynomials of a higher degree can be derived similarly.

2.2.3 De Casteljau algorithm for Bézier curve

The *De Casteljau algorithm* is geometric interpretation of the above mentioned derivation of Bernstein polynomials. De Casteljau algorithm is widely used to construct point $\mathbf{P}(\alpha)$, $\alpha \in [0, 1]$ on Bézier curve $\mathbf{P}(t)$, $t \in [0, 1]$. The constructions of points on Bézier curves of first, second and third degree at $\alpha = \frac{1}{4}$, $\frac{1}{2}$ and $\frac{3}{4}$ are shown in Fig. 2.10, Fig. 2.11 and Fig. 2.12.

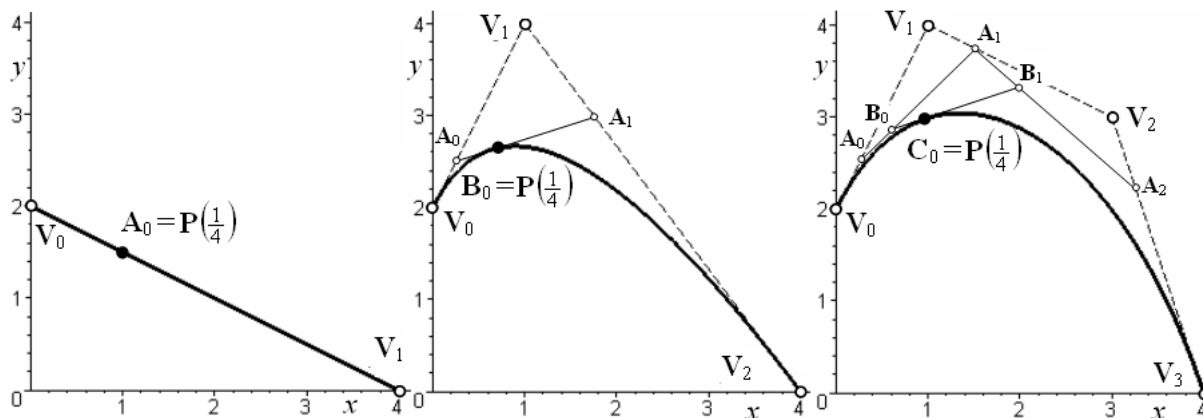


Figure 2.10: De Casteljau algorithm, $t = \frac{1}{4}$

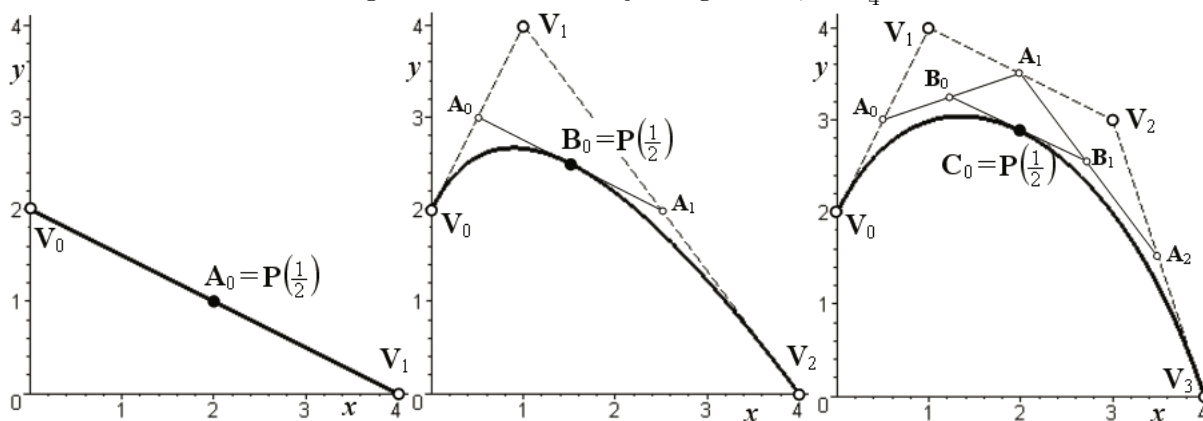


Figure 2.11: De Casteljau algorithm, $t = \frac{1}{2}$

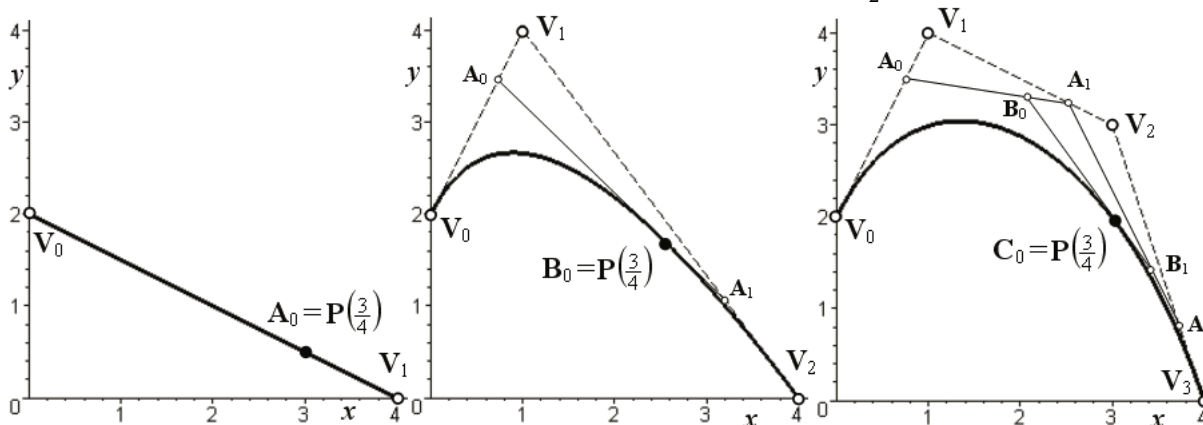


Figure 2.12: De Casteljau algorithm, $t = \frac{3}{4}$

De Casteljau algorithm includes the following steps (take a look at Fig. 2.10, Fig. 2.11 and Fig. 2.12):

1. Choose $\alpha \in [0, 1]$.
2. Divide all legs of control polygon in dividing ratio $\alpha : (1 - \alpha)$ to get points \mathbf{A}_i , $i = 0, 1, \dots, n - 1$.
3. Connect points \mathbf{A}_i , $i = 0, 1, \dots, n - 1$, by straight line segments. These straight line segments create a new control polygon with one leg less than the original control polygon.
4. Divide all legs of the new control polygon in dividing ratio $\alpha : (1 - \alpha)$ to get points \mathbf{B}_i , $i = 0, 1, \dots, n - 2$.
5. Repeat steps 3. and 4. until the last control polygon has only one leg.
6. The point constructed by means of the last dividing is point $\mathbf{P}(\alpha)$ on Bézier curve $\mathbf{P}(t)$. The last leg is a tangent line of Bézier curve at point $\mathbf{P}(\alpha)$. Tangent vector $\mathbf{P}'(\alpha)$ is equal to the n -multiple vector given by the last leg.

Obviously, it is necessary to accomplish n levels of dividing to construct a point on Bézier curve of n -th degree.

- **Example 2.7 De Casteljau algorithm, $n=6$.** Bézier curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_0 = (0, 2)$, $\mathbf{V}_1 = (7, 9)$, $\mathbf{V}_2 = (17, 9)$, $\mathbf{V}_3 = (20, 4)$, $\mathbf{V}_4 = (16, 0)$, $\mathbf{V}_5 = (10, 0)$, $\mathbf{V}_6 = (6, 4)$.

Construct point $\mathbf{P}(\frac{1}{2})$. Use de Casteljau algorithm.

Solution. Seven control points define sixth degree Bézier curve. De Casteljau algorithm has six levels of dividing, see Fig. 2.13 where the construction of point $\mathbf{P}(\frac{1}{2})$ on Bézier curve $\mathbf{P}(t)$, $t \in [0, 1]$, is construct.

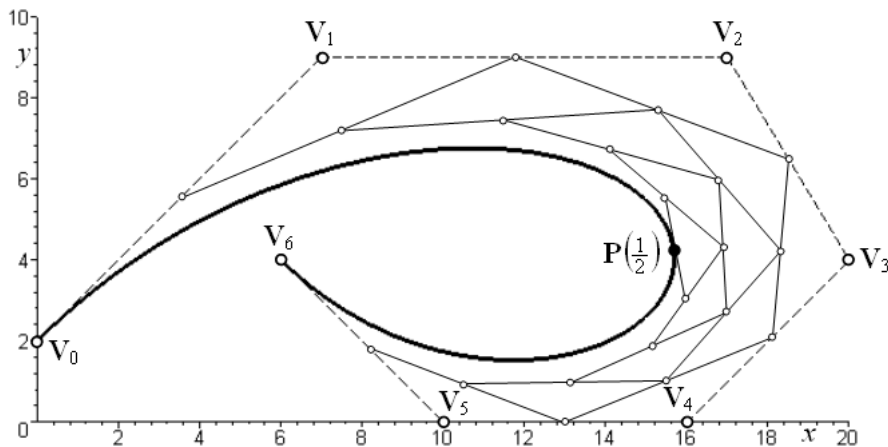


Figure 2.13: De Casteljau algorithm, $n = 6$, $t = \frac{1}{2}$

□

2.2.4 Continuity at common point of Bézier curves

Each Bézier curve is independent of any other Bézier curve. If we want to join two Bézier curves with any type of continuity, it is necessary to satisfy specific requirements on positions of certain control points of joined Bézier curves. Here, two Bézier cubic curves $\mathbf{P}(t)$, $t \in [0, 1]$ and $\mathbf{R}(s)$,

$s \in [0, 1]$ are considered. The first Bézier cubic curve $\mathbf{P}(t)$ is supposed to be defined by *known* control points \mathbf{V}_i , $i = 0, 1, 2, 3$

$$\mathbf{P}(t) = B_{0,3}(t)\mathbf{V}_0 + B_{1,3}(t)\mathbf{V}_1 + B_{2,3}(t)\mathbf{V}_2 + B_{3,3}(t)\mathbf{V}_3, \quad t \in [0, 1], \quad (2.36)$$

whereas the second Bézier cubic curve $\mathbf{R}(s)$ is supposed to be defined by *unknown* control points \mathbf{W}_i , $i = 0, 1, 2, 3$

$$\mathbf{R}(s) = B_{0,3}(s)\mathbf{W}_0 + B_{1,3}(s)\mathbf{W}_1 + B_{2,3}(s)\mathbf{W}_2 + B_{3,3}(s)\mathbf{W}_3, \quad s \in [0, 1]. \quad (2.37)$$

The task is to express the unknown control points \mathbf{W}_i , $i = 0, 1, 2, 3$ in terms of the known control points \mathbf{V}_i , $i = 0, 1, 2, 3$, in order to connect the initial point of Bézier cubic curve $\mathbf{R}(s)$ at the terminal point of Bézier cubic curve $\mathbf{P}(t)$ with C^0 , C^1 or C^2 continuity.

Vector equations of the first and the second derivatives of Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are

$$\begin{aligned} \mathbf{P}'(t) &= B'_{0,3}(t)\mathbf{V}_0 + B'_{1,3}(t)\mathbf{V}_1 + B'_{2,3}(t)\mathbf{V}_2 + B'_{3,3}(t)\mathbf{V}_3, \quad t \in [0, 1], \\ \mathbf{P}''(t) &= B''_{0,3}(t)\mathbf{V}_0 + B''_{1,3}(t)\mathbf{V}_1 + B''_{2,3}(t)\mathbf{V}_2 + B''_{3,3}(t)\mathbf{V}_3, \quad t \in [0, 1], \\ \mathbf{R}'(s) &= B'_{0,3}(s)\mathbf{W}_0 + B'_{1,3}(s)\mathbf{W}_1 + B'_{2,3}(s)\mathbf{W}_2 + B'_{3,3}(s)\mathbf{W}_3, \quad s \in [0, 1], \\ \mathbf{R}''(s) &= B''_{0,3}(s)\mathbf{W}_0 + B''_{1,3}(s)\mathbf{W}_1 + B''_{2,3}(s)\mathbf{W}_2 + B''_{3,3}(s)\mathbf{W}_3, \quad s \in [0, 1], \end{aligned} \quad (2.38)$$

where $B_{i,3}(u)$, $u = t$ or $u = s$, $i = 0, 1, 2, 3$ are third degree Bernstein polynomials, $B'_{i,3}(u)$ are their first derivatives and $B''_{i,3}(u)$ are their second derivatives. In Tab. 2.2, polynomials $B_{i,3}(u)$, $B'_{i,3}(u)$ and $B''_{i,3}(u)$ together with their function values at $u = 0$ and $u = 1$ are summarized.

Table 2.2: Third degree Bernstein polynomials and their derivatives

$B_{i,3}(u)$	$B_{i,3}(0)$	$B_{i,3}(1)$	$B'_{i,3}(u)$	$B'_{i,3}(0)$	$B'_{i,3}(1)$	$B''_{i,3}(u)$	$B''_{i,3}(0)$	$B''_{i,3}(1)$
$B_{0,3}(u) = (1 - u)^3$	1	0	$B'_{0,3}(u) = -3u^2 + 6u - 3$	-3	0	$B''_{0,3}(u) = -6u + 6$	6	0
$B_{1,3}(u) = 3u(1 - u)^2$	0	0	$B'_{1,3}(u) = 9u^2 - 12u + 3$	3	0	$B''_{1,3}(u) = 18u - 12$	-12	6
$B_{2,3}(u) = 3u^2(1 - u)$	0	0	$B'_{2,3}(u) = -9u^2 + 6u$	0	-3	$B''_{2,3}(u) = -18u + 6$	6	-12
$B_{3,3}(u) = u^3$	0	1	$B'_{3,3}(u) = 3u^2$	0	3	$B''_{3,3}(u) = 6u$	0	6

Continuity of the zeroth order. Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are joined with C^0 continuity, if the *function values of their vector functions* are equal, i.e. $\mathbf{R}(0) = \mathbf{P}(1)$. We obtain the condition of C^0 continuity when substituting function values of Bernstein polynomials for $t = 1$ from Tab. 2.2 into (2.36) and function values of Bernstein polynomials at $s = 0$ from Tab. 2.2 into (2.37)

$$\mathbf{W}_0 = \mathbf{V}_3. \quad (2.39)$$

Therefore, joining two Bézier curves with C^0 continuity requires that the curves share the last and the first control point. Position of remaining control points \mathbf{V}_i , $i = 0, 1, 2$ and \mathbf{W}_i , $i = 1, 2, 3$ is arbitrary.

Continuity of the first order. Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are joined with C^1 continuity if the condition of C^0 continuity is satisfied and the *function values of the first derivatives of their vector functions* are equal, i.e. $\mathbf{R}'(0) = \mathbf{P}'(1)$. We obtain the condition of C^1 continuity when substituting function values of first derivatives of Bernstein polynomials at $t = 1$ and $s = 0$ from Tab. 2.2 in (2.38)

$$-3\mathbf{W}_0 + 3\mathbf{W}_1 = -3\mathbf{V}_2 + 3\mathbf{V}_3. \quad (2.40)$$

After substituting $\mathbf{W}_0 = \mathbf{V}_3$ and modification in order to show a geometric interpretation of the required condition we get:

$$\mathbf{W}_1 - \mathbf{V}_3 = \mathbf{V}_3 - \mathbf{V}_2 \Rightarrow \mathbf{V}_3 = \frac{1}{2}(\mathbf{V}_2 + \mathbf{W}_1). \quad (2.41)$$

Vector $\overrightarrow{\mathbf{V}_3\mathbf{W}_1}$ is equal to vector $\overrightarrow{\mathbf{V}_2\mathbf{V}_3}$. Therefore, control point \mathbf{V}_3 is the centre of straight line segment $\mathbf{V}_2\mathbf{W}_1$; the position of reminding control points \mathbf{V}_0 , \mathbf{V}_1 , \mathbf{W}_2 and \mathbf{W}_3 is arbitrary. The formula

$$\mathbf{W}_1 = 2\mathbf{V}_3 - \mathbf{V}_2$$

is used when calculating the unknown coordinates of control point \mathbf{W}_1 .

Continuity of the second degree. Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are joined with C^2 continuity if the conditions of C^0 and C^1 continuity are satisfied and the *function values of the second derivatives of their vector functions* are equal, i.e. $\mathbf{R}''(0) = \mathbf{P}''(1)$. We obtain the condition of C^2 continuity when substituting function values of second derivatives of Bernstein polynomials at $t = 1$ and $s = 0$ from Tab. 2.2 in (2.38)

$$6\mathbf{W}_0 - 12\mathbf{W}_1 + 6\mathbf{W}_2 = 6\mathbf{V}_1 - 12\mathbf{V}_2 + 6\mathbf{V}_3. \quad (2.42)$$

After substituting $\mathbf{W}_0 = \mathbf{V}_3$ and $\mathbf{W}_1 = 2\mathbf{V}_3 - \mathbf{V}_2$ and modification in order to show a geometric interpretation of the required condition we get

$$\mathbf{W}_2 - \mathbf{V}_1 = 4(\mathbf{V}_3 - \mathbf{V}_2). \quad (2.43)$$

Vector $\overrightarrow{\mathbf{V}_1\mathbf{W}_2}$ is equal to 4-multiple of vector $\overrightarrow{\mathbf{V}_2\mathbf{V}_3}$, position of control points \mathbf{V}_0 and \mathbf{W}_3 is arbitrary. The formula

$$\mathbf{W}_2 = 4(\mathbf{V}_3 - \mathbf{V}_2) + \mathbf{V}_1$$

is used when calculating the unknown coordinates of control point \mathbf{W}_2 .

Note that the above derived conditions of C^0 and C^1 continuity of two Bézier cubic curves follow directly from properties of Bézier curves, see page 49. These conditions are valid for C^0 and C^1 continuity of two Bézier curves of any degree.

If the first control point \mathbf{V}_0 of Bézier cubic curve $\mathbf{P}(t)$ is the common point of two Bézier curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ joined with C^0 , C^1 or C^2 continuity, the same geometric rules are valid. However, the reverse orientation of control points is necessary to consider.

■ **Example 2.8 – Joining of Bézier quadratic curves.** Bézier quadratic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_0 = (-1, 1)$, $\mathbf{V}_1 = (0, 0)$, $\mathbf{V}_2 = (1, 1)$.

Draw control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2$. Construct the necessary number of control polygons of Bézier quadratic curves to create a closed curve k consisting of C^1 continuously joined Bézier quadratic curves. Sketch all Bézier curves. Verify the constructed positions of unknown control points and C^1 continuity of curve k at all common points by calculation.

Solution

Construction: Firstly, denote unknown control points of Bézier quadratic curve $\mathbf{R}(s)$, $s \in [0, 1]$ with \mathbf{W}_0 , \mathbf{W}_1 and \mathbf{W}_2 . To preserve the orientation of parameter along curve k , the common point of curves $\mathbf{R}(s)$ and $\mathbf{P}(t)$ is the initial point $\mathbf{R}(0)$ and the terminal point $\mathbf{P}(1)$. Therefore, $\mathbf{W}_0 = \mathbf{V}_2 = (1, 1)$. Control point $\mathbf{W}_1 = (2, 2)$ is radially symmetrical with respect to control point \mathbf{V}_1 , the centre of symmetry is \mathbf{V}_2 , see Fig. 2.14.

Secondly, denote unknown control points of Bézier quadratic curve $\mathbf{S}(u)$, $u \in [0, 1]$ with \mathbf{U}_0 , \mathbf{U}_1 and \mathbf{U}_2 . To preserve the orientation of parameter along curve k , the common point of curves $\mathbf{S}(u)$ and $\mathbf{P}(t)$ is the initial point $\mathbf{S}(0)$ and the terminal point $\mathbf{P}(1)$. Therefore, $\mathbf{U}_2 = \mathbf{V}_0 = (-1, -1)$. Control point $\mathbf{U}_1 = (-2, 2)$ is radially symmetrical with respect to control point \mathbf{V}_1 , the centre of symmetry is \mathbf{V}_0 .

The centre of straight line segment $\mathbf{W}_1\mathbf{U}_1$ is the common control point $\mathbf{W}_2 = \mathbf{U}_0 = (0, 2)$ of curves $\mathbf{S}(u)$ and $\mathbf{R}(s)$. Consequently, the curve k is closed. It is obvious that at least three Bézier quadratic curves are necessary to obtain closed C^1 continuous piecewise curve k .

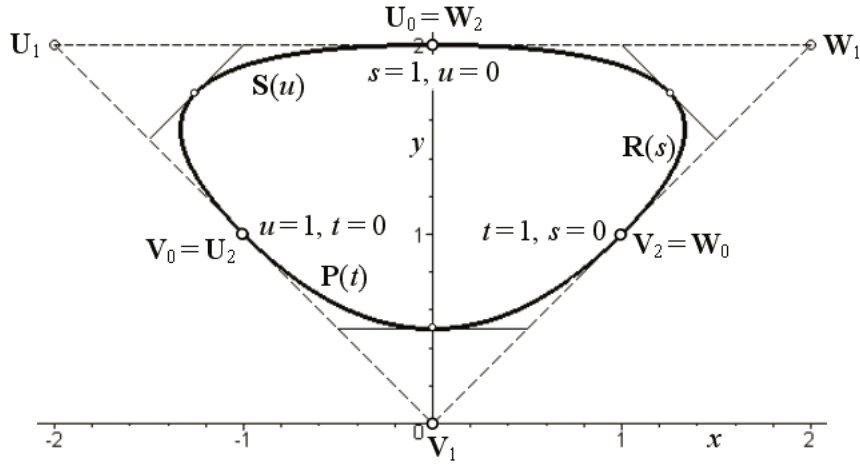


Figure 2.14: C^1 continuously joined Bézier quadratic curves

Calculation: From condition of C^0 continuity of curve k , it follows that $\mathbf{U}_2 = (-1, 1)$ and $\mathbf{W}_0 = (1, 1)$. From condition of C^1 continuity of curve k , we get

$$\begin{aligned}\mathbf{U}_1 &= 2\mathbf{V}_0 - \mathbf{V}_1 = 2(-1, 1) - (0, 0) = (-2, 2), \\ \mathbf{W}_1 &= 2\mathbf{V}_2 - \mathbf{V}_1 = 2(1, 1) - (0, 0) = (2, 2), \\ \mathbf{U}_0 = \mathbf{W}_2 &= \frac{1}{2}(\mathbf{U}_1 + \mathbf{W}_1) = \frac{1}{2}[(-2, 2) + (2, 2)] = (0, 2).\end{aligned}$$

Verification of C^1 continuity of curve k :

$$\begin{array}{lll}\mathbf{P}(t) = (2t - 1, 2t^2 - 2t + 1), & \mathbf{P}(0) = (-1, 1), & \mathbf{P}(1) = (1, 1), \\ \mathbf{R}(s) = (-3s^2 + 2s + 1, -s^2 + 2s + 1), & \mathbf{R}(0) = (1, 1), & \mathbf{R}(1) = (0, 2), \\ \mathbf{S}(u) = (3u^2 - 4u, -u^2 + 2), & \mathbf{S}(0) = (0, 2), & \mathbf{S}(1) = (-1, 1), \\ \mathbf{P}'(t) = (2, 4t - 2), & \mathbf{P}'(0) = (2, -2), & \mathbf{P}'(1) = (2, 2), \\ \mathbf{R}'(s) = (-6s + 2, -2s + 2), & \mathbf{R}'(0) = (2, 2), & \mathbf{R}'(1) = (-4, 0), \\ \mathbf{S}'(u) = (6u - 4, -2u), & \mathbf{S}'(0) = (-4, 0), & \mathbf{S}'(1) = (2, -2).\end{array}$$

□

- **Example 2.9 – Continuity of two Bézier cubic curves.** Bézier cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_0 = (1, 2)$, $\mathbf{V}_1 = (0, 1)$, $\mathbf{V}_2 = (0, 0)$ and $\mathbf{V}_3 = (1, 0)$. Bézier cubic curve $\mathbf{R}(s)$, $s \in [0, 1]$ is given by control points $\mathbf{W}_0 = (1, 0)$, $\mathbf{W}_1 = (2, 0)$, $\mathbf{W}_2 = (2, 1)$ and $\mathbf{W}_3 = (2, 2)$.

What is the continuity of these two Bézier cubic curves?

Solution. Vector equations of Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ and their first and second derivatives are as follows

$$\begin{aligned}\mathbf{P}(t) &= (3t^2 - 3t + 1, t^3 - 3t + 2), t \in [0, 1], \\ \mathbf{R}(s) &= (s^3 - 3s^2 + 3s + 1, -s^3 + 3s^2), s \in [0, 1],\end{aligned}\tag{2.44}$$

$$\begin{aligned}\mathbf{P}'(t) &= (6t - 3, 3t^2 - 3), t \in [0, 1], \\ \mathbf{R}'(s) &= (3s^2 - 6s + 3, -3s^2 + 6s), s \in [0, 1],\end{aligned}\tag{2.45}$$

$$\begin{aligned}\mathbf{P}''(t) &= (6, 6t), t \in [0, 1], \\ \mathbf{R}''(s) &= (6s - 6, -6s + 6), s \in [0, 1].\end{aligned}\tag{2.46}$$

Since $\mathbf{V}_0 \neq \mathbf{W}_3$, the curves are not joined at this point.

Substituting $t = 1$ and $s = 0$ (2.44), we get

$$\mathbf{P}(1) = (1, 0), \mathbf{R}(0) = (1, 0).$$

Therefore, the curves are joined at least with C^0 continuity at point $\mathbf{V}_3 = \mathbf{W}_0$.

Substituting $t = 1$ and $s = 0$ in (2.45), we get

$$\mathbf{P}'(1) = (3, 0), \mathbf{R}'(0) = (3, 0).$$

It follows that curves are joined with C^1 continuity at common point.

Substituting $t = 1$ and $s = 0$ in (2.46), we get

$$\mathbf{P}''(1) = (6, 6), \mathbf{R}''(0) = (-6, 6).$$

Obviously, the curves are not joined with C^2 continuity at common point. However, these curves can be still joined with G^2 continuity according to (1.30). The different vectors of the second derivatives this type of continuity do not exclude. Thus, we express the first curvature according to (1.22) of both curves at common point

$$\begin{aligned}{}^1k_{\mathbf{P}}(1) &= \frac{|\mathbf{P}'(1) \times \mathbf{P}''(1)|}{|\mathbf{P}'(1)|^3} = \frac{|(3, 0, 0) \times (6, 6, 0)|}{|(3, 0, 0)|^3} = \frac{2}{3}, \\ {}^1k_{\mathbf{R}}(0) &= \frac{|\mathbf{R}'(0) \times \mathbf{R}''(0)|}{|\mathbf{R}'(0)|^3} = \frac{|(3, 0, 0) \times (-6, 6, 0)|}{|(3, 0, 0)|^3} = \frac{2}{3}.\end{aligned}$$

Since ${}^1k_{\mathbf{P}}(1) = {}^1k_{\mathbf{R}}(0)$, the curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are joined with G^2 continuity at common point. The osculation circles of both curves are identical. The radius of the first curvature of both curves at common point is given by

$$r = \frac{1}{{}^1k_{\mathbf{P}}(1)} = \frac{1}{{}^1k_{\mathbf{R}}(0)} = \frac{3}{2}$$

and centre of the first curvature located on the normal line is then

$$\mathbf{S} = \left(1, \frac{3}{2}\right),$$

see Fig. 2.15 (the vectors of the second derivatives $\mathbf{R}''(0)$ and $\mathbf{P}''(1)$ are drawn in the scale 1:2).

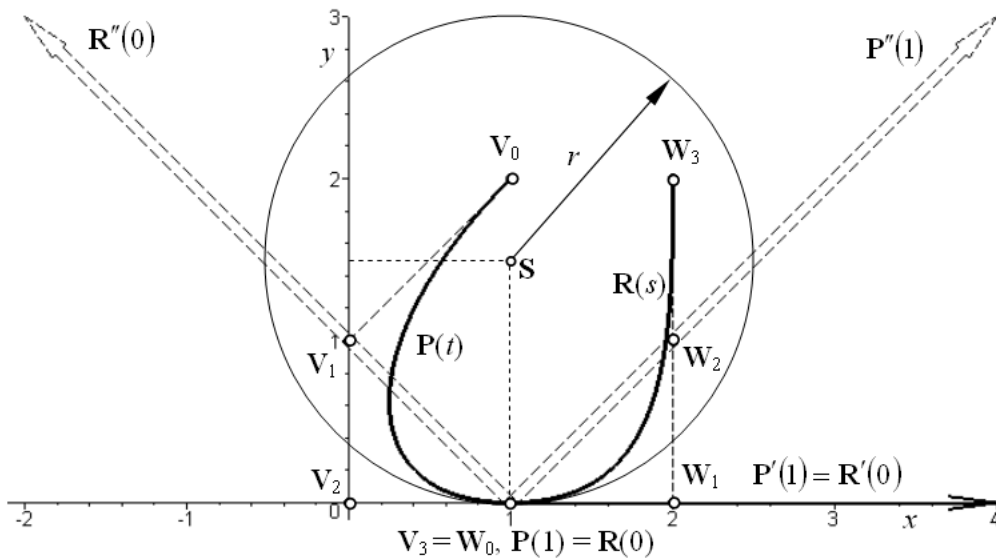


Figure 2.15: Bézier cubic curves preserving the conditions of C^0 , C^1 and G^2 continuity at common point □

■ **Example 2.10 – Joining of Bézier cubic curves.** Bézier cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{V}_3 = (4, 1)$, $\mathbf{V}_4 = (5, 0)$, $\mathbf{V}_5 = (7, 0)$ and $\mathbf{V}_6 = (8, 2)$. Additionally, control point $\mathbf{V}_0 = (0, 4)$ is given.

Draw control points \mathbf{V}_0 , \mathbf{V}_3 , \mathbf{V}_4 , \mathbf{V}_5 and \mathbf{V}_6 . Determine by construction and verify by calculation the coordinates of control points \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{V}_7 and \mathbf{V}_8 , so that Bézier cubic curves $\mathbf{R}(s)$, $s \in [0, 1]$ given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$ and $\mathbf{S}(u)$, $u \in [0, 1]$ given by control polygon $\mathbf{V}_6\mathbf{V}_7\mathbf{V}_8\mathbf{V}_0$ are C^2 continuously joined with curve $\mathbf{P}(t)$.

Using de Casteljau algorithm, construct all control polygons, construct points $\mathbf{R}(\frac{1}{2})$, $\mathbf{P}(\frac{1}{2})$ and $\mathbf{S}(\frac{1}{2})$ and sketch Bézier cubic curves $\mathbf{R}(s)$, $\mathbf{P}(t)$ and $\mathbf{S}(u)$.

Verify the required continuity at common points of Bézier curves.

Solution

Construction: Common point \mathbf{V}_3 of curves $\mathbf{R}(s)$ and $\mathbf{P}(t)$ guarantees C^0 continuity at this point. To join these curves with C^1 continuity requires that control point \mathbf{V}_2 is radially symmetrical with respect to point \mathbf{V}_4 , the centre of symmetry is \mathbf{V}_3 . The coordinates $\mathbf{V}_2 = (3, 2)$ can be read from the graph, see Fig. 2.16. C^2 continuity requires that control point \mathbf{V}_1 lies on parallel line passing through \mathbf{V}_5 at the oriented distance equal to 4-multiple $\overrightarrow{\mathbf{V}_4 - \mathbf{V}_3}$ from point \mathbf{V}_4 . The distance orientation is identical with the orientation of vector $\overrightarrow{\mathbf{V}_4\mathbf{V}_3}$. Therefore $\mathbf{V}_1 = (3, 4)$.

Common point \mathbf{V}_6 of curves $\mathbf{P}(t)$ and $\mathbf{S}(u)$ guarantees C^0 continuity at this point. From condition of C^1 continuity, it follows that control point \mathbf{V}_7 is radially symmetrical with respect to point \mathbf{V}_5 , the centre of symmetry is \mathbf{V}_6 . Therefore $\mathbf{V}_7 = (9, 4)$. From condition of C^2 continuity, it follows that control point \mathbf{V}_8 lies on parallel line passing through \mathbf{V}_4 at the oriented distance equal to 4-multiple of $\overrightarrow{\mathbf{V}_6 - \mathbf{V}_5}$ from point \mathbf{V}_4 . The distance orientation is identical with the orientation of vector $\overrightarrow{\mathbf{V}_6\mathbf{V}_5}$. Therefore $\mathbf{V}_8 = (9, 8)$.

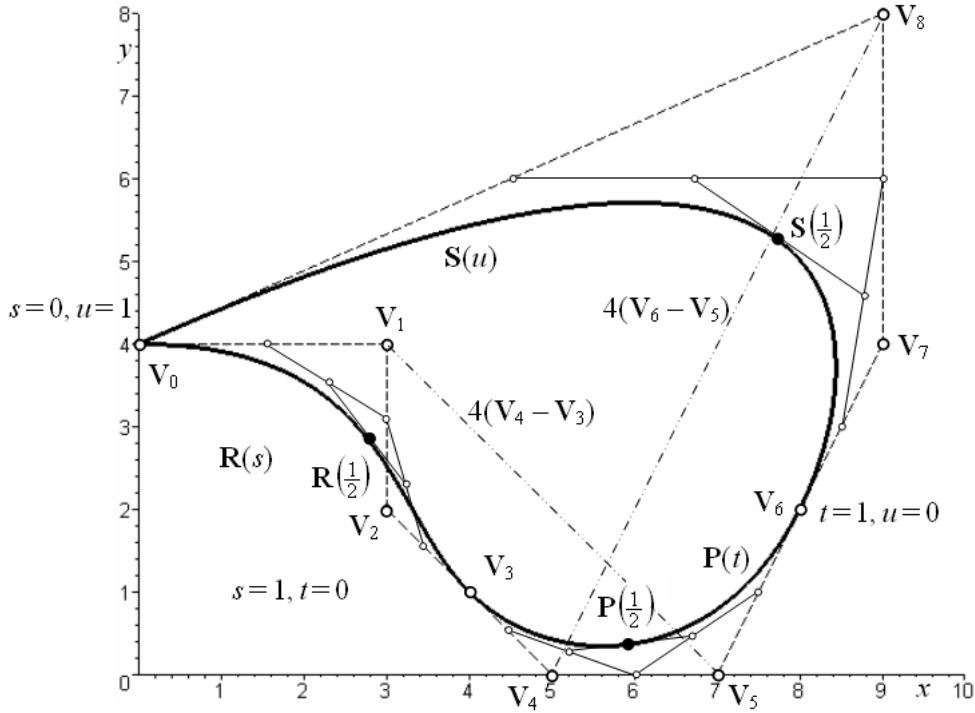


Figure 2.16: Joining of Bézier cubic curves with C^2 continuity

Calculation: Considering conditions of C^1 continuity at common points \mathbf{V}_3 and \mathbf{V}_6 , we get

$$\begin{aligned}\mathbf{V}_2 &= 2\mathbf{V}_3 - \mathbf{V}_4 = 2(4, 1) - (5, 0) = (3, 2), \\ \mathbf{V}_1 &= \mathbf{V}_5 - 4(\mathbf{V}_4 - \mathbf{V}_3) = (7, 0) - 4[(5, 0) - (4, 1)] = (3, 4).\end{aligned}$$

Considering condition of C^2 continuity at common points \mathbf{V}_3 and \mathbf{V}_6 , we get

$$\begin{aligned}\mathbf{V}_7 &= 2\mathbf{V}_6 - \mathbf{V}_5 = 2(8, 2) - (7, 0) = (9, 4), \\ \mathbf{V}_8 &= \mathbf{V}_4 + 4(\mathbf{V}_6 - \mathbf{V}_5) = (5, 0) + 4[(8, 2) - (7, 0)] = (9, 8).\end{aligned}$$

Verification of C^0 , C^1 and C^2 continuity at common points \mathbf{V}_3 and \mathbf{V}_6 :

$$\begin{aligned}\mathbf{R}(s) &= (4s^3 - 9s^2 + 9s, 3s^3 - 6s^2 + 4), & \mathbf{R}(0) &= (0, 4), & \mathbf{R}(1) &= (4, 1), \\ \mathbf{P}(t) &= (-2t^3 + 3t^2 + 3t + 4, t^3 + 3t^2 - 3t + 1), & \mathbf{P}(0) &= (4, 1), & \mathbf{P}(1) &= (8, 2), \\ \mathbf{S}(u) &= (-8u^3 - 3u^2 + 3u + 8, -10u^3 + 6u^2 + 6u + 2), & \mathbf{S}(0) &= (8, 2), & \mathbf{S}(1) &= (0, 4), \\ \\ \mathbf{R}'(s) &= (12s^2 - 18s + 9, 9s^2 - 12s), & \mathbf{R}'(0) &= (9, 0), & \mathbf{R}'(1) &= (3, -3), \\ \mathbf{P}'(t) &= (-6t^2 + 6t + 3, 3t^2 + 6t - 3), & \mathbf{P}'(0) &= (3, -3), & \mathbf{P}'(1) &= (3, 6), \\ \mathbf{S}'(u) &= (-24u^2 - 6u + 3, -30u^2 + 12u + 6), & \mathbf{S}'(0) &= (3, 6), & \mathbf{S}'(1) &= (-27, -12), \\ \\ \mathbf{R}''(s) &= (24s - 18, 18s - 12), & \mathbf{R}''(0) &= (-18, -12), & \mathbf{R}''(1) &= (6, 6), \\ \mathbf{P}''(t) &= (-12t + 6, 6t + 6), & \mathbf{P}''(0) &= (6, 6), & \mathbf{P}''(1) &= (-6, 12), \\ \mathbf{S}''(u) &= (-48u - 6, -60u + 12), & \mathbf{S}''(0) &= (-6, 12), & \mathbf{S}''(1) &= (-54, -48).\end{aligned}$$

It is obvious that Bézier cubic curves $\mathbf{R}(s)$ and $\mathbf{P}(t)$ are C^2 continuously joined at point \mathbf{V}_3 , curves $\mathbf{P}(t)$ and $\mathbf{S}(u)$ are C^2 continuously joined at point \mathbf{V}_6 and curves $\mathbf{S}(u)$ and $\mathbf{R}(s)$ are only C^0 continuously joined at point \mathbf{V}_0 . \square

- **Exercise 2.4** Bézier quadratic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2$.

Find the parametric expression and vector equation of Bézier curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0$, $t = \frac{1}{2}$ and $t = 1$, calculate the coordinates of points on Bézier curve and tangent vectors at these points. Draw control points, construct control polygon, construct tangent vector at endpoints of control polygon. Construct point $\mathbf{P}(\frac{1}{2})$ on Bézier curve and tangent vector $\mathbf{P}'(\frac{1}{2})$ by de Casteljau algorithm. Sketch Bézier curve. Consider the following sets of control points:

- a) $\mathbf{V}_0 = (0, 0)$, $\mathbf{V}_1 = (1, 1)$, $\mathbf{V}_2 = (0, 1)$, b) $\mathbf{V}_0 = (0, 0)$, $\mathbf{V}_1 = (1, -1)$, $\mathbf{V}_2 = (0, -1)$,
 c) $\mathbf{V}_0 = (0, 0)$, $\mathbf{V}_1 = (-1, 1)$, $\mathbf{V}_2 = (0, 1)$, d) $\mathbf{V}_0 = (0, 0)$, $\mathbf{V}_1 = (-1, -1)$, $\mathbf{V}_2 = (0, -1)$.

- **Exercise 2.5** Bézier cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control point $\mathbf{V}_0 = (0, 0)$, $\mathbf{V}_1 = (3, 2)$, $\mathbf{V}_2 = (1, 3)$ and $\mathbf{V}_3 = (0, 2)$. Bézier cubic curve $\mathbf{R}(s)$, $s \in [0, 1]$ is given by control points \mathbf{V}_3 , $\mathbf{V}_4 = (-1, 3)$, $\mathbf{V}_5 = (-3, 2)$ and \mathbf{V}_0 .

Find the parametric expression and vector equation of these Bézier curves and their tangent vectors. Calculate the coordinates of points on Bézier curves at $t = 0$, $t = \frac{1}{2}$ and $t = 1$ and $s = 0$, $s = \frac{1}{2}$ and $s = 1$ and coordinates of tangent vectors at these points. Draw control points, construct control polygons and tangent vectors at endpoints of Bézier curves. Verify the calculated coordinates of points $\mathbf{P}(\frac{1}{2})$ and $\mathbf{R}(\frac{1}{2})$ as well as the calculated coordinates of tangent vectors $\mathbf{P}'(\frac{1}{2})$ and $\mathbf{R}'(\frac{1}{2})$ by de Casteljau algorithm. Sketch both Bézier curves. What is the continuity of Bézier cubic curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$?

- **Exercise 2.6** Suppose, the individual segments of curves k from Exercise 2.2 are Bézier cubic curves. Determine their control points, draw these control points, construct control polygons. Using de Casteljau algorithm, construct points on individual curves at $t = \frac{1}{2}$. Sketch all Bézier cubic curves. What is the continuity at all common points?

- **Exercise 2.7** Find vector equations of individual Bézier cubic curves creating “@” sign in Example 2.6. Verify a continuity of their joining. Compare vector equations of Bézier cubic curves and vector equations of Ferguson cubic curves from Example 2.2.

2.2.5 Bézier curve in Rhinoceros

Bézier curve of degree n – Command: *Control Point Curve* → Start of curve: type degree n in command prompt; enter initial point of Bézier curve → Next point: enter $n+1$ control points → press Enter. The Bézier curve of degree n is drawn.

Remarks:

1. If $m < (n + 1)$ control points is entered, Bézier curve of degree $(m - 1)$ is drawn. If $m > (n + 1)$ control points is entered, a uniform clamped B-spline curve (see Chapter 2.5) of degree n with $(m - n)$ segments is drawn.
2. If the first and the last control point of Bézier curve are equal, it is necessary choose *Sharp=Yes* in command prompt before entering the last control point.

Ferguson cubic curve – Command: *Control Point Curve* → Start of curve: type degree 3 in command prompt; enter the initial definition point of Ferguson cubic curve → Next point: enter point at the first third of tangent vector at the initial point of Ferguson cubic curve → Next point: enter point at the first third of opposite tangent vector at the terminal point of Ferguson cubic curve → Next point: enter the terminal definition point of Ferguson cubic curve. Bézier cubic curve identical with Ferguson cubic curve is drawn.

Control points and control polygon – It is possible to display/hide control points and control polygon of a drawn Bézier curve using command *Control Points On/Points Off* → Select objects for control point display: click on the curve. Control points are drawn as temporarily visible points, control polygon is displayed as a temporarily visible dotted line.

Additionally, it is possible to create the control polygon of a drawn Bézier curve using command *Extract Mesh from NURBS Control Polygon* → Select curves or surface to extract control polygon: click on the curve → press Enter. The polyline representing a control polygon is drawn.

Shape of Bézier curve modification – The shape of Bézier curve is possible to modify in the following recommended ways:

1. Move displayed control points. It is possible to delete control points which causes the change of both the shape and degree of Bézier curve. However, it is impossible to add other control points.
2. Move displayed edit points. It is possible to display/hide edit points (points lying on the curve) using command *Edit Points On/Points Off* → Select curves for edit point display: click on the curve → press Enter. Edit points on the curve are displayed. The number and initial positions of edit points are generated by software. After position of edit points changing, the control polygon of the curve is changed automatically.

Attention! Do not use commands *Insert Knot* and *Remove Knot* to modify the shape of Bézier curve. These commands cause segmentation of a modified curve (a piecewise curve arises) and nonuniform parametrization of a modified curve (domains of individual segments are different, not $[0, 1]$, see [?], [2]). Do not use the command *Edit Control Point Weight*, either. This command causes rational parametrization of the curve (not polynomial). After this modification, the original curve is not Bézier curve anymore, but, in general, NURBS curve (see [?], [2]).

De Casteljau algorithm – It is impossible to realize De Casteljau algorithm directly. However, the following procedure can be used to simplify this task.

1. Express the value of parameter α , at which a point on Bézier curve has to be constructed, as a fraction $\alpha = \frac{a}{b}$. For example, $\alpha = 0.3 \Rightarrow \alpha = \frac{3}{10}$.
2. Create control polygon (see above) and explode it on individual straight line segments using the command *Explode*.
3. Draw dividing points on all legs of the control polygon using command *Divide curve by number of segments* → Number of segments: type b in command prompt → press Enter. $(b + 1)$ equidistantly placed points on each leg of control polygon are drawn as individual entities.
4. Activate *Point object snap* and draw control polygons of subsequent dividing level using command *Polyline* → Start of polyline: click at a -th dividing point on the first leg of control polygon from previous dividing level → Next point of polyline: click at

a -th dividing point on the next leg $\rightarrow \dots \rightarrow$ click at a -th dividing point on the last leg \rightarrow press Enter.

5. All subsequent dividing levels are realized similarly.

Joining of Bézier cubic curves – According to the conditions of continuity derived in section 2.2.4, we assume that Bézier cubic curve $\mathbf{P}(t)$ given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$ is already drawn, its control points are displayed and *Point object snap* and *Endpoint object snap* are activated. The construction of control points $\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2$ and \mathbf{W}_3 of Bézier cubic curve $\mathbf{R}(s)$ which has to be joined at \mathbf{V}_3 to Bézier cubic curve $\mathbf{P}(t)$ can be accomplished in the following way:

- C^0 continuity: according to condition (2.39), control point \mathbf{W}_0 has to be identified with control point \mathbf{V}_3 . Therefore, *Endpoints object snap* is used when placing the position of control point \mathbf{W}_0 . The position of control points $\mathbf{W}_1, \mathbf{W}_2$ and \mathbf{W}_3 is arbitrary.
- C^1 continuity: according to condition (2.41), point \mathbf{V}_3 lies at the centre of straight line segment $\mathbf{V}_2\mathbf{W}_1$. This requirement can be realized by using command *Line: form Midpoint* \rightarrow Middle of line: click at point $\mathbf{V}_3 \rightarrow$ End of line: click at point \mathbf{V}_2 . A straight line with the centre at point \mathbf{V}_3 and endpoints at \mathbf{V}_2 and \mathbf{W}_1 is drawn. The position of control points \mathbf{W}_2 and \mathbf{W}_3 is arbitrary.
- C^2 continuity: according to condition (2.43), vector $\overrightarrow{\mathbf{V}_1\mathbf{W}_2}$ is 4-multiple of vector $\overrightarrow{\mathbf{V}_2\mathbf{V}_3}$. This requirement can be realized by double copying straight line segment $\mathbf{V}_2\mathbf{W}_1$ (which represents 2-multiple of vector $\overrightarrow{\mathbf{V}_2\mathbf{V}_3}$) drawn in the previous step. Command: *Copy* \rightarrow Select object to copy: click on straight line segment $\mathbf{V}_2\mathbf{W}_1 \rightarrow$ press Enter \rightarrow Point to copy from: click at point $\mathbf{V}_2 \rightarrow$ Point to copy to: click at point $\mathbf{V}_1 \rightarrow$ Point to copy to: click at the free endpoint of previously copied straight line segment \rightarrow press Enter. A position of free endpoint of the second time copied straight line segment determines the position of point \mathbf{W}_2 . The position of control points \mathbf{W}_3 is arbitrary.

Remark: It is possible to proceed similarly when assuring C^0, C^1 continuity of Bézier quadratic curves and C^0, C^1 and C^2 continuity of Bézier curves of degrees higher than three.

- **Exercise 2.8** In Rhinoceros, draw Ferguson cubic curves from Example 2.1 and from Exercise 2.1. At the given parameter values, draw the points on the curves. Find Cartesian coordinates of these points and compare them with the calculated values.
- **Exercise 2.9** In Rhinoceros, draw Ferguson cubic curves from Exercise 2.2. Display the curvature graphs of all Ferguson cubic curves. What is the continuity of these curves?
- **Exercise 2.10** In Rhinoceros, draw Bézier curves from Examples 2.3 and 2.4. Using de Casteljau algorithm, construct points on these curves at the values of parameter given in Example 2.3. Find Cartesian coordinates of these points and compare them with calculation.
- **Exercise 2.11** In Rhinoceros, draw “@” sign from Example 2.6. Display the curvature graphs of all Bézier cubic curves. What is the continuity at common points of curves $k_i, i = 0, \dots, 5$ and $k_i^*, i = 0, 1, 2$?
- **Exercise 2.12** In Rhinoceros, draw Bézier curve of sixth degree from example 2.7. Using de Casteljau algorithm, construct point $\mathbf{P}(\frac{1}{2})$.

- **Exercise 2.13** In Rhinoceros, draw Bézier curves from Examples 2.8 and 2.10. Using de Casteljau algorithm, construct point at $t = \frac{1}{2}$ and display curvature graph. What is the continuity of these curves?
- **Exercise 2.14** In Rhinoceros, draw Bézier curves from Example 2.9. Draw the osculation circle at the common point and find Cartesian coordinates of its centre. Determine the radius of the first curvature at common point. Compare the centre coordinates and radius of osculation circle with the calculated values.

2.3 Coons cubic curve

Coons cubic curve approximates four control points.

- **Definition 2.3 – Coons cubic curve.** Let \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 are control points in space. Then, the vector equation of *Coons cubic curve* is

$$\mathbf{P}(t) = C_0(t)\mathbf{P}_0 + C_1(t)\mathbf{P}_1 + C_2(t)\mathbf{P}_2 + C_3(t)\mathbf{P}_3, \quad t \in [0, 1], \quad (2.47)$$

where the basis functions

$$\begin{aligned} C_0(t) &= \frac{1}{6}(1-t)^3, \\ C_1(t) &= \frac{1}{6}(3t^3 - 6t^2 + 4), \\ C_2(t) &= \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1), \\ C_3(t) &= \frac{1}{6}t^3, \end{aligned}$$

are *Coons polynomials*. □

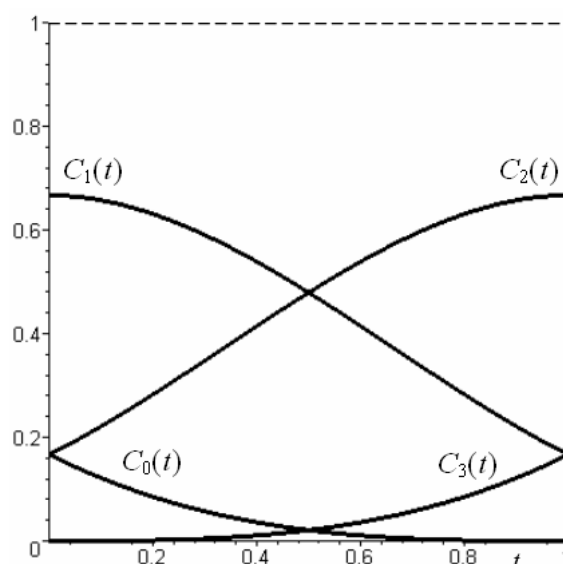


Figure 2.17: Basis functions (Coons polynomials) of Coons cubic curve

Graphs of Coons polynomials are depicted in Fig. 2.17. An example of Coons cubic curve is drawn in Fig. 2.19.

■ **Example 2.11 – Coons cubic curve.** Coons cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ is given by control points $\mathbf{P}_0 = (-6, 0)$, $\mathbf{P}_1 = (6, 0)$, $\mathbf{P}_2 = (6, 6)$ and $\mathbf{P}_3 = (0, 6)$.

Find the parametric expression and vector equation of Coons cubic curve $\mathbf{P}(t)$ and its tangent vector $\mathbf{P}'(t)$. At $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and 1, calculate the coordinates of points on Coons cubic curve and tangent vectors at these points. Draw control points, construct control polygon, draw calculated points and construct tangent vectors at these points. Sketch Coons cubic curve.

Solution.

The parametric expression of Coons cubic curve and its tangent vector is

$$\begin{aligned}x(t) &= -6C_0(t) + 6C_1(t) + 6C_2(t) = t^3 - 6t^2 + 6t + 4, \\y(t) &= 6C_2(t) + 6C_3(t) = -2t^3 + 3t^2 + 3t + 1, \quad t \in [0, 1], \\x'(t) &= 3t^2 - 12t + 6, \\y'(t) &= -6t^2 + 6t + 3, \quad t \in [0, 1].\end{aligned}$$

Vector equation of Coons cubic curve and its tangent vector is

$$\begin{aligned}\mathbf{P}(t) &= (t^3 - 6t^2 + 6t + 4, -2t^3 + 3t^2 + 3t + 1), \quad t \in [0, 1], \\ \mathbf{P}'(t) &= (3t^2 - 12t + 6, -6t^2 + 6t + 3), \quad t \in [0, 1].\end{aligned}$$

All calculated points and tangent vectors are summarized in the following table. Coons cubic curve and tangent vectors along it are drawn in Fig. 2.18.

	$t = 0$	$t = \frac{1}{4}$	$t = \frac{1}{2}$	$t = \frac{3}{4}$	$t = 1$
$\mathbf{P}(t)$	(4, 1)	$(\frac{329}{64}, \frac{61}{32})$	$(\frac{45}{8}, 3)$	$(\frac{355}{64}, \frac{131}{32})$	(5, 5)
$\mathbf{P}'(t)$	(6, 3)	$(\frac{51}{16}, \frac{33}{8})$	$(\frac{3}{4}, \frac{9}{2})$	$(-\frac{21}{16}, \frac{33}{8})$	(-3, 3)

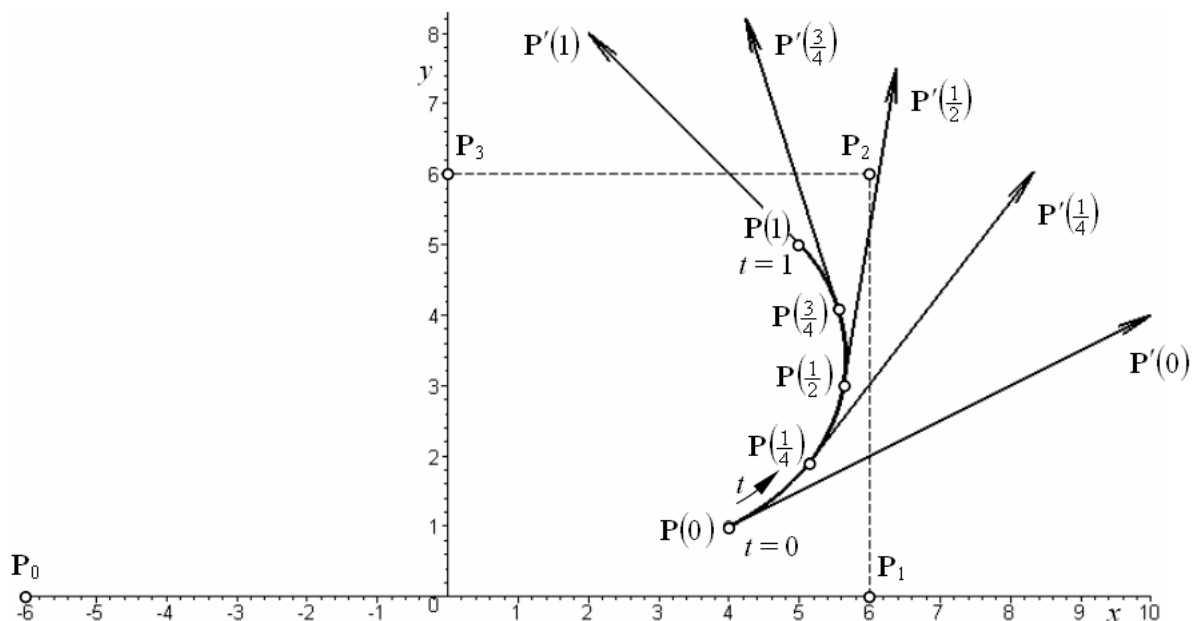


Figure 2.18: Coons cubic curve and tangent vectors at its points

□

2.3.1 Properties of Coons cubic curve

Coons cubic curve has the following interesting geometric properties (take a look at Fig. 2.19).

- Function values of Coons polynomials are less than 1 for any value of parameter $t \in [0, 1]$. Consequently, Coons cubic curve does not pass through any given control point.
- The initial point $\mathbf{P}(0)$ of Coons cubic curve lies at the “ant centroid” of triangle $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$ constructed with respect to control point \mathbf{P}_1 , i.e. at one third of median $\mathbf{P}_1\mathbf{S}_0$ of triangle $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$ constructed from control point \mathbf{P}_1 .
- The terminal point $\mathbf{P}(1)$ of Coons cubic curve lies at the “ant centroid” of triangle $\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ constructed with respect to control point \mathbf{P}_2 , i.e. at one third of median $\mathbf{P}_2\mathbf{S}_1$ of triangle $\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ constructed from control point \mathbf{P}_2 .

- Tangent vector $\mathbf{P}'(0)$ at the initial point of Coons cubic curve is given by

$$\mathbf{P}'(0) = \frac{1}{2} \overrightarrow{\mathbf{P}_0\mathbf{P}_2} = \frac{1}{2}(\mathbf{P}_2 - \mathbf{P}_0).$$

- Tangent vector $\mathbf{P}'(1)$ at the terminal point of Coons cubic curve is given by

$$\mathbf{P}'(1) = \frac{1}{2} \overrightarrow{\mathbf{P}_1\mathbf{P}_3} = \frac{1}{2}(\mathbf{P}_3 - \mathbf{P}_1).$$

- Tangent line at the initial point of Coons cubic curve intersects legs $\mathbf{P}_0\mathbf{P}_1$ and $\mathbf{P}_1\mathbf{P}_2$ at one third from control point \mathbf{P}_1 (points \mathbf{C} , \mathbf{E}). Consequently, leg $\mathbf{P}_1\mathbf{P}_2$ intersects the tangent vector $\mathbf{P}'(0)$ at one third from point $\mathbf{P}(0)$ (point \mathbf{E})
- Tangent line at the terminal point of Coons cubic curve intersects legs $\mathbf{P}_1\mathbf{P}_2$ and $\mathbf{P}_2\mathbf{P}_3$ at one third from control point \mathbf{P}_2 (Points \mathbf{F} , \mathbf{D}). Consequently, leg $\mathbf{P}_2\mathbf{P}_3$ intersects the tangent vector $\mathbf{P}'(1)$ at one third from point $\mathbf{P}(1)$ (point \mathbf{D}).

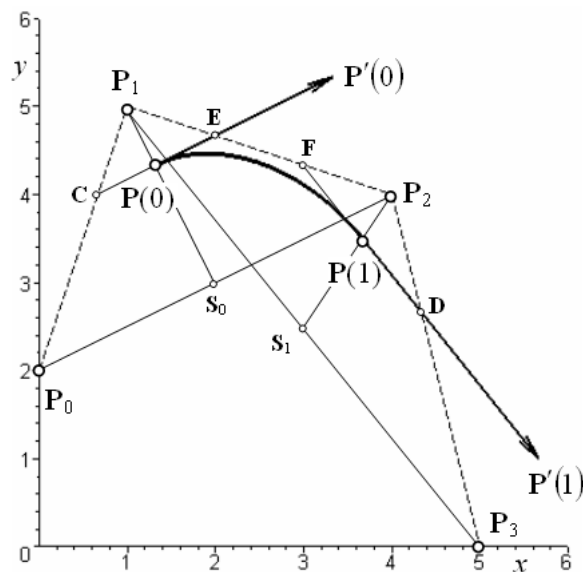


Figure 2.19: Properties of Coons cubic curve

Coons, Bézier and Ferguson cubic curves interrelation

Considering above mentioned properties of Coons cubic curve, we can deduce the mutual relation between Coons, Bézier and Ferguson cubic curves. Let $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ be known control polygon of Coons cubic curve. Let $\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2$ and \mathbf{V}_3 be control points of corresponding Bézier cubic curve expressed in terms of control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ and \mathbf{P}_3 (see Fig. 2.19)

$$\begin{aligned}
\mathbf{V}_0 &= \mathbf{P}(0) = \mathbf{P}_1 + \frac{1}{3}(\mathbf{S}_0 - \mathbf{P}_1) = \mathbf{P}_1 + \frac{1}{3} \left[\frac{1}{2}(\mathbf{P}_0 + \mathbf{P}_2) - \mathbf{P}_1 \right] = \frac{1}{6}\mathbf{P}_0 + \frac{2}{3}\mathbf{P}_1 + \frac{1}{6}\mathbf{P}_2, \\
\mathbf{V}_1 &= \mathbf{E} = \mathbf{P}_1 + \frac{1}{3}(\mathbf{P}_2 - \mathbf{P}_1) = \frac{2}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2, \\
\mathbf{V}_2 &= \mathbf{F} = \mathbf{P}_2 + \frac{1}{3}(\mathbf{P}_1 - \mathbf{P}_2) = \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_1, \\
\mathbf{V}_3 &= \mathbf{P}(1) = \mathbf{P}_2 + \frac{1}{3}(\mathbf{S}_1 - \mathbf{P}_2) = \mathbf{P}_2 + \frac{1}{3} \left[\frac{1}{2}(\mathbf{P}_1 + \mathbf{P}_3) - \mathbf{P}_2 \right] = \frac{1}{6}\mathbf{P}_1 + \frac{2}{3}\mathbf{P}_2 + \frac{1}{6}\mathbf{P}_3,
\end{aligned} \tag{2.48}$$

i.e. in the following well-arranged form

$$\begin{aligned}
\mathbf{V}_0 &= \frac{1}{6}\mathbf{P}_0 + \frac{2}{3}\mathbf{P}_1 + \frac{1}{6}\mathbf{P}_2, \\
\mathbf{V}_1 &= \frac{2}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2, \\
\mathbf{V}_2 &= \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_1, \\
\mathbf{V}_3 &= \frac{1}{6}\mathbf{P}_1 + \frac{2}{3}\mathbf{P}_2 + \frac{1}{6}\mathbf{P}_3.
\end{aligned} \tag{2.49}$$

After substituting control points (2.49) in (2.19), we get

$$\begin{aligned}
\mathbf{P}(t) &= B_{0,3}(t) \left[\frac{1}{6}\mathbf{P}_0 + \frac{2}{3}\mathbf{P}_1 + \frac{1}{6}\mathbf{P}_2 \right] + B_{1,3}(t) \left[\frac{2}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2 \right] + \\
&\quad + B_{2,3}(t) \left[\frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_1 \right] + B_{3,3}(t) \left[\frac{1}{6}\mathbf{P}_1 + \frac{2}{3}\mathbf{P}_2 + \frac{1}{6}\mathbf{P}_3 \right] = \\
&= \frac{1}{6}B_{0,3}(t)\mathbf{P}_0 + \left\{ \frac{2}{3}[B_{0,3}(t) + B_{1,3}(t)] + \frac{1}{3}B_{2,3}(t) + \frac{1}{6}B_{3,3}(t) \right\} \mathbf{P}_1 + \\
&\quad + \left\{ \frac{1}{6}B_{0,3}(t) + \frac{1}{3}B_{1,3}(t) + \frac{2}{3}[B_{2,3}(t) + B_{3,3}(t)] \right\} \mathbf{P}_2 + \frac{1}{6}B_{3,3}(t)\mathbf{P}_3 = \\
&= \frac{1}{6}(1-t)^3\mathbf{P}_0 + \frac{1}{6}(3t^3 - 6t^2 + 4)\mathbf{P}_1 + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)\mathbf{P}_2 + \frac{1}{6}t^3\mathbf{P}_3.
\end{aligned} \tag{2.50}$$

The polynomials in the last row of (2.50) are Coons polynomials (2.48). Therefore, (2.50) is the equation of Coons cubic curve given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$.

When solving the set of equations (2.49) with respect to control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ and \mathbf{P}_3 , we get a reverse transformation of control points

$$\begin{aligned}
\mathbf{P}_0 &= 6\mathbf{V}_0 - 7\mathbf{V}_1 + 2\mathbf{V}_2, \\
\mathbf{P}_1 &= 2\mathbf{V}_1 - \mathbf{V}_2, \\
\mathbf{P}_2 &= 2\mathbf{V}_2 - \mathbf{V}_1, \\
\mathbf{P}_3 &= 2\mathbf{V}_1 - 7\mathbf{V}_2 + 6\mathbf{V}_3.
\end{aligned} \tag{2.51}$$

- **Example 2.12 – Coons, Bézier and Ferguson cubic curves interrelation.** Suppose that Coons cubic curve $\mathbf{P}(t) = (x(t), y(t))$ from Example 2.11 is Bézier cubic curve $\mathbf{P}_B(t) = (x_B(t), y_B(t))$, $t \in [0, 1]$ as well as Ferguson cubic curve $\mathbf{P}_F(t) = (x_F(t), y_F(t))$, $t \in [0, 1]$.

Draw control points of Coons cubic curve $\mathbf{P}(t)$ and construct its control polygon. Construct control points of Bézier cubic curve $\mathbf{P}_B(t)$. Construct input data Ferguson cubic curve $\mathbf{P}_F(t)$. Sketch the curve.

Determine control points $\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2$ and \mathbf{V}_3 of Bézier cubic curve $\mathbf{P}_B(t)$ and find its parametric expression. Determine definition points \mathbf{A} and \mathbf{B} and tangent vectors \mathbf{a} and \mathbf{b} of Ferguson cubic curve $\mathbf{P}_F(t)$ and find its parametric expression. Compare parametric expressions of Coons, Bézier and Ferguson cubic curves.

Solution. Construction is shown in Fig. 2.20.

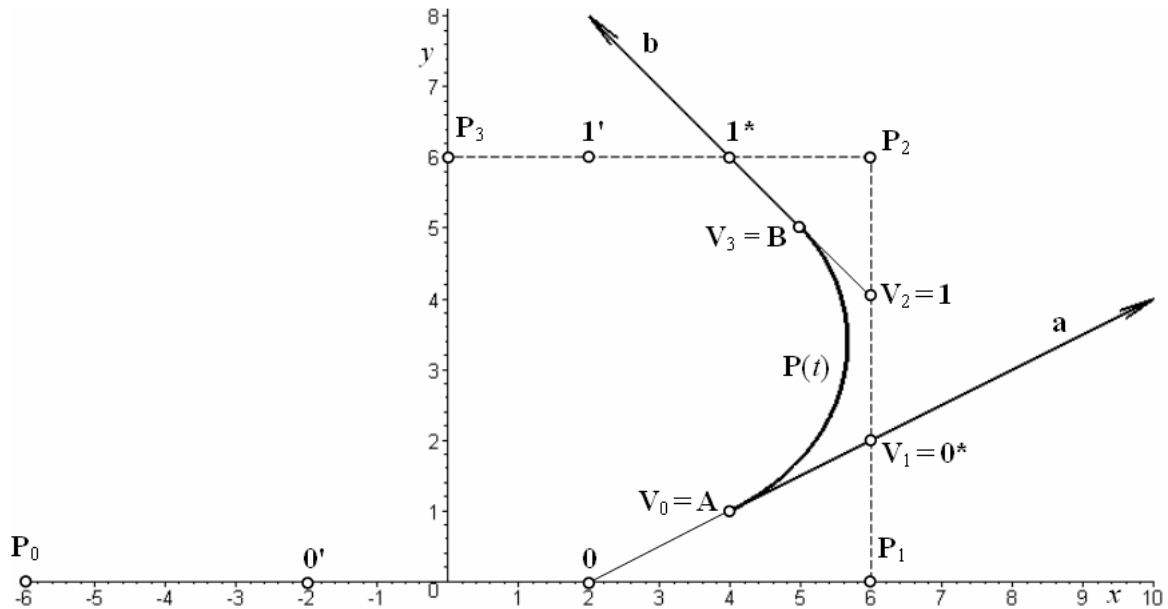


Figure 2.20: Coons, Bézier and Ferguson cubic curves interrelation

According to (2.49), control points of Bézier cubic curve $\mathbf{P}_B(t)$ are

$$\begin{aligned} \mathbf{V}_0 &= \frac{1}{6}(-6, 0) + \frac{2}{3}(6, 0) + \frac{1}{6}(6, 6) = (4, 1), \\ \mathbf{V}_1 &= \frac{2}{3}(6, 0) + \frac{1}{3}(6, 6) = (6, 2), \\ \mathbf{V}_2 &= \frac{2}{3}(6, 6) + \frac{1}{3}(6, 0) = (6, 4), \\ \mathbf{V}_3 &= \frac{1}{6}(6, 0) + \frac{2}{3}(6, 6) + \frac{1}{6}(0, 6) = (5, 5). \end{aligned}$$

The parametric expression of Bézier cubic curve $\mathbf{P}_B(t)$ is

$$\begin{aligned} x_B(t) &= 4B_{0,3}(t) + 6B_{1,3}(t) + 6B_{2,3}(t) + 5B_{3,3}(t) = t^3 - 6t^2 + 6t + 4, \\ y_B(t) &= B_{0,3}(t) + 2B_{1,3}(t) + 4B_{2,3}(t) + 5B_{3,3}(t) = -2t^3 + 3t^2 + 3t + 1, \quad t \in [0, 1] \end{aligned}$$

and input data of Ferguson cubic curve $\mathbf{P}_F(t)$ according to (2.24) is

$$\begin{aligned}\mathbf{A} &= \mathbf{V}_0 = (4, 1), \\ \mathbf{B} &= \mathbf{V}_3 = (5, 5), \\ \mathbf{a} &= 3(\mathbf{V}_1 - \mathbf{V}_0) = 3[(6, 2) - (4, 1)] = (6, 3), \\ \mathbf{b} &= 3(\mathbf{V}_3 - \mathbf{V}_2) = 3[(5, 5) - (6, 4)] = (-3, 3).\end{aligned}$$

The parametric expression of Ferguson cubic curve $\mathbf{P}_F(t)$ is

$$\begin{aligned}x_F(t) &= 4F_0(t) + 5F_1(t) + 6F_2(t) - 3F_3(t) = t^3 - 6t^2 + 6t + 4, \\ y_F(t) &= F_0(t) + 5F_1(t) + 3F_2(t) + 3F_3(t) = -2t^3 + 3t^2 + 3t + 1, \quad t \in [0, 1].\end{aligned}$$

Since $x_B(t) = x_F(t) = x(t)$ and $y_B(t) = y_F(t) = y(t)$, the three cubic curves are identical. \square

2.3.2 Coons polynomials derivation

Coons polynomials are a special case of B-spline basis functions but their derivation as B-spline basis functions is beyond the scope of this text. We can take the relation between Coons and Bézier cubic curves into consideration and regard (2.50) as the derivation of Coons polynomials.

2.3.3 Construction of Coons cubic curve endpoints

Endpoints of Coons cubic curve can be constructed as “anticentroids” mentioned on page 66. This construction can be simplified in the following way (take a look at Fig. 2.20):

1. Divide all legs of control polygon in thirds to get points $\mathbf{0}'$, $\mathbf{0}$, $\mathbf{0}^*$, $\mathbf{1}$, $\mathbf{1}^*$ and $\mathbf{1}'$.
2. Construct straight line segments $\mathbf{00}^*$ and $\mathbf{11}^*$.
3. The initial point of Coons cubic curve lies at the centre of straight line segment $\mathbf{00}^*$.
4. The terminal point of Coons cubic curve lies at the centre of straight line segment $\mathbf{11}^*$.
5. It is obvious that one half of vector $\mathbf{00}^*$ is equal to one third of tangent vector at the initial point of Coons cubic curve, as well as one half of vector $\mathbf{11}^*$ is equal to one third of tangent vector at the terminal point of Coons cubic curve.

■ **Exercise 2.15** Control points $\mathbf{P}_0 = (12, 6)$, $\mathbf{P}_1 = (6, 0)$, $\mathbf{P}_2 = (0, 0)$ and $\mathbf{P}_3 = (0, 12)$ are given. Find the vector equation of Coons cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$, given by these control points. Suppose that Coons cubic curve $\mathbf{P}(t)$ is Bézier cubic curve $\mathbf{P}_B(t)$, $t \in [0, 1]$, given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$. Using construction and calculation, determine the positions of control points \mathbf{V}_0 , \mathbf{V}_1 , \mathbf{V}_2 and \mathbf{V}_3 of Bézier cubic curve and verify that vector equations of both cubic curves are the same. Draw a picture.

■ **Exercise 2.16** Suppose that Bézier cubic curve and Ferguson cubic curve from Example 2.5 is a Coons cubic curve. Construct and calculate its control points \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 and find its parametric expression.

2.3.4 Coons cubic curve in Rhinoceros

In Rhinoceros, it is possible to draw Coons cubic curve only indirectly using the relation between Coons cubic curve and Bézier cubic curve. The first and the last control point of Bézier cubic curve can be constructed according to the procedure mentioned in Section 2.3.3. The points at the thirds of the middle leg of Coons cubic curve control polygon are the internal control points of Bézier cubic curve, see Fig. 2.20.

■ **Exercise 2.17** Consider the assignment of example 2.12. Construct two pictures in Rhinoceros:

1. Draw the control polygon of Coons cubic curve (use command *Line Segments*), construct control points of Bézier cubic curve and draw Bézier cubic curve.
2. Draw the control polygon of Bézier cubic curve (use command *Line Segments*), construct control points of Coons cubic curve and draw Coons cubic curve as Bézier cubic curve.

2.4 Coons cubic B-spline

Now, we will consider two Coons cubic curves with overlapping control polygons: $\mathbf{P}(t)$, $t \in [0, 1]$ given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ and $\mathbf{R}(s)$, $s \in [0, 1]$ given by control polygon $\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4$. In Fig. 2.21, an example of such a situation is depicted. The coordinates of control points \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 are the same as in Example 2.11, $\mathbf{P}_4 = (-3, 3)$. The overlap of control polygons automatically guarantees C^2 continuous joining of corresponding Coons cubic curves.

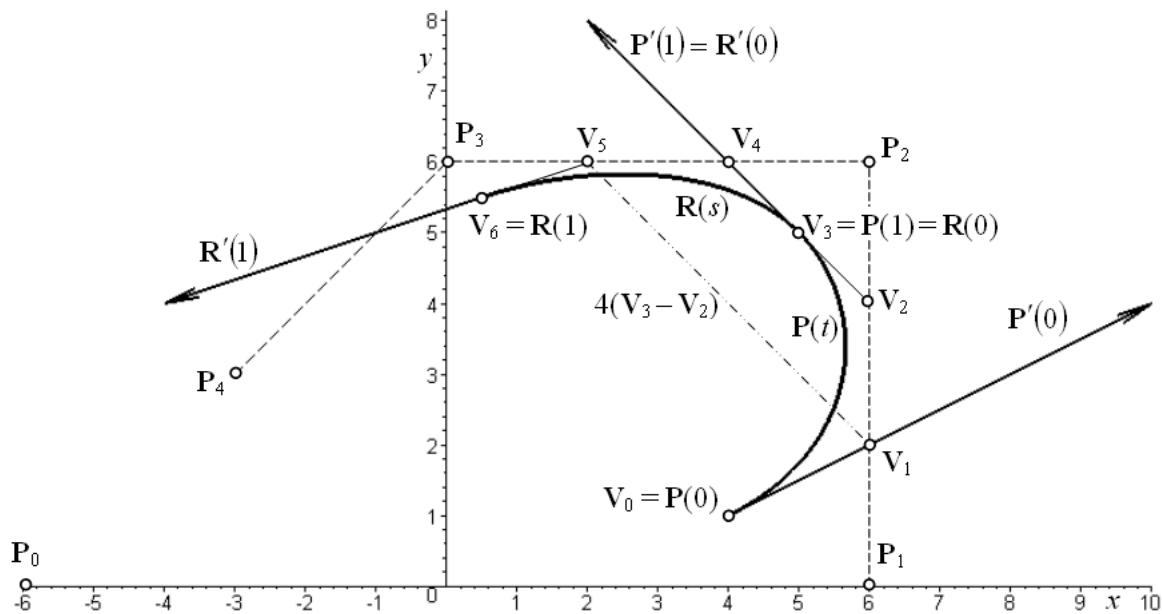


Figure 2.21: Coons cubic B-spline

To understand C^2 continuity of two Coons cubic curves given by overlapping control polygons, it is useful to consider both Coons cubic curves to be Bézier cubic curves. The first Bézier cubic curve is given by control polygon $\mathbf{V}_0\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$, the second one is given by control polygon $\mathbf{V}_3\mathbf{V}_4\mathbf{V}_5\mathbf{V}_6$. Then, from the conditions derived in section 2.2.4, it follows that curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are joined with:

- C^0 continuity, because point \mathbf{V}_3 is the common point of both Bézier cubic curves,
- C^1 continuity, because $\mathbf{V}_3 - \mathbf{V}_2 = \mathbf{V}_4 - \mathbf{V}_3$ is satisfied,
- C^2 continuity, because $\mathbf{V}_5 - \mathbf{V}_1 = 4(\mathbf{V}_3 - \mathbf{V}_2)$ is satisfied.

Moreover, it is possible to add new control points \mathbf{P}_i in both directions (i.e. $i < 0$ as well as $i > 4$) and construct endpoints of the further Coons cubic curves. Thus, a *piecewise C^2 continuous curve* arises. Endpoints of a piecewise curve are called *knots*. This curve is a so called *uniform B-spline curve of third degree*, known as a *Coons cubic B-spline*, too. Mathematical description of uniform B-spline curve of third degree is beyond the scope of this text. Here, we use the following definition.

■ **Definition 2.4 – Coons cubic B-spline.** Given a sequence of control points $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$, $n \geq 4$ in space, a uniform B-spline curve of third degree $\mathbf{R}(t)$ compounded from $n - 2$ Coons cubic curves with vector equations

$$\begin{aligned} \mathbf{R}_0(t) &= C_0(t)\mathbf{P}_0 + C_1(t)\mathbf{P}_1 + C_2(t)\mathbf{P}_2 + C_3(t)\mathbf{P}_3, \quad t \in [0, 1], \\ \mathbf{R}_1(t) &= C_0(t)\mathbf{P}_1 + C_1(t)\mathbf{P}_2 + C_2(t)\mathbf{P}_3 + C_3(t)\mathbf{P}_4, \quad t \in [0, 1], \\ &\vdots \\ \mathbf{R}_{n-3}(t) &= C_0(t)\mathbf{P}_{n-3} + C_1(t)\mathbf{P}_{n-2} + C_2(t)\mathbf{P}_{n-1} + C_3(t)\mathbf{P}_n, \quad t \in [0, 1] \end{aligned} \quad (2.52)$$

is called *Coons cubic B-spline*. □

■ **Example 2.13 – Open and closed Coons cubic B-spline.** The following control points of Coons cubic B-spline k are given:

$$\begin{aligned} \mathbf{P}_0 &= (0, 0), \quad \mathbf{P}_1 = (-6, 6), \quad \mathbf{P}_2 = (-6, 0), \\ \mathbf{P}_3 &= (0, -6), \quad \mathbf{P}_4 = (6, 0), \quad \mathbf{P}_5 = (6, 6), \quad \mathbf{P}_6 = \mathbf{P}_0 = (0, 0). \end{aligned}$$

a) How many Coons cubic curves create this Coons cubic B-spline k ? Write control polygons of individual Coons cubic curves. Draw a picture.

b) Modify the given control polygon of Coons cubic B-spline k to create a closed curve. Draw a picture.

Solution.

a) Since $n = 6$, the open Coons Cubic B-spline k contains four Coons cubic curves. Denote these Coons cubic curves with k_0, k_1, k_2 and k_3 , see Fig. 2.22. Their control polygons are

$$k_0 : \mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3, \quad k_1 : \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4, \quad k_2 : \mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5, \quad k_3 : \mathbf{P}_3\mathbf{P}_4\mathbf{P}_5\mathbf{P}_6.$$

b) It is necessary to add two more control points \mathbf{P}_7 and \mathbf{P}_8 , see Fig. 2.23. Their position is not arbitrary but the following conditions

$$\mathbf{P}_7 = \mathbf{P}_1, \quad \mathbf{P}_8 = \mathbf{P}_2,$$

have to be satisfied to create a closed curve. Then, $n = 8$ and the closed Coons cubic B-spline k is created by six Coons cubic curves. Their control polygons are

$$\begin{aligned} k_0 &: \mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3, \quad k_1 : \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4, \quad k_2 : \mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5, \\ k_3 &: \mathbf{P}_3\mathbf{P}_4\mathbf{P}_5\mathbf{P}_6, \quad k_4 : \mathbf{P}_4\mathbf{P}_5\mathbf{P}_6\mathbf{P}_7, \quad k_5 : \mathbf{P}_5\mathbf{P}_6\mathbf{P}_7\mathbf{P}_8. \end{aligned}$$

The control polygons of Coons cubic curves k_0, k_1, k_2 and k_3 of closed Coons cubic B-spline are the same as the control polygons of Coons cubic curves k_0, k_1, k_2 and k_3 of open Coons cubic B-spline.

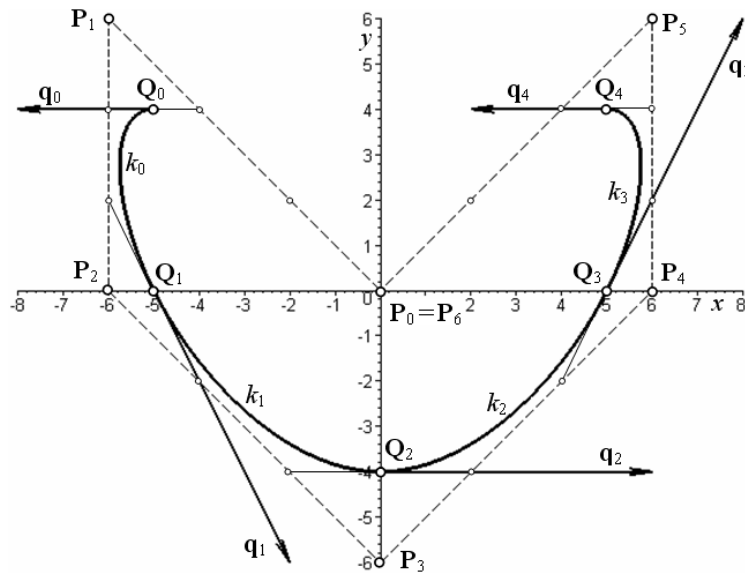


Figure 2.22: Open Coons cubic B-spline

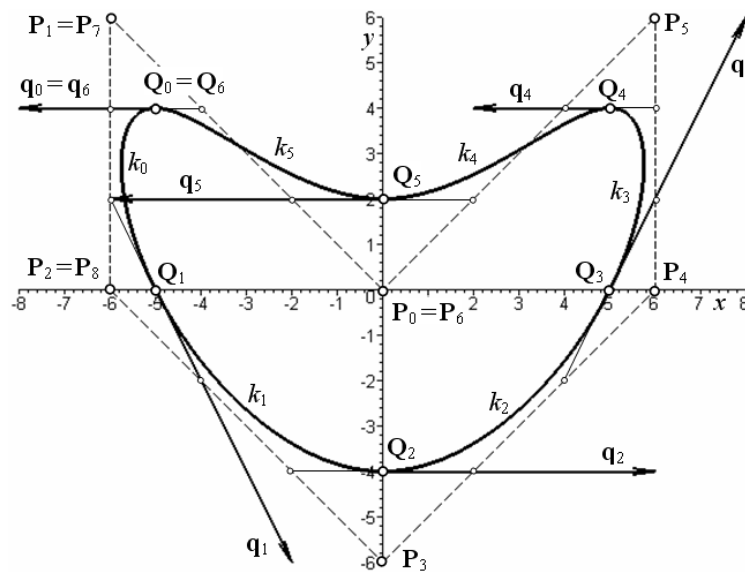


Figure 2.23: Closed Coons cubic B-spline

□

2.4.1 Properties of Coons cubic B-spline

- Control polygon of Coons cubic B-spline is created by at least five control points.
- If the last three control points are identical with the first three control points, i.e. $\mathbf{P}_n = \mathbf{P}_2$, $\mathbf{P}_{n-1} = \mathbf{P}_1$, $\mathbf{P}_{n-2} = \mathbf{P}_0$, Coons cubic B-spline is closed. Otherwise, Coons cubic B-spline is open.
- Coons cubic B-spline does not pass through any control point of its control polygon.

- Coons cubic B-spline is created by $n - 2$ C^2 continuously joined Coons cubic curves. Endpoints of these Coons cubic curves are called knots of Coons cubic B-spline.
- Knots of Coons cubic B-spline and tangent vectors at these knots can be constructed according to the properties of Coons cubic curve mentioned in Section 2.3.1.
- Coons cubic B-spline is a piecewise defined curve by partially overlapping control polygons. Consequently, a change of position of one control point does not cause the change of whole Coons cubic B-spline. The change of one control point influences the shape of those individual Coons cubic curves, of which vector equation contains the changing control point – i.e. of four Coons cubic curves at the most. This property is very useful when modelling the shape of curve given by a huge number of control points.
- The domain of each individual Coons cubic curve of Coons cubic B-spline is $t \in [0, 1]$, i.e. the length of all domains is the same (unit). If a piecewise curve has this property, it is called a *uniform* curve or curve with a *uniform parametrization*.

In CAD/CAM systems, the commands to create a closed Coons cubic B-spline are only available. An open Coons cubic B-spline is unusable for practical applications. The reason is that the curve cannot be drawn until a user enter the first three control points. Furthermore, the position of the initial point of the drawn curve (see above mentioned “ant centroid”) can be slightly confusing for a user without knowledge of open Coons cubic B-spline properties.

■ **Example 2.14 “@” sign created by Coons cubic B-spline.** Suppose curves k^* and k from Example 2.6 are Coons cubic B-splines.

Determine the control points of Coons cubic B-splines k^* and k , draw these control points, construct control polygon and sketch the Coons cubic B-splines.

Solution. Constructions of Coons cubic B-splines control points is shown in Fig. 2.24 (the constructions are not drawn in all details).

To calculate Cartesian coordinates of Coons cubic B-spline control points, we use interrelation between control points of Coons cubic curve and Bézier cubic curve. Denote with \mathbf{P}_i^* the control points of Coons cubic B-spline k^* . Then, according to (2.51), we get

$$\mathbf{P}_0^* = 6\mathbf{V}_0^* - 7\mathbf{V}_1^* + 2\mathbf{V}_2^* = 6(47, 46) - 7(29, 53) + 2(23, 42) = (125, -11),$$

$$\mathbf{P}_1^* = 2\mathbf{V}_1^* - \mathbf{V}_2^* = 2(29, 53) - (23, 42) = (35, 64),$$

$$\mathbf{P}_2^* = 2\mathbf{V}_2^* - \mathbf{V}_1^* = 2(23, 42) - (29, 53) = (17, 31),$$

$$\mathbf{P}_3^* = 2\mathbf{V}_1^* - 7\mathbf{V}_2^* + 6\mathbf{V}_3^* = 2(29, 53) - 7(23, 42) + 6(22, 34) = (29, 16).$$

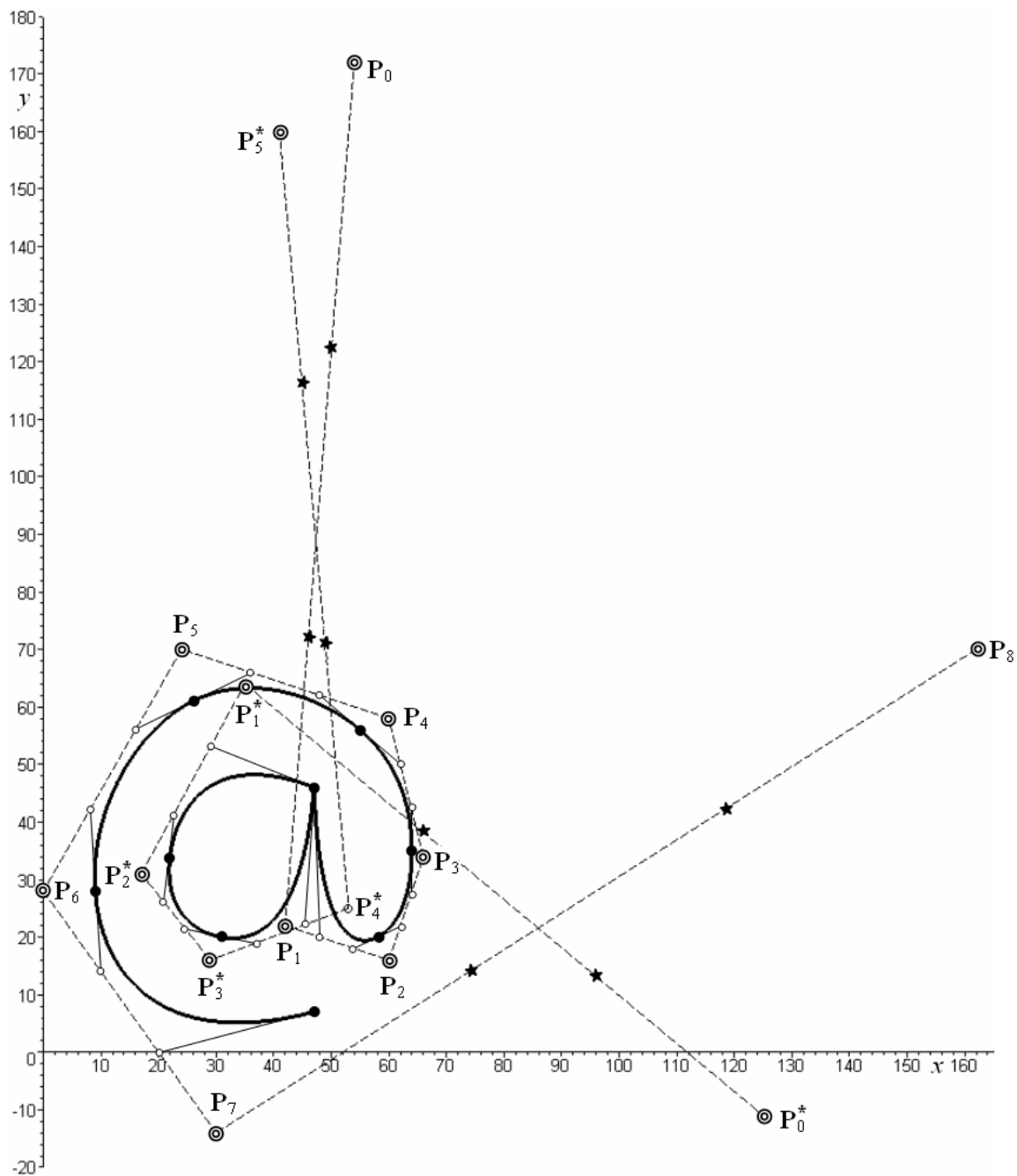
Coons cubic curve k_1^* is given by control polygon $\mathbf{P}_1^*\mathbf{P}_2^*\mathbf{P}_3^*\mathbf{P}_4^*$. Due to overlapping of control polygons of Coons cubic curves including in Coons cubic B-spline, only control point \mathbf{P}_4^* is unknown

$$\mathbf{P}_4^* = 2\mathbf{V}_4^* - 7\mathbf{V}_5^* + 6\mathbf{V}_6^* = 2(21, 26) - 7(25, 21) + 6(31, 20) = (53, 25).$$

Similarly, in control polygon $\mathbf{P}_2^*\mathbf{P}_3^*\mathbf{P}_4^*\mathbf{P}_5^*$ of Coons cubic curve k_2^* , only control point \mathbf{P}_5^* is unknown

$$\mathbf{P}_5^* = 2\mathbf{V}_7^* - 7\mathbf{V}_8^* + 6\mathbf{V}_0^* = 2(37, 19) - 7(45, 22) + 6(47, 46) = (41, 160).$$

It is possible to proceed in similar way to calculate Cartesian coordinates \mathbf{P}_i of Coons cubic B-spline k : $\mathbf{P}_0 = (54, 172)$, $\mathbf{P}_1 = (42, 22)$, $\mathbf{P}_2 = (60, 16)$, $\mathbf{P}_3 = (66, 34)$, $\mathbf{P}_4 = (60, 58)$, $\mathbf{P}_5 = (24, 70)$, $\mathbf{P}_6 = (0, 28)$, $\mathbf{P}_7 = (30, -14)$ and $\mathbf{P}_8 = (162, 70)$.



- ⊙ Control points of Coons cubic B-spline,
- Endpoints of control polygons of Bézier cubic curves identical with endpoints of control polygon of Coons cubic curves,
- Inner points of control polygons of Bézier cubic curves,
- ★ Points located at thirds of end legs of Coons cubic B-spline control polygon.

Figure 2.24: “@” sign created by Coons cubic B-spline

- **Exercise 2.18** Find vector equation of Coons cubic curve $\mathbf{R}(s)$, $s \in [0, 1]$, from Fig. 2.21 and prove C^2 continuous joining with Coons cubic curve $\mathbf{P}(t)$, $t \in [0, 1]$ (use the results of Example 2.11, too).
- **Exercise 2.19** Coons cubic B-spline is given by control polygon $\mathbf{P}_0 = (0, -6)$, $\mathbf{P}_1 = (-6, 0)$, $\mathbf{P}_2 = (-6, 6)$, $\mathbf{P}_3 = (6, 6)$, $\mathbf{P}_4 = (6, 0)$, $\mathbf{P}_5 = (12, 0)$. Draw control points, construct control polygon, construct knots of Coons cubic B-spline and tangent vectors at these knots. Sketch Coons cubic B-spline. How many curve segments create Coons cubic B-spline? What is the continuity between individual curve segments of Coons cubic B-spline?
- **Exercise 2.20** Coons cubic B-spline is given by control polygon $\mathbf{P}_0 = (0, 0)$, $\mathbf{P}_1 = (0, 6)$, $\mathbf{P}_2 = (6, 6)$, $\mathbf{P}_3 = (6, 0)$, $\mathbf{P}_4 = (0, 0)$, $\mathbf{P}_5 = (0, 6)$, $\mathbf{P}_6 = (6, 6)$. Draw control points, construct control polygon, construct knots of Coons cubic B-spline and tangent vectors at these knots. Sketch Coons cubic B-spline. How many curve segments create Coons cubic B-spline? What is the continuity between individual curve segments of Coons cubic B-spline?
- **Exercise 2.21** Bézier cubic curve k is given by control points $\mathbf{V}_0 = (-3, 0)$, $\mathbf{V}_1 = (-2, 2)$, $\mathbf{V}_2 = (2, 2)$, $\mathbf{V}_3 = (3, 0)$. Draw control points, construct control polygon and sketch Bézier cubic curve. Suppose that Bézier cubic curve k is Coons cubic curve. Construct control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ of Coons cubic curve and verify their coordinates by calculation. Modify control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ to create a closed Coons cubic B-spline. Determine the number of its Coons cubic curves. Write control polygons of individual Coons cubic curves. Suppose that Coons cubic curves of closed Coons cubic B-spline are Bézier curves. Construct control points of these Bézier curves. What is the minimal number of curve segments created this closed Coons cubic B-spline? What is the shape of control polygon of closed Coons cubic B-spline which is created by minimal number of curve segments?
- **Exercise 2.22** Bézier cubic curve is given by control points $\mathbf{V}_0 = (1, 5)$, $\mathbf{V}_1 = (2, 6)$, $\mathbf{V}_2 = (4, 6)$, $\mathbf{V}_3 = (6, 5)$. Construct the necessary number of control polygons of Bézier cubic curves to create a closed curve consisting of C^2 continuously joined Bézier cubic curves. Help: use interrelation of Bézier cubic curve, Coons cubic curve and Coons cubic B-spline.
- **Exercise 2.23** Coons cubic B-spline is given by control polygon $\mathbf{P}_0 = (3, 9)$, $\mathbf{P}_1 = (9, 3)$, $\mathbf{P}_2 = (3, -3)$, $\mathbf{P}_3 = (0, 0)$, $\mathbf{P}_4 = (3, 3)$, $\mathbf{P}_5 = (9, -3)$, $\mathbf{P}_6 = (3, -9)$, $\mathbf{P}_7 = (-3, -3)$, $\mathbf{P}_8 = (0, 0)$, $\mathbf{P}_9 = (3, -3)$, $\mathbf{P}_{10} = (-3, -9)$, $\mathbf{P}_{11} = (-9, -3)$, $\mathbf{P}_{12} = (-3, 3)$, $\mathbf{P}_{13} = (0, 0)$, $\mathbf{P}_{14} = (-3, -3)$, $\mathbf{P}_{15} = (-9, 3)$, $\mathbf{P}_{16} = (-3, 9)$, $\mathbf{P}_{17} = (3, 3)$, $\mathbf{P}_{18} = (0, 0)$, $\mathbf{P}_{19} = (-3, 3)$, $\mathbf{P}_{20} = (3, 9)$. Draw control points, construct control polygon, construct knots of Coons cubic B-spline and tangent vector at these knots. Sketch Coons cubic B-spline. How many curve segments create the Coons cubic B-spline? What is the continuity between individual curve segments of the Coons cubic B-spline?
- **Exercise 2.24** Modify the control polygon of Coons cubic B-spline from Exercise 2.23 to create a closed Coons cubic B-spline.

2.4.2 Coons cubic B-spline in Rhinoceros

Closed Coons cubic B-spline – Command: *Control Point Curve* → Start of curve: type degree 3 in command prompt; enter the first control point of closed Coons cubic B-spline → Next point: enter all control points but the last → Next point: click near the initial point of drawn curve. The curve is automatically closed with C^2 continuity and closed Coons cubic B-spline is drawn. The last two control points are not entered.

Remarks:

1. The commands concerning control points and control polygons of closed Coons cubic B-spline are the same as the commands used in the case of Bézier curve described in section 2.2.5.
2. The rules for modification of the shape of closed Coons cubic B-spline are the same as the rules used in the case of Bézier curve described in section 2.2.5.

Knots of Coons cubic B-spline – Command: *Multiple Points* → Location of point object: activate *Knot object snap*, possibly deactivate *Snap*; move cursor (appearance of which is crosshair) along closed Coons cubic B-spline and click when information *Knot* is displayed. The points representing knots of closed Coons cubic B-spline are drawn as individual entities.

Attention! Do not use commands *Insert Knot* and *Remove Knot* to modify the shape of a closed Coons cubic B-spline. These commands cause nonuniform parametrization of the curve. After this modification, the original curve is not a closed Coons cubic B-spline anymore but a third degree nonuniform B-spline curve (see [?], [2]). Do not use the command *Edit Control Point Weight*, either. This command causes rational parametrization of the curve (not polynomial). After this modification, the original curve is not a closed Coons cubic B-spline anymore, but, in general, a NURBS curve (see [?], [2]).

■ **Exercise 2.25** Realize Exercises 2.20, 2.21, 2.22, 2.23 and 2.24 in Rhinoceros.

2.5 Uniform clamped B-spline curve of third degree

In CAD systems, a piecewise curve interpolating the first and the last control point and approximating all internal control points of its control polygon is widely used. A special case of such a curve is a so called *uniform clamped B-spline curve of third degree*, shortly *clamped curve*. Mathematical description of clamped curve is beyond the scope of this text. However, as we see in this section, curve segments of clamped curve are created by Bézier cubic curves as well as by Coons cubic curves. Therefore, we use corresponding interrelations to describe clamped curve in a simplified way.

In Fig. 2.25, clamped curve k identical with open Coons cubic B-spline from Example 2.13 is drawn. Control points of clamped curve are denoted with $\mathbf{P}_0, \dots, \mathbf{P}_6$, control points of original Coons cubic B-spline are denoted with $\tilde{\mathbf{P}}_0, \dots, \tilde{\mathbf{P}}_6$. Obviously, control polygon of clamped curve and control polygon of Coons cubic B-spline differ *only in the first two and in the last two control points*. The positions of the first two and the last two control points of clamped curve can be deduced as follows:

1. The initial point of clamped curve has to be “ant centroid” \mathbf{Q}_0 of triangle $\tilde{\mathbf{P}}_0\tilde{\mathbf{P}}_1\tilde{\mathbf{P}}_2$ constructed with respect to point $\tilde{\mathbf{P}}_1$, to interpolate the first control point of its control polygon.

2. The terminal point of clamped curve has to be the “ant centroid” Q_4 of triangle $\tilde{P}_4\tilde{P}_5\tilde{P}_6$ constructed with respect to point \tilde{P}_5 , to interpolate the last control point of its control polygon.
3. Control point P_1 has to lie at the first third of tangent vector q_0 , therefore, on the leg $\tilde{P}_1\tilde{P}_2$ at one third from point \tilde{P}_1 , see properties of Coons cubic curve.
4. Similarly, control point P_5 has to lie at the second third of leg $\tilde{P}_4\tilde{P}_5$.

Transformation formulas between control points P_0, \dots, P_n of clamped curve and control points of open Coons cubic B-spline $\tilde{P}_0, \dots, \tilde{P}_n$ are as follows

$$\begin{aligned}
 P_0 &= Q_0 = \frac{1}{6}\tilde{P}_0 + \frac{2}{3}\tilde{P}_1 + \frac{1}{6}\tilde{P}_2, \\
 P_1 &= \frac{2}{3}\tilde{P}_1 + \frac{1}{3}\tilde{P}_2, \\
 P_i &= \tilde{P}_i, \quad i = 2, \dots, n-2, \\
 P_{n-1} &= \frac{1}{3}\tilde{P}_{n-2} + \frac{2}{3}\tilde{P}_{n-1}, \\
 P_n &= Q_{n-2} = \frac{1}{6}\tilde{P}_{n-2} + \frac{2}{3}\tilde{P}_{n-1} + \frac{1}{6}\tilde{P}_n.
 \end{aligned}
 \tag{2.53}$$

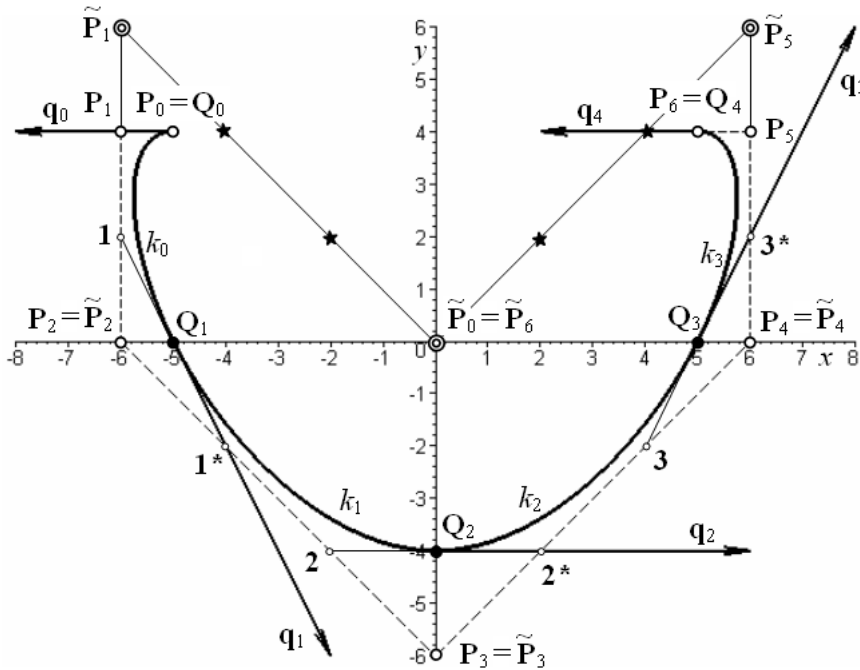


Figure 2.25: Clamped curve

Generally, the first two and the last two curve segments of clamped curve are created by Bézier curves and all inner curve segments of clamped curve are created by Coons cubic curves, i.e. Coons cubic B-spline. Specially, clamped curve in Fig. 2.25 is created only by four curve segments – Bézier cubic curves with the following control polygons

$$k_0: P_0P_11Q_1, \quad k_1: Q_11^*2Q_2, \quad k_2: Q_22^*3Q_3, \quad k_3: Q_33^*P_5P_6.$$

2.5.1 Properties of clamped curve

Properties of clamped curve can be deduced from properties of Bézier cubic curve, Coons cubic curve and Coons cubic B-spline because all these curves can create a certain part of clamped curve. The concrete situation depends on the number of control points of clamped curve. Now, we will analyze individual cases in detail.

$n = 3$: Clamped curve is created by only one curve segment k – Bézier cubic curve. Its control polygon is identical with control polygon of clamped curve

$$k: \mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3.$$

$n = 4$: An example of clamped curve given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4$ is shown in Fig. 2.26 a). In this case, clamped curve is created by two curve segments k_0 and k_1 . Both curve segments are Bézier cubic curves given by control polygons

$$k_0: \mathbf{P}_0\mathbf{P}_1\mathbf{1}\mathbf{Q}_1, \quad k_1: \mathbf{Q}_1\mathbf{1}^*\mathbf{P}_3\mathbf{P}_4.$$

The common point \mathbf{Q}_1 lies at the centre of straight line segment $\mathbf{11}^*$. Point $\mathbf{1}$ divides the second leg of control polygon in halves. Point $\mathbf{1}^*$ divides the last but one (the third in this case) leg of control polygon in halves.

$n = 5$: An example of clamped curve given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5$ is shown in Fig. 2.26 b). In this case, clamped curve is created by three curve segments k_0 , k_1 and k_2 . All the three curve segments are Bézier cubic curves with control polygons

$$k_0: \mathbf{P}_0\mathbf{P}_1\mathbf{1}\mathbf{Q}_1, \quad k_1: \mathbf{Q}_1\mathbf{1}^*\mathbf{2}\mathbf{Q}_2, \quad k_2: \mathbf{Q}_2\mathbf{2}^*\mathbf{P}_4\mathbf{P}_5.$$

The construction of knots \mathbf{Q}_1 and \mathbf{Q}_2 of clamped curve is as follows: divide the second and the fourth (last but one) leg of control polygon in halves to get points $\mathbf{1}$ and $\mathbf{2}^*$. Divide the third leg of control polygon in thirds to get points $\mathbf{1}^*$ and $\mathbf{2}$. Knot \mathbf{Q}_1 lies at the centre of straight line segment $\mathbf{11}^*$. Knot \mathbf{Q}_2 lies at the centre of straight line segment $\mathbf{22}^*$.

$n = 6$: An example of clamped curve given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5\mathbf{P}_6$ is shown in Fig. 2.26 c). In this case, clamped curve is created by four curve segments k_0 , k_1 , k_2 and k_3 . All the four curve segments are Bézier cubic curves with control polygons

$$k_0: \mathbf{P}_0\mathbf{P}_1\mathbf{1}\mathbf{Q}_1, \quad k_1: \mathbf{Q}_1\mathbf{1}^*\mathbf{2}\mathbf{Q}_2, \quad k_2: \mathbf{Q}_2\mathbf{2}^*\mathbf{3}\mathbf{Q}_3, \quad k_3: \mathbf{Q}_3\mathbf{3}^*\mathbf{P}_5\mathbf{P}_6.$$

Construction of knots \mathbf{Q}_1 , \mathbf{Q}_2 and \mathbf{Q}_3 is as follows: divide the second and the last but one leg in halves to get points $\mathbf{1}$ and $\mathbf{3}^*$. Divide the remaining legs of control polygon in thirds to get points $\mathbf{1}^*$, $\mathbf{2}$, $\mathbf{2}^*$ and $\mathbf{3}$. Knots \mathbf{Q}_1 , \mathbf{Q}_2 , \mathbf{Q}_3 lie at centers of straight line segments $\mathbf{11}^*$, $\mathbf{22}^*$, $\mathbf{33}^*$ in the given order.

$n = 7$: An example of clamped curve given by control polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5\mathbf{P}_6\mathbf{P}_7$ is shown in Fig. 2.26 d). In this case, clamped curve is created by five curve segments k_0 , k_1 , k_2 , k_3 and k_4 . All five curve segments are Bézier cubic curves with control polygons

$$k_0: \mathbf{P}_0\mathbf{P}_1\mathbf{1}\mathbf{Q}_1, \quad k_1: \mathbf{Q}_1\mathbf{1}^*\mathbf{2}\mathbf{Q}_2, \quad k_2: \mathbf{Q}_2\mathbf{2}^*\mathbf{3}\mathbf{Q}_3, \quad k_3: \mathbf{Q}_3\mathbf{3}^*\mathbf{4}\mathbf{Q}_4, \quad k_4: \mathbf{Q}_4\mathbf{4}^*\mathbf{P}_6\mathbf{P}_7.$$

However, the middle curve segment k_2 is simultaneously Coons cubic curve given by control polygon

$$k_2: \mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5.$$

Construction of knots \mathbf{Q}_1 , \mathbf{Q}_2 , \mathbf{Q}_3 and \mathbf{Q}_4 is similar to the previous case, see Fig. 2.26 d).

$n > 7$: Clamped curve given by control polygon $\mathbf{P}_0, \dots, \mathbf{P}_n$ is created by $n - 2$ curve segments k_0, \dots, k_{n-3} . The first two and the last two curve segments are Bézier cubic curves. The remaining $n - 6$ internal curve segments are C^2 continuously joined Coons cubic curves which create open Coons cubic B-spline. Control polygon of this Coons cubic B-spline is $\mathbf{P}_2 \dots \mathbf{P}_{n-2}$. An example of such a curve is shown in Fig. 2.27.

Obviously, for $n \geq 3$ it is possible to create the clamped curve as a set of C^2 continuously joined Bézier cubic curves. C^2 continuity follows from the position of control points of two adjacent Bézier cubic curves because the conditions (2.36), (2.37) and (2.38) are satisfied.

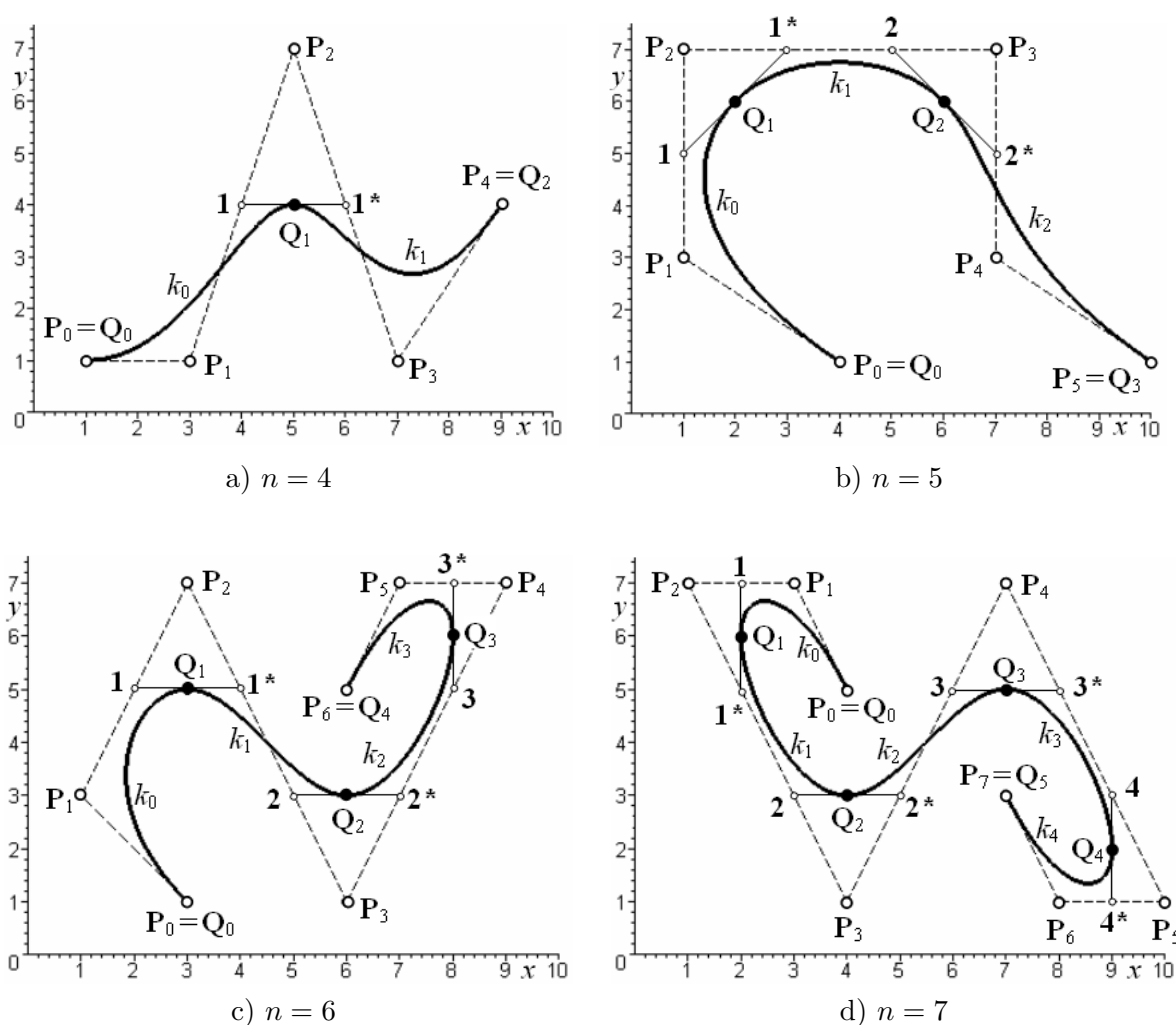


Figure 2.26: Clamped curve

2.5.2 Construction of knots of clamped curve

The construction of knots of clamped curve for $n > 7$ is as follows (take a look at Fig. 2.27):

1. The initial point \mathbf{Q}_0 is equal to the first control point \mathbf{P}_0 .
2. The terminal point \mathbf{Q}_{n-2} is equal to the last control point \mathbf{P}_n .

3. Do not divide the first and the last leg of control polygon.
4. Divide the second and the last but one leg of the control polygon in halves – you get points $\mathbf{1}$ and $(\mathbf{n} - \mathbf{3})^*$.
5. Divide the remaining internal legs of control polygon in thirds – you get points $\mathbf{1}^*, \mathbf{2}, \mathbf{2}^*, \mathbf{3}, \mathbf{3}^*, \dots$
6. Construct straight line segments $\mathbf{11}^*, \mathbf{22}^*, \dots$
7. Knots $\mathbf{Q}_1, \mathbf{Q}_2, \dots$ lie at the centers of straight line segments $\mathbf{11}^*, \mathbf{22}^*, \dots$

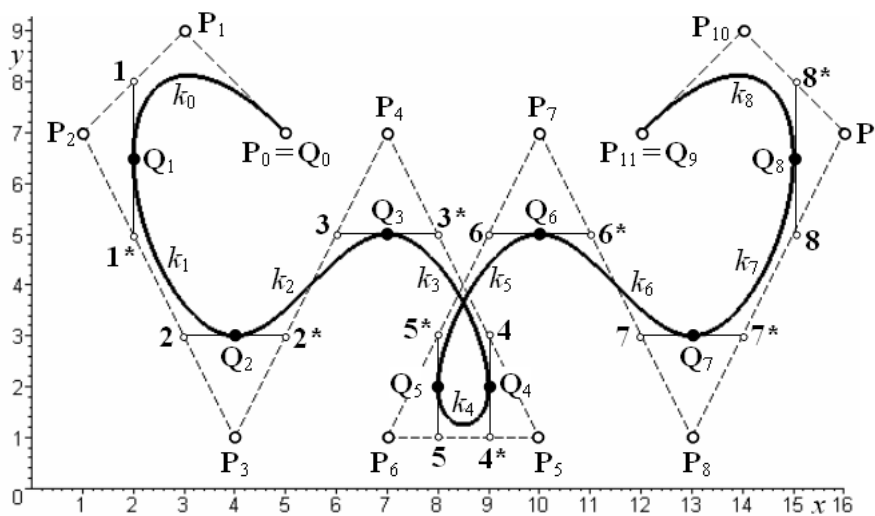


Figure 2.27: Construction of knots of clamped curve

- **Example 2.15 – “@” sign created by clamped curve.** Suppose that curves k^* and k from Example 2.6 are clamped curves.

Construct and read from the graph Cartesian coordinates of control points of clamped curves k and k^* . Sketch the curves.

Solution. Construction of control points of clamped curve is shown in Fig. 2.28. Cartesian coordinates of control points of clamped curves k and k^* are as follows:

$$k: \mathbf{P}_0 = (47, 46), \mathbf{P}_1 = (48, 20), \mathbf{P}_2 = (60, 16),$$

$$\mathbf{P}_3 = (66, 34), \mathbf{P}_4 = (60, 58), \mathbf{P}_5 = (24, 70),$$

$$\mathbf{P}_6 = (0, 28), \mathbf{P}_7 = (20, 0), \mathbf{P}_8 = (47, 7);$$

$$k^*: \mathbf{P}_0^* = (47, 46), \mathbf{P}_1^* = (29, 53), \mathbf{P}_2^* = (17, 31),$$

$$\mathbf{P}_3^* = (29, 16), \mathbf{P}_4^* = (45, 22), \mathbf{P}_5^* = (47, 46).$$

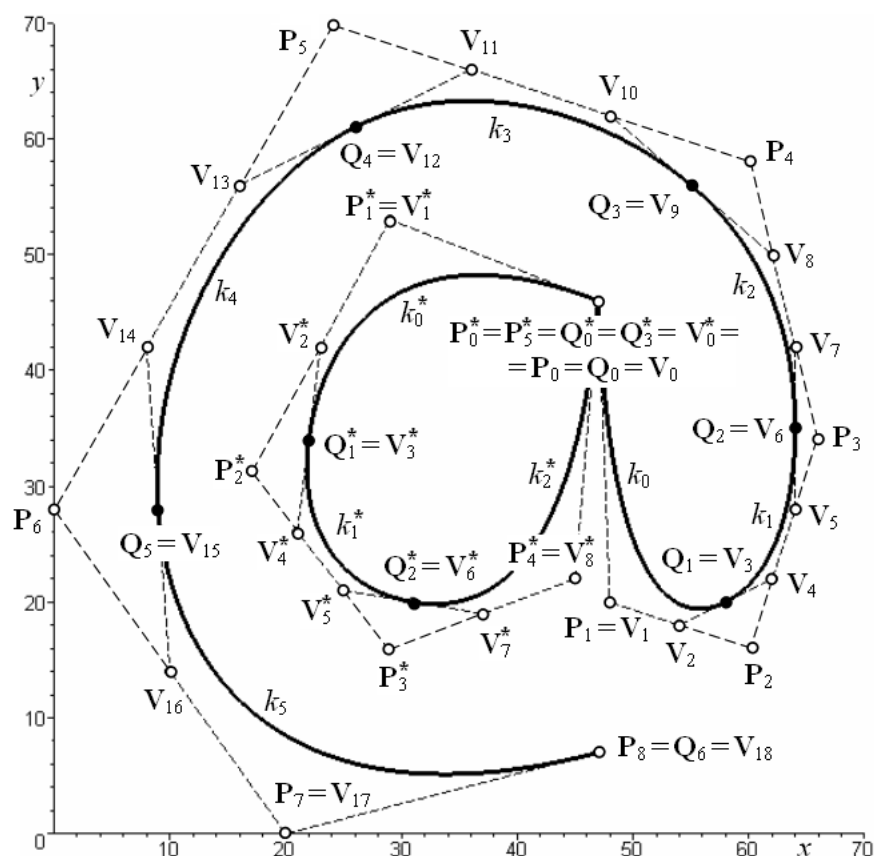


Figure 2.28: “@” sign from clamped curve

□

A set of suitable designed clamped curves are widely applied in engineering. It allows to create arbitrary free-form shapes as is shown in Fig. 2.29. Here, the set of clamped curves is used for handwritten-like letters.

■ **Exercise 2.26** Control polygons of clamped curves depicted in Fig. 2.26 are as follows:

- $\mathbf{P}_0 = (1, 1), \mathbf{P}_1 = (3, 1), \mathbf{P}_2 = (5, 7), \mathbf{P}_3 = (7, 1), \mathbf{P}_4 = (9, 4);$
- $\mathbf{P}_0 = (4, 1), \mathbf{P}_1 = (1, 3), \mathbf{P}_2 = (1, 7), \mathbf{P}_3 = (7, 7), \mathbf{P}_4 = (7, 3), \mathbf{P}_5 = (10, 1);$
- $\mathbf{P}_0 = (3, 1), \mathbf{P}_1 = (1, 3), \mathbf{P}_2 = (3, 7), \mathbf{P}_3 = (6, 1), \mathbf{P}_4 = (9, 7), \mathbf{P}_5 = (7, 7), \mathbf{P}_6 = (6, 5);$
- $\mathbf{P}_0 = (4, 5), \mathbf{P}_1 = (3, 7), \mathbf{P}_2 = (1, 7), \mathbf{P}_3 = (4, 1), \mathbf{P}_4 = (7, 7), \mathbf{P}_5 = (10, 1), \mathbf{P}_6 = (8, 1), \mathbf{P}_7 = (7, 3).$

Suppose that these clamped curves are created from Bézier cubic curves. Determine by construction and calculation the Cartesian coordinates of control points of all these Bézier cubic curves and find their vector equations. Using vector equations of Bézier cubic curves and verify C^2 continuity of clamped curves.

■ **Exercise 2.27** Suppose that clamped curves depicted in Fig. 2.26 (control polygons are given in Exercise 2.26) are Coons cubic B-splines. Construct control points of these Coons cubic B-splines. Find vector equations of individual Coons cubic curves of Coons cubic B-splines and compare them with the vector equations of Bézier cubic curves in Exercise 2.26.

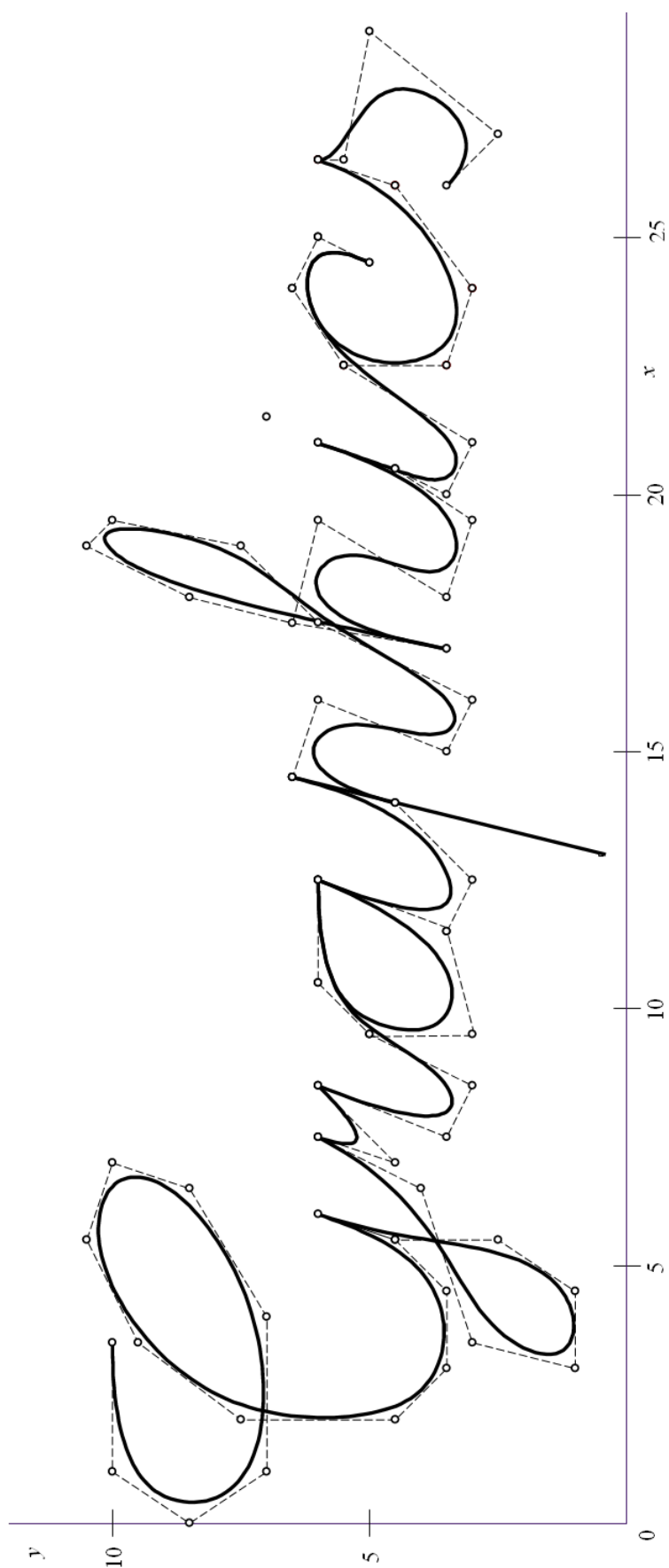


Figure 2.29: Clamped curves application

2.5.3 Clamped curve in Rhinoceros

Uniform clamped B-spline curve of p -th degree with $n + 1$ control points and $p < n$ is widely used in CAD/CAM systems.

Clamped curve of third degree – Command: *Control Point Curve* → Start of curve: type degree 3 in command prompt; enter all control points → press Enter.

Remarks:

1. The commands concerning control points and control polygon of clamped curve are the same as the commands used in the case of Bézier curve described in Section 2.2.5.
2. The rules for shape of clamped curve modification are the same as the rules used in the case of Bézier curve described in Section 2.2.5.

Knots of clamped curve – Command: *Multiple Points* → Location of point object: activate *Knot object snap*, possibly deactivate *Snap*; move cursor along clamped curve and click when information *Knot* is displayed. The points representing knots of clamped curve are drawn as individual entities.

Attention! Do not use commands *Insert Knot* and *Remove Knot* to modify the shape of clamped curve. These commands cause nonuniform parametrization of the curve. After this modification, the original curve is not a third degree uniform clamped curve anymore but nonuniform clamped curve (see [?], [2]). Do not use the command *Edit Control Point Weight*, either. This command causes rational parametrization of the curve (not polynomial). After this modification, the original curve is not clamped curve anymore, but, in general, a NURBS curve (see [?], [2]).

List of data structure of clamped curve – Command: *List* → Select objects to list: click on the clamped curve → press Enter. The dialog box *List* with detailed information about clamped curve is shown. For example, the useful information given in list of data structure of clamped curve k from Fig. 2.26 d) is:

- **order** = 4: the fourth order curve (i.e. the third degree curve),
- **cv_count** = 8: number of control points is 8,
- **index value**: list of Cartesian coordinates of control points ("CV" means Control Vertex – equal to control points):

index	value
CV[0]	(4,5,0)
CV[1]	(3,7,0)
CV[2]	(1,7,0)
CV[3]	(4,1,0)
CV[4]	(7,7,0)
CV[5]	(10,1,0)
CV[6]	(8,1,0)
CV[7]	(7,3,0)

- **Exercise 2.28** In Rhinoceros, draw control polygons of clamped curves depicted in Fig. 2.26 (control polygons are given in Exercise 2.26). Construct control points of individual Bézier curves creating clamped curves. Display *Curvature Graphs*.

Change the layer and draw clamped curves and their knots (*Knots Object Snap* is activated). Display *Curvature Graphs* of these clamped curves.

Compare the shape of Bézier cubic curves and clamped curves, knots positions and *Curvature Graphs*.

- **Exercise 2.29** Inspiring by Fig. 2.29, design your name in Rhinoceros.

Chapter 3

Surfaces modelling

Input data of a modelled free-form surface consists of a set of points or curves and geometrical conditions (constraints) which have to be satisfied at the given points or along the given curves. The given set of points is a so called *points mesh*, particularly a *control points mesh* (shortly *control mesh*) when modelling an *approximation* surface (resulting surface does not pass through the given points) or a *definition points mesh* (shortly *definition mesh*) when modelling an *interpolation* surface (the resulting surface passes through the given points). An interpolation surface can be given by a *set of definition curves*, too.

Surfaces of a more complicated shape are created by a huge number of surface elements, *patches*, which are joined according to given geometrical conditions to satisfy the required continuity along the whole surface. The joining of individual patches with the required order of continuity is called *patching*.

Here, the easiest models of free-form surfaces parameterized by polynomial functions and their patching will be discussed: a ruled surface (interpolation surface given by two boundary curves), a surface of hyperbolic paraboloid (interpolation surface given by four points – corners), Coons bilinear surface (interpolation surface given by four boundary curves) and Bézier surface (approximation surface given by a control points mesh).

3.1 Vector equation of a surface

Vector equation of a surface

$$\mathbf{P}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in [0, 1]^2$$

parameterized by polynomial functions is a linear combination of basis functions – polynomials of two variables $(u, v) \in [0, 1]^2$ (bivariate polynomials). Coefficients of individual basis functions are the given input data – definition points, boundary curves or control points. If the input data is the boundary curves of a patch, the basis functions are only univariate polynomials (analytical representation of a boundary curve is a univariate vector function itself).

The input data is ordered into a matrix \mathbf{M} called a *map of surface*. The elements $\mathbf{M}_{i,j}$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$ of the map have three coordinate components $\mathbf{M}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$. The concrete form of map elements depends on the chosen mathematical model of the modelled surface.

Bivariate basis function is a product of two univariate basis functions. Denote generally $H_i(u)$, $u \in [0, 1]$, $i = 0, 1, \dots, m$ basis functions of u parameter and $H_j(v)$, $v \in [0, 1]$, $j = 0, 1, \dots, n$ basis functions of v parameter. In the resulting linear combination, the element

$\mathbf{M}_{i,j}$ is a coefficient of bivariate basis function $H_{i,j}(u, v)$ created by the product of two univariate basis functions of corresponding subscripts

$$H_{i,j}(u, v) = H_i(u)H_j(v), \quad (u, v) \in [0, 1]^2, \quad i = 0, 1, \dots, m, \quad j = 0, 1, \dots, n. \quad (3.1)$$

This means that basis function $H_{i,j}(u, v)$ is *associated* with element $\mathbf{M}_{i,j}$.

In the case of mathematical models of free-form surfaces given by more points (curves, geometrical constrains), we will prefer the following matrix form of the vector equation of a surface

$$\begin{aligned} \mathbf{P}(u, v) &= \mathbf{H}(u) \cdot \mathbf{M} \cdot \mathbf{H}(v) = \\ &= (H_0(u), H_1(u), \dots, H_m(u)) \cdot \begin{pmatrix} \mathbf{M}_{0,0} & \mathbf{M}_{0,1} & \dots & \mathbf{M}_{0,n} \\ \mathbf{M}_{1,0} & \mathbf{M}_{1,1} & \dots & \mathbf{M}_{1,n} \\ \vdots & \vdots & & \vdots \\ \mathbf{M}_{m,0} & \mathbf{M}_{m,1} & \dots & \mathbf{M}_{m,n} \end{pmatrix} \cdot \begin{pmatrix} H_0(v) \\ H_1(v) \\ \vdots \\ H_n(v) \end{pmatrix}, \\ &\quad (u, v) \in [0, 1]^2. \end{aligned} \quad (3.2)$$

After multiplying the matrices in (3.2), we obtain the vector equation in the form of a linear combination of basis functions

$$\mathbf{P}(u, v) = H_0(u)H_0(v)\mathbf{M}_{0,0} + H_0(u)H_1(v)\mathbf{M}_{0,1} + \dots + H_m(u)H_n(v)\mathbf{M}_{m,n}, \quad (u, v) \in [0, 1]^2. \quad (3.3)$$

This form is useful for the mathematical model of surfaces that is given by less points (curves, geometrical constrains).

The vector equation of a surface can also be written in a form of two summations

$$\mathbf{P}(u, v) = \sum_{i=0}^m \sum_{j=0}^n H_i(u)H_j(v)\mathbf{M}_{i,j}, \quad (u, v) \in [0, 1]^2. \quad (3.4)$$

Here, this form will not be used.

All three forms of the vector equation of a surface – matrix form, linear combination of basis functions and expression by summation – are equivalent.

If the parametric expression of the surface is required, it is necessary to substitute the concrete coordinate components $x_{i,j}$, $y_{i,j}$, $z_{i,j}$ instead of $\mathbf{M}_{i,j}$ (in the matrix form):

$$\begin{aligned} x(u, v) &= (H_0(u), H_1(u), \dots, H_m(u)) \cdot \begin{pmatrix} x_{0,0} & x_{0,1} & \dots & x_{0,n} \\ x_{1,0} & x_{1,1} & \dots & x_{1,n} \\ \vdots & \vdots & & \vdots \\ x_{m,0} & x_{m,1} & \dots & x_{m,n} \end{pmatrix} \cdot \begin{pmatrix} H_0(v) \\ H_1(v) \\ \vdots \\ H_n(v) \end{pmatrix}, \quad (u, v) \in [0, 1]^2, \\ y(u, v) &= (H_0(u), H_1(u), \dots, H_m(u)) \cdot \begin{pmatrix} y_{0,0} & y_{0,1} & \dots & y_{0,n} \\ y_{1,0} & y_{1,1} & \dots & y_{1,n} \\ \vdots & \vdots & & \vdots \\ y_{m,0} & y_{m,1} & \dots & y_{m,n} \end{pmatrix} \cdot \begin{pmatrix} H_0(v) \\ H_1(v) \\ \vdots \\ H_n(v) \end{pmatrix}, \quad (u, v) \in [0, 1]^2, \\ z(u, v) &= (H_0(u), H_1(u), \dots, H_m(u)) \cdot \begin{pmatrix} z_{0,0} & z_{0,1} & \dots & z_{0,n} \\ z_{1,0} & z_{1,1} & \dots & z_{1,n} \\ \vdots & \vdots & & \vdots \\ z_{m,0} & z_{m,1} & \dots & z_{m,n} \end{pmatrix} \cdot \begin{pmatrix} H_0(v) \\ H_1(v) \\ \vdots \\ H_n(v) \end{pmatrix}, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

3.2 Ruled surface

A ruled surface interpolates two given curves which can be regarded as opposite boundaries of a patch. Analytical representation of the given curves are univariate vector functions with identical parametrization domain. We obtain the analytical representation of a ruled surface – bivariate vector function – by *linear interpolation* between the given boundaries. Then, the ruled surface is a set of straight line segments connecting the points of the same parameter value on the boundary curves.

The following short designation will be used for the *given* corners of a patch

$$\mathbf{P}_{0,0}, \mathbf{P}_{0,1}, \mathbf{P}_{1,0}, \mathbf{P}_{1,1},$$

where

$$\mathbf{P}_{0,0} = \mathbf{P}(0, 0), \mathbf{P}_{0,1} = \mathbf{P}(0, 1), \mathbf{P}_{1,0} = \mathbf{P}(1, 0), \mathbf{P}_{1,1} = \mathbf{P}(1, 1).$$

Similarly, the *given* boundaries of a patch will be designated by

$$\mathbf{P}_0(u), \mathbf{P}_1(u), \mathbf{P}_0(v), \mathbf{P}_1(v),$$

where

$$\mathbf{P}_0(u) = \mathbf{P}(u, 0), \mathbf{P}_1(u) = \mathbf{P}(u, 1), \mathbf{P}_0(v) = \mathbf{P}(0, v), \mathbf{P}_1(v) = \mathbf{P}(1, v).$$

■ **Definition 3.1 – Ruled surface.** Let $\mathbf{P}_0(u)$ and $\mathbf{P}_1(u)$, $u \in [0, 1]$ be the given boundaries in u -direction, each of them created by one curve segment. Then, the vector equation of a ruled surface given by boundaries in u -direction is

$$\mathbf{P}(u, v) = (1 - v)\mathbf{P}_0(u) + v\mathbf{P}_1(u), \quad (u, v) \in [0, 1]^2. \quad (3.5)$$

Let $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$, $v \in [0, 1]$ be the given boundaries in v -direction, each of them created by one curve segment. Then, the vector equation of a ruled surface given by boundaries in v -direction is

$$\mathbf{P}(u, v) = (1 - u)\mathbf{P}_0(v) + u\mathbf{P}_1(v), \quad (u, v) \in [0, 1]^2. \quad (3.6)$$

If the given boundaries of a ruled surface are created by one curve segment, the ruled surface is created by one patch. \square

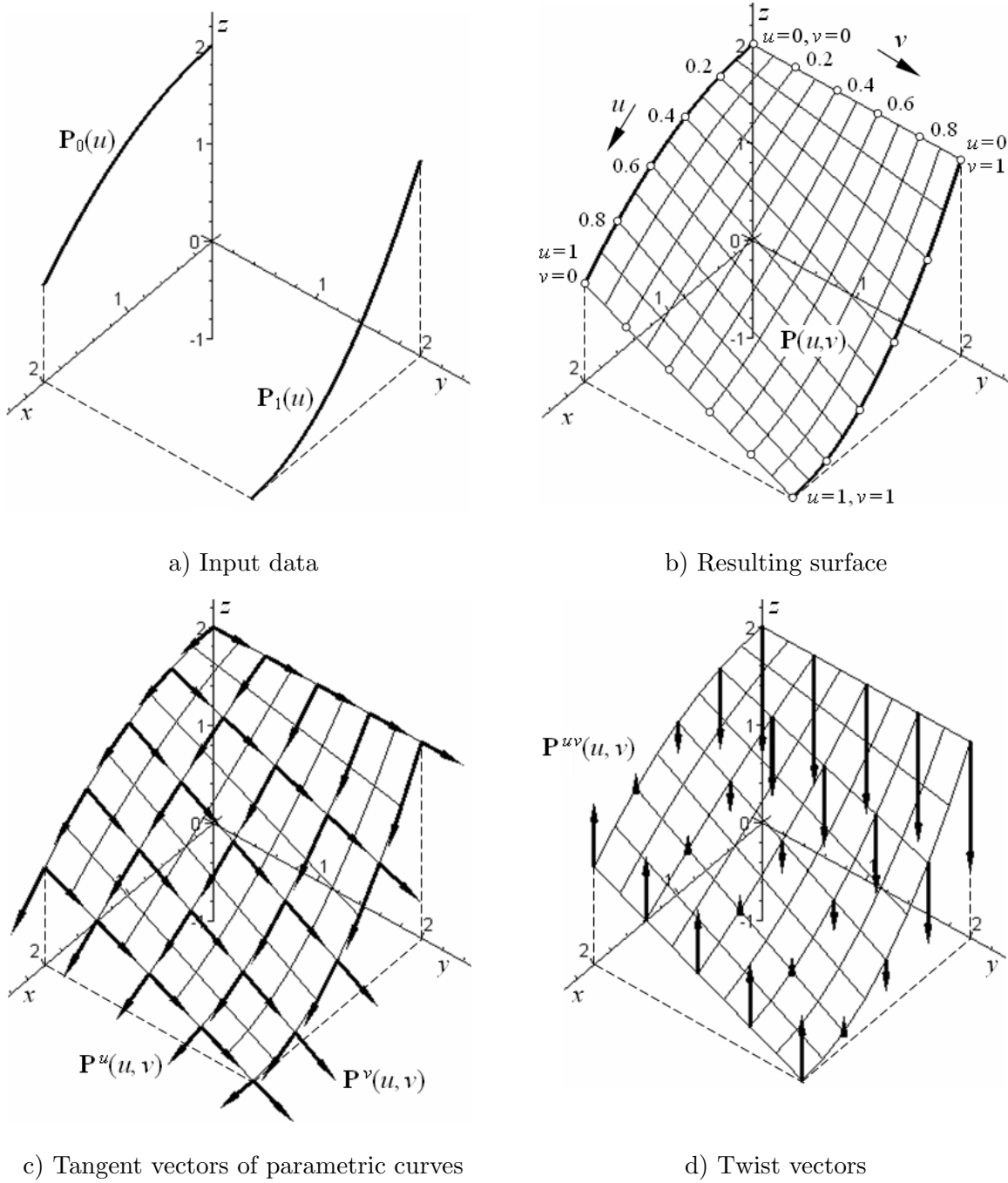
In (3.5), the given boundaries are functions of u variable, therefore, the basis functions are linear polynomials of v variable: $1 - v$ and v . In (3.6), the given boundaries are functions of v variable, therefore, the basis functions are linear polynomials of u variable: $1 - u$, u .

■ **Example 3.1 – Ruled surface given by boundaries in u -direction.** Boundaries $\mathbf{P}_0(u) = (2u, 0, -u^2 + 2)$ and $\mathbf{P}_1(u) = (2u, 2, 2u^2 - 4u + 2)$, $u \in [0, 1]$ are given.

Find parametric expression and vector equation of ruled surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vector $\mathbf{P}^{uv}(u, v)$. Substitute parameter values $u, v = 0, 1$ in all vector equations and determine the elements which you obtain.

Solution. We obtain the coordinate functions $x(u, v)$, $y(u, v)$ and $z(u, v)$ of the surface $\mathbf{P}(u, v)$ when substituting the corresponding coordinate functions of the given boundaries in (3.5):

$$\begin{aligned} x(u, v) &= (1 - v) \cdot 2u + v \cdot 2u = 2u, \\ y(u, v) &= (1 - v) \cdot 0 + v \cdot 2 = 2v, \\ z(u, v) &= (1 - v)(-u^2 + 2) + v(2u^2 - 4u + 2) = 3u^2v - u^2 - 4uv + 2, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Figure 3.1: Ruled surface given by boundaries in u -direction

Vector equations of ruled surface, tangent vectors of parametric curves and twist vectors are

$$\mathbf{P}(u, v) = (2u, 2v, 3u^2v - u^2 - 4uv + 2), \quad (u, v) \in [0, 1]^2, \quad (3.7)$$

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, 6uv - 2u - 4v), \quad (u, v) \in [0, 1]^2, \quad (3.8)$$

$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 2, 3u^2 - 4u), \quad (u, v) \in [0, 1]^2, \quad (3.9)$$

$$\mathbf{P}^{uv}(u, v) = \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = (0, 0, 6u - 4), \quad (u, v) \in [0, 1]^2. \quad (3.10)$$

After substitution $u = 0$, $u = 1$, $v = 0$ and $v = 1$ in (3.7), we obtain the boundaries of the patch

$$\begin{aligned} \mathbf{P}(u, 0) &= (2u, 0, -u^2 + 2), \quad u \in [0, 1] \dots\dots\dots && \text{the given boundary } \mathbf{P}_0(u), \\ \mathbf{P}(u, 1) &= (2u, 2, 2u^2 - 4u + 2), \quad u \in [0, 1] \dots\dots\dots && \text{the given boundary } \mathbf{P}_1(u), \\ \mathbf{P}(0, v) &= (0, 2v, 2), \quad v \in [0, 1] \dots\dots\dots && \text{straight line segment in plane } x = 0, \\ \mathbf{P}(1, v) &= (2, 2v, -v + 1), \quad v \in [0, 1] \dots\dots\dots && \text{straight line segment in plane } x = 2, \end{aligned}$$

After substitution $u, v = 0, 1$ in (3.7), (3.8), (3.9) and (3.10), we obtain the corners of the patch, tangent vectors of parametric curves and twist vectors at corner points of the patch, see the following table.

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	(0, 0, 2)	(0, 2, 2)	(2, 0, 1)	(2, 2, 0)
$\mathbf{P}^u(u, v)$	(2, 0, 0)	(2, 0, -4)	(2, 0, -2)	(2, 0, 0)
$\mathbf{P}^v(u, v)$	(0, 2, 0)	(0, 2, 0)	(0, 2, -1)	(0, 2, -1)
$\mathbf{P}^{uv}(u, v)$	(0, 0, -4)	(0, 0, -4)	(0, 0, 2)	(0, 0, 2)

The given boundaries are drawn in Fig. 3.1 a), the resulting ruled surface (3.7) is drawn in Fig. 3.1 b). Tangent vectors (3.8) and (3.9) of parametric curves at several points of ruled surface (3.7) are drawn in Fig. 3.1 c). Twist vectors (3.10) at the same points of ruled surface (3.7) are drawn Fig. 3.1 d). To keep the readability of the picture, the length of the drawn vectors is shortened (tangent vectors are drawn in the scale 1:5, twist vectors in the scale 1:3). \square

■ **Example 3.2 – Ruled surface given by boundaries in v -direction.** The boundaries $\mathbf{P}_0(v) = (0, 2v, 4v^2 - 4v + 2)$ and $\mathbf{P}_1(v) = (2, 2v, v^2 - 2v + 1)$, $v \in [0, 1]$, are given.

Find parametric expression and vector equation of ruled surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vector $\mathbf{P}^{uv}(u, v)$. Substitute parameter values $u, v = 0, 1$ in all vector equations and determine the elements which you obtain.

Solution. We obtain the coordinate functions $x(u, v)$, $y(u, v)$ and $z(u, v)$ of the surface $\mathbf{P}(u, v)$ when substituting the corresponding coordinate functions of the given boundaries in (3.6):

$$\begin{aligned} x(u, v) &= (1 - u) \cdot 0 + u \cdot 2 = 2u, \\ y(u, v) &= (1 - u) \cdot 2v + u \cdot 2v = 2v, \\ z(u, v) &= (1 - u)(4v^2 - 4v + 2) + u(v^2 - 2v + 1) = \\ &= -3uv^2 + 2uv - u + 4v^2 - 4v + 2, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

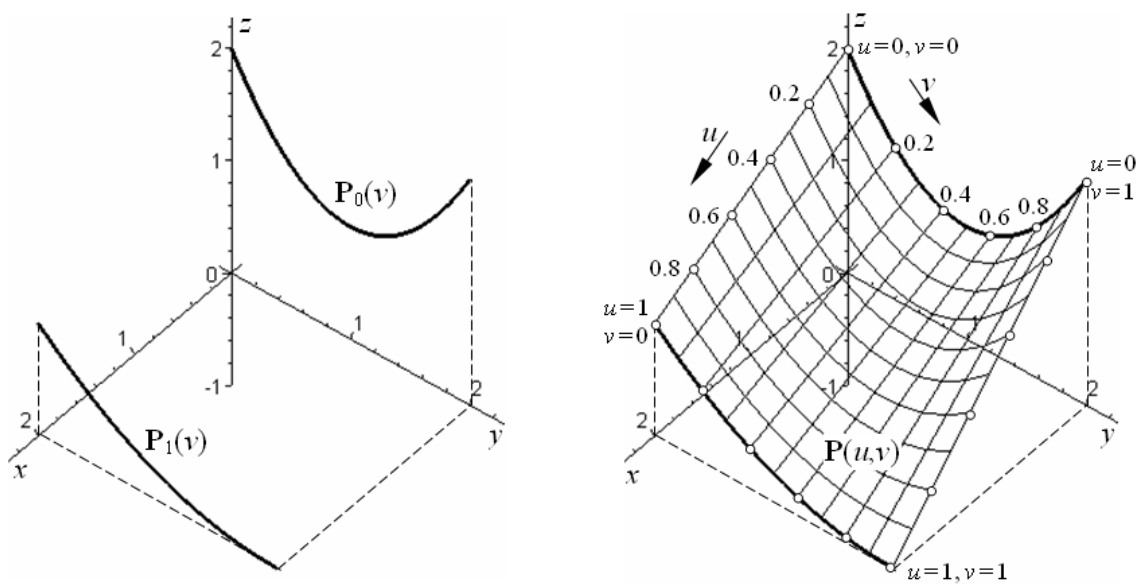
Vector equations of the ruled surface, tangent vectors of parametric curves and twist vectors are

$$\mathbf{P}(u, v) = (2u, 2v, -3uv^2 + 2uv - u + 4v^2 - 4v + 2), \quad (u, v) \in [0, 1]^2, \quad (3.11)$$

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, -3v^2 + 2v - 1), \quad (u, v) \in [0, 1]^2, \quad (3.12)$$

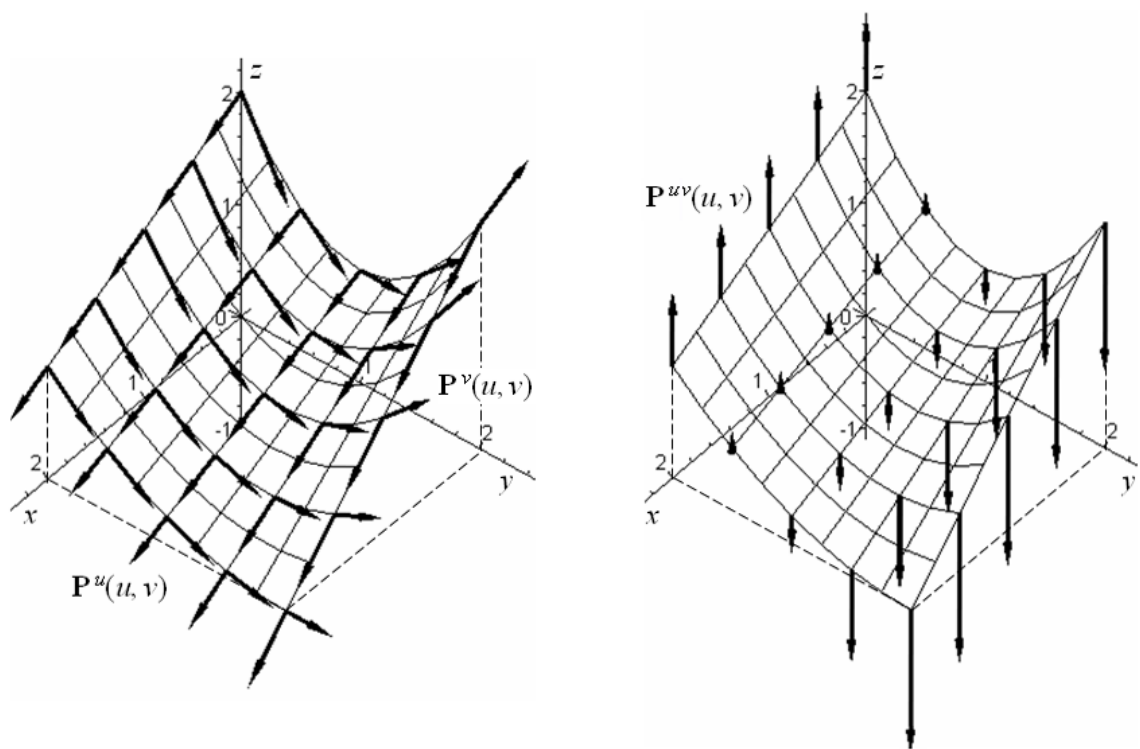
$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 2, -6uv + 2u + 8v - 4), \quad (u, v) \in [0, 1]^2, \quad (3.13)$$

$$\mathbf{P}^{uv}(u, v) = \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = (0, 0, -6v + 2), \quad (u, v) \in [0, 1]^2. \quad (3.14)$$



a) Input data

b) Resulting surface



c) Tangent vectors of parametric curves

d) Twist vectors

Figure 3.2: Ruled surface given by boundaries in v direction

After substitution $u = 0, u = 1, v = 0$ and $v = 1$ in (3.11), we obtain the boundaries of the patch

$$\begin{aligned}
 \mathbf{P}(u, 0) &= (2u, 0, -u + 2), \quad u \in [0, 1] \quad \dots\dots\dots && \text{straight line segment in plane } y = 0, \\
 \mathbf{P}(u, 1) &= (2u, 2, -2u + 2), \quad u \in [0, 1] \quad \dots\dots\dots && \text{straight line segment in plane } y = 2, \\
 \mathbf{P}(0, v) &= (0, 2v, 4v^2 - 4v + 2), \quad v \in [0, 1] \quad \dots\dots && \text{the given boundary } \mathbf{P}_0(v), \\
 \mathbf{P}(1, v) &= (2, 2v, v^2 - 2v + 1), \quad v \in [0, 1] \quad \dots\dots && \text{the given boundary } \mathbf{P}_1(v).
 \end{aligned}$$

After substitution $u, v = 0, 1$ in (3.11), (3.12), (3.13) and (3.14), we obtain the corners of the patch and tangent and twist vectors at the corners of the patch, see the following table.

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	$(0, 0, 2)$	$(0, 2, 2)$	$(2, 0, 1)$	$(2, 2, 0)$
$\mathbf{P}^u(u, v)$	$(2, 0, -1)$	$(2, 0, -2)$	$(2, 0, -1)$	$(2, 0, -2)$
$\mathbf{P}^v(u, v)$	$(0, 2, -4)$	$(0, 2, 4)$	$(0, 2, -2)$	$(0, 2, 0)$
$\mathbf{P}^{uv}(u, v)$	$(0, 0, 2)$	$(0, 0, -4)$	$(0, 0, 2)$	$(0, 0, -4)$

The given boundaries are drawn in Fig. 3.2 a), the resulting ruled surface (3.11) is drawn in Fig. 3.2 b). Tangent vectors (3.12) and (3.13) of parametric curves at several points of ruled surface (3.11) are drawn in Fig. 3.2 c). Twist vectors (3.14) at the same points are drawn in Fig. 3.2 d) (tangent vectors are drawn in the scale 1:5, twist vectors in the scale 1:3). \square

3.2.1 Properties of ruled surface

The important geometrical properties of ruled surface follow directly from definition 3.1 and from pictures of ruled surfaces drawn in examples 3.1 and 3.2:

- A ruled surface interpolates the given boundary curves.
- The boundaries in v -direction of ruled surface given by boundaries in u -direction are straight line segments.
- The boundaries in u -direction of ruled surface given by boundaries in v -direction are straight line segments.

3.2.2 Patching – patches from ruled surface

When joining patches from ruled surface, a continuity of a resulted surface is given by continuity of boundaries in u - and v -directions. Thus, if two patches from ruled surface are given by boundaries in u direction, it is possible to satisfy only C^0 continuity in v -direction. The continuity in u -direction depends on the type of coordinate functions of the given boundaries in u -direction. Similarly, if two patches from ruled surface are given by boundaries in v -direction, it is possible to satisfy only C^0 continuous patching in u -direction. The continuity in v -direction depends on the type of coordinate functions of the given boundaries in v -direction. It follows directly from the definition of ruled surface.

■ **Exercise 3.1** The following boundaries are given

$$\mathbf{P}_0(u) = (3u, 0, 7u^3 - 12u^2 + 3u + 2), \quad \mathbf{P}_1(u) = (3u, 3, 3u^3 - 9u^2 + 6u), \quad u \in [0, 1].$$

Find parametric expression and vector equation of ruled surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vector $\mathbf{P}^{uv}(u, v)$. For parameter values $u, v = 0, 1$, determine corner points and boundaries of the patch and tangent and twist vectors at corner points of the patch.

■ **Exercise 3.2** The following boundaries are given

$$\mathbf{P}_0(v) = (0, 3v, -2v^3 + 2), \mathbf{P}_1(v) = (3, 3v, -3v^3 + 3v^2), v \in [0, 1].$$

Find parametric expression and vector equation of ruled surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vector $\mathbf{P}^{uv}(u, v)$. For parameters values $u, v = 0, 1$, determine corner points and boundaries of the patch and tangent and twist vectors at the corner points of the patch.

3.2.3 Ruled surface in Rhinoceros

Before creating a ruled surface, two boundaries – planar or spatial curves – have to be drawn.

Ruled surface – Command: *Surface from 2, 3 or 4 Edge Curves* → Select 2, 3 or 4 curves: click on both boundaries → press Enter. The ruled surface determined by the given boundaries is drawn.

Remarks:

1. The orientation of curves representing the given boundaries of ruled surface has to be the same and, at the same time, it is necessary to click near to corresponding endpoints of boundaries when selecting 2, 3 or 4 curves. Otherwise, a wrongly twisted surface is drawn.
2. The direction and orientation of surface of ruled surface is given by system. If necessary, it is possible to modify it by command *Direction*, see 1.6.

Point $\mathbf{P}(\alpha, \beta)$ on ruled surface – Command: *Point from UV coordinates* → Select surface to evaluate: choose *CreatePoint=Yes* and *Normalized=Yes* in command prompt → click on the ruled surface → Enter U value between 0.0 and 1.0 → type α in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type β in command prompt → press Enter → press Enter. The point with parametric coordinates $u = \alpha$ and $v = \beta$ on the ruled surface is drawn, see example 3.3.

Parametric curves of ruled surface – Command: *Extract Isocurve* → Select surface for isocurve extraction: click on ruled surface; the marker moves only along the surface and the parametric curve corresponding to its actual position is dynamically displayed → Select isocurve to extract: choose in command prompt *Direction=U*, *Toggle Direction=V* or *Direction=Both* → click to place the required parametric curve or the pair of parametric curves → press Enter. Depending on the chosen option in the command prompt, the parametric u -curve, parametric v -curve or both parametric curves as individual entities are drawn.

■ **Example 3.3 Ruled surface in Rhinoceros.** Control polygons of two spatial cubic Bézier curves $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$, $v \in [0, 1]$ are given. Bézier curves $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$ are boundaries of ruled surface.

In Rhinoceros, draw cubic Bézier curves $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$. Using de Casteljau algorithm, construct points for $v = \frac{1}{2}$. Create ruled surface $\mathbf{P}(u, v)$ given by boundaries $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$. Draw points on ruled surface for parameter values $u = 0, v = 0, 0.2, \dots, 1$ and $v = 0, u = 0, 0.2, \dots, 1$. Draw parametric curves passing through these points.

Consider the following control polygons of cubic Bézier curves $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$:

- a) $\mathbf{P}_0(v)$: $\mathbf{V}_0 = (1, 0, 1)$, $\mathbf{V}_1 = (2, 1, 1)$, $\mathbf{V}_2 = (3, 0, 0)$, $\mathbf{V}_3 = (4, 1, 0)$,
 $\mathbf{P}_1(v)$: $\mathbf{W}_0 = (0, 3, 2)$, $\mathbf{W}_1 = (0, 4, 1)$, $\mathbf{W}_2 = (2, 3, 0)$, $\mathbf{W}_3 = (3, 4, -1)$,
- b) $\mathbf{P}_0(v)$: $\mathbf{V}_0 = (-1, 4, 1)$, $\mathbf{V}_1 = (-2, 2, 2)$, $\mathbf{V}_2 = (-1, 1, -1)$, $\mathbf{V}_3 = (-1, 0, 2)$,
 $\mathbf{P}_1(v)$: $\mathbf{W}_0 = (3, 0, 3)$, $\mathbf{W}_1 = (5, 1, 0)$, $\mathbf{W}_2 = (4, 3, -1)$, $\mathbf{W}_3 = (2, 1, -1)$.

Solution.

a) Ruled surface $\mathbf{P}(u, v)$, points and parametric curves are drawn in Fig. 3.3 a).

b) Ruled surface $\mathbf{P}(u, v)$, points and parametric curves are drawn in Fig. 3.3 b).

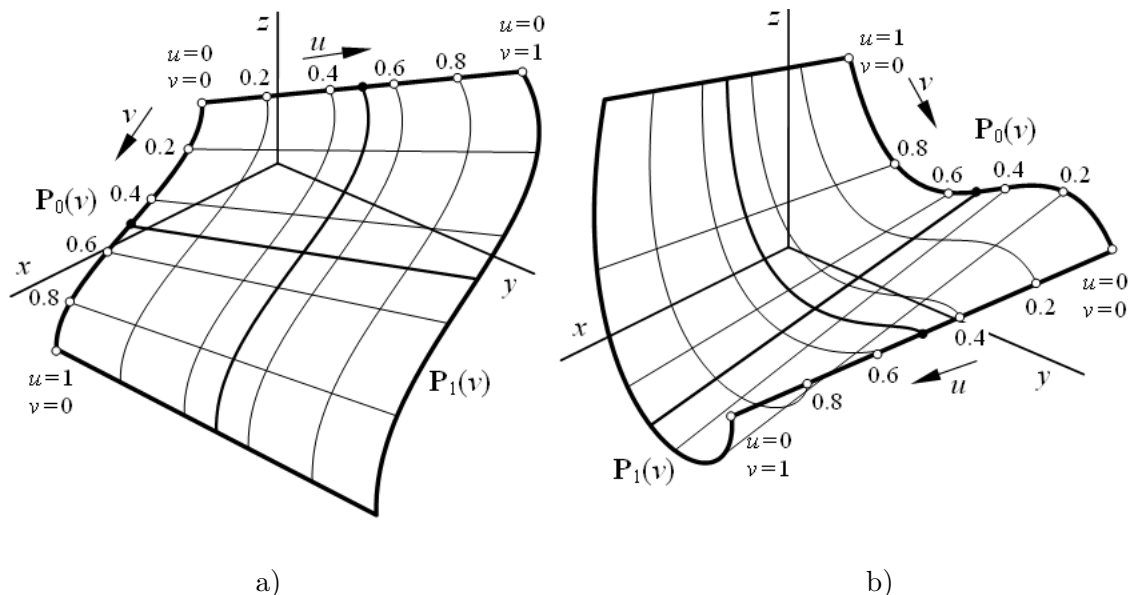


Figure 3.3: Ruled surface in Rhinoceros

Remarks:

1. The parametric curves drawn according to the task setting are drawn by thin lines, the parametric curves automatically displayed for parameter values $u, v = \frac{1}{2}$ are drawn by thicker lines and the given boundaries are drawn by thick lines.
2. Points on boundaries constructed by de Casteljau algorithm are designated by symbol \bullet .
3. To keep readability of the pictures, neither the control polygons nor constructions needed for de Casteljau algorithm are drawn. \square

3.3 Surface of hyperbolic paraboloid

A surface of hyperbolic paraboloid interpolates four given points – corner points of a patch. We obtain the analytical representation of surface of hyperbolic paraboloid – bivariate vector function – by *bilinear interpolation* of the given points.

■ **Definition 3.2 – Surface of hyperbolic paraboloid.** Given four corners of the patch $\mathbf{P}_{0,0}$, $\mathbf{P}_{0,1}$, $\mathbf{P}_{1,0}$ and $\mathbf{P}_{1,1}$, vector equation of *surface of hyperbolic paraboloid* given by these corners is

$$\mathbf{P}(u, v) = (1 - u)(1 - v)\mathbf{P}_{0,0} + (1 - u)v\mathbf{P}_{0,1} + u(1 - v)\mathbf{P}_{1,0} + uv\mathbf{P}_{1,1}, \quad (u, v) \in [0, 1]^2. \quad (3.15)$$

Basis functions are bivariate polynomials – products of two univariate linear polynomials. □

■ **Example 3.4 – Surface of hyperbolic paraboloid.** The following corners of the patch are given $\mathbf{P}_{0,0} = (0, 0, 2)$, $\mathbf{P}_{0,1} = (0, 2, 2)$, $\mathbf{P}_{1,0} = (2, 0, 1)$ and $\mathbf{P}_{1,1} = (2, 2, 0)$.

Find parametric expression and vector equation of surface of hyperbolic paraboloid $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vectors $\mathbf{P}^{uv}(u, v)$. Substitute parameters values $u, v = 0, 1$ in all vector equations and determine the elements of the surface which you obtain.

Solution. We obtain the coordinate functions $x(u, v)$, $y(u, v)$ and $z(u, v)$ of the surface $\mathbf{P}(u, v)$ when substituting the corresponding coordinates of the given corners in (3.15):

$$\begin{aligned} x(u, v) &= (1 - u)(1 - v) \cdot 0 + (1 - u)v \cdot 0 + u(1 - v) \cdot 2 + uv \cdot 2 = 2u, \\ y(u, v) &= (1 - u)(1 - v) \cdot 0 + (1 - u)v \cdot 2 + u(1 - v) \cdot 0 + uv \cdot 2 = 2v, \\ z(u, v) &= (1 - u)(1 - v) \cdot 2 + (1 - u)v \cdot 2 + u(1 - v) \cdot 1 + uv \cdot 0 = \\ &= -uv - u + 2, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Vector equations of the surface of hyperbolic paraboloid, tangent vectors of parametric curves and twist vectors are

$$\mathbf{P}(u, v) = (2u, 2v, -uv - u + 2), \quad (u, v) \in [0, 1]^2, \quad (3.16)$$

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, -v - 1), \quad (u, v) \in [0, 1]^2, \quad (3.17)$$

$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 2, -u), \quad (u, v) \in [0, 1]^2, \quad (3.18)$$

$$\mathbf{P}^{uv}(u, v) = \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = (0, 0, -1), \quad (u, v) \in [0, 1]^2. \quad (3.19)$$

After substitution $u, v = 0, 1$ in (3.16), we obtain the corners of the patch

$$\begin{aligned} \mathbf{P}(0, 0) &= (0, 0, 2) \dots \text{the given corner } \mathbf{P}_{0,0}, \\ \mathbf{P}(0, 1) &= (0, 2, 2) \dots \text{the given corner } \mathbf{P}_{0,1}, \\ \mathbf{P}(1, 0) &= (2, 0, 1) \dots \text{the given corner } \mathbf{P}_{1,0}, \\ \mathbf{P}(1, 1) &= (2, 2, 0) \dots \text{the given corner } \mathbf{P}_{1,1}. \end{aligned}$$

After substitution $u = 0$, $u = 1$, $v = 0$ and $v = 1$ in (3.16), we obtain boundaries of the patch

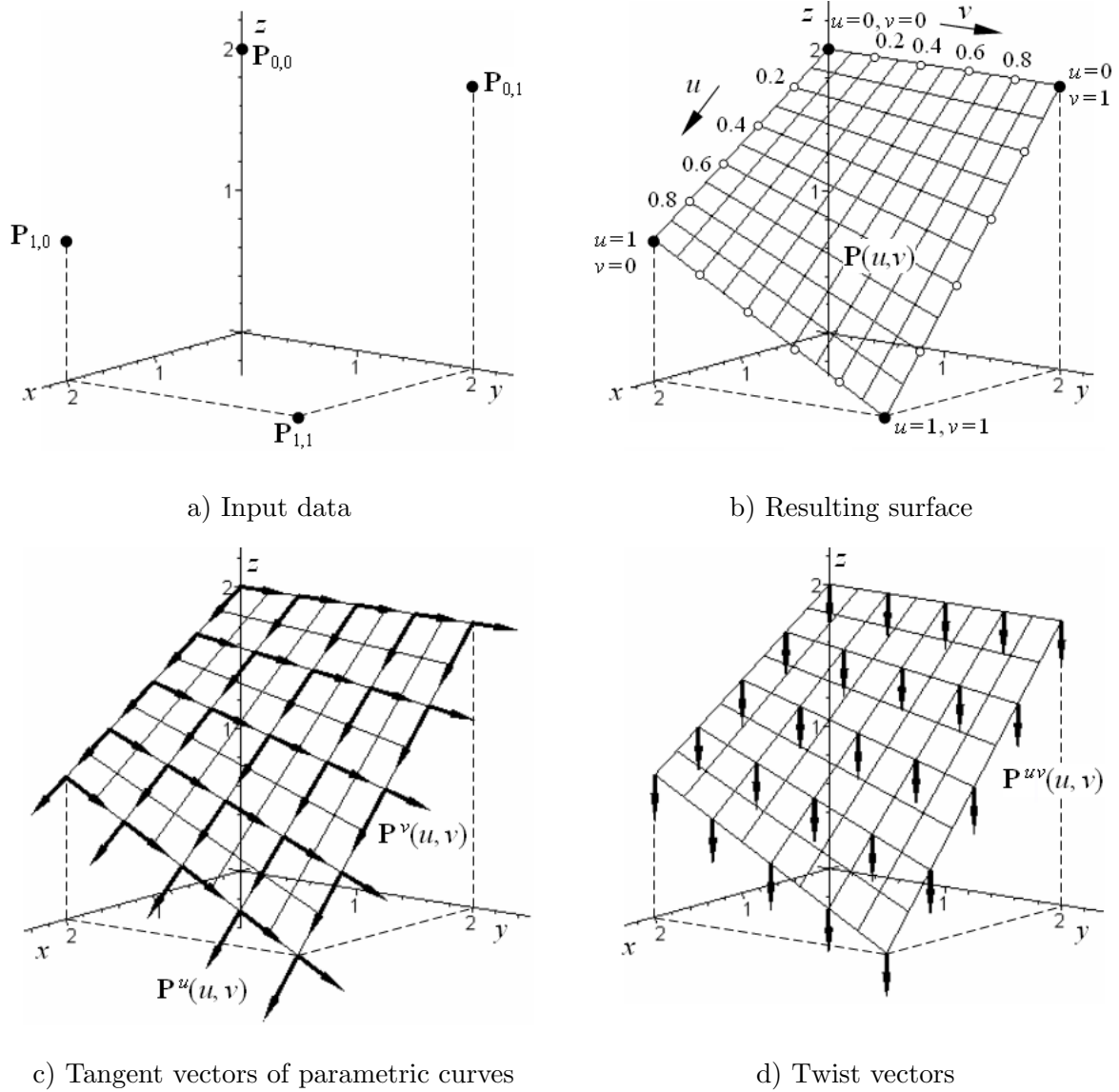


Figure 3.4: Surface of hyperbolic paraboloid

$\mathbf{P}(u, 0) = (2u, 0, -u + 2), u \in [0, 1]$	straight line segment in plane $y = 0$,
$\mathbf{P}(u, 1) = (2u, 2, -2u + 2), u \in [0, 1]$	straight line segment in plane $y = 2$,
$\mathbf{P}(0, v) = (0, 2v, 2), v \in [0, 1]$	straight line segment in plane $x = 0$,
$\mathbf{P}(1, v) = (2, 2v, -v + 1), v \in [0, 1]$	straight line segment in plane $x = 2$.

After substitution $u, v = 0, 1$ in (3.17), (3.18) and (3.19), we obtain tangent and twist vectors at the corners of the patch, see the following table (for completeness, the corners of the patch obtaining by substitution $u, v = 0, 1$ in (3.16) are repeated in this table, too). Twist vectors are constant along the whole patch, see (3.19). From geometrical point of view, the constant twist vectors mean that the deviation of the surface from the tangent plane at each point is the same.

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	$(0, 0, 2)$	$(0, 2, 2)$	$(2, 0, 1)$	$(2, 2, 0)$
$\mathbf{P}^u(u, v)$	$(2, 0, -1)$	$(2, 0, -2)$	$(2, 0, -1)$	$(2, 0, -2)$
$\mathbf{P}^v(u, v)$	$(0, 2, 0)$	$(0, 2, 0)$	$(0, 2, -1)$	$(0, 2, -1)$
$\mathbf{P}^{uv}(u, v)$	$(0, 0, -1)$	$(0, 0, -1)$	$(0, 0, -1)$	$(0, 0, -1)$

The given corners of the patch are drawn in Fig. 3.4 a), the resulting surface of hyperbolic paraboloid (3.16) is drawn in Fig. 3.4 b). Tangent vectors (??) of parametric curves at several points of the surface of hyperbolic paraboloid (3.16) are drawn in Fig. 3.4 c), twist vectors (3.19) at the same points are drawn in Fig. 3.4 d) (tangent vector are drawn in the scale 1:5, twist vectors are drawn in the scale 1:3). \square

3.3.1 Properties of surface of hyperbolic paraboloid

Considering the definition of surface of hyperbolic paraboloid as linear interpolation among four points, the following geometrical properties of surface of hyperbolic paraboloid are obvious.

- Surface of hyperbolic paraboloid interpolates the given corners of the patch.
- Parametric curves in both directions are straight line segments. Therefore, all four boundaries are straight line segments as well.
- If the given boundaries of ruled surface are straight line segments, the ruled surface and surface of hyperbolic paraboloid given by endpoints of these straight line segments are identical.

3.3.2 Patching – patches from surface of hyperbolic paraboloid

Surface of hyperbolic paraboloid is not useful for modelling the surface of a complicated shape given by a mesh of definition points because it allows only C^0 continuous patching in both directions.

■ **Exercise 3.3** The following corners of the patch are given

$$\mathbf{P}_{0,0} = (0, 0, 2), \mathbf{P}_{0,1} = (0, 3, 0), \mathbf{P}_{1,0} = (3, 0, 0) \text{ and } \mathbf{P}_{1,1} = (3, 3, 0).$$

Find parametric expression and vector equation of surface of hyperbolic paraboloid $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$, $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vectors $\mathbf{P}^{uv}(u, v)$. For parameter values $u, v = 0, 1$ determine corners and boundaries of the patch and tangent and twist vectors at the corners of the patch.

3.3.3 Surface of hyperbolic paraboloid in Rhinoceros

Surface of hyperbolic paraboloid – Command: *Surface from 3 or 4 corner points* → First corner of surface: → enter first corner → Second corner of surface: → enter second corner → Third corner of surface: → enter third corner → Fourth corner of surface: → enter fourth corner. The surface of hyperbolic paraboloid is drawn.

Remarks:

1. The corners have to be entered clockwise ($\mathbf{P}_{0,0}$, $\mathbf{P}_{0,1}$, $\mathbf{P}_{1,1}$ and $\mathbf{P}_{1,0}$) or anticlockwise ($\mathbf{P}_{0,0}$, $\mathbf{P}_{1,0}$, $\mathbf{P}_{1,1}$ and $\mathbf{P}_{0,1}$). Otherwise, a wrongly twisted surface is drawn.

2. The direction and orientation of surface of hyperbolic paraboloid is given by system. If necessary, it is possible to modify it by command *Direction*, see 1.6.

Point $\mathbf{P}(\alpha, \beta)$ on surface of hyperbolic paraboloid – Command: *Point from UV coordinates* → Select surface to evaluate: choose *CreatePoint=Yes* and *Normalized=Yes* in command prompt → click on the surface of hyperbolic paraboloid → Enter U value between 0.0 and 1.0 → type α in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type β in command prompt → press Enter → press Enter. The point with parametric coordinates $u = \alpha$ and $v = \beta$ on the surface of hyperbolic paraboloid is drawn, see example 3.3.

Parametric curves of surface of hyperbolic paraboloid – Command: *Extract Isocurve* → Select surface for isocurve extraction: click on surface of hyperbolic paraboloid; the marker moves only along the surface and the parametric curve corresponding to its actual position is dynamically displayed → Select isocurve to extract: choose in command prompt *Direction=U*, *Toggle Direction=V* or *Direction=Both* → click to place the required parametric curve or the pair of parametric curves → press Enter. Depending on the chosen option in the command prompt, the parametric u -curve, parametric v -curve or both parametric curves as individual entities are drawn.

- **Exercise 3.4** In Rhinoceros, create the surfaces of hyperbolic paraboloid from example 3.4 and from exercise 3.3. Draw points on surfaces of hyperbolic paraboloid for parameter values $u = 0, v = 0, 0.2, \dots, 1$ and $v = 0, u = 0, 0.2, \dots, 1$. Draw parametric curves passing through these points.

3.4 Coons bilinear surface

Coons bilinear surface is an interpolation surface created by one patch. This patch interpolates four given boundaries with common endpoints at the corners of the patch. Considering this condition, the corners of the patch belong to input data, even though, their coordinates are practically obtained by substitution $u, v = 0, 1$ in vector equations of the given boundaries.

- **Definition 3.3 – Coons bilinear surface.** Given boundaries $\mathbf{P}_0(u)$, $\mathbf{P}_1(u)$, $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$ with common points at the corners $\mathbf{P}_{0,0}$, $\mathbf{P}_{0,1}$, $\mathbf{P}_{1,0}$ and $\mathbf{P}_{1,1}$ of the patch, vector equation of *Coons bilinear surface* is

$$\mathbf{P}(u, v) = (1 - u, 1, u) \cdot \begin{pmatrix} -\mathbf{P}_{0,0} & \mathbf{P}_0(v) & -\mathbf{P}_{0,1} \\ \mathbf{P}_0(u) & \mathbf{0} & \mathbf{P}_1(u) \\ -\mathbf{P}_{1,0} & \mathbf{P}_1(v) & -\mathbf{P}_{1,1} \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix}, \quad (u, v) \in [0, 1]^2. \quad (3.20)$$

□

- **Example 3.5 – Coons bilinear surface.** The following boundaries are given:

$$\mathbf{P}_0(u) = (2u, 0, -u^2 + 2), \quad \mathbf{P}_1(u) = (2u, 2, 2u^2 - 4u + 2), \quad u \in [0, 1],$$

$$\mathbf{P}_0(v) = (0, 2v, 4v^2 - 4v + 2), \quad \mathbf{P}_1(v) = (2, 2v, v^2 - 2v + 1), \quad v \in [0, 1].$$

Find parametric expression and vector equation of Coons bilinear surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vectors $\mathbf{P}^{uv}(u, v)$. Substitute parameters values $u, v = 0, 1$ in all vector equations and determine the elements of the surface which you obtain.

Solution. First of all, we calculate the coordinates of the corners of the patch. There are two possibilities: we either substitute $u = 0, 1$ in $\mathbf{P}_0(u)$ and $\mathbf{P}_1(u)$ or $v = 0, 1$ in $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$. In both cases, we obtain the following corners of the patch

$$\mathbf{P}_{0,0} = (0, 0, 2), \quad \mathbf{P}_{0,1} = (0, 2, 2), \quad \mathbf{P}_{1,0} = (2, 0, 1), \quad \mathbf{P}_{1,1} = (2, 2, 0)$$

which belong to input data of Coons bilinear surface.

After that, we obtain the coordinate functions $x(u, v)$, $y(u, v)$ and $z(u, v)$ of the surface $\mathbf{P}(u, v)$ when substituting corresponding calculate coordinates of corners and corresponding coordinate functions of the given boundaries in (3.20):

$$x(u, v) = (1 - u, 1, u) \cdot \begin{pmatrix} 0 & 0 & 0 \\ 2u & 0 & 2u \\ -2 & 2 & -2 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = 2u,$$

$$y(u, v) = (1 - u, 1, u) \cdot \begin{pmatrix} 0 & 2v & -2 \\ 0 & 0 & 2 \\ 0 & 2v & -2 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = 2v,$$

$$\begin{aligned} z(u, v) &= (1 - u, 1, u) \cdot \begin{pmatrix} -2 & 4v^2 - 4v + 2 & -2 \\ -u^2 + 2 & 0 & 2u^2 - 4u + 2 \\ -1 & v^2 - 2v + 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = \\ &= 3u^2v - u^2 - 3uv^2 - uv + 4v^2 - 4v + 2, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Vector equations of Coons bilinear surface, tangent vectors of parametric curves and twist vectors are

$$\mathbf{P}(u, v) = (2u, 2v, 3u^2v - u^2 - 3uv^2 - uv + 4v^2 - 4v + 2), \quad (u, v) \in [0, 1]^2, \quad (3.21)$$

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, 6uv - 2u - 3v^2 - v), \quad (u, v) \in [0, 1], \quad (3.22)$$

$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 2, 3u^2 - 6uv - u + 8v - 4), \quad (u, v) \in [0, 1], \quad (3.23)$$

$$\mathbf{P}^{uv}(u, v) = (0, 0, 6u - 6v - 1), \quad (u, v) \in [0, 1]. \quad (3.24)$$

After substitution $u, v = 0, 1$ in (3.21), we obtain

$$\mathbf{P}(0, 0) = (0, 0, 2) \quad \dots \quad (\text{the given corner } \mathbf{P}_{0,0},$$

$$\mathbf{P}(0, 1) = (0, 2, 2) \quad \dots \quad (\text{the given corner } \mathbf{P}_{0,1},$$

$$\mathbf{P}(1, 0) = (2, 0, 1) \quad \dots \quad (\text{the given corner } \mathbf{P}_{1,0},$$

$$\mathbf{P}(1, 1) = (2, 2, 0) \quad \dots \quad (\text{the given corner } \mathbf{P}_{1,1}.$$

After substitution $u = 0, u = 1, v = 0$ and $v = 1$ in (3.21), we obtain the given boundaries of the patch

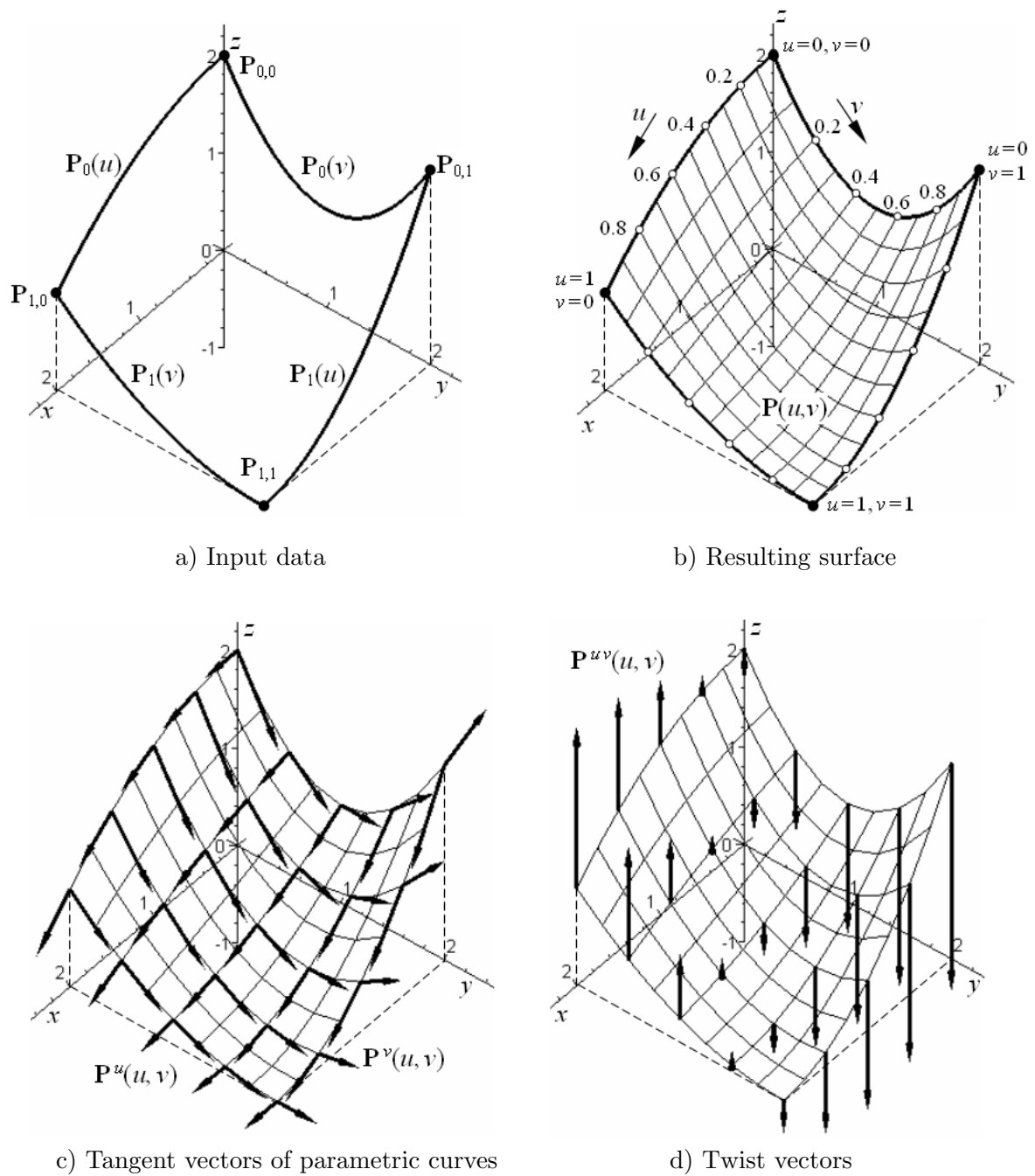


Figure 3.5: Coons bilinear surface

$$\begin{aligned}
 \mathbf{P}(u, 0) &= (2u, 0, -u^2 + 2), \quad u \in [0, 1] \quad \dots\dots\dots && \text{the given boundary } \mathbf{P}_0(u), \\
 \mathbf{P}(u, 1) &= (2u, 2, 2u^2 - 4u + 2), \quad u \in [0, 1] \quad \dots\dots && \text{the given boundary } \mathbf{P}_1(u), \\
 \mathbf{P}(0, v) &= (0, 2v, 4v^2 - 4v + 2), \quad v \in [0, 1] \quad \dots\dots && \text{the given boundary } \mathbf{P}_0(v), \\
 \mathbf{P}(1, v) &= (2, 2v, v^2 - 2v + 1), \quad v \in [0, 1] \quad \dots\dots && \text{the given boundary } \mathbf{P}_1(v).
 \end{aligned}$$

After substitution $u, v = 0, 1$ in (3.21), (3.22), (3.23) and (3.23), we obtain the corners of

the patch and tangent and twist vectors at corners of the patch, see the following table (for completeness, the corners of the patch obtaining by substitution $u, v = 0, 1$ in (3.21) are repeated in this table, too).

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	$(0, 0, 2)$	$(0, 2, 2)$	$(2, 0, 1)$	$(2, 2, 0)$
$\mathbf{P}^u(u, v)$	$(2, 0, -4)$	$(2, 0, -2)$	$(2, 0, -3)$	$(2, 0, 0)$
$\mathbf{P}^v(u, v)$	$(0, 2, -4)$	$(0, 2, 4)$	$(0, 2, -2)$	$(0, 2, 0)$
$\mathbf{P}^{uv}(u, v)$	$(0, 0, -1)$	$(0, 0, -7)$	$(0, 0, 5)$	$(0, 0, -1)$

The given boundaries and corners of the patch are drawn in Fig. 3.5 a), the resulting Coons bilinear surface is drawn in Fig. 3.5 b). Tangent vectors (3.22) and (3.23) of parametric curves at several points of Coons bilinear surface are drawn in Fig. 3.5 c), twist vectors (3.24) at the same points are drawn in Fig. 3.5 d) (tangent vectors are drawn in the scale 1:5, twist vectors are drawn in the scale 1:3). \square

3.4.1 Properties of Coons bilinear surface

The important properties of Coons bilinear surface follow from definition 3.3 and are obvious from examples 3.1, 3.2, 3.4 and 3.5, see section 3.4.2, too.

- Coons bilinear surface interpolates all the given boundaries.
- Coons bilinear surface interpolates the corners of the patch.

If the boundaries of Coons bilinear surface are special, the following interrelations are obvious:

- A ruled surface given by boundaries in u -direction and Coons bilinear surface with straight line segments as boundaries in v -direction are identical.
- A ruled surface given by boundaries in v -direction and Coons bilinear surface with straight line segments as boundaries in u -direction are identical.
- The surface of hyperbolic paraboloid and Coons bilinear surface with straight line segments as all four boundaries are identical.

3.4.2 Vector equation of Coons bilinear surface derivation

Knowing that Coons bilinear surface is given not only by two pairs of opposite boundaries (input data for two ruled surfaces), but also by four corners of a patch (input data for surface of hyperbolic paraboloid), the slightly strange form of the map of Coons bilinear surface as well as the basis functions in definition 3.3 will be comprehensive. Denote

$$\mathbf{S}(u, v) = (1 - v)\mathbf{P}_0(u) + v\mathbf{P}_1(u), \quad (u, v) \in [0, 1]^2$$

the ruled surface given by boundaries of Coons bilinear surface in u -direction,

$$\mathbf{T}(u, v) = (1 - u)\mathbf{P}_0(v) + u\mathbf{P}_1(v), \quad (u, v) \in [0, 1]^2$$

the ruled surface given by boundaries of Coons bilinear surface in v -direction and

$$\mathbf{R}(u, v) = (1 - u)(1 - v)\mathbf{P}_{0,0} + (1 - u)v\mathbf{P}_{0,1} + u(1 - v)\mathbf{P}_{1,0} + uv\mathbf{P}_{1,1}, \quad (u, v) \in [0, 1]^2$$

the surface of hyperbolic paraboloid given by corner points of Coons bilinear surface.

The sum of both of the ruled surfaces $\mathbf{S}(u, v) + \mathbf{T}(u, v)$ is a surface containing Coons bilinear surface as well as the surface of hyperbolic paraboloid $\mathbf{R}(u, v)$, see Fig. 3.6. The given boundaries have common points at corners and parametrization domain of each boundary is $[0, 1]$ (i.e. corners including). Therefore, double corner points appear. Consequently, the vector equation of Coons bilinear surface $\mathbf{P}(u, v)$ is obtained after subtraction of redundant surface of hyperbolic paraboloid from the sum of two ruled surfaces

$$\mathbf{P}(u, v) = \mathbf{S}(u, v) + \mathbf{T}(u, v) - \mathbf{R}(u, v), \quad (u, v) \in [0, 1]^2. \quad (3.25)$$

The proof is elementary – we carry out the matrix product in (3.20) and we get

$$\begin{aligned} \mathbf{P}(u, v) &= \underbrace{(1 - v)\mathbf{P}_0(u) + v\mathbf{P}_1(u)}_{\mathbf{S}(u, v)} + \underbrace{(1 - u)\mathbf{P}_0(v) + u\mathbf{P}_1(v)}_{\mathbf{T}(u, v)} - \\ &\quad - \underbrace{[(1 - u)(1 - v)\mathbf{P}_{0,0} + (1 - u)v\mathbf{P}_{0,1} + u(1 - v)\mathbf{P}_{1,0} + uv\mathbf{P}_{1,1}]}_{\mathbf{R}(u, v)} = \\ &= \mathbf{S}(u, v) + \mathbf{T}(u, v) - \mathbf{R}(u, v), \quad (u, v) \in [0, 1]^2. \end{aligned}$$

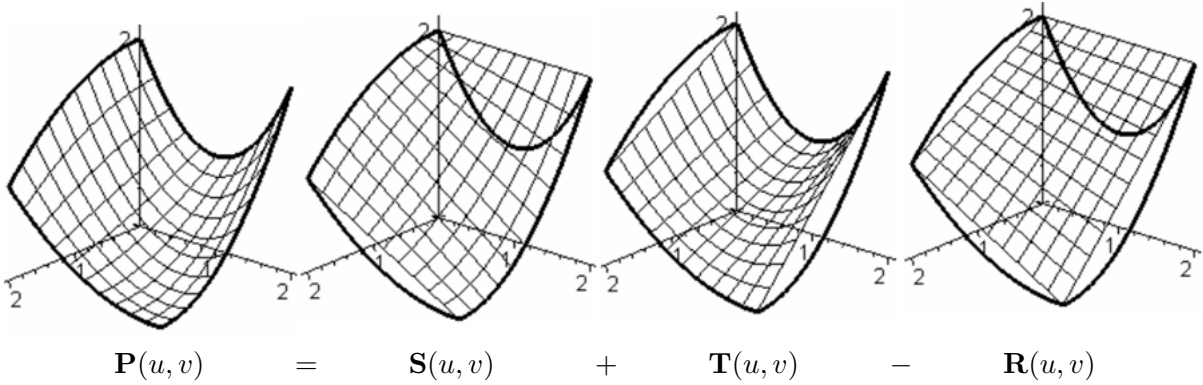


Figure 3.6: Vector equation of Coons bilinear surface derivation

Remarks:

1. The given data in Examples 3.1, 3.2, 3.4 and 3.5 has been chosen in order to obtain Coons bilinear surface from Example 3.5 as the sum of ruled surfaces from Examples 3.1 and 3.2 and subtraction of surface of hyperbolic paraboloid from Example 3.4.
2. In Examples 3.1, 3.2, 3.4 and 3.5, the given data is very simplified: the given boundaries were chosen as planar curves in all cases and the corner points of the patch always lie above the vertices of an axis-aligned square in (x, y) plane. In general, the given boundaries can be spatial curves and corner points of the patch can lie above irregular quadrilateral in (x, y) plane. In such case, the calculation is slightly complicated – coordinate functions $x(u, v)$ and $y(u, v)$ are bivariate functions as well. Thus, the shape of surfaces modeled by Coons bilinear surface can be of a fully general shape.

3.4.3 Patching – patches from Coons bilinear surface

Coons bilinear surface allows only C^0 continuity along the common boundary. Even if coordinate functions of all boundaries allow a continuity of a higher order, the resulting continuity of two patches from Coons bilinear surfaces is only C^0 . In spite of this fact, Coons bilinear surface are very important in technical practice, because the situation when the boundaries are known (or it is possible to find their mathematical representation) is very frequent.

■ **Exercise 3.5** According to the input data, determine the type of surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ (ruled surface, surface of hyperbolic paraboloid or Coons bilinear surface). Find vector equation of surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$, tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and twist vectors $\mathbf{P}^{uv}(u, v)$. Substitute parameters values $u, v = 0, 1$ in all vector equations and determine the elements of the surface which you obtain. Calculate the coordinate of point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$.

In oblique projection ($\omega = 135^\circ, q = 2\sqrt{2} : 3$, unit = 6 cm), use de Casteljau algorithm to construct points $\mathbf{P}(0, \frac{1}{2})$, $\mathbf{P}(1, \frac{1}{2})$, $\mathbf{P}(\frac{1}{2}, 0)$ and $\mathbf{P}(\frac{1}{2}, 1)$ and sketch boundaries $\mathbf{P}_0(u)$, $\mathbf{P}_1(u)$, $\mathbf{P}_0(v)$ and $\mathbf{P}_1(v)$. Draw point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$.

Consider the following enter data:

1. The following corners are given:

- (a) $\mathbf{P}_{0,0} = (0, 0, 2)$, $\mathbf{P}_{0,1} = (0, 2, 1)$, $\mathbf{P}_{1,0} = (2, 0, 2)$, $\mathbf{P}_{1,1} = (2, 2, 1)$;
- (b) $\mathbf{P}_{0,0} = (0, 0, 2)$, $\mathbf{P}_{0,1} = (0, 3, 1)$, $\mathbf{P}_{1,0} = (3, 0, 1)$, $\mathbf{P}_{1,1} = (3, 3, 1)$.

2. Boundaries are Bézier curves given by control polygons:

- (a) $\mathbf{P}_0(u)$: $\mathbf{V}_{0,0} = (0, 0, 2)$, $\mathbf{V}_{1,0} = (1, 0, 0)$, $\mathbf{V}_{2,0} = (2, 0, 2)$,
 $\mathbf{P}_1(u)$: $\mathbf{V}_{0,2} = (0, 2, 1)$, $\mathbf{V}_{1,2} = (1, 2, 0)$, $\mathbf{V}_{2,2} = (2, 2, 1)$;
- (b) $\mathbf{P}_0(v)$: $\mathbf{V}_{0,0} = (0, 0, 2)$, $\mathbf{V}_{0,1} = (0, 1, 2)$, $\mathbf{V}_{0,2} = (0, 2, 1)$,
 $\mathbf{P}_1(v)$: $\mathbf{V}_{2,0} = (2, 0, 2)$, $\mathbf{V}_{2,1} = (2, 1, 0)$, $\mathbf{V}_{2,2} = (2, 2, 1)$;
- (c) $\mathbf{P}_0(u)$: $\mathbf{V}_{0,0} = (0, 0, 2)$, $\mathbf{V}_{1,0} = (1, 0, 0)$, $\mathbf{V}_{2,0} = (2, 0, 2)$,
 $\mathbf{P}_1(u)$: $\mathbf{V}_{0,2} = (0, 2, 1)$, $\mathbf{V}_{1,2} = (1, 2, 0)$, $\mathbf{V}_{2,2} = (2, 2, 1)$,
 $\mathbf{P}_0(v)$: $\mathbf{V}_{0,0} = (0, 0, 2)$, $\mathbf{V}_{0,1} = (0, 1, 2)$, $\mathbf{V}_{0,2} = (0, 2, 1)$,
 $\mathbf{P}_1(v)$: $\mathbf{V}_{2,0} = (2, 0, 2)$, $\mathbf{V}_{2,1} = (2, 1, 0)$, $\mathbf{V}_{2,2} = (2, 2, 1)$;
- (d) $\mathbf{P}_0(u)$: $\mathbf{V}_{0,0} = (0, 0, 3)$, $\mathbf{V}_{1,0} = (1, 0, 0)$, $\mathbf{V}_{2,0} = (2, 0, 0)$, $\mathbf{V}_{3,0} = (3, 0, 1)$,
 $\mathbf{P}_1(u)$: $\mathbf{V}_{0,3} = (0, 3, 2)$, $\mathbf{V}_{1,3} = (1, 3, 0)$, $\mathbf{V}_{2,3} = (2, 3, 1)$, $\mathbf{V}_{3,3} = (3, 3, 0)$;
- (e) $\mathbf{P}_0(v)$: $\mathbf{V}_{0,0} = (0, 0, 3)$, $\mathbf{V}_{0,1} = (0, 1, 2)$, $\mathbf{V}_{0,2} = (0, 2, 0)$, $\mathbf{V}_{0,3} = (0, 3, 2)$,
 $\mathbf{P}_1(v)$: $\mathbf{V}_{3,0} = (3, 0, 1)$, $\mathbf{V}_{3,1} = (3, 1, 0)$, $\mathbf{V}_{3,2} = (3, 2, 1)$, $\mathbf{V}_{3,3} = (3, 3, 0)$;
- (f) $\mathbf{P}_0(u)$: $\mathbf{V}_{0,0} = (0, 0, 3)$, $\mathbf{V}_{1,0} = (1, 0, 0)$, $\mathbf{V}_{2,0} = (2, 0, 0)$, $\mathbf{V}_{3,0} = (3, 0, 1)$,
 $\mathbf{P}_1(u)$: $\mathbf{V}_{0,3} = (0, 3, 2)$, $\mathbf{V}_{1,3} = (1, 3, 0)$, $\mathbf{V}_{2,3} = (2, 3, 1)$, $\mathbf{V}_{3,3} = (3, 3, 0)$,
 $\mathbf{P}_0(v)$: $\mathbf{V}_{0,0} = (0, 0, 3)$, $\mathbf{V}_{0,1} = (0, 1, 2)$, $\mathbf{V}_{0,2} = (0, 2, 0)$, $\mathbf{V}_{0,3} = (0, 3, 2)$,
 $\mathbf{P}_1(v)$: $\mathbf{V}_{3,0} = (3, 0, 1)$, $\mathbf{V}_{3,1} = (3, 1, 0)$, $\mathbf{V}_{3,2} = (3, 2, 1)$, $\mathbf{V}_{3,3} = (3, 3, 0)$.

■ **Exercise 3.6** Use input data and solution of Examples 3.1, 3.2 and 3.3 and find vector equation of Coons bilinear surface according to (3.25). Compare the obtained result with vector equation from Example 3.5.

3.4.4 Coons bilinear surface in Rhinoceros

Before creating Coons bilinear surface, four boundaries – planar or spatial curves – have to be drawn. The common points of adjacent boundaries have to lie at corners of the patch.

Coons bilinear surface – Command: *Surface from 2, 3 or 4 Edge Curves* → Select 2, 3 or 4 curves: click on the boundaries in this order: $\mathbf{P}_0(v)$, $\mathbf{P}_1(v)$, $\mathbf{P}_0(u)$ and $\mathbf{P}_1(u)$ → press Enter. Coons bilinear surface determined by the given boundaries is drawn.

Remark: If the order of selected boundaries corresponds to the order recommended above, the parameters direction and orientation of the created Coons bilinear surface is as expected, see Example 3.6. However, it is possible to modify an unsuitable orientation by command *Direction*, see 1.6.

Point $\mathbf{P}(\alpha, \beta)$ on Coons bilinear surface – Command: *Point from UV coordinates* → Select surface to evaluate: choose *CreatePoint=Yes* and *Normalized=Yes* in command prompt → click on Coons bilinear surface → Enter U value between 0.0 and 1.0 → type α in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type β in command prompt → press Enter → press Enter. The point with parametric coordinates $u = \alpha$ and $v = \beta$ on Coons bilinear surface is drawn, see example 3.3.

Parametric curves of Coons bilinear surface – Command: *Extract Isocurve* → Select surface for isocurve extraction: click on Coons bilinear surface; the marker moves only along the surface and the parametric curve corresponding to its actual position is dynamically displayed → Select isocurve to extract: choose in command prompt *Direction=U*, *Toggle Direction=V* or *Direction=Both* → click to place the required parametric curve or the pair of parametric curves → press Enter. Depending on the chosen option in the command prompt, the parametric u -curve, parametric v -curve or both parametric curves as individual entities are drawn.

■ **Example 3.6 Coons bilinear surface in Rhinoceros.** Coons bilinear surface is given by the following control polygons of boundary Bézier cubic curves:

$$\mathbf{P}_0(u): \quad \mathbf{V}_{0,0} = (0, 0, 2), \quad \mathbf{V}_{1,0} = (1, 0, 0), \quad \mathbf{V}_{2,0} = (2, 0, 0), \quad \mathbf{V}_{3,0} = (3, 0, 1),$$

$$\mathbf{P}_1(u): \quad \mathbf{V}_{0,3} = (0, 3, 1), \quad \mathbf{V}_{1,3} = (1, 3, 0), \quad \mathbf{V}_{2,3} = (2, 3, 1), \quad \mathbf{V}_{3,3} = (3, 3, 1),$$

$$\mathbf{P}_0(v): \quad \mathbf{V}_{0,0} = (0, 0, 2), \quad \mathbf{V}_{0,1} = (0, 1, 2), \quad \mathbf{V}_{0,2} = (0, 2, 0), \quad \mathbf{V}_{0,3} = (0, 3, 1),$$

$$\mathbf{P}_1(v): \quad \mathbf{V}_{3,0} = (3, 0, 1), \quad \mathbf{V}_{3,1} = (3, 1, 0), \quad \mathbf{V}_{3,2} = (3, 2, 1), \quad \mathbf{V}_{3,3} = (3, 3, 1).$$

In Rhinoceros, draw all the given boundaries and create Coons bilinear surface. Draw points on Coons bilinear surface for parameters values $u = 0$, $v = 0, 0.2, \dots, 1$ and $v = 0$, $u = 0, 0.2, \dots, 1$. Draw parametric curves passing through these points.

Solution. Coons bilinear surface is drawn in Fig. 3.7. The parametric curves according to the task setting are drawn by thin lines, the parametric curves automatically displayed for parameters values $u, v = \frac{1}{2}$ are drawn by thicker lines and the given boundaries are drawn by thick lines. □

■ **Exercise 3.7** Consider input data from Exercise 3.5. In Rhinoceros, draw all the given input data and create a surface of appropriate type. Draw point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$ on the surface and find its coordinates (command *Analyze* → *Point* →: activate *PointObjectSnap* → *Point* to evaluate: click at drawn point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$ → read Cartesian coordinates of point in command prompt. Use these coordinates to check your results from Exercise 3.5.

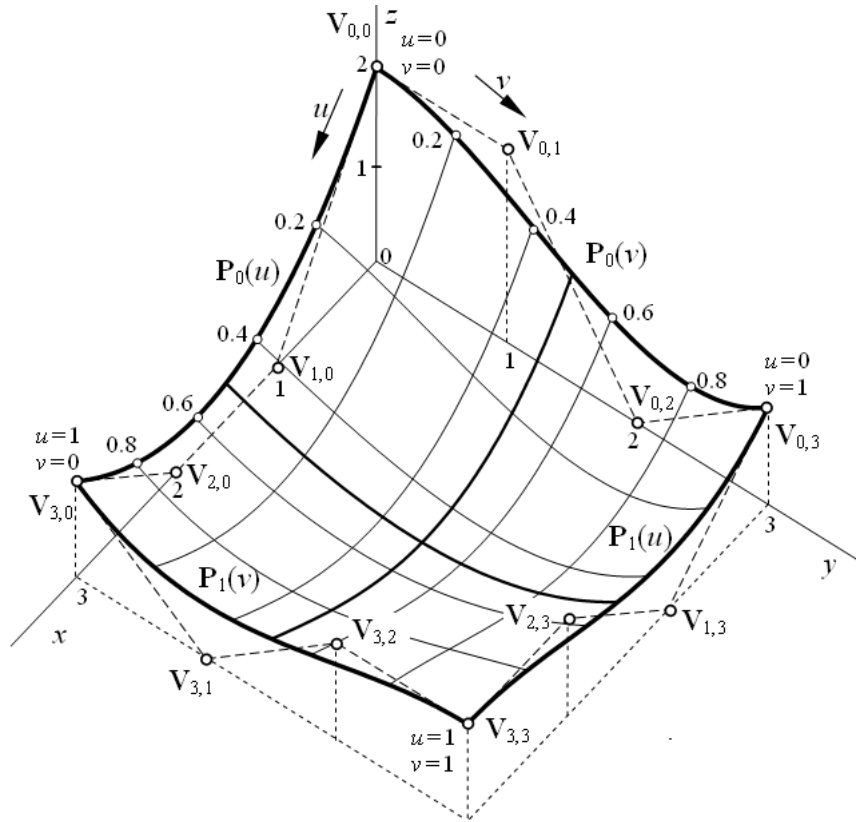


Figure 3.7: Coons bilinear surface given by boundary Bézier cubic curves

3.5 Bézier surface

Bézier surface is given by a mesh of $(m + 1) \times (n + 1)$ control points $\mathbf{V}_{i,j}$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$, arranged into a map of surface:

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \dots & \mathbf{V}_{0,n} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \dots & \mathbf{V}_{1,n} \\ \vdots & \vdots & & \vdots \\ \mathbf{V}_{m,0} & \mathbf{V}_{m,1} & \dots & \mathbf{V}_{m,n} \end{pmatrix}. \quad (3.26)$$

The control points are arranged in u direction along the columns of map \mathbf{M} and in v direction along the rows of map \mathbf{M} . Bézier surface approximates the map \mathbf{M} by one patch.

Control points $\mathbf{V}_{0,0}$, $\mathbf{V}_{0,n}$, $\mathbf{V}_{m,0}$ and $\mathbf{V}_{m,n}$ are called *corners of control mesh*. Straight line segment – connection of two consecutive control points in the row or in the column is called *leg*. Polygons $\mathbf{V}_{i,0}\mathbf{V}_{i,1}\dots\mathbf{V}_{i,n}$, $i = 0, 1, \dots, m$, are called *row control polygons*, polygons $\mathbf{V}_{0,j}\mathbf{V}_{1,j}\dots\mathbf{V}_{m,j}$, $j = 0, 1, \dots, n$, are called *column control polygons*. The row control polygons given by the first and the last row and the column control polygon given by the first and the last column are called *boundary control polygons*.

■ **Definition 3.4 – Bézier surface.** Given a map (3.26), the vector equation of Bézier surface determined by this map is

$$\mathbf{P}(u, v) = \mathbf{B}(u) \cdot \mathbf{M} \cdot \mathbf{B}(v), \quad (u, v) \in [0, 1]^2, \quad (3.27)$$

where

$$\mathbf{B}(u) = (B_{0,m}(u), B_{1,m}(u), \dots, B_{m,m}(u)),$$

and

$$\mathbf{B}^T(v) = (B_{0,n}(v), B_{1,n}(v), \dots, B_{n,n}(v))^T$$

are vectors of basis functions – Bernstein polynomials. $B_{i,m}(u)$, $i = 0, 1, \dots, m$ are Bernstein polynomials of m -th degree

$$B_{i,m}(u) = \binom{m}{i} u^i (1-u)^{m-i}, \quad u \in [0, 1], \quad i = 0, \dots, m,$$

and $B_{j,n}(v)$, $j = 0, 1, \dots, n$ are Bernstein polynomials of n -th degree

$$B_{j,n}(v) = \binom{n}{j} v^j (1-v)^{n-j}, \quad v \in [0, 1], \quad j = 0, \dots, n.$$

□

Bézier surface is created by one patch with analytical representation in the form of bivariate polynomial vector function of m -th degree in u variable and n -th degree in v variable. If the map \mathbf{M} (3.26) is a squared matrix, the degree of both variables is the same. In such a case we distinguish *bilinear* Bézier surface ($m = n = 1$, the control mesh is created by four control points), *biquadratic* Bézier surface ($m = n = 2$, the control mesh is created by nine control points) and *bicubic* Bézier surface ($m = n = 3$, the control mesh is created by sixteen control points). Here, Bézier surfaces given by a higher number of control points will not be considered.

If the map \mathbf{M} (3.26) is a rectangular matrix, the degree of both variables is different.

3.5.1 Bézier surface with control mesh above unit squared grid

With no loss of generality, we will consider Bézier surface given by control points lying above unit squared mesh in (x, y) plane. Thus, the expression of coordinate functions $x(u, v)$ and $y(u, v)$ are simple. Suppose that the control mesh is placed in the first octant of coordinate system. Then, the map of Bézier surface is given by

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \dots & \mathbf{V}_{0,n} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \dots & \mathbf{V}_{1,n} \\ \vdots & \vdots & & \vdots \\ \mathbf{V}_{m,0} & \mathbf{V}_{m,1} & \dots & \mathbf{V}_{m,n} \end{pmatrix} = \begin{pmatrix} (0, 0, z_{0,0}) & (0, 1, z_{0,1}) & \dots & (0, n, z_{0,n}) \\ (1, 0, z_{1,0}) & (1, 1, z_{1,1}) & \dots & (1, n, z_{1,n}) \\ \vdots & \vdots & & \vdots \\ (m, 0, z_{m,0}) & (m, 1, z_{m,1}) & \dots & (m, n, z_{m,n}) \end{pmatrix},$$

and for x -coordinate function and y -coordinate function we get

$$\begin{aligned} x(u, v) &= (B_{0,m}(u), B_{1,m}(u), \dots, B_{m,m}(u)) \cdot \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ m & m & \dots & m \end{pmatrix} \cdot \begin{pmatrix} B_{0,n}(v) \\ B_{1,n}(v) \\ \vdots \\ B_{n,n}(v) \end{pmatrix} = \\ &= mu, \quad (u, v) \in [0, 1]^2, \\ y(u, v) &= (B_{0,m}(u), B_{1,m}(u), \dots, B_{m,m}(u)) \cdot \begin{pmatrix} 0 & 1 & \dots & n \\ 0 & 1 & \dots & n \\ \vdots & \vdots & & \vdots \\ 0 & 1 & \dots & n \end{pmatrix} \cdot \begin{pmatrix} B_{0,n}(v) \\ B_{1,n}(v) \\ \vdots \\ B_{n,n}(v) \end{pmatrix} = \\ &= nv, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Vector equation of Bézier surface given by a mesh above unit squared grid is

$$\mathbf{P}(u, v) = (mu, nv, z(u, v)), \quad (u, v) \in [0, 1]^2, \quad (3.28)$$

and its determination is reduced to the finding z -coordinate function $z(u, v)$.

■ **Example 3.7 – Biquadratic Bézier surface.** The following map of Bézier surface is given

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} \end{pmatrix} = \begin{pmatrix} (0, 0, 2) & (0, 1, 1) & (0, 2, 3) \\ (1, 0, 3) & (1, 1, 0) & (1, 2, 0) \\ (2, 0, 1) & (2, 1, 0) & (2, 2, 0) \end{pmatrix}.$$

Find parametric expression and vector equation of Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vectors $\mathbf{P}^{uv}(u, v)$. Substitute parameters values $u, v = 0, 1$ in vector equations and calculate the corners of the patch, boundary curves of the patch, tangent and twist vectors at the corners of the patch.

Solution. Bézier surface is given by a mesh above unit squared mesh, see Fig. 3.8 a), containing 3×3 control points. Thus, we substitute Bernstein polynomials of second degree $B_{i,2}(u)$, $i = 0, 1, 2$, and $B_{j,2}(v)$, $j = 0, 1, 2$ in (3.27). The resulting surface is biquadratic. Coordinate functions are $x(u, v) = 2u$, $y(u, v) = 2v$ and

$$\begin{aligned} z(u, v) &= (B_{0,2}(u), B_{1,2}(u), B_{2,2}(u)) \cdot \begin{pmatrix} 2 & 1 & 3 \\ 3 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} B_{0,2}(v) \\ B_{1,2}(v) \\ B_{2,2}(v) \end{pmatrix} = \\ &= ((1-u)^2, 2u(1-u), u^2) \cdot \begin{pmatrix} 2 & 1 & 3 \\ 3 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} (1-v)^2 \\ 2v(1-v) \\ v^2 \end{pmatrix} = \\ &= -2u^2v^2 + 8u^2v - 8uv - 3u^2 + 2u + 3v^2 - 2v + 2, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Vector equations of Bézier biquadratic surface, tangent vectors of parametric curves and twist vectors are

$$\mathbf{P}(u, v) = (2u, 2v, -2u^2v^2 + 8u^2v - 8uv - 3u^2 + 2u + 3v^2 - 2v + 2), \quad (u, v) \in [0, 1]^2, \quad (3.29)$$

$$\mathbf{P}^u(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, -4uv^2 + 16uv - 6u - 8v + 2), \quad (u, v) \in [0, 1]^2, \quad (3.30)$$

$$\mathbf{P}^v(u, v) = \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 2, -4u^2v + 8u^2 - 8u + 6v - 2), \quad (u, v) \in [0, 1]^2, \quad (3.31)$$

$$\mathbf{P}^{uv}(u, v) = \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = (0, 0, -8uv + 16u - 8), \quad (u, v) \in [0, 1]^2. \quad (3.32)$$

After substitution $u, v = 0, 1$ in (3.29), (3.30), (3.31) and (3.32), we obtain the corners of the patch, tangent vectors of parametric curves and twist vectors at corners of the patch, see the following table.

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	(0, 0, 2)	(0, 2, 3)	(2, 0, 1)	(2, 2, 0)
$\mathbf{P}^u(u, v)$	(2, 0, 2)	(2, 0, -6)	(2, 0, -4)	(2, 0, 0)
$\mathbf{P}^v(u, v)$	(0, 2, -2)	(0, 2, 4)	(0, 2, -2)	(0, 2, 0)
$\mathbf{P}^{uv}(u, v)$	(0, 0, -8)	(0, 0, -8)	(0, 0, 8)	(0, 0, 0)

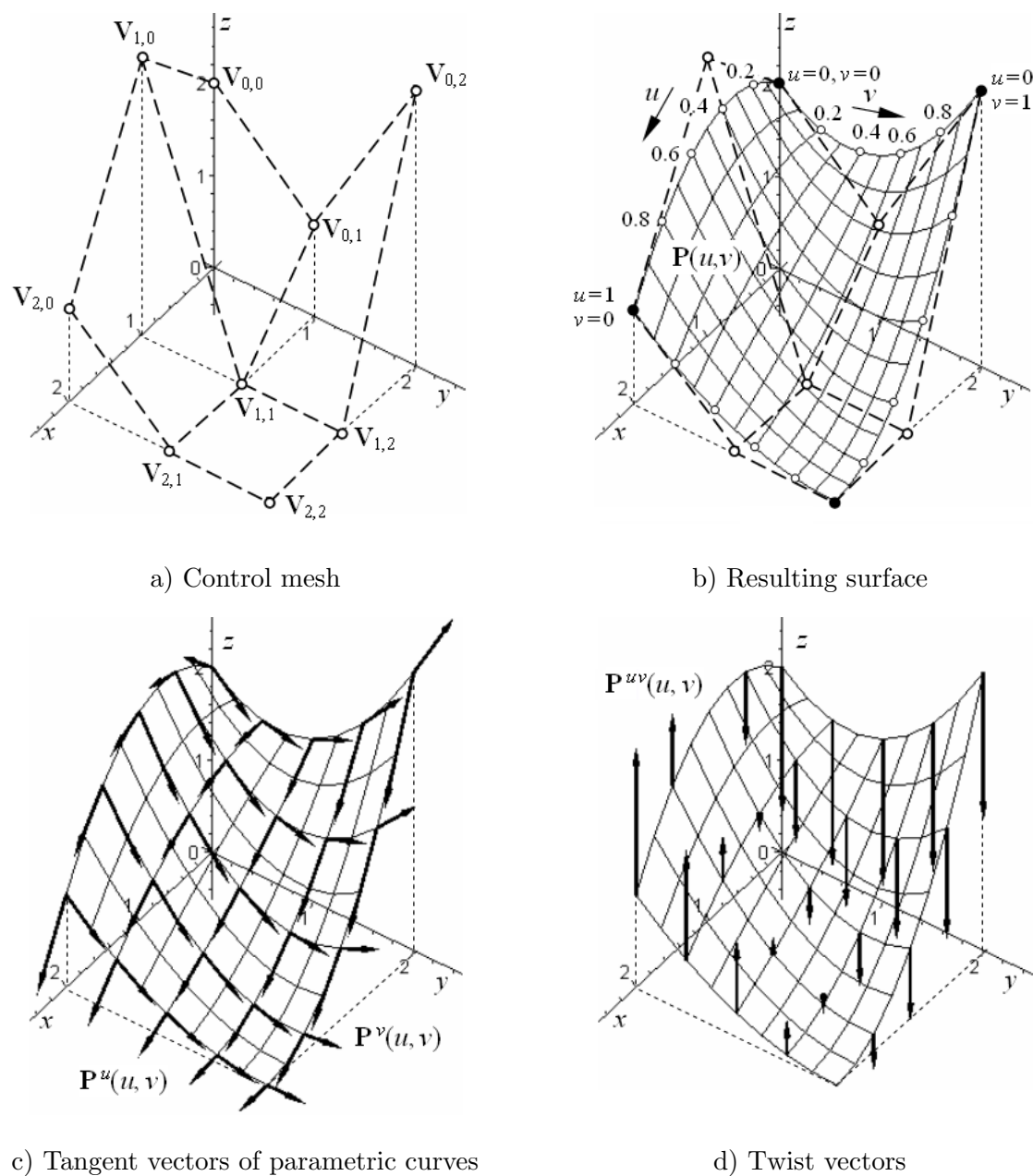


Figure 3.8: Biquadratic Bézier surface

It is obvious that corners $\mathbf{P}(0,0)$, $\mathbf{P}(0,1)$, $\mathbf{P}(1,0)$ and $\mathbf{P}(1,1)$ of the patch are the given control points $\mathbf{V}_{0,0}$, $\mathbf{V}_{0,2}$, $\mathbf{V}_{2,0}$ and $\mathbf{V}_{2,2}$ – corners of the control mesh. After substitution $u = 0$, $u = 1$, $v = 0$ and $v = 1$ in (3.29) we obtain the boundaries of the patch which are quadratic Bézier curves given by boundary control polygons of control mesh

$$\mathbf{P}(u, 0) = (2u, 0, -3u^2 + 2u + 2), \quad u \in [0, 1],$$

$$\mathbf{P}(u, 1) = (2u, 2, 3u^2 - 6u + 3), \quad u \in [0, 1],$$

$$\mathbf{P}(0, v) = (0, 2v, 3v^2 - 2v + 2), \quad v \in [0, 1],$$

$$\mathbf{P}(1, v) = (2, 2v, v^2 - 2v + 1), \quad v \in [0, 1].$$

The given control mesh is drawn in Fig. 3.8 a), the resulting biquadratic Bézier surface (3.29) is drawn in Fig. 3.8 b). Tangent vectors (3.30) and (3.31) of parametric curves at several points of biquadratic Bézier surface (3.29) are drawn in Fig. 3.8 c), twist vectors (3.32) at the same points are drawn in Fig. 3.8 d) (tangent and twist vectors are drawn in the scale 1:5). \square

■ **Example 3.8 – Bézier surface with a rectangular map.** The following map of Bézier surface is given

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 5) & (0, 1, 5) & (0, 2, 2) & (0, 3, 5) \\ (1, 0, 2) & (1, 1, 1) & (1, 2, 0) & (1, 3, 2) \\ (2, 0, 5) & (2, 1, 2) & (2, 2, 2) & (2, 3, 2) \end{pmatrix}.$$

Find parametric expression and vector equation of Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Next, find vector equation of tangent vectors $\mathbf{P}^u(u, v)$ and $\mathbf{P}^v(u, v)$ of parametric curves and vector equation of twist vectors $\mathbf{P}^{uv}(u, v)$. Substitute parameters values $u, v = 0, 1$ in vector equations and calculate the corners of the patch, boundary curves of the patch, tangent and twist vectors at the corners of the patch.

Solution. Bézier surface is given by a mesh above unit squared mesh, see Fig. 3.9 a), containing 3×4 control points. Thus, we substitute second degree Bernstein polynomials $B_{i,2}(u)$, $i = 0, 1, 2$, and third degree Bernstein polynomials $B_{j,3}(v)$, $j = 0, 1, 2, 3$ in (3.27). Coordinate functions are $x(u, v) = 2u$, $y(u, v) = 3v$ and

$$\begin{aligned} z(u, v) &= (B_{0,2}(u), B_{1,2}(u), B_{2,2}(u)) \cdot \begin{pmatrix} 5 & 5 & 2 & 5 \\ 2 & 1 & 0 & 2 \\ 5 & 2 & 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} B_{0,3}(v) \\ B_{1,3}(v) \\ B_{2,3}(v) \\ B_{3,3}(v) \end{pmatrix} = \\ &= ((1-u)^2, 2u(1-u), u^2) \cdot \begin{pmatrix} 5 & 5 & 2 & 5 \\ 2 & 1 & 0 & 2 \\ 5 & 2 & 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} (1-v)^3 \\ 3v(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{pmatrix} = \\ &= -3u^2v + 6u^2 - 12uv^3 + 18uv^2 - 6uv - 6u + 9v^3 - 9v^2 + 5, \quad (u, v) \in [0, 1]^2. \end{aligned}$$

Vector equation of Bézier surface, tangent vectors of parametric curves and twist vectors is

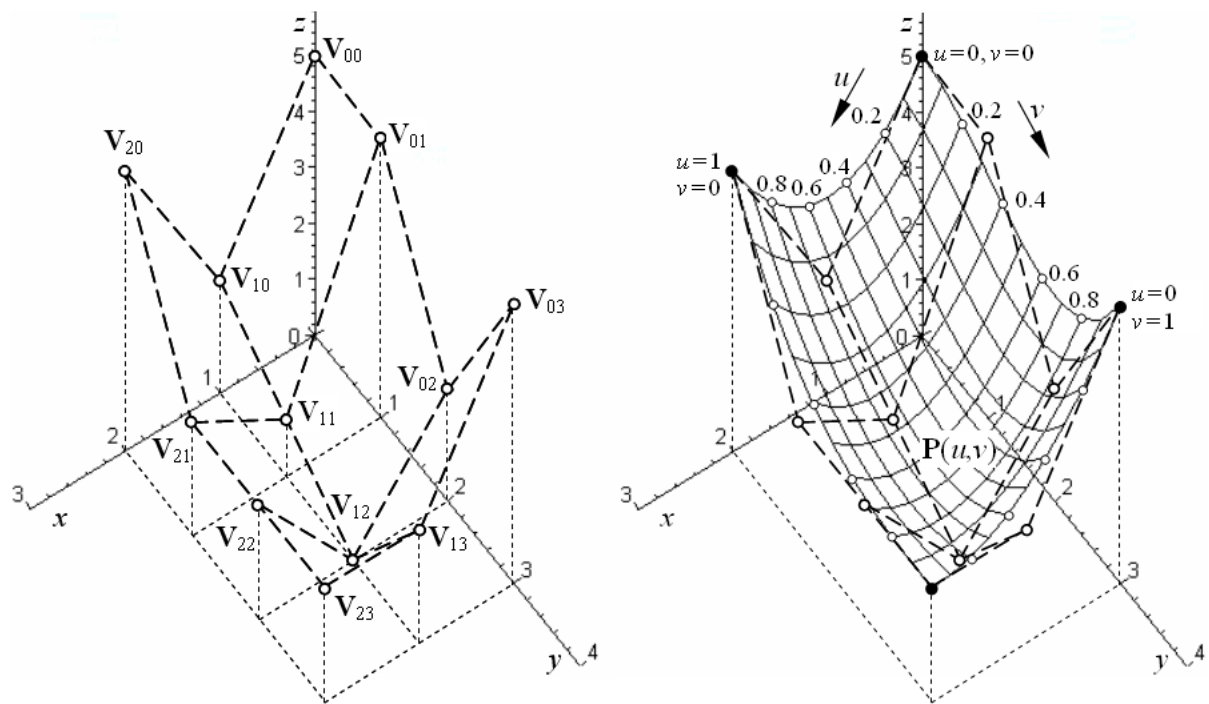
$$\begin{aligned} \mathbf{P}(u, v) &= (2u, 3v, -3u^2v + 6u^2 - 12uv^3 + 18uv^2 - 6uv - 6u + 9v^3 - 9v^2 + 5), \\ &(u, v) \in [0, 1]^2, \end{aligned} \quad (3.33)$$

$$\begin{aligned} \mathbf{P}^u(u, v) &= \frac{\partial \mathbf{P}(u, v)}{\partial u} = (2, 0, -6uv + 12u - 12v^3 + 18v^2 - 6v - 6), \\ &(u, v) \in [0, 1]^2, \end{aligned} \quad (3.34)$$

$$\begin{aligned} \mathbf{P}^v(u, v) &= \frac{\partial \mathbf{P}(u, v)}{\partial v} = (0, 3, -36uv^2 + 36uv - 3u^2 - 6u + 27v^2 - 18v), \\ &(u, v) \in [0, 1]^2, \end{aligned} \quad (3.35)$$

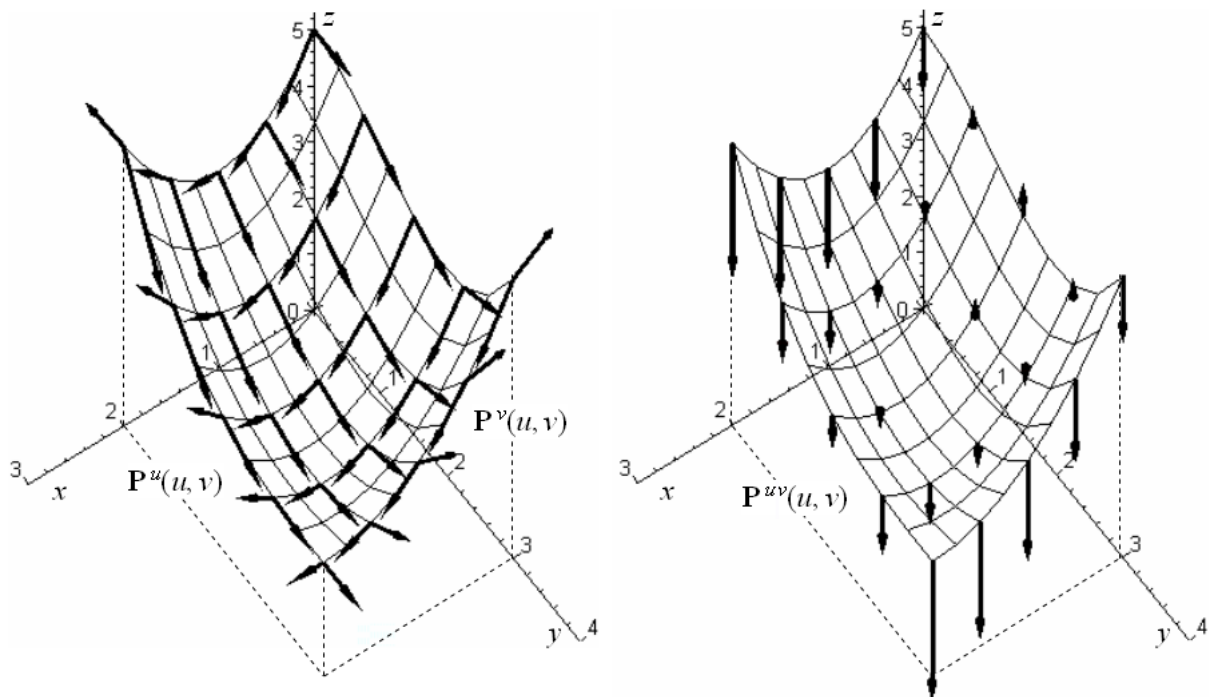
$$\mathbf{P}^{uv}(u, v) = \frac{\partial^2 \mathbf{P}(u, v)}{\partial u \partial v} = (0, 0, -6u - 36v^2 + 36v - 6), \quad (u, v) \in [0, 1]^2. \quad (3.36)$$

After substitution $u, v = 0, 1$ in (3.33), (3.34), (3.35) and (3.36) we obtain the corners of the patch, tangent vectors of parametric curves and twist vectors at the corners of the patch, see the following table.



a) Control mesh

b) Resulting surface



c) Tangent vectors of parametric curves

d) Twist vectors of parametric curves

Figure 3.9: Bézier surface with a rectangular mesh

	$u = 0, v = 0$	$u = 0, v = 1$	$u = 1, v = 0$	$u = 1, v = 1$
$\mathbf{P}(u, v)$	$(0, 0, 5)$	$(0, 3, 5)$	$(2, 0, 5)$	$(2, 3, 2)$
$\mathbf{P}^u(u, v)$	$(2, 0, -6)$	$(2, 0, -6)$	$(2, 0, 6)$	$(2, 0, 0)$
$\mathbf{P}^v(u, v)$	$(0, 3, 0)$	$(0, 3, 9)$	$(0, 3, -9)$	$(0, 3, 0)$
$\mathbf{P}^{uv}(u, v)$	$(0, 0, -6)$	$(0, 0, -6)$	$(0, 0, -12)$	$(0, 0, -12)$

The corners $\mathbf{P}(0, 0)$, $\mathbf{P}(0, 1)$, $\mathbf{P}(1, 0)$ and $\mathbf{P}(1, 1)$ are the given control points $\mathbf{V}_{0,0}$, $\mathbf{V}_{0,3}$, $\mathbf{V}_{2,0}$ and $\mathbf{V}_{2,3}$ – corners of control mesh. After substitution $u = 0$, $u = 1$, $v = 0$ and $v = 1$ in (3.33) we obtain the boundaries of the patch which are quadratic Bézier curves in u -direction and cubic Bézier curves in v -direction given by boundary control polygons of control mesh.

$$\begin{aligned}\mathbf{P}(u, 0) &= (2u, 0, 6u^2 - 6u + 5), \quad u \in [0, 1], \\ \mathbf{P}(u, 1) &= (2u, 3, 3u^2 - 6u + 5), \quad u \in [0, 1], \\ \mathbf{P}(0, v) &= (0, 3v, 9v^3 - 9v^2 + 5), \quad v \in [0, 1], \\ \mathbf{P}(1, v) &= (2, 3v, -3v^3 + 9v^2 - 9v + 5), \quad v \in [0, 1].\end{aligned}$$

The given control mesh is drawn in Fig. 3.9 a), the resulting Bézier surface (3.33) is drawn in Fig. 3.9 b). Tangent vectors (3.34) and (3.35) of parametric curves at several points of Bézier surface (3.33) are drawn in Fig. 3.9 c), twist vectors (3.36) at the same points are drawn in Fig. 3.9 d) (tangent and twist vectors are drawn in the scale 1:5). \square

3.5.2 Properties of Bézier surface

The important properties of Bézier surface are obvious from examples 3.7 and 3.8.

- Bézier surface interpolates the corners of control mesh.
- The boundaries of patch from Bézier surface are Bézier curves given by boundary control polygons.
- Tangent vectors of parametric u -curves at the corners of the patch are equal to m -multiple of the vectors given by the first/the last leg of boundary control polygon in u direction. Tangent vectors of parametric v -curves at the corners of the patch are equal to n -multiple of the vectors given by the first/the last leg of boundary control polygon in v -direction. Specially, for bicubic Bézier surface we have

$$\begin{aligned}\mathbf{P}^u(0, 0) &= 3(\mathbf{V}_{1,0} - \mathbf{V}_{0,0}), & \mathbf{P}^u(0, 1) &= 3(\mathbf{V}_{1,3} - \mathbf{V}_{0,3}), \\ \mathbf{P}^u(1, 0) &= 3(\mathbf{V}_{3,0} - \mathbf{V}_{2,0}), & \mathbf{P}^u(1, 1) &= 3(\mathbf{V}_{3,3} - \mathbf{V}_{2,3}), \\ \mathbf{P}^v(0, 0) &= 3(\mathbf{V}_{0,1} - \mathbf{V}_{0,0}), & \mathbf{P}^v(0, 1) &= 3(\mathbf{V}_{0,3} - \mathbf{V}_{0,2}), \\ \mathbf{P}^v(1, 1) &= 3(\mathbf{V}_{3,3} - \mathbf{V}_{3,2}), & \mathbf{P}^v(1, 0) &= 3(\mathbf{V}_{3,1} - \mathbf{V}_{3,0}).\end{aligned}\tag{3.37}$$

This property follows from the previous property of Bézier surface and from properties of Bézier curves, see Section 2.2.1. It is obvious that tangent vectors of parametric curves at corners of a patch are given only by control points of boundary control polygons. Moreover, the tangent planes at corners of a patch are given only by end legs of boundary control polygons.

- Twist vectors at corners of a patch from bicubic Bézier surface are given by

$$\begin{aligned}
 \mathbf{P}^{uv}(0, 0) &= 9[(\mathbf{V}_{0,0} - \mathbf{V}_{1,0}) + (\mathbf{V}_{1,1} - \mathbf{V}_{0,1})] = 9[(\mathbf{V}_{0,0} - \mathbf{V}_{0,1}) + (\mathbf{V}_{1,1} - \mathbf{V}_{1,0})], \\
 \mathbf{P}^{uv}(0, 1) &= 9[(\mathbf{V}_{0,2} - \mathbf{V}_{1,2}) + (\mathbf{V}_{1,3} - \mathbf{V}_{0,3})] = 9[(\mathbf{V}_{0,2} - \mathbf{V}_{0,3}) + (\mathbf{V}_{1,3} - \mathbf{V}_{1,2})], \\
 \mathbf{P}^{uv}(1, 0) &= 9[(\mathbf{V}_{2,0} - \mathbf{V}_{3,0}) + (\mathbf{V}_{3,1} - \mathbf{V}_{2,1})] = 9[(\mathbf{V}_{2,0} - \mathbf{V}_{2,1}) + (\mathbf{V}_{3,1} - \mathbf{V}_{3,0})], \\
 \mathbf{P}^{uv}(1, 1) &= 9[(\mathbf{V}_{2,2} - \mathbf{V}_{3,2}) + (\mathbf{V}_{3,3} - \mathbf{V}_{2,3})] = 9[(\mathbf{V}_{2,2} - \mathbf{V}_{2,3}) + (\mathbf{V}_{3,3} - \mathbf{V}_{3,2})].
 \end{aligned}
 \tag{3.38}$$

- Bézier surface given by a mesh containing 2×2 control points (i.e. corners of the patch) and the surface of hyperbolic paraboloid given by these corners are identical.

■ **Example 3.9 – Bézier quadratic-cubic surface.** The following map of Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ is given

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 2) & (0, 1, 0) & (0, 2, 0) & (0, 3, 1) \\ (1, 0, 2) & (1, 1, 0) & (1, 2, 0) & (1, 3, 0) \\ (2, 0, 0) & (2, 1, 0) & (2, 2, 0) & (2, 3, 0) \end{pmatrix}.$$

In oblique projection ($\omega = 135^\circ, q = 2\sqrt{2} : 3$, unit = 3 cm) draw control points $\mathbf{V}_{i,j}$, $i = 0, 1, 2, j = 0, 1, 2, 3$ and construct control mesh. Using de Casteljau algorithm, construct points $\mathbf{P}(\frac{1}{2}, 0)$, $\mathbf{P}(\frac{1}{2}, 1)$, $\mathbf{P}(0, \frac{1}{2})$ and $\mathbf{P}(1, \frac{1}{2})$ on boundaries of the patch from Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Sketch boundaries of the patch. Mark tangent planes $\tau_{0,0}$, $\tau_{0,3}$, $\tau_{2,0}$ and $\tau_{2,3}$ at corners of the patch.

Solution. The solution is drawn in Fig. 3.10.

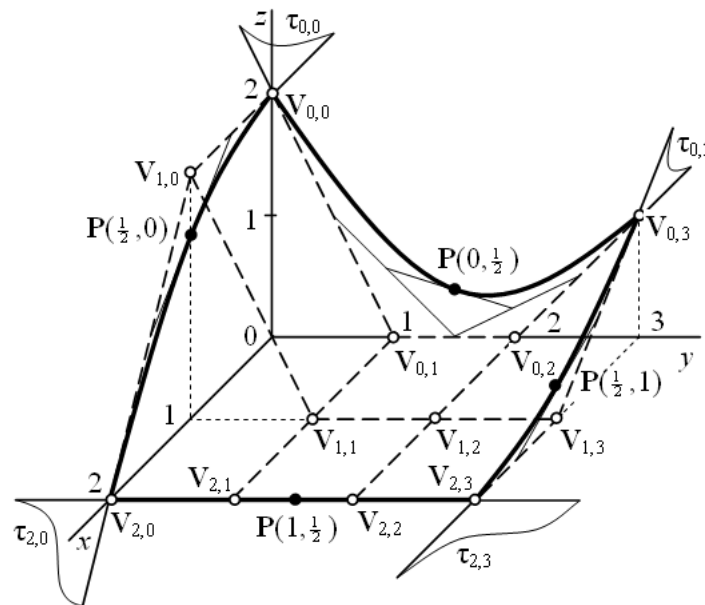


Figure 3.10: Bézier quadratic-cubic surface

□

■ **Example 3.10 – Bézier bicubic surface.** The following map of Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ is given

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \\ \mathbf{V}_{3,0} & \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \mathbf{V}_{3,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 2) & (0, 1, 2) & (0, 2, 0) & (0, 3, 2) \\ (1, 0, 2) & (1, 1, 0) & (1, 2, 0) & (1, 3, 0) \\ (2, 0, 2) & (2, 1, 0) & (2, 2, 0) & (2, 3, 0) \\ (3, 0, 1) & (3, 1, 0) & (3, 2, 0) & (3, 3, 0) \end{pmatrix}.$$

In oblique projection ($\omega = 135^\circ, q = 2\sqrt{2} : 3$, unit = 3 cm), draw control points $\mathbf{V}_{i,j}$, $i, j = 0, 1, 2, 3$ and construct control mesh. Using de Casteljau algorithm, construct points $\mathbf{P}(\frac{1}{2}, 0)$, $\mathbf{P}(\frac{1}{2}, 1)$, $\mathbf{P}(0, \frac{1}{2})$ and $\mathbf{P}(1, \frac{1}{2})$ on boundaries of the patch from Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$. Sketch boundaries of the patch. Mark tangent planes $\tau_{0,0}$, $\tau_{0,3}$, $\tau_{3,0}$ and $\tau_{3,3}$ at corners of the patch.

Solution. The solution is drawn in Fig. 3.11.

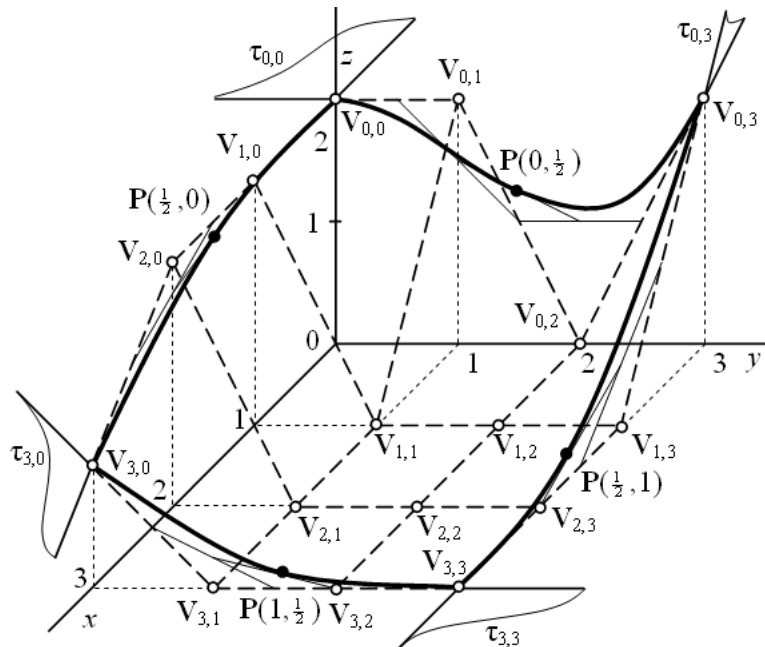


Figure 3.11: Bicubic Bézier surface

□

Bézier bicubic surface and Coons bilinear surface interrelation

To formulate an interesting interrelation between Coons bilinear surface given by four boundary Bézier cubic curves and Bézier bicubic surface, it is useful to summarize the following properties of both surfaces.

- Coons bilinear surface is given by four boundary curves with common points at corners of a patch.
- Bézier bicubic surface interpolates corners of a patch.
- Boundaries of Bézier bicubic surface are Bézier cubic curves given by boundary control polygons.

The shape of Coons bilinear surface is given only by boundary curves. However, the shape of Bézier bicubic surface is given not only by boundary control polygons but also by four inner control points ($\mathbf{V}_{1,1}$, $\mathbf{V}_{1,2}$, $\mathbf{V}_{2,1}$ and $\mathbf{V}_{2,2}$). It follows that there exist infinitely many Bézier bicubic surfaces with identical boundaries but vary in shape of surface itself. This difference is given by position of four inner control points. Therefore, it is possible to find such positions of control points $\mathbf{V}_{1,1}$, $\mathbf{V}_{1,2}$, $\mathbf{V}_{2,1}$ and $\mathbf{V}_{2,2}$ that Bézier bicubic surface and Coons bilinear surface are identical.

■ **Theorem 3.1 – Bézier bicubic and Coons bilinear surface interrelation.** Let $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ be Coons bilinear surface given by boundary Bézier cubic curves with the following control polygons

$$\begin{aligned} \mathbf{P}_0(u): \mathbf{V}_{0,0} \mathbf{V}_{1,0} \mathbf{V}_{2,0} \mathbf{V}_{3,0}, & \quad \mathbf{P}_1(u): \mathbf{V}_{0,3} \mathbf{V}_{1,3} \mathbf{V}_{2,3} \mathbf{V}_{3,3}, \\ \mathbf{P}_0(v): \mathbf{V}_{0,0} \mathbf{V}_{0,1} \mathbf{V}_{0,2} \mathbf{V}_{0,3}, & \quad \mathbf{P}_1(v): \mathbf{V}_{3,0} \mathbf{V}_{3,1} \mathbf{V}_{3,2} \mathbf{V}_{3,3}. \end{aligned}$$

Then, Coons bilinear surface $\mathbf{P}(u, v)$ and Bézier bicubic surface given by the following map

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \\ \mathbf{V}_{3,0} & \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \mathbf{V}_{3,3} \end{pmatrix},$$

are identical provided that the inner control points comply with the following relations

$$\begin{aligned} \mathbf{V}_{1,1} &= \mathbf{V}_{0,0} + \frac{1}{3}\mathbf{P}^u(0, 0) + \frac{1}{3}\mathbf{P}^v(0, 0) + \frac{1}{9}\mathbf{P}^{uv}(0, 0), \\ \mathbf{V}_{1,2} &= \mathbf{V}_{0,3} + \frac{1}{3}\mathbf{P}^u(0, 1) - \frac{1}{3}\mathbf{P}^v(0, 1) - \frac{1}{9}\mathbf{P}^{uv}(0, 1), \\ \mathbf{V}_{2,1} &= \mathbf{V}_{3,0} - \frac{1}{3}\mathbf{P}^u(1, 0) + \frac{1}{3}\mathbf{P}^v(1, 0) - \frac{1}{9}\mathbf{P}^{uv}(1, 0), \\ \mathbf{V}_{2,2} &= \mathbf{V}_{3,3} - \frac{1}{3}\mathbf{P}^u(1, 1) - \frac{1}{3}\mathbf{P}^v(1, 1) + \frac{1}{9}\mathbf{P}^{uv}(1, 1), \end{aligned} \tag{3.39}$$

where $\mathbf{P}^u(0, 0)$, $\mathbf{P}^u(0, 1)$, $\mathbf{P}^u(1, 0)$ and $\mathbf{P}^u(1, 1)$ are tangent vectors of parametric u -curves at the corners of the patch from Coons bilinear surface, $\mathbf{P}^v(0, 0)$, $\mathbf{P}^v(0, 1)$, $\mathbf{P}^v(1, 0)$ and $\mathbf{P}^v(1, 1)$ are tangent vectors of parametric v -curves at the corners of the patch from Coons bilinear surface and $\mathbf{P}^{uv}(0, 0)$, $\mathbf{P}^{uv}(0, 1)$, $\mathbf{P}^{uv}(1, 0)$ and $\mathbf{P}^{uv}(1, 1)$ are twist vectors of the patch from Coons bilinear surface.

Proof: Firstly, we solve (3.37) with respect to the boundary control points of Bézier bicubic surface. Secondly, we substitute the obtained boundary control points in (3.38). Finally, we solve the obtained set of equations with respect to the unknown inner control points of Bézier bicubic surface. \square

■ **Example 3.11 – Bézier bicubic surface and Coons bilinear surface interrelation.** The following control polygon of Bézier cubic curves are given

$$\begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & & & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & & & \mathbf{V}_{2,3} \\ \mathbf{V}_{3,0} & \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \mathbf{V}_{3,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 4) & (0, 1, 4) & (0, 2, 1) & (0, 3, 1) \\ (1, 0, 1) & & & (1, 3, 1) \\ (2, 0, 1) & & & (2, 3, 1) \\ (3, 0, 4) & (3, 1, 1) & (3, 2, 1) & (3, 3, 1) \end{pmatrix}.$$

Suppose these Bézier cubic curves are boundaries of Coons bilinear surface $\mathbf{P}_C(u, v)$, $(u, v) \in [0, 1]^2$.

Find vector equation of Coons bilinear surface $\mathbf{P}_C(u, v)$, $(u, v) \in [0, 1]^2$. Suppose Coons bilinear surface $\mathbf{P}_C(u, v)$, $(u, v) \in [0, 1]^2$ is identical with Bézier bicubic surface $\mathbf{P}_B(u, v)$, $(u, v) \in [0, 1]^2$ given by control mesh above unit squared grid. Determine the inner control points of Bézier bicubic surface $\mathbf{P}_B(u, v)$, $(u, v) \in [0, 1]^2$ and find its vector equation. Compare the results.

Solution. Vector equations of boundary Bézier cubic curves are

$$\begin{aligned}\mathbf{P}_0(u) &= (3u, 0, 9u^2 - 9u + 4), \quad u \in [0, 1], \\ \mathbf{P}_1(u) &= (3u, 3, 1), \quad u \in [0, 1], \\ \mathbf{P}_0(v) &= (0, 3v, 6v^3 - 9v^2 + 4), \quad v \in [0, 1], \\ \mathbf{P}_1(v) &= (3, 3v, -3v^3 + 9v^2 - 9v + 4), \quad v \in [0, 1].\end{aligned}$$

Parametric expression of Coons bilinear surface according to (3.20) is

$$\begin{aligned}x_C(u, v) &= (1 - u, 1, u) \cdot \begin{pmatrix} 0 & 0 & 0 \\ 3u & 0 & 3u \\ -3 & 3 & -3 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = 3u, \quad (u, v) \in [0, 1]^2, \\ y_C(u, v) &= (1 - u, 1, u) \cdot \begin{pmatrix} 0 & 3v & -3 \\ 0 & 0 & 3 \\ 0 & 3v & -3 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = 3v, \quad (u, v) \in [0, 1]^2, \\ z_C(u, v) &= (1 - u, 1, u) \cdot \begin{pmatrix} -4 & 6v^3 - 9v^2 + 4 & -1 \\ 9u^2 - 9u + 4 & 0 & 1 \\ -4 & -3v^3 + 9v^2 - 9v + 4 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 - v \\ 1 \\ v \end{pmatrix} = \\ &= -9u^2v + 9u^2 - 9uv^3 + 18uv^2 - 9u + 6v^3 - 9v^2 + 4, \quad (u, v) \in [0, 1]^2.\end{aligned}$$

Vector equation of Coons bilinear surface is

$$\mathbf{P}_C(u, v) = (3u, 3v, -9u^2v + 9u^2 - 9uv^3 + 18uv^2 - 9u + 6v^3 - 9v^2 + 4), \quad (u, v) \in [0, 1]^2. \quad (3.40)$$

The given control polygons and boundary Bézier cubic curves are drawn in Fig. 3.12 a).

Vector equations of tangent vectors of parametric curves of Coons bilinear surface are

$$\begin{aligned}\mathbf{P}_C^u(u, v) &= (3, 0, -18uv + 18u - 9v^3 + 18v^2 - 9), \quad (u, v) \in [0, 1]^2, \\ \mathbf{P}_C^v(u, v) &= (0, 3, -9u^2 - 27uv^2 + 36uv + 18v^2 - 18v), \quad (u, v) \in [0, 1]^2.\end{aligned} \quad (3.41)$$

After substitution $u, v = 0, 1$ in (3.41), we obtain tangent vectors of parametric curves at the corners of the patch from Coons bilinear surface

$$\begin{aligned}\mathbf{P}_C^u(0, 0) &= (3, 0, -9), \quad \mathbf{P}_C^u(0, 1) = (3, 0, 0), \quad \mathbf{P}_C^u(1, 0) = (3, 0, 9), \quad \mathbf{P}_C^u(1, 1) = (3, 0, 0), \\ \mathbf{P}_C^v(0, 0) &= (0, 3, 0), \quad \mathbf{P}_C^v(0, 1) = (0, 3, 0), \quad \mathbf{P}_C^v(1, 0) = (0, 3, -9), \quad \mathbf{P}_C^v(1, 1) = (0, 3, 0).\end{aligned}$$

Vector equation of twist vectors of Coons bilinear surface is

$$\mathbf{P}_C^{uv}(u, v) = (0, 0, -18u - 27v^2 + 36v), \quad (u, v) \in [0, 1]^2. \quad (3.42)$$

After substitution $u, v = 0, 1$ in (3.42), we obtain twist vectors at the corners of the patch from Coons bilinear surface

$$\mathbf{P}_C^{uv}(0, 0) = (0, 0, 0), \quad \mathbf{P}_C^{uv}(0, 1) = (0, 0, 9), \quad \mathbf{P}_C^{uv}(1, 0) = (0, 0, -18), \quad \mathbf{P}_C^{uv}(1, 1) = (0, 0, -9).$$

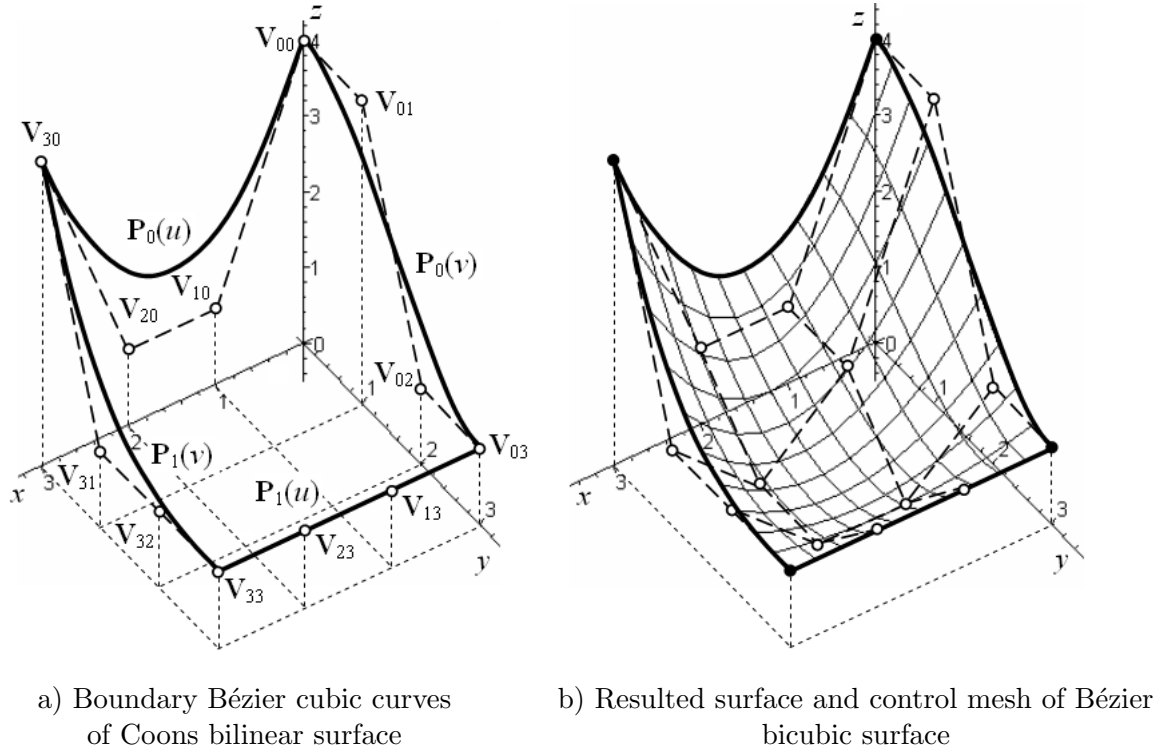


Figure 3.12: Bézier bicubic and Coons bilinear surface interrelation

Now, it is possible to determine the inner control points of Bézier bicubic surface $\mathbf{P}_{\mathbf{B}}(u, v)$. According to (3.39) we have

$$\begin{aligned} \mathbf{V}_{1,1} &= (0, 0, 4) + \frac{1}{3}(3, 0, -9) + \frac{1}{3}(0, 3, 0) + \frac{1}{9}(0, 0, 0) = (1, 1, 1), \\ \mathbf{V}_{1,2} &= (0, 3, 1) + \frac{1}{3}(3, 0, 0) - \frac{1}{3}(0, 3, 0) - \frac{1}{9}(0, 0, 9) = (1, 2, 0), \\ \mathbf{V}_{2,1} &= (3, 0, 4) - \frac{1}{3}(3, 0, 9) + \frac{1}{3}(0, 3, -9) - \frac{1}{9}(0, 0, -18) = (2, 1, 0), \\ \mathbf{V}_{2,2} &= (3, 3, 1) - \frac{1}{3}(3, 0, 0) - \frac{1}{3}(0, 3, 0) + \frac{1}{9}(0, 0, -9) = (2, 2, 0). \end{aligned}$$

Then, the map \mathbf{M} of Bézier bicubic surface $\mathbf{P}_{\mathbf{B}}(u, v)$ is

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \\ \mathbf{V}_{3,0} & \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \mathbf{V}_{3,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 4) & (0, 1, 4) & (0, 2, 1) & (0, 3, 1) \\ (1, 0, 1) & (1, 1, 1) & (1, 2, 0) & (1, 3, 1) \\ (2, 0, 1) & (2, 1, 0) & (2, 2, 0) & (2, 3, 1) \\ (3, 0, 4) & (3, 1, 1) & (3, 2, 1) & (3, 3, 1) \end{pmatrix}.$$

Bézier bicubic surface $\mathbf{P}_{\mathbf{B}}(u, v)$, $(u, v) \in [0, 1]^2$ is given by control mesh above unit squared grid. Therefore, the coordinate function of Bézier bicubic surface are

$$\begin{aligned} x_{\mathbf{B}}(u, v) &= 3u, \\ y_{\mathbf{B}}(u, v) &= 3v \end{aligned}$$

and

$$\begin{aligned}
z_{\mathbf{B}}(u, v) &= (B_{0,3}(u), B_{1,3}(u), B_{2,3}(u), B_{3,3}(u)) \cdot \begin{pmatrix} 4 & 4 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 4 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} B_{0,3}(v) \\ B_{1,3}(v) \\ B_{2,3}(v) \\ B_{3,3}(v) \end{pmatrix} = \\
&= ((1-u)^3, 3u(1-u)^2, 3u^2(1-u), u^3) \cdot \begin{pmatrix} 4 & 4 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 4 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} (1-v)^3 \\ 3v(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{pmatrix} = \\
&= -9u^2v + 9u^2 - 9uv^3 + 18uv^2 - 9u + 6v^3 - 9v^2 + 4, \quad (u, v) \in [0, 1]^2.
\end{aligned}$$

Vector equation of Bézier bicubic surface $\mathbf{P}_{\mathbf{B}}(u, v)$, $(u, v) \in [0, 1]^2$ is

$$\mathbf{P}_{\mathbf{B}}(u, v) = (3u, 3v, -9u^2v + 9u^2 - 9uv^3 + 18uv^2 - 9u + 6v^3 - 9v^2 + 4), \quad (u, v) \in [0, 1]^2.$$

The resulted Bézier bicubic surface and its control mesh are drawn in Fig. 3.12 b).

Since $x_{\mathbf{B}}(u, v) = x_{\mathbf{C}}(u, v)$, $y_{\mathbf{B}}(u, v) = y_{\mathbf{C}}(u, v)$ and $z_{\mathbf{B}}(u, v) = z_{\mathbf{C}}(u, v)$, Bézier bicubic surface and Coons bilinear surface are identical. \square

3.5.3 Vector equation of Bézier surface derivation

To derive vector equation of Bézier surface (3.27), we will use two approaches. Firstly, we will respect a *surface character* of a Bézier surface and obtain its vector equation by repeated linear interpolation of four points. Secondly, we will respect a *curve character* of a Bézier surface and express its vector equation as an equation of Bézier curve with variable control points. Both approaches are equivalent and the results obtained are the same. Note that there exist more ways to derive a vector equation of a Bézier surface but they will not be discussed here.

Repeated linear interpolation of four points

Repeated linear interpolation of four points corresponds to the linear interpolation of two points – endpoints of a leg of control polygon – used in Section 2.2.2 to derive Bernstein polynomials and vector equation of Bézier curve. In general, a spatial quadrilateral – four adjacent control points of control mesh – corresponds to the leg of control polygon.

If a control mesh is given by only four control points, the resulted Bézier surface and surface of hyperbolic paraboloid given by these four points are identical, see Section 3.5.2. The surface of hyperbolic paraboloid is created just by linear interpolation of four points.

If a control mesh is given by higher number of control points, it is necessary to repeat the linear interpolation of four points. For example, see Fig. 3.13 a). Here, a control mesh of Bézier biquadratic surface is drawn. This control mesh is created by four quadrilaterals with the following maps

$$\mathbf{M}_0 = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} \end{pmatrix}, \quad \mathbf{M}_1 = \begin{pmatrix} \mathbf{V}_{0,1} & \mathbf{V}_{0,2} \\ \mathbf{V}_{1,1} & \mathbf{V}_{1,2} \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} \mathbf{V}_{1,0} & \mathbf{V}_{1,1} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} \mathbf{V}_{1,1} & \mathbf{V}_{1,2} \\ \mathbf{V}_{2,1} & \mathbf{V}_{2,2} \end{pmatrix}. \quad (3.43)$$

Each quadrilateral is considered to be a surface of hyperbolic paraboloid. Then the point $(u, v) = (\alpha, \beta) \in [0, 1]^2$ at each quadrilateral can be expressed as a point on a surface of hyper-

bolic paraboloid (in Fig. 3.13 a), the procedure for $\alpha = \beta = \frac{1}{2}$ is shown)

$$\begin{aligned}\mathbf{W}_{00} &= (1 - \alpha)(1 - \beta)\mathbf{V}_{0,0} + (1 - \alpha)\beta\mathbf{V}_{0,1} + \alpha(1 - \beta)\mathbf{V}_{1,0} + \alpha\beta\mathbf{V}_{1,1}, \\ \mathbf{W}_{01} &= (1 - \alpha)(1 - \beta)\mathbf{V}_{0,1} + (1 - \alpha)\beta\mathbf{V}_{0,2} + \alpha(1 - \beta)\mathbf{V}_{1,1} + \alpha\beta\mathbf{V}_{1,2}, \\ \mathbf{W}_{10} &= (1 - \alpha)(1 - \beta)\mathbf{V}_{1,0} + (1 - \alpha)\beta\mathbf{V}_{1,1} + \alpha(1 - \beta)\mathbf{V}_{2,0} + \alpha\beta\mathbf{V}_{2,1}, \\ \mathbf{W}_{11} &= (1 - \alpha)(1 - \beta)\mathbf{V}_{1,1} + (1 - \alpha)\beta\mathbf{V}_{1,2} + \alpha(1 - \beta)\mathbf{V}_{2,1} + \alpha\beta\mathbf{V}_{2,2}.\end{aligned}\quad (3.44)$$

Points \mathbf{W}_{00} , \mathbf{W}_{01} , \mathbf{W}_{10} and \mathbf{W}_{11} are considered to be corner points of a new quadrilateral of a reduced control mesh (in Fig. 3.13 a), this quadrilateral is filled by dotted pattern).

Point $\mathbf{P}(\alpha, \beta)$ on Bézier biquadratic surface is expressed as a point on the surface of hyperbolic paraboloid given by corners \mathbf{W}_{00} , \mathbf{W}_{01} , \mathbf{W}_{10} and \mathbf{W}_{11}

$$\mathbf{P}(\alpha, \beta) = (1 - \alpha)(1 - \beta)\mathbf{W}_{0,0} + (1 - \alpha)\beta\mathbf{W}_{0,1} + \alpha(1 - \beta)\mathbf{W}_{1,0} + \alpha\beta\mathbf{W}_{1,1}.\quad (3.45)$$

Considering parameter values u and v in the whole parametrization domain and substituting (3.44) in (3.45), we get the vector equation of Bézier biquadratic surface.

The same procedure can be applied to derive vector equation of Bézier surface given by a control mesh with arbitrary number of rows and columns of control points. In case the number of rows and columns is equal (squared map), the reduced control mesh in the last step is represented by one quadrilateral. In case the number of rows and columns is different (rectangular map), the reduced control mesh in the last step is represented by only one leg of control polygon.

Bézier curve with variable control points

Consider Bézier curve $\mathbf{Q}(u)$, $u \in [0, 1]$ of which control points are not constant but a function of parameter v , see Fig. 3.13 b)

$$\mathbf{Q}(u) = B_{0,2}(u)\mathbf{V}_{0,0}(v) + B_{1,2}(u)\mathbf{V}_{1,0}(v) + B_{2,2}(u)\mathbf{V}_{2,0}(v), \quad u \in [0, 1].\quad (3.46)$$

Thus, control point $\mathbf{V}_{0,0}$ moves along Bézier curve $\mathbf{R}(v)$, control point $\mathbf{V}_{1,0}$ moves along Bézier curve $\mathbf{S}(v)$ and control point $\mathbf{V}_{2,0}$ moves along $\mathbf{T}(v)$

$$\begin{aligned}\mathbf{V}_{0,0}(v) &= \mathbf{R}(v) = B_{0,2}(v)\mathbf{V}_{0,0} + B_{1,2}(v)\mathbf{V}_{0,1} + B_{2,2}(v)\mathbf{V}_{0,2}, \quad v \in [0, 1], \\ \mathbf{V}_{1,0}(v) &= \mathbf{S}(v) = B_{0,2}(v)\mathbf{V}_{1,0} + B_{1,2}(v)\mathbf{V}_{1,1} + B_{2,2}(v)\mathbf{V}_{1,2}, \quad v \in [0, 1], \\ \mathbf{V}_{2,0}(v) &= \mathbf{T}(v) = B_{0,2}(v)\mathbf{V}_{2,0} + B_{1,2}(v)\mathbf{V}_{2,1} + B_{2,2}(v)\mathbf{V}_{2,2}, \quad v \in [0, 1].\end{aligned}\quad (3.47)$$

After substitution (3.47) in (3.46), we obtain vector equation of Bézier biquadratic surface.

In the case of arbitrary number of rows and columns of a control mesh it is possible to proceed in a similar way. Obviously, Bernstein polynomials of a corresponding degree have to be used.

3.5.4 De Casteljau algorithm for Bézier surface

De Casteljau algorithm is widely applied when constructing a point on Bézier surface. De Casteljau algorithm can be geometrically interpreted by the two above mentioned procedures of vector equation of Bézier surface derivation. Therefore, two different approaches can be distinguished.

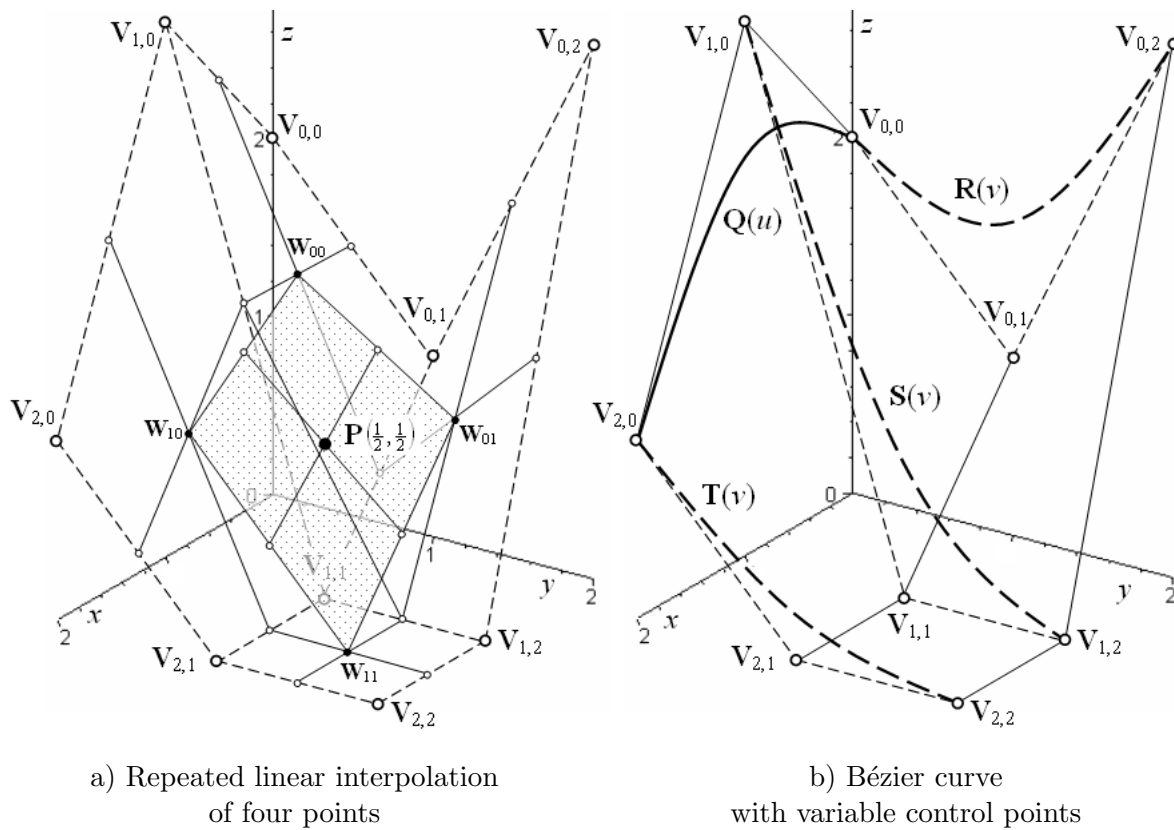


Figure 3.13: Vector equation of Bézier surface derivation

De Casteljau algorithm based on a surface character

Construction of point $\mathbf{P}(\alpha, \beta)$, $(\alpha, \beta) \in [0, 1]^2$ on Bézier surface by means of de Casteljau algorithm based on repeated linear interpolation of four points is drawn in Fig. 3.14 a).

De Casteljau algorithm based on surface character of Bézier surface includes the following steps (see Fig. 3.14 a))

1. Choose $(\alpha, \beta) \in [0, 1]^2$; in Fig. 3.14 a), the chosen values are $\alpha = \beta = \frac{1}{2}$.
2. Divide all legs of control mesh in u -direction in dividing ratio $\alpha : (1 - \alpha)$ to get points **1**, **1'**, **2**, **2'**, **3** and **3'**.
3. Divide all legs of control mesh in v -direction in dividing ratio $\beta : (1 - \beta)$ to get points **4**, **4'**, **5**, **5'**, **6** and **6'**.
4. Corners of the new quadrilateral of reduced control mesh lie at the following intersections $\mathbf{W}_{0,0} = \mathbf{12} \cap \mathbf{45}$, $\mathbf{W}_{0,1} = \mathbf{23} \cap \mathbf{4'5'}$, $\mathbf{W}_{1,0} = \mathbf{1'2'} \cap \mathbf{56}$, $\mathbf{W}_{1,1} = \mathbf{2'3'} \cap \mathbf{5'6'}$.
5. Repeat steps 2 to 4 for each reduced control mesh until the reduced control mesh contains only one quadrilateral (in the case of a squared map of Bézier surface) or only one leg of control polygon (in the case of a rectangular map of Bézier surface).

Thus, in Fig. 3.14 a), we obtain points \mathbf{A}_0 and \mathbf{A}_1 by dividing legs $\mathbf{W}_{00}\mathbf{W}_{10}$ and $\mathbf{W}_{01}\mathbf{W}_{11}$ and points \mathbf{B}_0 and \mathbf{B}_1 by dividing legs $\mathbf{W}_{00}\mathbf{W}_{01}$ and $\mathbf{W}_{10}\mathbf{W}_{11}$.

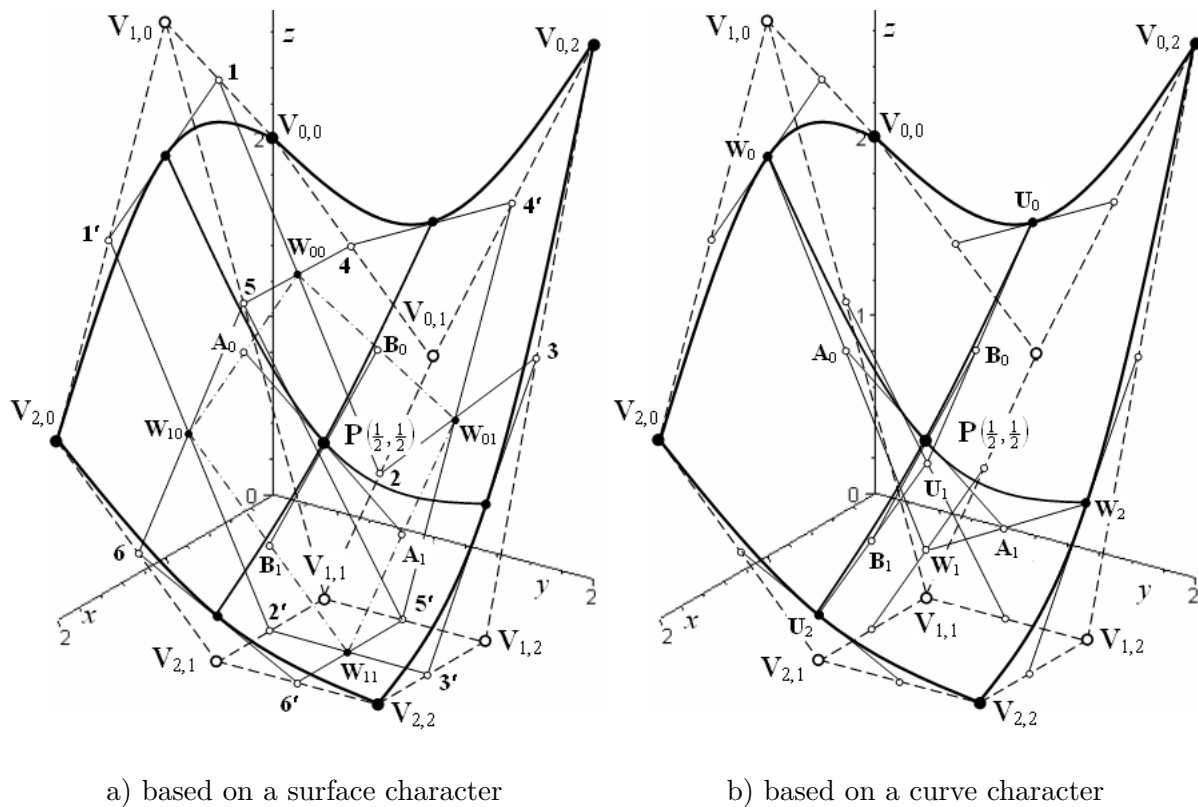


Figure 3.14: De Casteljau algorithm for Bézier surface

6. In the case of squared map of Bézier surface, point $\mathbf{P}(\alpha, \beta)$ lies at the intersection of straight line segments $\mathbf{A}_0\mathbf{A}_1$ and $\mathbf{B}_0\mathbf{B}_1$ connecting the dividing points of the last quadrilateral. These straight line segments determine tangent lines of parametric curves at point $\mathbf{P}(\alpha, \beta)$.

In the case of a rectangular map of Bézier surface, point $\mathbf{P}(\alpha, \beta)$ lies at the dividing point on the only leg of the last reduced control polygon. This leg determines the tangent line of the parametric curve of the corresponding parameter, see Fig. 3.15. If it is necessary to construct the tangent line of the second parametric curve, it has to be done by means of de Casteljau algorithm based on a curve character of Bézier surface, see below.

De Casteljau algorithm based on a curve character

If a curve character of Bézier surface is preferred, only de Casteljau algorithm described in Section 2.2.3 is applied, see Fig. 3.14 b).

1. Choose $(\alpha, \beta) \in [0, 1]^2$; in Fig. 3.14 b), the chosen values are $\alpha = \beta = \frac{1}{2}$.
2. Using de Casteljau algorithm, construct points for $u = \alpha$ on Bézier curves given by control polygons in u -direction. We obtain points \mathbf{W}_0 , \mathbf{W}_1 and \mathbf{W}_2 .
3. Using de Casteljau algorithm, construct point for $v = \beta$ on Bézier curve given by control polygon $\mathbf{W}_0\mathbf{W}_1\mathbf{W}_2$. We obtain point $\mathbf{P}(\alpha, \beta)$ on Bézier surface. The last leg of control polygon $\mathbf{A}_0\mathbf{A}_1$ determines the tangent line of parametric v -curve at point $\mathbf{P}(\alpha, \beta)$.

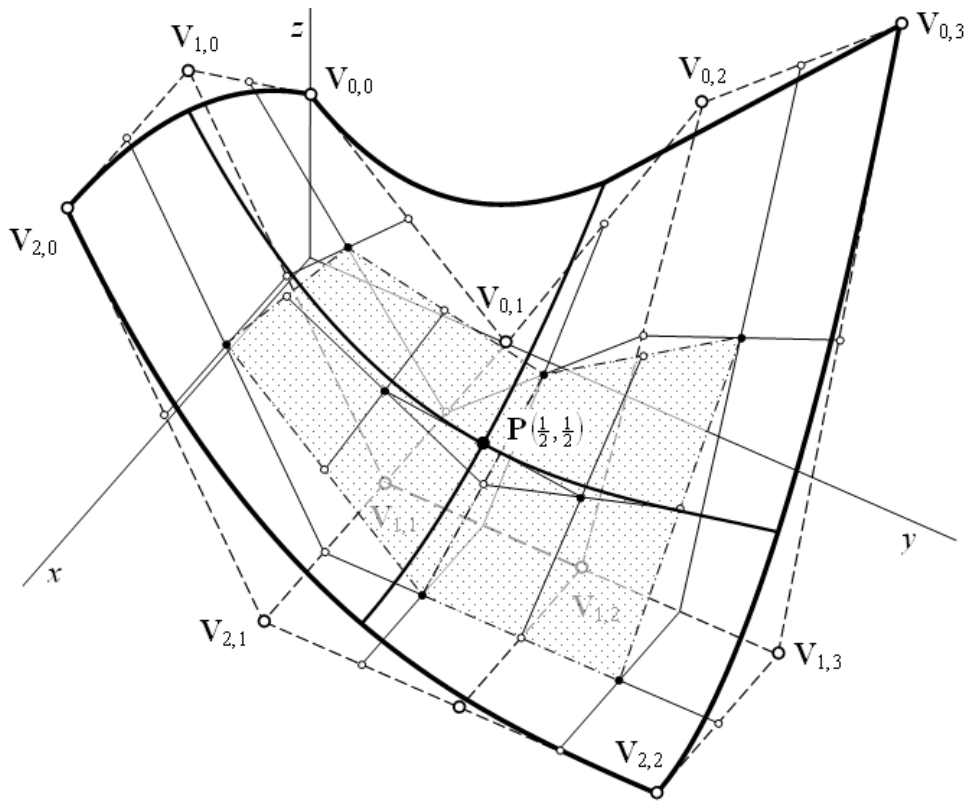


Figure 3.15: De Casteljau algorithm based on a surface character of Bézier surface (Bézier surface is given by a rectangular map)

4. Using de Casteljau algorithm, construct points for $v = \beta$ on Bézier curves given by control polygons in v -direction. We get points \mathbf{U}_0 , \mathbf{U}_1 and \mathbf{U}_2 .
5. Using de Casteljau algorithm, construct point for $u = \alpha$ on Bézier curve given by control polygon $\mathbf{U}_0\mathbf{U}_1\mathbf{U}_2$. This point is point $\mathbf{P}(\alpha, \beta)$ on Bézier surface, too. The last leg of control polygon $\mathbf{B}_0\mathbf{B}_1$ determines the tangent line of parametric u -curve at point $\mathbf{P}(\alpha, \beta)$.

In Fig. 3.14, constructions of the points on boundary curves are drawn, too. De Casteljau algorithm described in Section 2.2.3 is used for these constructions.

Remark: Straight line segments $\mathbf{A}_0\mathbf{A}_1$ and $\mathbf{B}_0\mathbf{B}_1$ determining the tangent lines of parametric curves at point $\mathbf{P}(\alpha, \beta)$ drawn in Fig. 3.14 a) are the same as straight line segments $\mathbf{A}_0\mathbf{A}_1$ and $\mathbf{B}_0\mathbf{B}_1$ drawn in Fig. 3.14 b).

■ **Example 3.12 – Point on Bézier surface construction by means of de Casteljau algorithm.** Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ is given by the following map

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} \end{pmatrix} = \begin{pmatrix} (0, 0, 2) & (0, 1, 0) & (0, 2, 1) \\ (1, 0, 0) & (1, 1, 0) & (1, 2, 0) \\ (2, 0, 1) & (2, 1, 0) & (2, 2, 0) \end{pmatrix}.$$

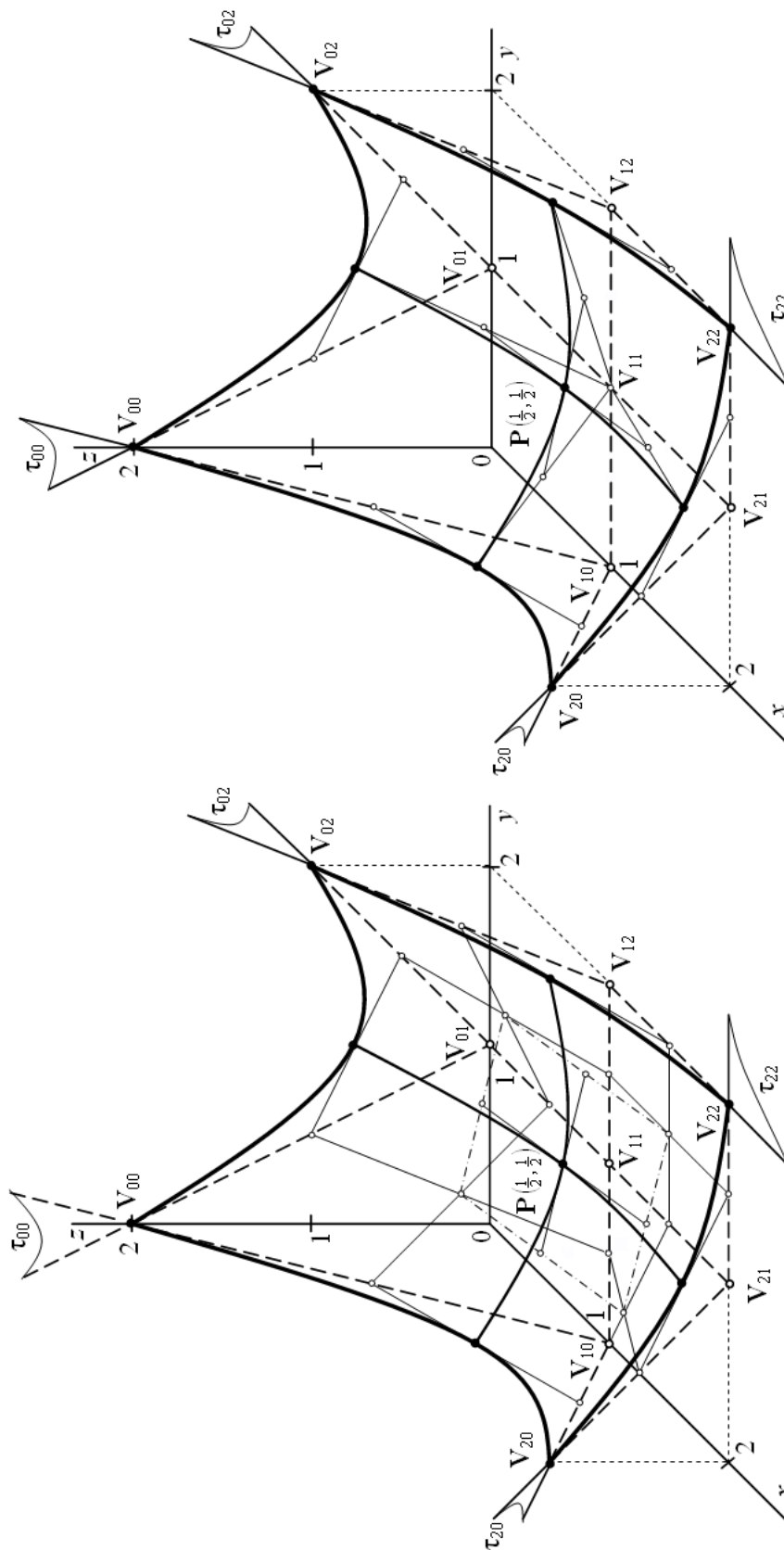


Figure 3.16: Point on Bézier surface construction by means of de Casteljau algorithm

In oblique projection ($\omega = 135^\circ, q = 2\sqrt{2} : 3$, unit = 3 cm), draw the control mesh. Construct points on Bézier surface by means of de Casteljau algorithm for the following parameter values $u, v = 0, \frac{1}{2}, 1$. Use

a) de Casteljau algorithm based on surface character,

b) de Casteljau algorithm based on curve character.

Sketch boundary Bézier curves and parametric curves passing through point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$. Indicate tangent planes at all corners.

Solution. Construction of points on Bézier surface by means of de Casteljau algorithm based on surface character is drawn in Fig. 3.16 a). Construction of points on Bézier surface by means of de Casteljau algorithm based on curve character is drawn in Fig. 3.16 b). \square

3.5.5 Patching – patches from Bézier surface

Each Bézier surface is independent of any other Bézier surface. If we want to join two Bézier surfaces with any type of continuity, it is necessary to satisfy specific requirements on positions of certain control points of joined Bézier surfaces. Here, two patches from bicubic Bézier surfaces $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$, and $\mathbf{R}(s, t)$, $(s, t) \in [0, 1]^2$, are considered. The first Bézier surface $\mathbf{P}(u, v)$ is supposed to be defined by *known* control points arranged in map \mathbf{M} , whereas the second Bézier surface $\mathbf{R}(s, t)$ is supposed to be defined by *unknown* control points arranged in map \mathbf{N} . The task is to express the unknown control points $\mathbf{W}_{i,j}$, $i, j = 0, 1, 2, 3$, in terms of the known control points $\mathbf{V}_{i,j}$, $i, j = 0, 1, 2, 3$.

Vector equations of both patches are

$$\mathbf{P}(u, v) = \mathbf{B}(u) \cdot \mathbf{M} \cdot \mathbf{B}(v), \quad (u, v) \in [0, 1]^2, \quad (3.48)$$

$$\mathbf{R}(s, t) = \mathbf{B}(s) \cdot \mathbf{N} \cdot \mathbf{B}(t), \quad (s, t) \in [0, 1]^2, \quad (3.49)$$

where $\mathbf{B}(w)$, $w = u, v, s, t$, are vectors of third degree Bernstein polynomials,

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \\ \mathbf{V}_{3,0} & \mathbf{V}_{3,1} & \mathbf{V}_{3,2} & \mathbf{V}_{3,3} \end{pmatrix} \quad \text{and} \quad \mathbf{N} = \begin{pmatrix} \mathbf{W}_{0,0} & \mathbf{W}_{0,1} & \mathbf{W}_{0,2} & \mathbf{W}_{0,3} \\ \mathbf{W}_{1,0} & \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \mathbf{W}_{1,3} \\ \mathbf{W}_{2,0} & \mathbf{W}_{2,1} & \mathbf{W}_{2,2} & \mathbf{W}_{2,3} \\ \mathbf{W}_{3,0} & \mathbf{W}_{3,1} & \mathbf{W}_{3,2} & \mathbf{W}_{3,3} \end{pmatrix} \quad (3.50)$$

is the map of the first and the second patch, respectively.

It is possible to derive the following formulas to express control points $\mathbf{W}_{i,j}$ in terms of control points $\mathbf{V}_{i,j}$

$$\mathbf{N} = \begin{pmatrix} \mathbf{W}_{0,0} = \mathbf{V}_{0,3} & \mathbf{W}_{0,1} = 2\mathbf{V}_{0,3} - \mathbf{V}_{0,2} & \mathbf{W}_{0,2} = 4(\mathbf{V}_{0,3} - \mathbf{V}_{0,2}) + \mathbf{V}_{0,1} & \mathbf{W}_{0,3} \\ \mathbf{W}_{1,0} = \mathbf{V}_{1,3} & \mathbf{W}_{1,1} = 2\mathbf{V}_{1,3} - \mathbf{V}_{1,2} & \mathbf{W}_{1,2} = 4(\mathbf{V}_{1,3} - \mathbf{V}_{1,2}) + \mathbf{V}_{1,1} & \mathbf{W}_{1,3} \\ \mathbf{W}_{2,0} = \mathbf{V}_{2,3} & \mathbf{W}_{2,1} = 2\mathbf{V}_{2,3} - \mathbf{V}_{2,2} & \mathbf{W}_{2,2} = 4(\mathbf{V}_{2,3} - \mathbf{V}_{2,2}) + \mathbf{V}_{2,1} & \mathbf{W}_{2,3} \\ \mathbf{W}_{3,0} = \mathbf{V}_{3,3} & \mathbf{W}_{3,1} = 2\mathbf{V}_{3,3} - \mathbf{V}_{3,2} & \mathbf{W}_{3,2} = 4(\mathbf{V}_{3,3} - \mathbf{V}_{3,2}) + \mathbf{V}_{3,1} & \mathbf{W}_{3,3} \end{pmatrix}, \quad (3.51)$$

so that C^2 continuity between both patches in v -direction is satisfied. In particularly:

- If C^0 continuity between both patches in v -direction is required, the position of control points $\mathbf{W}_{i,0}$, $i = 0, 1, 2, 3$, is given by the zeroth column of the map (3.51). Position of remaining control points in map \mathbf{N} is arbitrary.

- If C^1 continuity between both patches in v -direction is required, the position of control points $\mathbf{W}_{i,0}$, $i = 0, 1, 2, 3$, is given by zeroth column of the map (3.51) and the position of control points $\mathbf{W}_{i,1}$, $i = 0, 1, 2, 3$, is given by the first column of the map (3.51). Position of remaining control points in map \mathbf{N} is arbitrary.
- If C^2 continuity between both patches in v -direction is required, the position of control points $\mathbf{W}_{i,0}$, $i = 0, 1, 2, 3$, is given by zeroth column of the map (3.51), the position of control points $\mathbf{W}_{i,1}$, $i = 0, 1, 2, 3$, is given by the first column of the map (3.51) and the position of control points $\mathbf{W}_{i,2}$, $i = 0, 1, 2, 3$ is given by the second column of the map (3.51). Position of remaining control points in map \mathbf{N} is arbitrary.

The continuity in u -direction can be express in similar way.

- **Example 3.13 – Patching – patches from Bézier surface.** The following map of Bézier surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$, is given

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{0,0} & \mathbf{V}_{0,1} & \mathbf{V}_{0,2} & \mathbf{V}_{0,3} \\ \mathbf{V}_{1,0} & \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \mathbf{V}_{1,3} \\ \mathbf{V}_{2,0} & \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \mathbf{V}_{2,3} \end{pmatrix} = \begin{pmatrix} (0, 0, 1) & (0, 1, 0) & (0, 2, 2) & (0, 3, 3) \\ (1, 0, 2) & (1, 1, 0) & (1, 2, 0) & (1, 3, 0) \\ (2, 0, 2) & (2, 1, 0) & (2, 2, 0) & (2, 3, 0) \end{pmatrix}.$$

Determine by construction in oblique projection ($\omega = 135^\circ, q = 1 : 1$) and calculation positions of control points $\mathbf{W}_{i,j}$ of patch $\mathbf{R}(u, v)$ and control points $\mathbf{U}_{i,j}$ of patch $\mathbf{S}(u, v)$ from Bézier surfaces with 3×4 control points above unit squared grid so that:

- Patches $\mathbf{P}(u, v)$ and $\mathbf{R}(u, v)$ are C^2 continuously joined along the boundary given by control polygon $\mathbf{V}_{0,3}\mathbf{V}_{1,3}\mathbf{V}_{2,3}$.
- Patches $\mathbf{P}(u, v)$ and $\mathbf{S}(u, v)$ are C^1 continuously joined along the boundary given by control polygon $\mathbf{V}_{2,0}\mathbf{V}_{2,1}\mathbf{V}_{2,2}\mathbf{V}_{2,3}$.
- z -coordinates of control points without any influence on the required continuity are equal to zero.

Use de Casteljau algorithm to construct points on all boundary curves of all patches for corresponding parameter value equals $\frac{1}{2}$. Sketch boundary curves.

Solution. The constructive solution is drawn in Fig. 3.17.

Let $\mathbf{W}_{i,j}$, $i = 0, 1, 2, j = 0, 1, 2, 3$ be control points of patch $\mathbf{R}(u, v)$, $(u, v) \in [0, 1]^2$. Firstly, considering the condition of C^0 continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{R}(u, v)$, we have

$$\mathbf{W}_{0,0} = \mathbf{V}_{0,3} = (0, 3, 3), \quad \mathbf{W}_{1,0} = \mathbf{V}_{1,3} = (1, 3, 0), \quad \mathbf{W}_{2,0} = \mathbf{V}_{2,3} = (2, 3, 0).$$

Secondly, considering condition of C^1 continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{R}(u, v)$, we have

$$\begin{aligned} \mathbf{W}_{0,1} &= 2\mathbf{V}_{0,3} - \mathbf{V}_{0,2} = 2(0, 3, 3) - (0, 2, 2) = (0, 4, 4), \\ \mathbf{W}_{1,1} &= 2\mathbf{V}_{1,3} - \mathbf{V}_{1,2} = 2(1, 3, 0) - (1, 2, 0) = (1, 4, 0), \\ \mathbf{W}_{2,1} &= 2\mathbf{V}_{2,3} - \mathbf{V}_{2,2} = 2(2, 3, 0) - (2, 2, 0) = (2, 4, 0). \end{aligned}$$

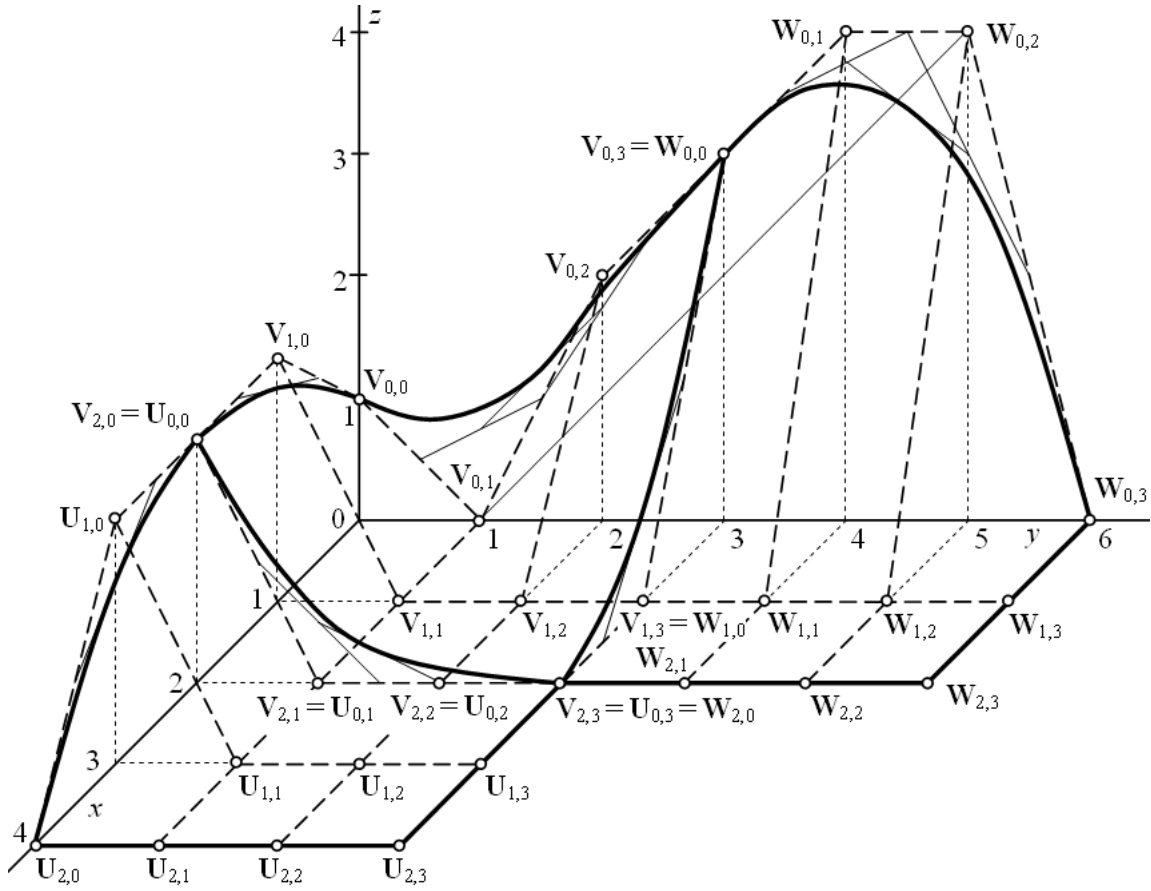


Figure 3.17: Patching – patches from Bézier surface

Third, considering condition of C^2 continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{R}(u, v)$, we have

$$\begin{aligned}\mathbf{W}_{0,2} &= 4(\mathbf{V}_{0,3} - \mathbf{V}_{0,2}) + \mathbf{V}_{0,1} = 4((0, 3, 3) - (0, 2, 2)) + (0, 1, 0) = (0, 5, 4), \\ \mathbf{W}_{1,2} &= 4(\mathbf{V}_{1,3} - \mathbf{V}_{1,2}) + \mathbf{V}_{1,1} = 4((1, 3, 0) - (1, 2, 0)) + (1, 1, 0) = (1, 5, 0), \\ \mathbf{W}_{2,2} &= 4(\mathbf{V}_{2,3} - \mathbf{V}_{2,2}) + \mathbf{V}_{2,1} = 4((2, 3, 0) - (2, 2, 0)) + (2, 1, 0) = (2, 5, 0).\end{aligned}$$

Thus, the map of patch $(u, v) \in [0, 1]^2$ is given by

$$\begin{pmatrix} \mathbf{W}_{0,0} & \mathbf{W}_{0,1} & \mathbf{W}_{0,2} & \mathbf{W}_{0,3} \\ \mathbf{W}_{1,0} & \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \mathbf{W}_{1,3} \\ \mathbf{W}_{2,0} & \mathbf{W}_{2,1} & \mathbf{W}_{2,2} & \mathbf{W}_{2,3} \end{pmatrix} = \begin{pmatrix} (0, 3, 3) & (0, 4, 4) & (0, 5, 4) & (0, 6, 0) \\ (1, 3, 0) & (1, 4, 0) & (1, 5, 0) & (1, 6, 0) \\ (2, 3, 0) & (2, 4, 0) & (2, 5, 0) & (2, 6, 0) \end{pmatrix}.$$

z -coordinate of control points $\mathbf{W}_{0,3}$, $\mathbf{W}_{1,3}$ and $\mathbf{W}_{2,3}$ are zero because position of these control points does not influence the required continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{R}(u, v)$.

Let $\mathbf{U}_{i,j}$, $i = 0, 1, 2, 3$, $j = 0, 1, 2$ $\mathbf{S}(u, v)$, $(u, v) \in [0, 1]^2$ be control points of patch $\mathbf{S}(u, v)$. Firstly, considering condition of C^0 continuity along the common boundary of patches $\mathbf{P}(u, v)$

and $\mathbf{S}(u, v)$, we have

$$\begin{aligned}\mathbf{U}_{0,0} &= \mathbf{V}_{2,0} = (2, 0, 2), & \mathbf{U}_{0,1} &= \mathbf{V}_{2,1} = (2, 1, 0), \\ \mathbf{U}_{0,2} &= \mathbf{V}_{2,2} = (2, 2, 0), & \mathbf{U}_{0,3} &= \mathbf{V}_{2,3} = (2, 3, 0).\end{aligned}$$

Secondly, considering condition of C^1 continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{S}(u, v)$, we have

$$\begin{aligned}\mathbf{U}_{1,0} &= 2\mathbf{V}_{2,0} - \mathbf{V}_{1,0} = 2(2, 0, 2) - (1, 0, 2) = (3, 0, 2), \\ \mathbf{U}_{1,1} &= 2\mathbf{V}_{2,1} - \mathbf{V}_{1,1} = 2(2, 1, 0) - (1, 1, 0) = (3, 1, 0), \\ \mathbf{U}_{1,2} &= 2\mathbf{V}_{2,2} - \mathbf{V}_{1,2} = 2(2, 2, 0) - (1, 2, 0) = (3, 2, 0), \\ \mathbf{U}_{1,3} &= 2\mathbf{V}_{2,3} - \mathbf{V}_{1,3} = 2(2, 3, 0) - (1, 3, 0) = (3, 3, 0).\end{aligned}$$

Thus, the map of patch $\mathbf{S}(u, v)$, $(u, v) \in [0, 1]^2$ is given by

$$\begin{pmatrix} \mathbf{U}_{0,0} & \mathbf{U}_{0,1} & \mathbf{U}_{0,2} & \mathbf{U}_{0,3} \\ \mathbf{U}_{1,0} & \mathbf{U}_{1,1} & \mathbf{U}_{1,2} & \mathbf{U}_{1,3} \\ \mathbf{U}_{2,0} & \mathbf{U}_{2,1} & \mathbf{U}_{2,2} & \mathbf{U}_{2,3} \end{pmatrix} = \begin{pmatrix} (2, 0, 2) & (2, 1, 0) & (2, 2, 0) & (2, 3, 0) \\ (3, 0, 2) & (3, 1, 0) & (3, 2, 0) & (3, 3, 0) \\ (4, 0, 0) & (4, 1, 0) & (4, 2, 0) & (4, 3, 0) \end{pmatrix}.$$

z -coordinate of control points $\mathbf{U}_{2,0}$, $\mathbf{U}_{2,1}$, $\mathbf{U}_{2,2}$ and $\mathbf{U}_{2,3}$ are zero because position of these control points does not influence the required continuity along the common boundary of patches $\mathbf{P}(u, v)$ and $\mathbf{S}(u, v)$. \square

3.5.6 Bézier surface in Rhinoceros

Bézier surface of $(m \times n)$ degree – Command: *Surface from Control Point Grid* → choose *Degree* in command prompt → type m in command prompt → press Enter → Number of points in row → type $m + 1$ in command prompt → press Enter → choose *Degree* in command prompt → type n in command prompt → press Enter → Number of points in column → type $n + 1$ in command prompt → press Enter → Point (1 of $m + 1$, 1 of $n + 1$): enter control point $\mathbf{V}_{0,0}$ → Point(1 of $m + 1$, 2 of $n + 1$): enter control point $\mathbf{V}_{0,1}$ → ... → Point ($m + 1$ of $m + 1$, $n + 1$ of $n + 1$): enter control point $\mathbf{V}_{m,n}$. Bézier surface of $m \times n$ degree is drawn.

Remark: The input data of Bézier surface are purposely entered in form of transposed map. The orientation of such created Bézier surface is as expected according to Definition 3.4, i.e. u parameter is oriented along columns of the map and v parameter is oriented along rows of the map, see Fig. 3.18 b).

Control points and control mesh – It is possible to display/hide control points and control mesh of drawn Bézier surface using the command *Control Points On/Points Off* → Select object for control point display: click on the surface. Control points are drawn as temporarily visible points, control mesh is drawn as a temporarily visible dotted line.

If necessary, it is possible to draw legs of control mesh as straight line segments using command *Line*.

Shape of Bézier surface modification – The shape of Bézier surface is possible to modify in the following recommended way: Draw a point in a new position → command *Point* → activate *Point object snap* → click at displayed control point which has to be modified →

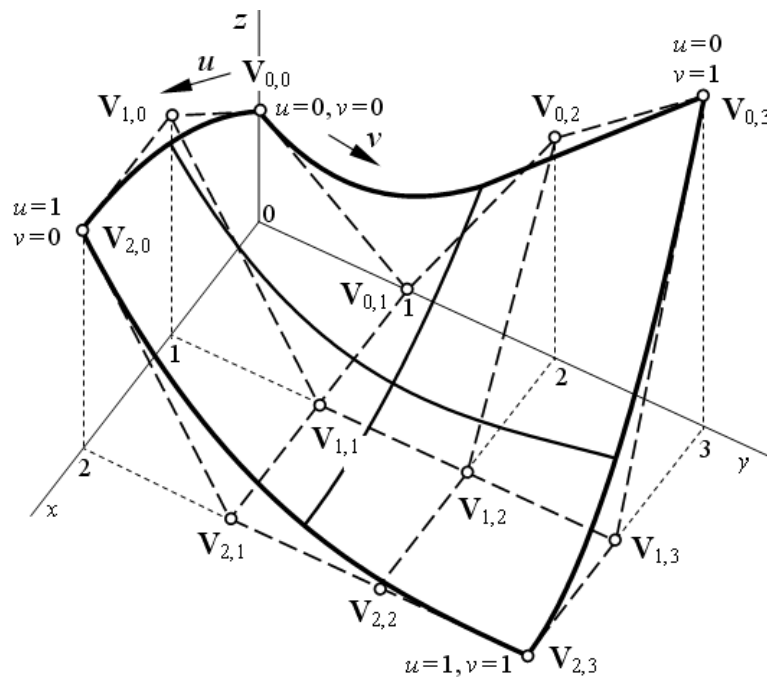


Figure 3.18: Bézier surface of $m \times n$ degree in Rhinoceros

drag the selected control point and drop it at the new position given by point drawn in previous step.

Attention! Do not use commands *Insert Knot* and *Remove Knot* to modify the shape of Bézier surface. These commands cause segmentation of the surface (piecewise surface) and nonuniform parametrization of the surface (domains of individual surface segments are different, not $[0,1]$). Do not use command *Edit Control Point Weight*, either. This command causes rational parametrization of the surface (not polynomial). After this modification, the original surface is not Bézier surface anymore, but, in general, NURBS surface (see [2]).

De Casteljau algorithm – It is possible to realize De Casteljau algorithm based on curve character according to the procedure given in Section 2.2.5. De Casteljau algorithm based on surface character supposes that all legs of control mesh are drawn as individual straight line segments. Then, the construction can be realized according to the procedure given in Section 3.5.4.

Joining Bézier surfaces – The construction of control points of joined Bézier surface according to the conditions of the required continuity is similar to the procedure given in 2.2.5. In the case of Bézier surface, we apply this construction on individual row or column control polygons.

Point $P(\alpha, \beta)$ on Bézier surface – Command: *Point from UV coordinates* → Select surface to evaluate: choose *CreatePoint=Yes* and *Normalized=Yes* in command prompt click on Bézier surface → Enter U value between 0.0 and 1.0 → type α in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type β in command prompt → press Enter → press Enter. The point with parametric coordinates $u = \alpha$ and $v = \beta$ on Bézier surface is drawn .

Parametric curves of Bézier surface – Command: *Extract Isocurve* → Select surface for isocurve extraction: click on Bézier surface; the marker moves only along the surface and the parametric curve corresponding to its actual position is dynamically displayed → Select isocurve to extract: choose *Direction=U* or *Toggle Direction=V* or *Direction=Both* → click to place the required parametric curve or the pair of parametric curves → press Enter. Depending on the chosen option in the command prompt, the parametric u -curve, parametric v -curve or both parametric curves as individual entities are drawn on.

- **Exercise 3.8** In Rhinoceros, draw surfaces from Examples 3.7 and 3.8. At $u = 0$, $v = 0, 0.1, \dots, 1$ and $u = 0, 0.1, \dots, 1$, $v = 0$, draw points on surfaces and parametric curves passing through these points.

Create tangent planes at corners of patches from Bézier surfaces. Determine, whether the corner is elliptical, hyperbolic or parabolic point.

- **Exercise 3.9** In Rhinoceros, realize Examples 3.9 and 3.10.

- **Exercise 3.10** In Rhinoceros, draw Coons bilinear surface from Example 3.11. Find Cartesian coordinates of inner control points of Bézier surface and compare them with calculated values.

- **Exercise 3.11** In Rhinoceros, draw Bézier surface from Example 3.12. Using both de Casteljau algorithm based on surface character and de Casteljau algorithm based on curve character, construct point $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$.

Practise these construction for another chosen parameters values.

- **Exercise 3.12** Realize Example 3.13 in Rhinoceros. At $u = 0$, $v = 0, 0.1, \dots, 1$ and $u = 0, 0.1, \dots, 1$, $v = 0$, draw points on Bézier surfaces and parameter curves passing through these points. Verify the required continuity by means of *Curvature Graph* and *Zebra*.

- **Example 3.14 – Patching – patches from Coons bilinear surface in Rhinoceros.** Consider patch from Coons cubic surface $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ from Example 3.6.

- (a) Draw patch $\mathbf{P}(u, v)$, $(u, v) \in [0, 1]^2$ from Coons bilinear surface.
- (b) Draw patch $\mathbf{R}(u, v)$, $(u, v) \in [0, 1]^2$ from Coons bilinear surface given by the following planar Bézier cubic curves as boundaries: $\mathbf{R}_0(u) = \mathbf{P}_1(u)$, $\mathbf{R}_0(v)$ is C^2 continuously joined with $\mathbf{P}_0(v)$, $\mathbf{R}_1(v)$ is C^2 continuously joined with $\mathbf{P}_1(v)$ and $\mathbf{R}_1(u)$ is a straight line segment in plane (x, y) drawn as Bézier cubic curve given by four collinear control points.
- (c) Draw patch $\mathbf{S}(u, v)$, $(u, v) \in [0, 1]^2$ from Coons bilinear surface given by the following planar Bézier cubic curves as boundaries: $\mathbf{S}_0(v) = \mathbf{P}_1(v)$, $\mathbf{S}_0(u)$ is C^2 continuously joined with $\mathbf{P}_0(u)$, $\mathbf{S}_1(u)$ is C^2 continuously joined with $\mathbf{P}_1(u)$ and $\mathbf{S}_1(v)$ is a straight line segment in plane (x, y) drawn as Bézier cubic curve given by four collinear control points.
- (d) Draw patch $\mathbf{T}(u, v)$, $(u, v) \in [0, 1]^2$ from Coons bilinear surface given by the following planar Bézier cubic curves as boundaries: $\mathbf{T}_0(u) = \mathbf{S}_1(u)$, $\mathbf{T}_0(v) = \mathbf{R}_1(v)$, $\mathbf{T}_1(u)$ is a straight line segment in plane (x, y) drawn as Bézier cubic curve given by four control points collinear with $\mathbf{R}_0(u)$ and $\mathbf{T}_1(v)$ is a straight line segment in plane (x, y) drawn as Bézier cubic curve given by four control points collinear with $\mathbf{S}_1(v)$.
- (e) Make a copy of all four patches. Using known interrelation of Bézier bicubic surface and Coons bilinear surface, modify inner control points of patches $\mathbf{R}(u, v)$, $\mathbf{S}(u, v)$ and $\mathbf{T}(u, v)$ so that they are C^2 continuously joined along the common boundaries.

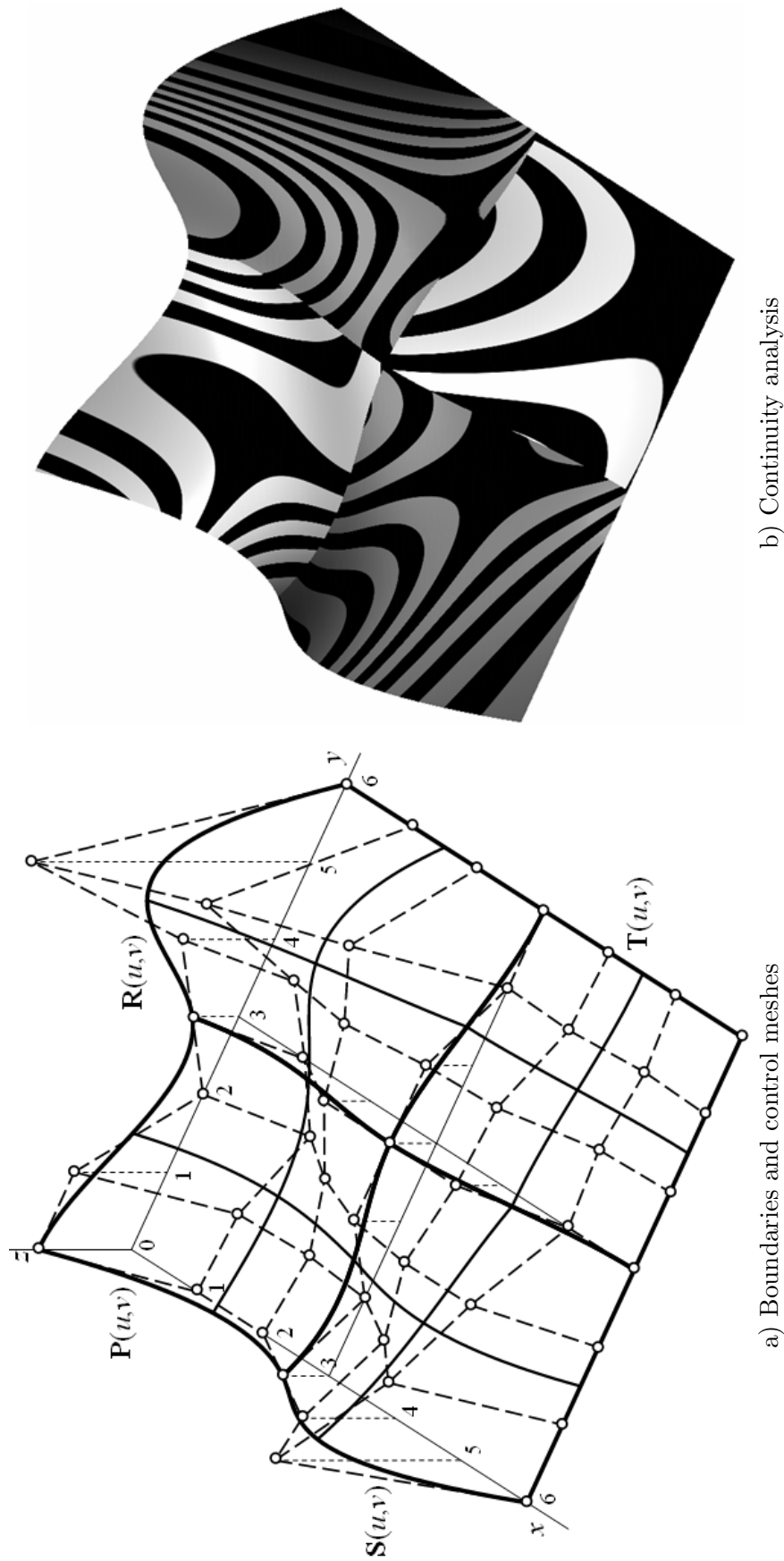


Figure 3.19: C^0 continuous patching – patches from Coons bilinear surface

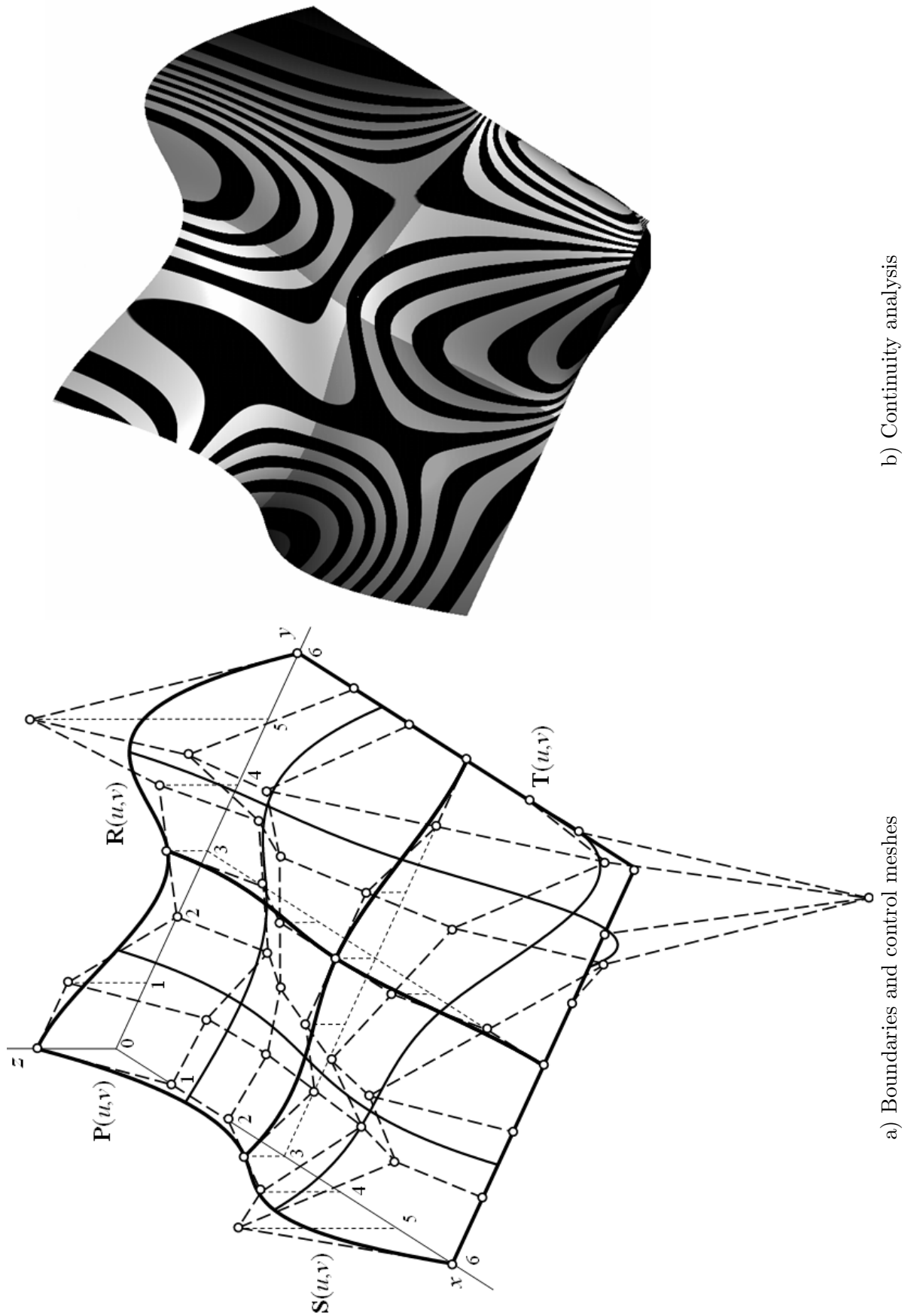


Figure 3.20: C^2 continuous patching – patches from Bézier bicubic surface

- (f) Using *Zebra* command, analyze a continuity of both original patches from Coons bilinear surfaces and modified patches from Bézier bicubic surfaces along the common boundaries.

Solution. Patches $\mathbf{P}(u, v)$, $\mathbf{R}(u, v)$, $\mathbf{S}(u, v)$ and $\mathbf{T}(u, v)$ from Coons bilinear surfaces are drawn in Fig. 3.19 a) together with control points of corresponding Bézier surfaces. Continuity analysis by means of zebra stripes is shown in Fig. 3.19 b). Obviously, the patches are joined with C^0 continuity.

Modified patches $\mathbf{P}(u, v)$, $\mathbf{R}(u, v)$, $\mathbf{S}(u, v)$ and $\mathbf{T}(u, v)$ from C^2 continuously joined Bézier bicubic surfaces are drawn in Fig. 3.20 a). Continuity analysis by means of zebra stripes is shown in Fig. 3.20 b). The patches are joined with C^2 continuity. \square

3.6 Uniform clamped bicubic B-spline surface

In hierarchy of mathematical modelling of surfaces, a *uniform clamped bicubic B-spline surface* (*clamped surface*) corresponds to a uniform clamped B-spline curve of third degree in hierarchy of mathematical modelling of curves. Clamped curve plays such a crucial role in CAD/CAM systems that it is necessary to know its properties and application possibilities. However, direct mathematical description of clamped surface is beyond the scope of this text. Therefore, neither definitions nor numerical examples are given here. A practical approach has been chosen to explain clamped curve and demonstrate its geometrical properties in Rhinoceros.

3.6.1 Properties of clamped curve

- A clamped surface is given by control mesh containing $(m + 1) \times (n + 1)$, $m, n > 4$ control points $\mathbf{P}_{i,j}$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$. Similarly to Bézier surface, the following elements can be determined on control mesh of clamped surface: corners, row and column control polygons and boundary control polygons.
- A clamped curve interpolates corners of control mesh and approximates all other control points.
- The boundaries of a patch from clamped surface are clamped curves given by boundary control polygons.
- From the last two properties it follows that Coons bilinear surface given by boundary clamped curves is a special type of a clamped surface.

Obviously, there are infinity many clamped surfaces having identical boundary curves but varying in the shape due to different position of inner control points. It is possible to find such a position of inner control points of clamped surface so that clamped surface and Coons bilinear surface given by boundaries of clamped surface are identical.

- A clamped surface is a piecewise surface because the number of its rows as well as columns is bigger than 4 (the minimal number of rows as well as columns necessary to create bicubic surface).
- A clamped surface consists of $(m - 2) \times (n - 2)$ surface segments – patches from Bézier bicubic surface.
- All patches of Bézier bicubic surface creating clamped surface are *automatically* C^2 continuously joined. Therefore, the clamped surface is C^2 continuous in the whole parametrization domain.

3.6.2 Clamped surface in Rhinoceros

Clamped surface given by control mesh – Command: *Surface from Control Point Grid* → chose *Degree* in command prompt → type 3 in command prompt → Number of points in row: type $m + 1$ in command prompt → press Enter → choose *Degree* in command prompt → type 3 in command prompt → Number of points in column: type $n + 1$ in command row → Point(1 from $m + 1$, 1 from $n + 1$): enter control point $\mathbf{P}_{0,0}$ → press Enter → Point(1 of $m + 1$, 2 of $n + 1$): enter control point $\mathbf{P}_{0,1}$ → ... → Point($m + 1$ from $m + 1$, $n + 1$ from $n + 1$): enter control point $\mathbf{V}_{m,n}$ → press Enter. Clamped curve given by $(m + 1) \times (n + 1)$ control points with expected direction as well as orientation of parameters u and v is drawn.

Remarks:

1. Parametric curves of clamped surface identical with boundaries of individual surface segments – patches from Bézier bicubic surface are automatically displayed.
2. The commands concerning control points and control mesh of clamped surface are the same as the commands used in the case of Bézier curve described in Section 3.5.6.
3. The rules for shape of clamped curve modification are the same as the rules used in the case of Bézier surface described in Section 3.5.6.

Clamped surface given by boundary curves – Before creating clamped surface by this way, four boundaries – planar or spatial curves – have to be drawn. The common points of adjacent boundaries have to lie at corners of the path.

Command: *Surface from 2, 3 or 4 Edge Curves* → Select 2, 3 or 4 curves: click on the previously drawn boundaries in the following order: $\mathbf{P}_0(v)$, $\mathbf{P}_1(v)$, $\mathbf{P}_0(u)$, $\mathbf{P}_1(u)$ → press Enter. Coons bilinear surface determined by the given boundaries is drawn. Simultaneously, control mesh of such a created surface is identical with control mesh of clamped surface given by boundaries $\mathbf{P}_0(v)$, $\mathbf{P}_1(v)$, $\mathbf{P}_0(u)$, $\mathbf{P}_1(u)$.

Attention! Obviously, the reverse procedure is not valid. Generally, a clamped surface is not identical with Coons bilinear surface given by identical boundaries.

Clamped surface decomposition – It is possible to decompose a drawn clamped surface in individual patches from Bézier bicubic surface. Activate *Endpoint object snap* → command: *Split Surface by Isocurve* → choose *Isocurve* in command prompt → choose *Shrink=Yes* → Select surface to split: click on clamped surface → Split point: click at endpoints of all automatically drawn parametric u -curves/ v -curves → choose *Toggle* in command prompt → click at endpoints of all automatically drawn parametric v -curves/ u -curves → press Enter. The clamped surface is split into $m \times n$ patches from Bézier bicubic surface. This information is displayed in command prompt, too.

Attention! C^2 continuity of decomposed clamped surface is disturbed when moving by arbitrary control points of individual patches from Bézier bicubic surface except corners of original clamped surface.

■ **Example 3.15 – Clamped surface as a composition of four patches from Bézier bicubic surface.** Consider patches $\mathbf{P}(u, v)$, $\mathbf{R}(u, v)$, $\mathbf{S}(u, v)$ and $\mathbf{T}(u, v)$ from Bézier bicubic surface from Example 3.14 drawn in Rhinoceros. Suppose that all these patches create clamped surface $\mathbf{P}^*(u, v)$.

In Rhinoceros, construct control points of clamped surface. Analyze a continuity of clamped surface by means of zebra stripes. Find Cartesian coordinates of control vertices and write a map of clamped surface.

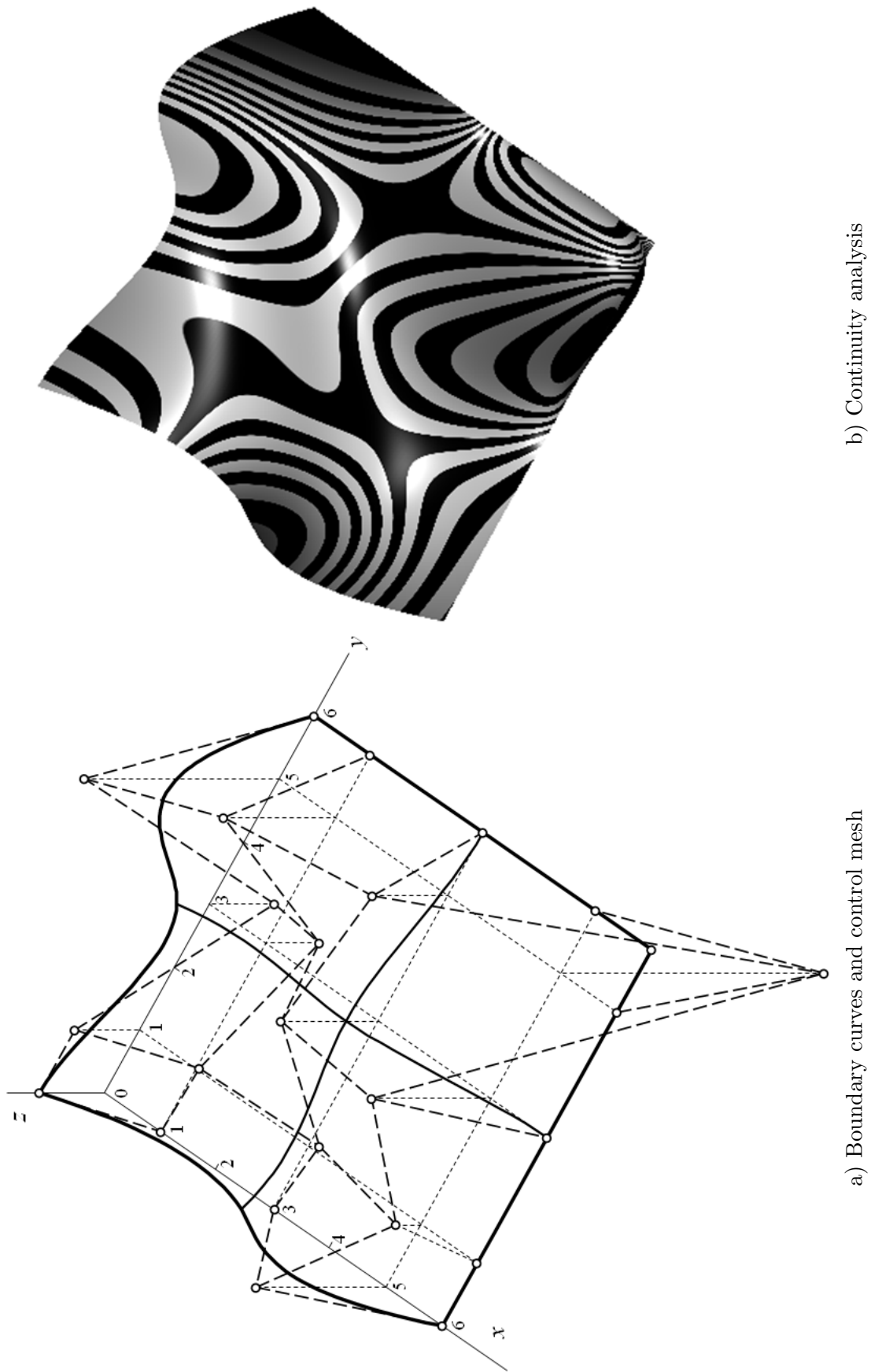


Figure 3.21: Clamped surface as a composition of four patches from Bézier bicubic surface

Solution. Clamped surface together with its control mesh is drawn in Fig. 3.23 a). Continuity analysis by means of zebra stripes is shown in Fig. 3.23 b).

Map \mathbf{M}^* of clamped surface is given by

$$\mathbf{M}^* = \begin{pmatrix} \mathbf{P}_{0,0}^* & \mathbf{P}_{0,1}^* & \mathbf{P}_{0,2}^* & \mathbf{P}_{0,3}^* & \mathbf{P}_{0,4}^* \\ \mathbf{P}_{1,0}^* & \mathbf{P}_{1,1}^* & \mathbf{P}_{1,2}^* & \mathbf{P}_{1,3}^* & \mathbf{P}_{1,4}^* \\ \mathbf{P}_{2,0}^* & \mathbf{P}_{2,1}^* & \mathbf{P}_{2,2}^* & \mathbf{P}_{2,3}^* & \mathbf{P}_{2,4}^* \\ \mathbf{P}_{3,0}^* & \mathbf{P}_{3,1}^* & \mathbf{P}_{3,2}^* & \mathbf{P}_{3,3}^* & \mathbf{P}_{3,4}^* \\ \mathbf{P}_{4,0}^* & \mathbf{P}_{4,1}^* & \mathbf{P}_{4,2}^* & \mathbf{P}_{4,3}^* & \mathbf{P}_{4,4}^* \end{pmatrix} = \begin{pmatrix} (0, 0, 2) & (0, 1, 2) & (0, 3, -2) & (0, 5, 6) & (0, 6, 0) \\ (1, 0, 0) & (1, 1, -\frac{1}{9}) & (1, 3, -\frac{5}{3}) & (1, 5, \frac{31}{9}) & (1, 6, 0) \\ (3, 0, 0) & (3, 1, -\frac{1}{3}) & (3, 3, 3) & (3, 5, \frac{7}{3}) & (3, 6, 0) \\ (5, 0, 4) & (5, 1, \frac{7}{9}) & (5, 3, \frac{11}{3}) & (5, 5, -\frac{82}{9}) & (5, 6, 0) \\ (6, 0, 0) & (6, 1, 0) & (6, 3, 0) & (6, 5, 0) & (6, 6, 0) \end{pmatrix}.$$

□

- **Example 3.16 – Clamped surface decomposition.** Boundary clamped curves of clamped surface $\mathbf{P}(u, v)$ are given by the following control polygons

$$\begin{aligned} \mathbf{P}_0(u) : \quad & \mathbf{V}_{0,0} = (0, 0, 3), \quad \mathbf{V}_{1,0} = (1, 0, 4), \quad \mathbf{V}_{2,0} = (2, 0, 3), \quad \mathbf{V}_{3,0} = (3, 0, 0), \quad \mathbf{V}_{4,0} = (4, 0, 2), \\ \mathbf{P}_1(u) : \quad & \mathbf{V}_{0,5} = (0, 5, 4), \quad \mathbf{V}_{1,5} = (1, 5, 0), \quad \mathbf{V}_{2,5} = (2, 5, 0), \quad \mathbf{V}_{3,5} = (3, 5, 0), \quad \mathbf{V}_{4,5} = (4, 5, 0), \\ \mathbf{P}_0(v) : \quad & \mathbf{V}_{0,0} = (0, 0, 3), \quad \mathbf{V}_{0,1} = (0, 1, 3), \quad \mathbf{V}_{0,2} = (0, 2, 3), \\ & \mathbf{V}_{0,3} = (0, 3, 1), \quad \mathbf{V}_{0,4} = (0, 4, 1), \quad \mathbf{V}_{0,5} = (0, 5, 4), \\ \mathbf{P}_1(v) : \quad & \mathbf{V}_{4,0} = (4, 0, 2), \quad \mathbf{V}_{4,1} = (4, 1, 0), \quad \mathbf{V}_{4,2} = (4, 2, 0), \\ & \mathbf{V}_{4,3} = (4, 3, 2), \quad \mathbf{V}_{4,4} = (4, 4, 2), \quad \mathbf{V}_{4,5} = (4, 5, 0). \end{aligned}$$

In Rhinoceros, draw boundary clamped curves and corresponding clamped surface. Analyze a continuity of the drawn clamped surface by means of zebra stripes.

How many patches from Bézier bicubic surface create clamped surface $\mathbf{P}(u, v)$?

Decompose this clamped surface into individual patches from Bézier bicubic surface. Analyze continuity of decomposed surface by means zebra stripes.

Solution. Clamped surface $\mathbf{P}(u, v)$ together with its control mesh is drawn in Fig. 3.22 a). Continuity analysis of clamped surface $\mathbf{P}(u, v)$ by means of zebra stripes is shown in Fig. 3.22 b). Since $m = 5$ and $n = 4$, the number of patches from Bézier surface creating clamped surface $\mathbf{P}(u, v)$ is

$$(m - 2) \cdot (n - 2) = 3 \cdot 2 = 6.$$

In Fig. 3.22 a), automatically displayed parametric curves – boundaries of individual patches from Bézier bicubic surfaces are drawn. It is obvious, that clamped surface $\mathbf{P}(u, v)$ is created by six patches from Bézier bicubic surface.

Decomposed clamped surface together with control meshes of individual patches from Bézier bicubic surface is drawn in Fig. 3.23 a). Continuity analysis of all individual patches by means of zebra stripes is shown in Fig. 3.23 b).

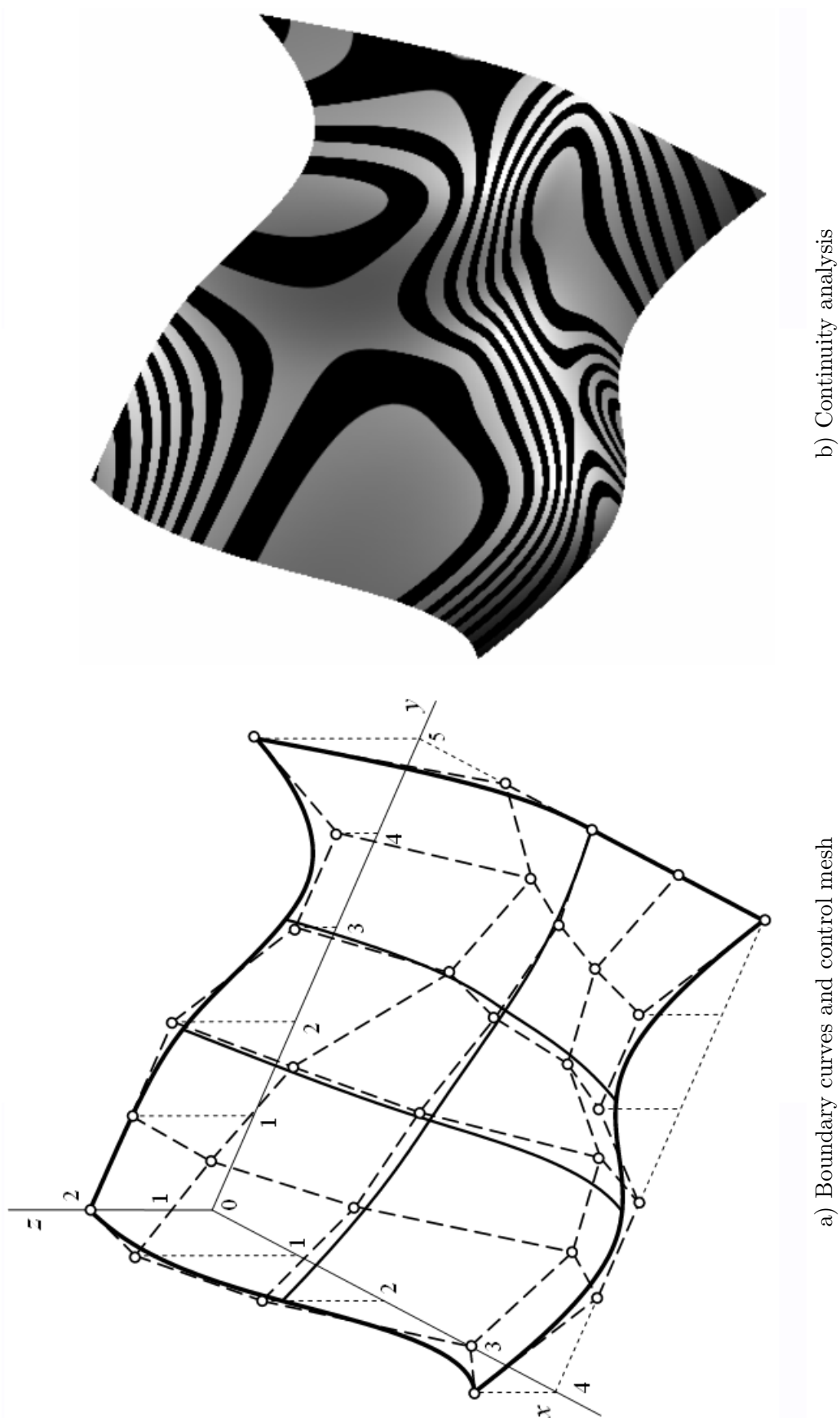


Figure 3.22: Clamped surface given by boundaries

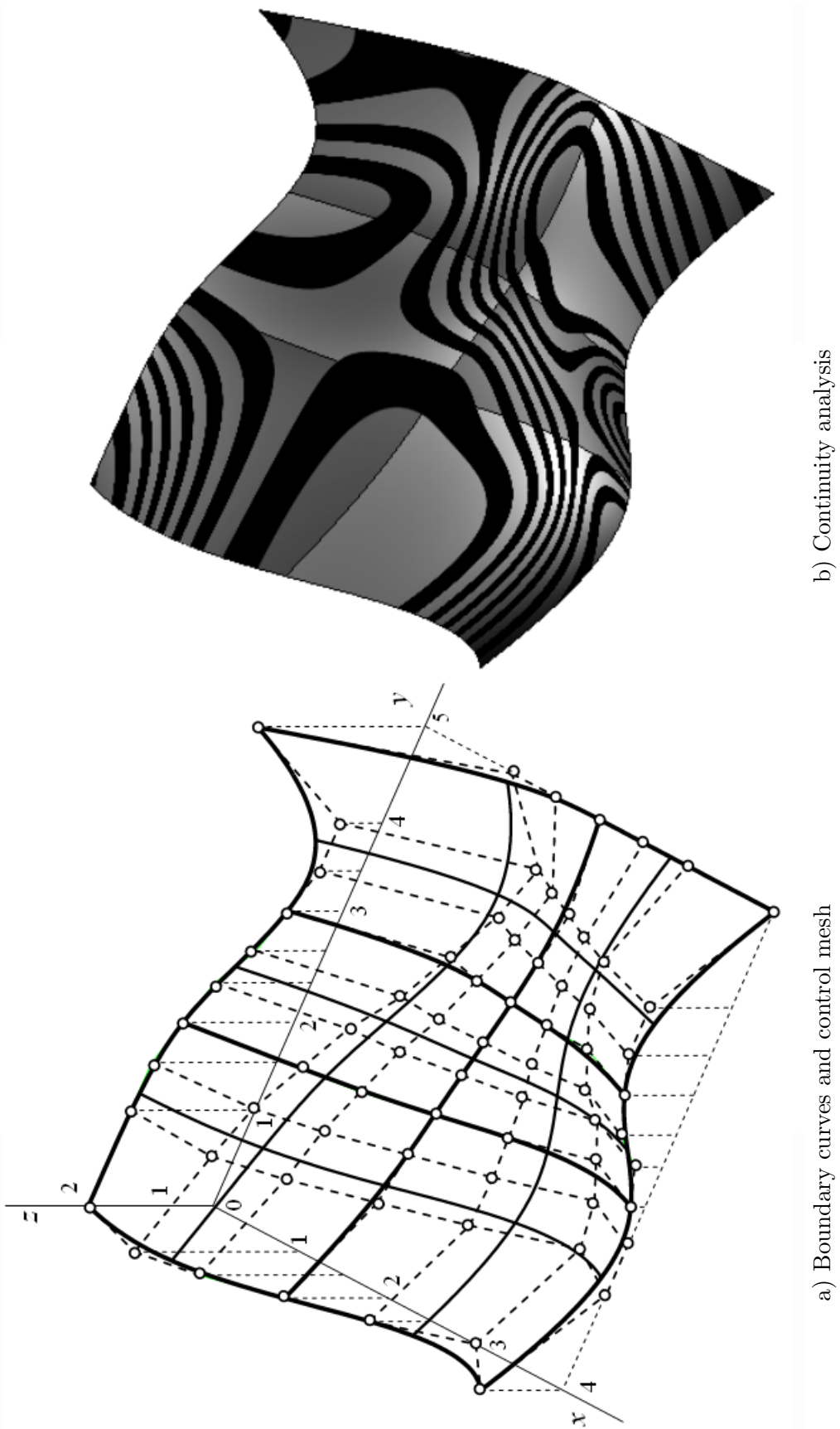


Figure 3.23: Clamped surface decomposition

□

■ **Exercise 3.13** Consider clamped surface from Example 3.16.

In Rhinoceros, determine by construction all control points of all individual patches from Bézier surface creating clamped surface. Display control points of individual patches of decomposed clamped surface and verify this construction.

3.7 Clamped surface application

In practical applications, surfaces of very complicated shape given by a huge number of control points or definition curves are widely used. In such a case, it is possible to join clamped surface together with suitable order or type of continuity to obtain free-form surface of arbitrary complicated shape. Based on interrelation between clamped surface and Bézier bicubic surface as well as Coons bilinear surface, conditions of continuity along common boundaries of individual clamped surface can be derived.

Here, a brief example of clamped surface practical application is described. In Fig. 3.24, definition curves of a part of human face are depicted. The problem is to create interpolation surface passing through all the given curves.

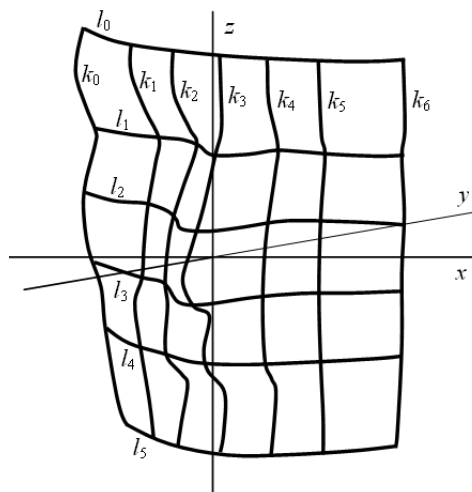


Figure 3.24: Definition curves of part of human face

To accomplish this task, all definition curves have been split at all common intersections, first. Thus, the boundaries of individual clamped surfaces have been obtained. Then, thirty individual clamped surfaces joined with C^2 continuity have been created, see Fig. 3.25. However, control mesh of this surface can be simplified. Therefore, all individual patches from clamped surface have been merged into one C^2 continuous clamped surface of the same shape, see Fig. 3.26. Here, the resulting clamped surface and its control mesh are drawn. Control mesh is drawn separately in front view, top view and profile view. The simplification of control mesh is obvious.

The surface described in this example is a part of surface for artificial human head simulator for measurements in physiological acoustics [1]. The simulator is used in an anechoic chamber located in the Acoustic Laboratory of the Department of Physics of Czech Technical University in Prague.

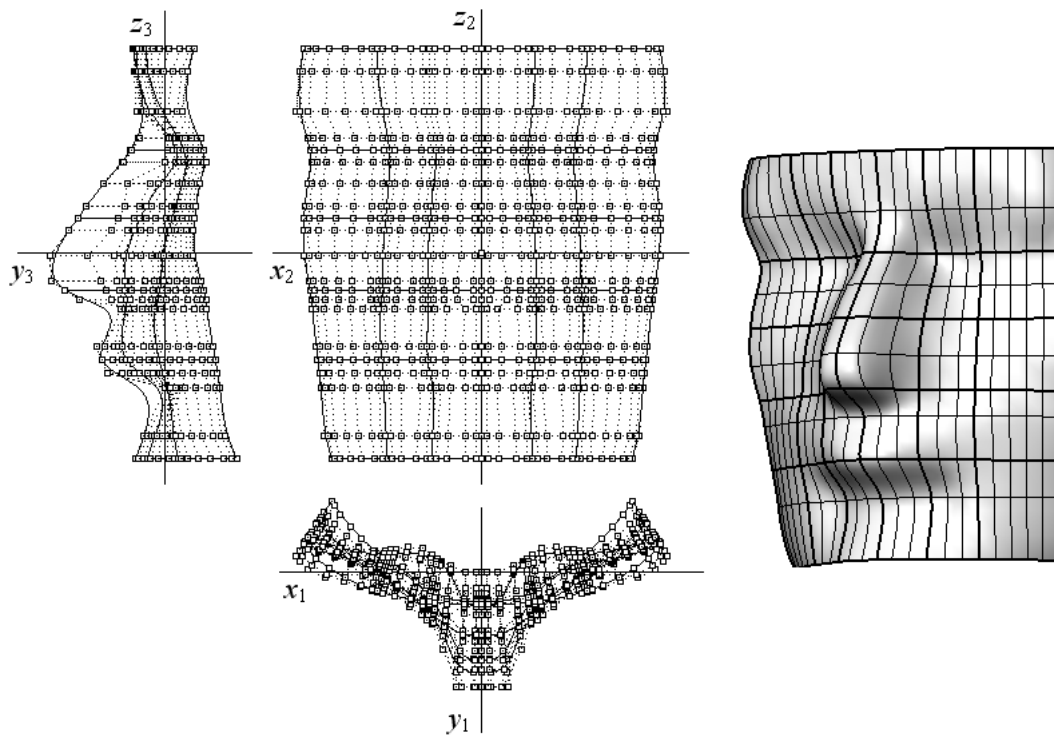


Figure 3.25: Part of human face modelling by C^2 continuously joined patches from clamped surface

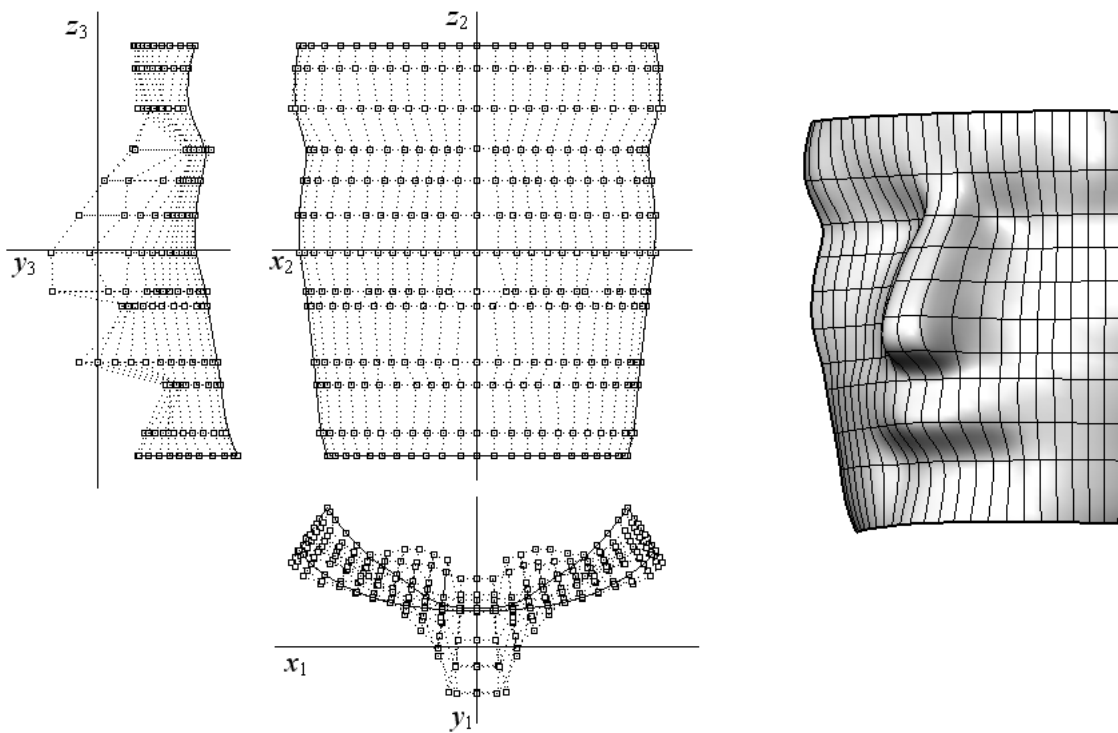


Figure 3.26: A part of human face modelling by one clamped surface

Appendix A

Rhinoceros – list of commands used

In the following list, Rhinoceros commands used in previous chapters are written in alphabetical order. For each command, the command title that appears in the tooltip for the toolbar button is given first together with corresponding toolbar title. Next, an image of toolbar button is drawn. Finally, the detailed menu path is described. Abbreviation RMB (Right Mouse Button) means to right mouse click on toolbar button or on viewport title.

It is possible to display a hidden toolbar by command *Tool* → *Toolbar Layout ...* → mark the required toolbar in checkbox.

Analyze Direction (Analyze)	 Analyze → Direction
Circle: Around Curve (Circle)	 Curve → Circle → Around Curve
Control Point Curve (Curve)	 Curve → Free-form → Control Points
Control Points On (Point Editing)	 Edit → Control Points → Control Points On
Copy (Transform)	 Transform → Copy
Curvature		RMB..Analyze → Curvature Circle
Curvature Analysis (Surface Analysis)	 Analyze → Surface → Curvature Analysis
Curvature Graph On (Analyze)	 Analyze → Curve → Curvature Graph On
Divide Curve by Number of Segments (Point)		RMB..Curve → Point Object → Divide Curve by → Number of Segments
Duplicate Border (Curve From Object)	 Curve → Curve From Objects → Duplicate Border
Duplicate Edge (Curve From Object)	 Curve → Curve From Objects → Duplicate Edge
Edit Points On (Point Editing)	 Edit → Control Points → Show Edit Points
Ellipse: From Center (Ellipse)	 Curve → Ellipse → From Center
Evaluate Point (Analyze)	 Analyze → Point
Explode (Main2)	 Edit → Explode
Extend by Arc to Points (Extend)	 Curve → Extend Curve → By Arc to Point
Extract Isocurve (Curve From Object)	 Curve → Curve From Objects → Extract Isocurve

Extract Mesh from NURBS Control Polygon (Mesh) ...		RMB..Mesh → From NURBS Control Polygon
Extract Wireframe (Curve From Object).....	 Curve → Curve From Objects → Extract Wireframe
Extrude Along Curve (Extrude)	 Surface → Extrude Curve → Along Curve
Extrude Straight (Extrude)	 Surface → Extrude Curve → Straight
Geometric Continuity of 2 Curves (Analyze)	 Analyze → Curve → Geometric Continuity
Line: from Midpoint (Lines)	 NONE
Line: Perpendicular from Curve (Lines).....	 Curve → Line → Perpendicular from Curve
Line Segments (Lines)		RMB..Curve → Line → Line Segments
Line: Surface Normal (Lines)	 Curve → Line → Normal to Surface
Line: Tangent from Curve (Lines).....	 Curve → Line → Tangent from Curve
List Object Database (Diagnostics)	 Analyze → Diagnostics → List
Mark Curve End (Point).....		RMB..Curve → Point → Mark Curve End
Mark Curve Start (Point).....	 Curve → Point → Mark Curve Start
Move (Transform)	 Transform → Move
Multiple Points (Point).....	 Curve → Point Object → Multiple Points
Object Intersection (Curve From Objects)	 Curve → Curve From Objects → Intersection
On Curve (Object Snap).....	 Tools → Object Snap → On Object → On Curve
On Surface (Object Snap)	 Tools → Object Snap → On Object → On Surface
Point from UV Coordinates (Surface Analysis).....	 Analyze → Surface → Point from UVCoordinates
Points Off (Point Editing)		RMB..Edit → Control Points → Control Points Off
Polyline (Lines).....	 Curve → Line → Line Segments
Radius (Analyze).....	 Analyze → Radius
Shaded Viewport (Standard).....	 Context menu of viewport title bar → Shaded
Single Point (Point).....	 Curve → Point Object → Single Point
Surface from 2, 3 or 4 Edge Curves (Surface).....	 Surface → Edge Curves
Surface from 3 or 4 Corner Points (Surface).....	 Surface → Corner Points
Surface from Control Point Grid (Surface)		RMB..NONE
Surface from Planar Curves (Surface)	 Surface → Planar Curves
Torus (Solid)	 Solid → Torus
UV Coordinates of a Point (Surface Analysis).....		RMB..Analyze → Surface → UV Coordinates of a Point
Zebra Analysis (Surface Analysis).....	 Analyze → Surface → Zebra

Bibliography

- [1] Linkeová, I.: Manufacturing an Artificial Human Head. In: Journal for Geometry and Graphics. 2003, vol. 7, no. 1, s. 89-99. ISSN 1433-8157.
- [2] L. Piegl, W. Tiller: The NURBS Book. Monografie. Springer, Berlin, 1997.