

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

**MODELLING AND ANALYSIS  
OF FREEFORM CURVES AND SURFACES  
IN RHINOCEROS**



Ivana Linkeová

Prague 2022

# 1 Curves analysis

**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. 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.

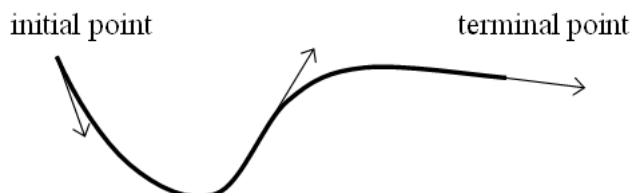


Figure 1: Curve orientation

**Curve point  $\mathbf{P}(\alpha)$**  – 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.

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. 2 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  $\alpha$  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. 2 b).

3. Remove the auxiliary cylindrical surface (select the surface and press Delete), see Fig. 2 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.

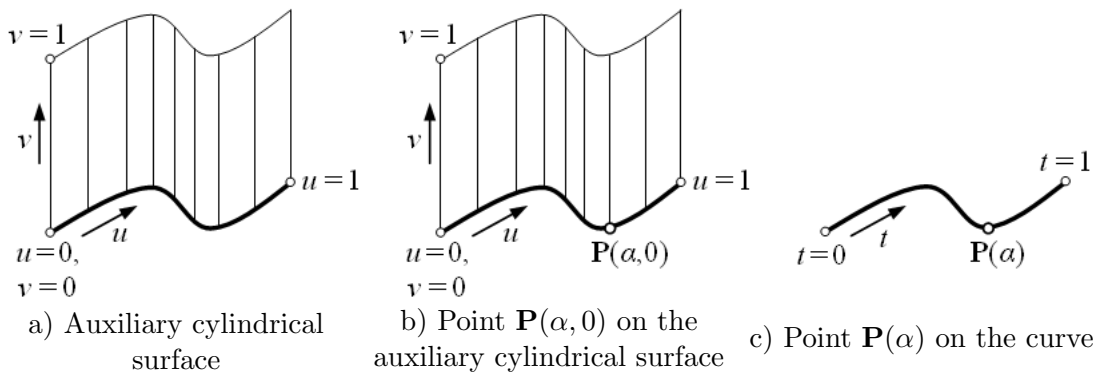


Figure 2: Construction of point  $\mathbf{P}(\alpha)$  on the curve  $\mathbf{P}(t)$ ,  $t \in [0, 1]$

**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. 3 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. 3 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. 3 c).

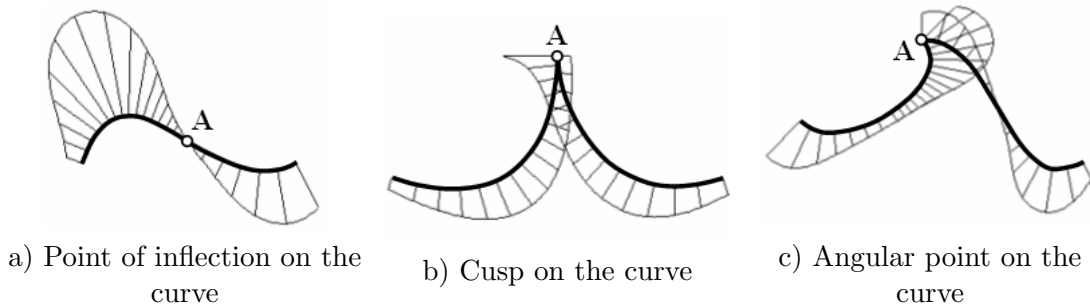


Figure 3: Classification of curve points via curvature graph

**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  $\alpha$  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. 4.

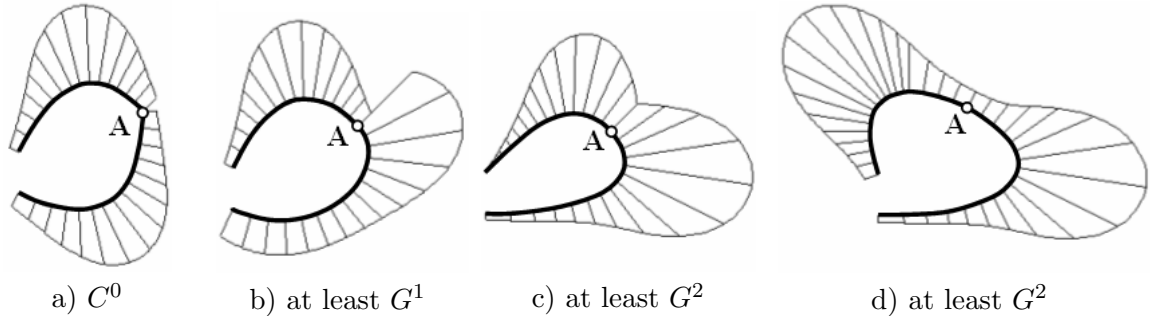


Figure 4: Continuity of two curves determination via curvature graph

In the case of  $C^0$  continuity of two curves, the perpendicular segments are intersecting, see Fig. 4 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. 4 b). In the case of  $G^2$  continuity, the curvature graphs are continuous, see Fig. 4 c) and d).

Remark: Depending on parametrization, the curves in Fig. 4 b) can be  $C^1$  continuously joined, as well as the curves in Fig. 4 c) and d) can be  $C^2$  continuously joined. However, the curvature graphs look the same way. Therefore, the words *at least* are used.

## 2 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 can be used to draw Ferguson cubic curve.

## 3 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 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  $[0, 1]$ ). 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.

**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  $\alpha$ , 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, 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: 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: 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: 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.

## 4 Coons cubic curve in Rhinoceros

In Rhinoceros, it is possible to draw Coons cubic curve only indirectly using the relation between Coons and Bézier cubic curves.

## 5 Coons cubic B-spline in Rhinoceros

**Closed Coons cubic B-spline** – Command: *Control Point Curve*  $\rightarrow$  Start of curve: type degree 3 in command prompt; enter the first control point of closed Coons cubic B-spline  $\rightarrow$  Next point: enter all control points but the last  $\rightarrow$  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.
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.



**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. 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.

## 6 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.
2. The rules for shape of clamped curve modification are the same as the rules used in the case of Bézier curve.

**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. 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.

**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 is as follows:

- **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)

## 7 Analysis of surfaces in Rhinoceros

**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  $\alpha$  in command prompt → Enter V value between 0.0 and 1.0: type the chosen value  $\beta$  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. 5 (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.

**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.

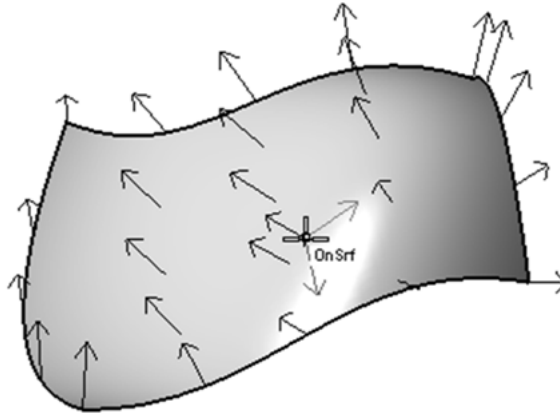


Figure 5: Surface orientation

**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  $(\alpha, \beta)$  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.

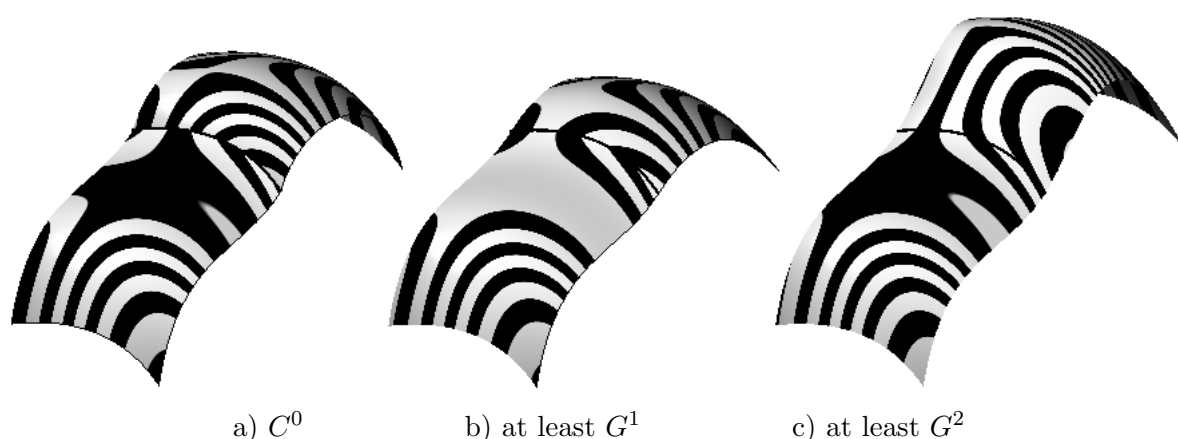


Figure 6: Analysis of continuity of two patches by means of zebra stripes

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. 6 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. 6 b).
- The patches are at least  $G^2$  continuously joined, if their zebra stripes continue smoothly as they cross the common boundary, see Fig. 6 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. 7.

## 8 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.

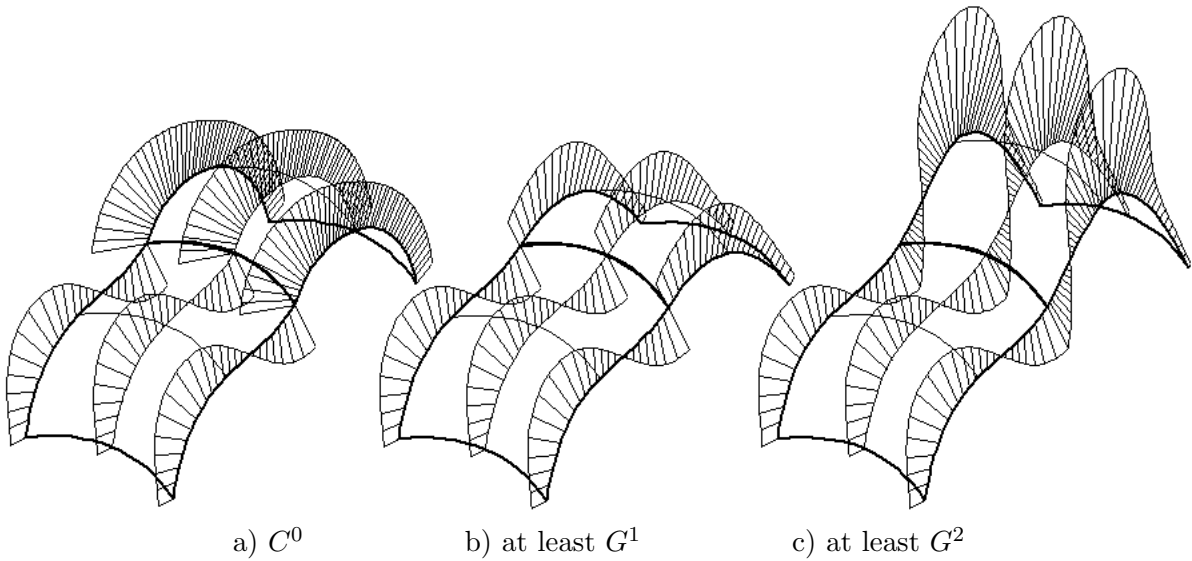


Figure 7: Analysis of continuity of two patches by means of curvature graph

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*.

**Point  $P(\alpha, \beta)$  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  $\alpha$  in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type  $\beta$  in command prompt → press Enter → press Enter. The point with parametric coordinates  $u = \alpha$  and  $v = \beta$  on the ruled surface is drawn.

**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.

## 9 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  $\mathbf{P}_{1,0}$ ). 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*.

**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  $\alpha$  in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type  $\beta$  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.

**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.

## 10 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 expectable. However, it is possible to modify an unsuitable orientation by command *Direction*.

**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  $\alpha$  in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type  $\beta$  in command prompt → press Enter → press Enter. The point with parametric coordinates  $u = \alpha$  and  $v = \beta$  on Coons bilinear surface is drawn.

**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.

## 11 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  $V_{0,0}$  → Point(1 of  $m + 1$ , 2 of  $n + 1$ ): enter control point  $V_{0,1}$  → ... → Point ( $m + 1$  of  $m + 1$ ,  $n + 1$  of  $n + 1$ ): enter control point  $V_{m,n}$ . 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, i.e.  $u$  parameter is oriented along columns of the map and  $v$  parameter is oriented along rows of the map, see Fig. 8 b).

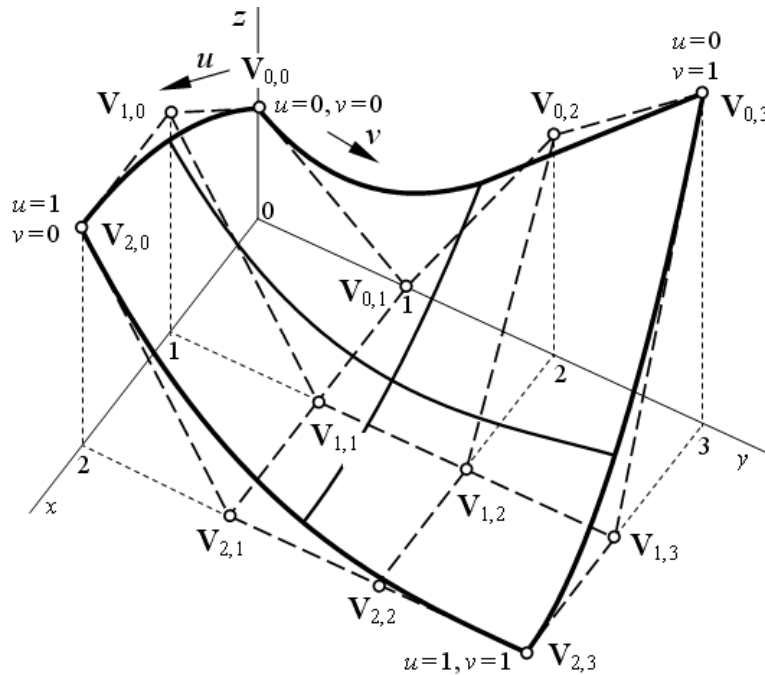


Figure 8: Bézier surface of  $m \times n$  degree in Rhinoceros

**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 → 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.

**De Casteljau algorithm** – It is possible to realize De Casteljau algorithm based on curve character. De Casteljau algorithm based on surface character supposes that all legs of control mesh are drawn as individual straight line segments.

**Joining Bézier surfaces** – The construction of control points of joined Bézier surface according to the conditions of the required continuity is similar to the procedures given in Bézier curve section. In the case of Bézier surface, we apply this construction on individual row or column control polygons.

**Point  $\mathbf{P}(\alpha, \beta)$  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  $\alpha$  in command prompt → press Enter → Enter V value between 0.0 and 1.0 → type  $\beta$  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.

## 12 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.
3. The rules for shape of clamped curve modification are the same as the rules used in the case of Bézier surface.

**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.

## 13 Rhinoceros – list of commands

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