Reprezentarea unor cazuri particulare de curbe și suprafețe

Mihai-Sorin Stupariu

Sem. al II-lea, 2022 - 2023

Conice - breviar teoretic

- **Convenție.** Din motive de simetrie, vom nota coordonatele din plan cu x_1, x_2, iar pe cele din spațiul tridimensional cu x_1, x_2, x_3. Descriem mai întâi conicele - locuri geometrice din plan, apoi, prin analogie, sunt introduse cuadricele.

Conice - breviar teoretic

- **Convenție.** Din motive de simetrie, vom nota coordonatele din plan cu x_1, x_2, iar pe cele din spațiul tridimensional cu x_1, x_2, x_3. Descriem mai întâi conicele - locuri geometrice din plan, apoi, prin analogie, sunt introduse cuadricele.
- **Definiții.** O **conică** (în planul $\mathbb{R}^2$) este o mulțime de puncte ale căror coordonate (x_1, x_2) verifică o ecuație de forma

$$a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 + 2a_{13}x_1 + 2a_{23}x_2 + a_{33} = 0, \qquad (1)$$

unde $(a_{ij})_{i,j}$ sunt coeficienți reali astfel ca $(a_{11}, a_{12}, a_{22}) \neq (0, 0, 0)$.

Conice - breviar teoretic

- **Convenție.** Din motive de simetrie, vom nota coordonatele din plan cu x_1, x_2, iar pe cele din spațiul tridimensional cu x_1, x_2, x_3. Descriem mai întâi conicele - locuri geometrice din plan, apoi, prin analogie, sunt introduse cuadricele.
- **Definiții.** O **conică** (în planul $\mathbb{R}^2$) este o mulțime de puncte ale căror coordonate (x_1, x_2) verifică o ecuație de forma

$$a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 + 2a_{13}x_1 + 2a_{23}x_2 + a_{33} = 0, \qquad (1)$$

unde $(a_{ij})_{i,j}$ sunt coeficienți reali astfel ca $(a_{11}, a_{12}, a_{22}) \neq (0, 0, 0)$.
- **Exemple.**

(i) $\dfrac{x_1^2}{9} + \dfrac{x_2^2}{25} - 1 = 0.$

(ii) $2x_1^2 + 8x_1x_2 + 10x_2^2 - 2x_1 + 2x_2 - 5 = 0.$

(iii) $x_1^2 - 2x_1x_2 - x_2^2 + 4x_1 - 4 = 0.$

(iv) $x_2^2 - 6x_1 = 0.$

(v) $x_1^2 + 2x_1x_2 + x_2^2 - 4 = 0.$

Clasificarea afină a conicelor

Pentru conica descrisă de ecuația (1) fie $a = (a_{ij})_{i,j=1,2}$ (**matricea conicei**) și $A = (a_{ij})_{i,j=1,2,3}$ (**matricea extinsă a conicei**); fie $r := \operatorname{rang} a$, $\quad R := \operatorname{rang} A$. Numerele r și R asociate unei ecuații de forma (1) nu se modifică în urma unei schimbări afine de coordonate.

Printr-o schimbare de coordonate convenabil aleasă și înmulțind, eventual, ecuația obținută cu o constantă, orice ecuație de forma (1) poate fi adusă la una din formele de mai jos:

R	r	Forma canonică afină a conicei	Denumire
3	2	$x_1^2 + x_2^2 - 1 = 0$	Elipsă
		$x_1^2 - x_2^2 - 1 = 0$	Hiperbolă
		$-x_1^2 - x_2^2 - 1 = 0$	Elipsă vidă
3	1	$x_1^2 - 2x_2 = 0$	Parabolă
2	2	$x_1^2 + x_2^2 = 0$	Punct dublu
		$x_1^2 - x_2^2 = 0$	Pereche de drepte secante
2	1	$x_1^2 - 1 = 0$	Pereche de drepte paralele
		$-x_1^2 - 1 = 0$	Pereche de drepte vidă
1	1	$x_1^2 = 0$	Dreaptă dublă

Cuadrice - breviar teoretic

O **cuadrică** este un loc geometric din spațiul $\mathbb{R}^3$ dat prin anularea unui polinom de gradul II, adică printr-o ecuație analoagă lui (1), în care apar coordonatele x_1, x_2, x_3. În mod similar se construiesc matricele a, A și definesc numerele r și R, fiind aplicate considerente și raționamente analoage celor din cazul conicelor.

Clasificarea afină a cuadricelor

R	r	Forma canonică afină a cuadricei	Denumire
4	3	$x_1^2 + x_2^2 + x_3^2 - 1 = 0$	Elipsoid
		$x_1^2 + x_2^2 - x_3^2 - 1 = 0$	Hiperboloid cu o pânză
		$x_1^2 - x_2^2 - x_3^2 - 1 = 0$	Hiperboloid cu două pânze
4	2	$x_1^2 - x_2^3 - 2x_3 = 0$	Paraboloid hiperbolic
3	3	$x_1^2 + x_2^2 - x_3^2 = 0$	Con
3	2	$x_1^2 + x_2^2 - 1 = 0$	Cilindru eliptic
		$x_1^2 - x_2^2 - 1 = 0$	Cilindru hiperbolic
3	1	$x_1^2 - 2x_2 = 0$	Cilindru parabolic
2	2	$x_1^2 + x_2^2 = 0$	Dreaptă dublă
		$x_1^2 - x_2^2 = 0$	Pereche de plane secante
2	1	$x_1^2 - 1 = 0$	Pereche de plane paralele
1	1	$x_1^2 = 0$	Plan dublu

Funcții OpenGL pentru obiecte cuadrice

- OpenGL are, în biblioteca GLU, funcții dedicate pentru următoarele obiecte cuadrice: sferă, cilindru (trunchi de con), coroană circulară și coroană circulară parțială.

Funcții OpenGL pentru obiecte cuadrice

- OpenGL are, în biblioteca GLU, funcții dedicate pentru următoarele obiecte cuadrice: sferă, cilindru (trunchi de con), coroană circulară și coroană circulară parțială.
- Pentru trasarea acestor obiecte sunt trei clase de funcții OpenGL:

Funcții OpenGL pentru obiecte cuadrice

- OpenGL are, în biblioteca GLU, funcții dedicate pentru următoarele obiecte cuadrice: sferă, cilindru (trunchi de con), coroană circulară și coroană circulară parțială.
- Pentru trasarea acestor obiecte sunt trei clase de funcții OpenGL:
  - Gestionarea obiectelor cuadrice

Funcții OpenGL pentru obiecte cuadrice

- OpenGL are, în biblioteca GLU, funcții dedicate pentru următoarele obiecte cuadrice: sferă, cilindru (trunchi de con), coroană circulară și coroană circulară parțială.
- Pentru trasarea acestor obiecte sunt trei clase de funcții OpenGL:
  - Gestionarea obiectelor cuadrice
  - Controlul proprietăților obiectului

Funcții OpenGL pentru obiecte cuadrice

- OpenGL are, în biblioteca GLU, funcții dedicate pentru următoarele obiecte cuadrice: sferă, cilindru (trunchi de con), coroană circulară și coroană circulară parțială.
- Pentru trasarea acestor obiecte sunt trei clase de funcții OpenGL:
  - Gestionarea obiectelor cuadrice
  - Controlul proprietăților obiectului
  - Desenarea propriu-zisă

Gestionarea obiectelor cuadrice

- Crearea unui nou obiect cuadric se face cu ajutorul setului de funcții

```
GLUquadricObj *qobj;
qobj=gluNewQuadric (  );
```

Gestionarea obiectelor cuadrice

- Crearea unui nou obiect cuadric se face cu ajutorul setului de funcții

  ```
  GLUquadricObj *qobj;
  qobj=gluNewQuadric (  );
  ```

- Pentru ștergerea obiectului `qobj` anterior creat se folosește funcția

  ```
  gluDeleteQuadric (qobj) ;
  ```

Controlul proprietăților obiectului

Sunt controlate două aspecte, iar funcțiile corespunzătoare trebuie să apară în codul sursă înaintea obiectelor propriu-zise.

Controlul proprietăților obiectului

Sunt controlate două aspecte, iar funcțiile corespunzătoare trebuie să apară în codul sursă înaintea obiectelor propriu-zise.

(i) Stilul de desenare a obiectului.

```
gluQuadricDrawStyle (qobj, drawstyle);
```

Controlul proprietăților obiectului

Sunt controlate două aspecte, iar funcțiile corespunzătoare trebuie să apară în codul sursă înaintea obiectelor propriu-zise.

(i) Stilul de desenare a obiectului.

```
gluQuadricDrawStyle (qobj, drawstyle);
```

Obiectul cuadric este indicat cu `obj`, iar `drawstyle` poate fi una dintre constantele simbolice `GLU_POINT, GLU_LINE, GLU_SILHOUETTE, GLU_FILL`. Stilul corespunzător lui `GLU_SILHOUETTE` este de a desena obiectul cu segmente de dreaptă, însă muchiile care separă fețe coplanare nu sunt desenate.

Controlul proprietăților obiectului

Sunt controlate două aspecte, iar funcțiile corespunzătoare trebuie să apară în codul sursă înaintea obiectelor propriu-zise.

(i) Stilul de desenare a obiectului.

```
gluQuadricDrawStyle (qobj, drawstyle);
```

Obiectul cuadric este indicat cu `obj`, iar `drawstyle` poate fi una dintre constantele simbolice `GLU_POINT`, `GLU_LINE`, `GLU_SILHOUETTE`, `GLU_FILL`. Stilul corespunzător lui `GLU_SILHOUETTE` este de a desena obiectul cu segmente de dreaptă, însă muchiile care separă fețe coplanare nu sunt desenate.

(ii) Orientarea normalelor.

```
gluQuadricNormals (qobj, orientation);
```

Controlul proprietăților obiectului

Sunt controlate două aspecte, iar funcțiile corespunzătoare trebuie să apară în codul sursă înaintea obiectelor propriu-zise.

(i) Stilul de desenare a obiectului.

```
gluQuadricDrawStyle (qobj, drawstyle);
```

Obiectul cuadric este indicat cu `obj`, iar `drawstyle` poate fi una dintre constantele simbolice `GLU_POINT, GLU_LINE, GLU_SILHOUETTE, GLU_FILL`. Stilul corespunzător lui `GLU_SILHOUETTE` este de a desena obiectul cu segmente de dreaptă, însă muchiile care separă fețe coplanare nu sunt desenate.

(ii) Orientarea normalelor.

```
gluQuadricNormals (qobj, orientation);
```

Obiectul cuadric este indicat prin `qobj`, iar orientarea (i.e. direcția spre care sunt îndreptate normalele) este indicată folosind una dintre constantele simbolice `GLU_OUTSIDE, GLU_INSIDE`.

Desenarea propriu-zisă

- O *sferă* cu centrul în $(0, 0, 0)$, de rază `r`, pentru care sunt desenate `nLong` meridiane și `nLat` paralele este desenată cu
  `gluSphere (qobj, r, nLong, nLat);`

Desenarea propriu-zisă

- O *sferă* cu centrul în $(0, 0, 0)$, de rază `r`, pentru care sunt desenate `nLong` meridiane și `nLat` paralele este desenată cu
  `gluSphere (qobj, r, nLong, nLat);`
- Un *trunchi de con* (în cazuri particulare devine cilindru) având razele celor două baze `rBase`, respectiv `rTop`, înălțimea `height`, pentru care discurile situate în partea superioară și inferioară nu sunt desenate, este randat cu
  `gluCylinder (qobj, rBase, rTop, height, nLong, nLat);`

Desenarea propriu-zisă

- O *sferă* cu centrul în $(0,0,0)$, de rază `r`, pentru care sunt desenate `nLong` meridiane și `nLat` paralele este desenată cu
  `gluSphere (qobj, r, nLong, nLat);`
- Un *trunchi de con* (în cazuri particulare devine cilindru) având razele celor două baze `rBase`, respectiv `rTop`, înălțimea `height`, pentru care discurile situate în partea superioară și inferioară nu sunt desenate, este randat cu
  `gluCylinder (qobj, rBase, rTop, height, nLong, nLat);`
- O *coroană circulară* situată în planul orizontal, având razele celor două cercuri care o determină `rInner` și `rOuter` și centrul în origine este desenată cu
  `gluDisk (qobj, rInner, rOuter, slices, rings);`

Desenarea propriu-zisă

- O *sferă* cu centrul în $(0, 0, 0)$, de rază `r`, pentru care sunt desenate `nLong` meridiane și `nLat` paralele este desenată cu
  `gluSphere (qobj, r, nLong, nLat);`
- Un *trunchi de con* (în cazuri particulare devine cilindru) având razele celor două baze `rBase`, respectiv `rTop`, înălțimea `height`, pentru care discurile situate în partea superioară și inferioară nu sunt desenate, este randat cu
  `gluCylinder (qobj, rBase, rTop, height, nLong, nLat);`
- O *coroană circulară* situată în planul orizontal, având razele celor două cercuri care o determină `rInner` și `rOuter` și centrul în origine este desenată cu
  `gluDisk (qobj, rInner, rOuter, slices, rings);`
- Mai general, se poate desena și un *sector al unei coroane circulare* ca mai sus, prin indicarea unghiurilor de în ceput și de sfârșit `startA`, `sweepA`, măsurate în grade, în sensul acelor de ceasornic:
  `gluDisk (qobj, rInner, rOuter, slices, rings, startA, sweepA);`

Curbe Bézier

- O curbă Bézier este definită de un poligon de control. Fie $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ o mulțime ordonată de puncte din $\mathbb{R}^m$, numită **poligon de control**. **Curba Bézier** $\mathrm{b} : [0,1] \to \mathbb{R}^m$ definită de poligonul de control $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ este dată de formula

$$\mathrm{b}(t) := \sum_{i=0}^{n} B_i^n(t)\mathrm{b}_i, \tag{2}$$

Definiții. unde, pentru $n \in \mathbb{N}$ fixat, **polinoamele Bernstein de grad** n sunt definite prin

$$B_i^n(t) = C_n^i t^i (1-t)^{n-i}, \quad i \in \{0, \ldots, n\},$$

unde $C_n^i = \frac{n!}{i!(n-i)!}$. Prin convenție, definim $B_i^n(t) = 0$, dacă $i \notin \{0, \ldots, n\}$.

Curbe Bézier

- O curbă Bézier este definită de un poligon de control. Fie $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ o mulțime ordonată de puncte din $\mathbb{R}^m$, numită **poligon de control**. **Curba Bézier** $\mathrm{b} : [0,1] \to \mathbb{R}^m$ definită de poligonul de control $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ este dată de formula
$$\mathrm{b}(t) := \sum_{i=0}^{n} B_i^n(t)\mathrm{b}_i, \tag{2}$$
Definiții. unde, pentru $n \in \mathbb{N}$ fixat, **polinoamele Bernstein de grad** n sunt definite prin
$$B_i^n(t) = C_n^i t^i (1-t)^{n-i}, \quad i \in \{0, \ldots, n\},$$
unde $C_n^i = \frac{n!}{i!(n-i)!}$. Prin convenție, definim $B_i^n(t) = 0$, dacă $i \notin \{0, \ldots, n\}$.
- **Exemplu** Considerăm poligonul de control
$$\mathrm{b}_0 = (1,0), \quad \mathrm{b}_1 = (1,1), \quad \mathrm{b}_2 = (0,2).$$
Curba Bézier asociată $\mathrm{b} : [0,1] \to \mathbb{R}^2$ se scrie sub forma Bernstein
$$\mathrm{b}(t) = \sum_{i=0}^{2} B_i^2(t)\mathrm{b}_i = (1-t)^2(1,0) + 2t(1-t)(1,1) + t^2(0,2) =$$
$$(1 - 2t + t^2 + 2t - 2t^2, 2t - 2t^2 + 2t^2) = (1 - t^2, 2t).$$
Avem, de exemplu, $\mathrm{b}(\frac{1}{3}) = (\frac{8}{9}, \frac{2}{3})$, $\mathrm{b}(\frac{1}{4}) = (\frac{15}{16}, \frac{1}{2})$, etc.

Curbe Bézier - Algoritmul de Casteljau

- Fie $\mathrm{b}_0, \mathrm{b}_1, \ldots, \mathrm{b}_n \in \mathbb{R}^m$. Pentru $t \in [0,1]$ se notează $\mathrm{b}_i^0(t) := \mathrm{b}_i$ $(i = 0, \ldots, n)$ și se definesc inductiv punctele de Casteljau

$$\mathrm{b}_i^r(t) := (1-t)\mathrm{b}_i^{r-1}(t) + t\mathrm{b}_{i+1}^{r-1}(t), \quad \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases} \tag{3}$$

Curbe Bézier - Algoritmul de Casteljau

- Fie $\mathrm{b}_0, \mathrm{b}_1, \ldots, \mathrm{b}_n \in \mathbb{R}^m$. Pentru $t \in [0,1]$ se notează $\mathrm{b}_i^0(t) := \mathrm{b}_i$ $(i = 0, \ldots, n)$ și se definesc inductiv punctele de Casteljau

$$\mathrm{b}_i^r(t) := (1-t)\mathrm{b}_i^{r-1}(t) + t\mathrm{b}_{i+1}^{r-1}(t), \quad \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases} \tag{3}$$

- **Teoremă.** Punctul $\mathrm{b}_0^n(t)$ descrie, când t variază, curba Bézier asociată poligonului de control $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ dată de ecuația (2). (altfel spus, cele două construcții conduc la același obiect geometric).

Curbe Bézier - Algoritmul de Casteljau

- Fie $\mathrm{b}_0, \mathrm{b}_1, \ldots, \mathrm{b}_n \in \mathbb{R}^m$. Pentru $t \in [0,1]$ se notează $\mathrm{b}_i^0(t) := \mathrm{b}_i$ $(i = 0, \ldots, n)$ și se definesc inductiv punctele de Casteljau

$$\mathrm{b}_i^r(t) := (1-t)\mathrm{b}_i^{r-1}(t) + t\mathrm{b}_{i+1}^{r-1}(t), \quad \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases} \tag{3}$$

- **Teoremă.** Punctul $\mathrm{b}_0^n(t)$ descrie, când t variază, curba Bézier asociată poligonului de control $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ dată de ecuația (2). (altfel spus, cele două construcții conduc la același obiect geometric).
- Ilustrare

Proprietăți elementare

Fie $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$ un poligon de control din $\mathbb{R}^m$. Curba Bézier asociată $\mathrm{b} : [0, 1] \to \mathbb{R}^m$ are următoarele proprietăți:

(i) b este o curbă polinomială, având gradul mai mic sau egal cu n;

(ii) curba b interpolează extremitățile poligonului de control, i.e. $\mathrm{b}(0) = \mathrm{b}_0$, $\mathrm{b}(1) = \mathrm{b}_n$; în particular, dacă poligonul de control este închis, curba Bézier asociată este închisă;

(iii) **proprietatea acoperirii convexe**: punctele curbei Bézier b se află în acoperirea convexă a punctelor de control;

(iv) **invarianța la schimbări afine de parametru**: dacă $\varphi : [0, 1] \to [\alpha, \beta]$, $\varphi(t) = \alpha + t(\beta - \alpha)$ este o schimbare afină de parametru și dacă $\mathrm{b}^{[\alpha,\beta]}$ este curba Bézier asociată poligonului de control $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$, dar definită pe intervalul $[\alpha, \beta]$, atunci $\mathrm{b} = \mathrm{b}^{[\alpha,\beta]} \circ \varphi$;

(v) **invarianță afină**: dacă $\tau : \mathbb{R}^m \to \mathbb{R}^m$ este o aplicație afină, atunci curba Bézier asociată poligonului de control date de $(\tau(\mathrm{b}_0), \ldots, \tau(\mathrm{b}_n))$ este curba $\tau(\mathrm{b}^n)$;

(vi) **(Invarianța la combinații baricentrice)**: fie $(\mathrm{b}_0, \ldots, \mathrm{b}_n)$, respectiv $(\widetilde{\mathrm{b}}_0, \ldots, \widetilde{\mathrm{b}}_n)$ două poligoane de control și b, respectiv $\widetilde{\mathrm{b}}$ curbele Bézier corespunzătoare. Pentru orice $\alpha \in \mathbb{R}$, curba Bézier asociată poligonului de control $((1-\alpha)\mathrm{b}_0 + \alpha\widetilde{\mathrm{b}}_0, \ldots, (1-\alpha)\mathrm{b}_n + \widetilde{\mathrm{b}}_n)$ este curba $(1-\alpha)\mathrm{b} + \alpha\widetilde{\mathrm{b}}$.

(vii) dacă $\widetilde{\mathrm{b}} : [0, 1] \to \mathbb{R}^m$ este curba Bézier asociată poligonului de control $(\mathrm{b}_n, \ldots, \mathrm{b}_0)$, atunci $\widetilde{\mathrm{b}}(t) = \mathrm{b}(1-t)$, în particular, cele două curbe au aceeași imagine geometrică.

Suprafețe Bézier

- O suprafață Bézier este determinată de o **rețea Bézier (poliedru de control)**.

Suprafețe Bézier

- O suprafață Bézier este determinată de o **rețea Bézier (poliedru de control)**.
- Fie $m, n \in \mathbb{N}^*$ două numere naturale nenule și

$$\begin{pmatrix} \mathrm{b}_{00} & \mathrm{b}_{01} & \dots & \mathrm{b}_{0n} \\ \mathrm{b}_{10} & \mathrm{b}_{11} & \dots & \mathrm{b}_{1n} \\ \dots & \dots & \dots & \dots \\ \mathrm{b}_{m0} & \mathrm{b}_{m1} & \dots & \mathrm{b}_{mn} \end{pmatrix}$$

  o matrice ale cărei elemente sunt puncte din $\mathbb{R}^3$, numită **rețea Bézier (poliedru de control)**. **Suprafața Bézier de tip produs tensorial** asociată acestor date este dată de formula:

$$\mathrm{s} : [0,1] \times [0,1] \to \mathbb{R}^3, \quad \mathrm{s}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u) B_j^n(v) \mathrm{b}_{ij}.$$

Suprafețe Bézier - Proprietăți elementare

(i) Prin analogie cu curbele Bézier, suprafața de tip produs tensorial are următoarele proprietăți:

- interpolarea punctelor $\mathrm{b}_{00} = \mathrm{s}(0,0), \mathrm{b}_{0n} = \mathrm{s}(0,1), \mathrm{b}_{m0} = \mathrm{s}(1,0), \mathrm{b}_{mn} = \mathrm{s}(1,1)$;
- imaginea suprafeței este inclusă în acoperirea convexă a punctelor poliedrului de control;
- invarianță la transformări afine.

(ii) Curbele frontieră, i.e. curbele de coordonate $\mathrm{s}(0,\cdot)$, $\mathrm{s}(1,\cdot)$, $\mathrm{s}(\cdot,0)$ și $\mathrm{s}(\cdot,1)$ sunt curbe Bézier având poligoane de control respectiv $(\mathrm{b}_{00}, \mathrm{b}_{01}, \ldots, \mathrm{b}_{0n})$, $(\mathrm{b}_{m0}, \mathrm{b}_{m1}, \ldots, \mathrm{b}_{mn})$, $(\mathrm{b}_{00}, \mathrm{b}_{10}, \ldots, \mathrm{b}_{m0})$, $(\mathrm{b}_{0n}, \mathrm{b}_{1n}, \ldots, \mathrm{b}_{mn})$. Restul curbelor de coordonate sunt, la rândul lor, curbe Bézier. Totuși, acestea din urmă *nu* au drept puncte de control linii sau coloane din matricea $(\mathrm{b}_{ij})_{i,j}$.

Exemplu. Dacă $m = n = 1$ avem

$$\mathrm{s}(u,v) = \sum_{i=0}^{1}\sum_{j=0}^{1} B_i^1(u)B_j^1(v)\mathrm{b}_{ij} = \begin{pmatrix} 1-u & u \end{pmatrix} \begin{pmatrix} \mathrm{b}_{00} & \mathrm{b}_{01} \\ \mathrm{b}_{10} & \mathrm{b}_{11} \end{pmatrix} \begin{pmatrix} 1-v \\ v \end{pmatrix}.$$

De exemplu, dacă

$$\mathrm{b}_{00} = (0,0,0), \quad \mathrm{b}_{01} = (1,0,0), \quad \mathrm{b}_{10} = (0,1,0), \quad \mathrm{b}_{11} = (1,1,1),$$

un calcul direct arată că $\mathrm{s}(u,v) = (v, u, uv)$.

Curbe și suprafețe Bézier în OpenGL

- OpenGL are funcții specifice pentru reprezentarea curbelor și suprafețelor Bézier. În codul sursă sunt indicate coordonate vârfurilor și proprietăți ale acestora.

Curbe și suprafețe Bézier în OpenGL

- OpenGL are funcții specifice pentru reprezentarea curbelor și suprafețelor Bézier. În codul sursă sunt indicate coordonate vârfurilor și proprietăți ale acestora.
- Atât curbele cât și suprafețele Bézier sunt obținute folosind tehnici de interpolare, care are loc la nivelul coordonatelor vârfurilor. Utilizând funcții specifice OpenGL în loc de `glVertex ( )` (de exemplu `glNormal ( )`, `glColor ( )`, `glTexCoord* ( )`), folosind aceeași regulă de interpolare sunt determinate, pe lângă punctele curbei, proprietățile lor, acestea fiind apoi manevrate ca niște vârfuri obișnuite.

Curbe și suprafețe Bézier în OpenGL

- OpenGL are funcții specifice pentru reprezentarea curbelor și suprafețelor Bézier. În codul sursă sunt indicate coordonate vârfurilor și proprietăți ale acestora.
- Atât curbele cât și suprafețele Bézier sunt obținute folosind tehnici de interpolare, care are loc la nivelul coordonatelor vârfurilor. Utilizând funcții specifice OpenGL în loc de `glVertex ( )` (de exemplu `glNormal ( )`, `glColor ( )`, `glTexCoord* ( )`), folosind aceeași regulă de interpolare sunt determinate, pe lângă punctele curbei, proprietățile lor, acestea fiind apoi manevrate ca niște vârfuri obișnuite.
- Atât în cazul curbelor, cât și al suprafețelor, funcțiile asociate pot fi grupate în două categorii: pentru definire și pentru evaluare. Mai jos sunt prezentate funcțiile folosite în cazul curbelor; în cazul suprafețelor acestea sunt similare.

Funcții pentru definire.

Pentru definirea unei curbe Bézier este apelată funcția

```
glMap1* (target, umin, umax, stride, order, *points);
```

unde `target` poate fi una dintre constantele simbolice `GL_MAP1_VERTEX3`, `GL_MAP1_COLOR4`, `GL_MAP1_NORMAL`, etc. (în funcție de datele care se interpolează); `umin`, `umax` sunt capetele intervalului de definiție, `stride`, `order` sunt parametri legați de gradul curbei, iar `*points` este un pointer catre primul punct de control (punctele de control sunt indicate în `points[][]`).
Funcția `glMap1*` este urmată de activare

```
glEnable (target);
```

Efectul funcțiilor este crearea unui evaluator care utilizează ecuațiile de definire a unei curbe Bézier.

Funcții pentru evaluare

- Funcția

  `glEvalCoord1*(u);`

  este apelată în cadrul unei funcții pentru trasarea unei primitive standard și are ca efect evaluarea coordonatelor punctului de pe curba Bézier corespunzător parametrului u (sau a altor evaluatori activați).

Funcții pentru evaluare

- Funcția

  `glEvalCoord1*(u);`

  este apelată în cadrul unei funcții pentru trasarea unei primitive standard și are ca efect evaluarea coordonatelor punctului de pe curba Bézier corespunzător parametrului u (sau a altor evaluatori activați).

- Alternativ, poate fi generată mai întâi o diviziune echidistantă a intervalului $[u_1, u_2]$ folosind funcția

  `glMapGrid*(n, u1,u2);`

  apoi se aplică diviziunea tuturor evaluatorilor activați folosind modul `mode` (`GL_POINTS`, `GL_LINE_STRIP`) pentru toți întregii dintre p_1 și p_2 cu funcția

  `glEvalMesh (mode, p1, p2);`

Funcții pentru evaluare

- Funcția

  `glEvalCoord1*(u);`

  este apelată în cadrul unei funcții pentru trasarea unei primitive standard și are ca efect evaluarea coordonatelor punctului de pe curba Bézier corespunzător parametrului u (sau a altor evaluatori activați).

- Alternativ, poate fi generată mai întâi o diviziune echidistantă a intervalului $[u_1, u_2]$ folosind funcția

  `glMapGrid*(n, u1,u2);`

  apoi se aplică diviziunea tuturor evaluatorilor activați folosind modul `mode` (`GL_ POINTS`, `GL_ LINE_STRIP`) pentru toți întregii dintre p_1 și p_2 cu funcția

  `glEvalMesh (mode, p1, p2);`

- Această funcție este echivalentă cu secvența

```
glBegin(mode);
for (i=p1; i<= p2; i++)
glEvalCoord(u1+i*(u2-u1)/n);
glEnd( );
```