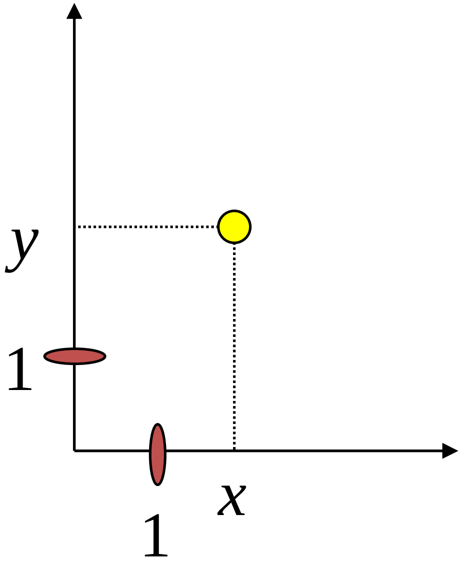
”A semmiből egy új, más világot teremtettem.”

Bolyai János

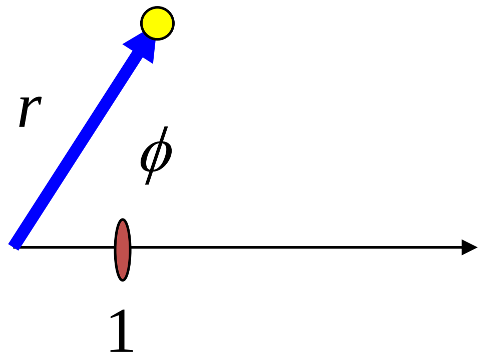
Geometriai modellezés
1. Pontok és klasszikus görbék

Szirmay-Kalos László

Pontok definíciója koordinátarendszerrel



Descartes

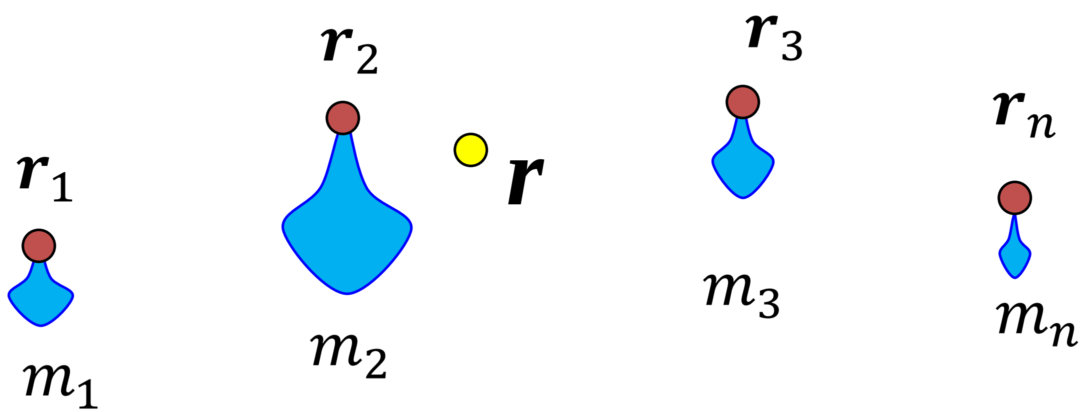


Polár

<u>Számokkal!</u>

1. Koordinátarendszer (=referencia geometria)
2. Koordináták(=mérés)

Baricentrikus (homogén) koordináták



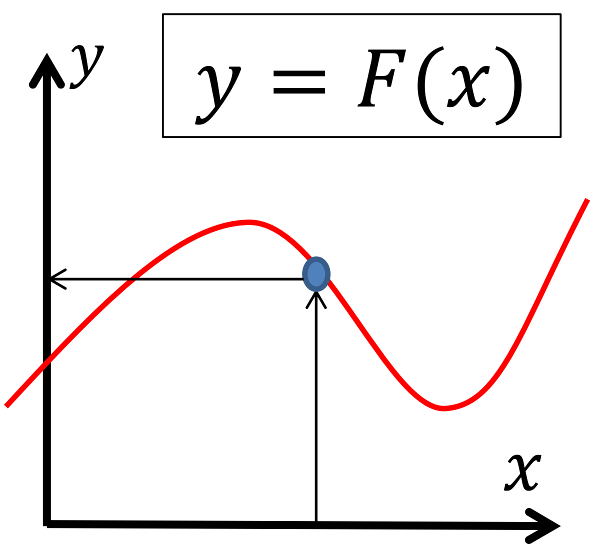
Forgatónyomaték zérus

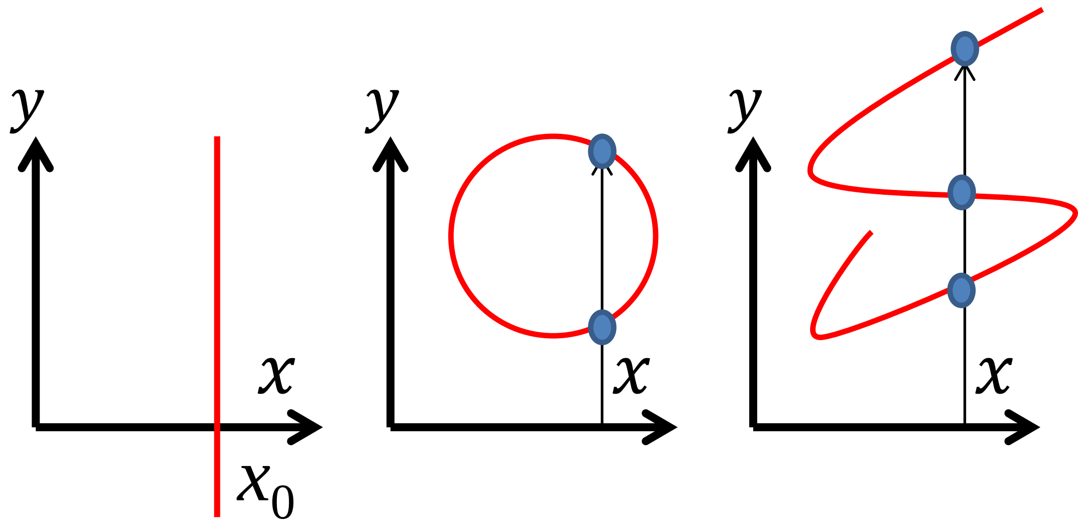
$$\sum_i (\boldsymbol{r}_i - \boldsymbol{r}) \times m_i\, \boldsymbol{g} = 0$$

$$\boxed{\boldsymbol{r} = \frac{\sum_i m_i \boldsymbol{r}_i}{\sum_i m_i}}$$

- $\boldsymbol{r}$ az $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n$ **pontok kombinációja**
- Ha a súlyok nem negatívak: **konvex kombináció**
- Konvex kombináció a **konvex burkon** belül van
- Egyenes (szakasz) = két pont (konvex) kombinációja
- Sík (háromszög) = három pont (konvex) kombinációja

Görbék: 1D ponthalmazok
Explicit egyenlet





<u>2D egyenes:</u>

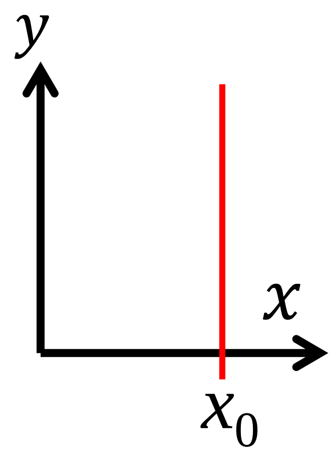
$$y = mx + b$$

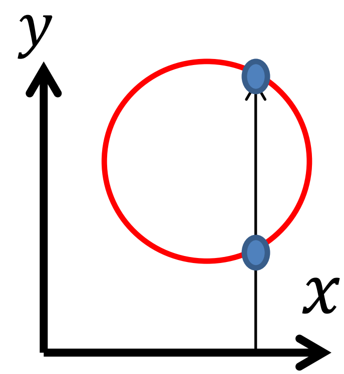
<u>Nem jó:</u>

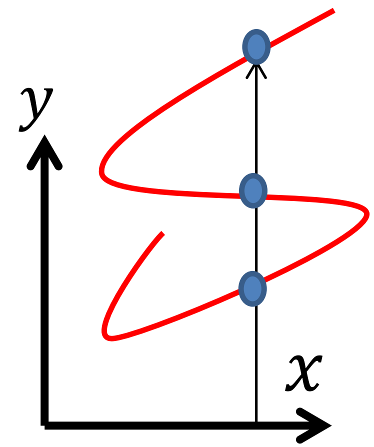
Egy x-hez nem pontosan egy y

Görbe: Implicit egyenlet

$$\boxed{f(x,y) = 0 \text{ vagy } f(\boldsymbol{r}) = 0}$$



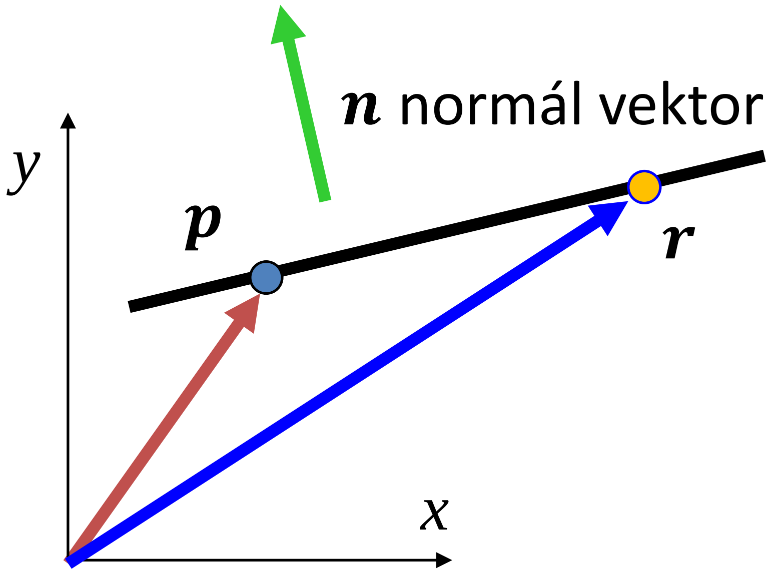




$$x - x_0 = 0$$

$$(x - c_x)^2 + \left(y - c_y\right)^2 - R^2 = 0$$

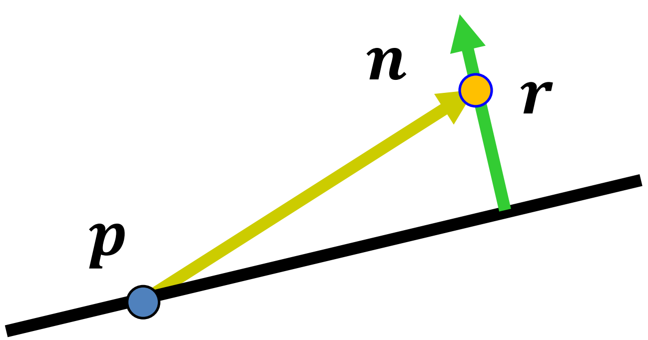
2D egyenes implicit egyenlete



$$\boldsymbol{n} \cdot (\boldsymbol{r} - \boldsymbol{p}) = 0$$
$$n_x(x - p_x) + n_y(y - p_y) = 0$$
$$ax + by + c = 0$$

Ez is: $(ax + by + c)^2 = 0$

<u>2D egyenestől mért távolság:</u>



$\boldsymbol{n} \cdot (\boldsymbol{r} - \boldsymbol{p}) =$ Vetület $\boldsymbol{n}$-re $\times$ az $\boldsymbol{n}$ hossza

Ha $\boldsymbol{n}$ egységvektor:
$\boldsymbol{n} \cdot (\boldsymbol{r} - \boldsymbol{p}) =$ az előjeles távolság!

Kvadratikus görbék

- **Kör:** Azon $\boldsymbol{r}(x,y)$ pontok mértani helye, amelyek a $\boldsymbol{c}(c_x,c_y)$ középpponttól R távolságra vannak: $|\boldsymbol{r}-\boldsymbol{c}| = R \leftrightarrow$ $(\boldsymbol{r}-\boldsymbol{c})^2 - R^2 = 0 \leftrightarrow \ (x-c_x)^2 + \left(y-c_y\right)^2 - R^2 = 0$
- **Ellipszis:** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{f}_1$és $\boldsymbol{f}_2$ fókuszpontoktól mért távolság összege állandó C: $|\boldsymbol{r}-\boldsymbol{f}_1| + |\boldsymbol{r}-\boldsymbol{f}_2| = C$
- **Hiperbola:** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{f}_1$és $\boldsymbol{f}_2$ fókuszpontoktól mért távolság különbsége állandó C: $|\boldsymbol{r}-\boldsymbol{f}_1| - |\boldsymbol{r}-\boldsymbol{f}_2| = C$
- **Parabola:** Azon $\boldsymbol{r}$ pontok, amelyek az $\boldsymbol{f}$ fókuszponttól mért távolsága megegyezik az $\boldsymbol{n}$ normálvektorú és $\boldsymbol{p}$ helyvektorú egyenestől mért távolsággal: $|\boldsymbol{r}-\boldsymbol{f}| = |\boldsymbol{n}^0 \cdot (\boldsymbol{r}-\boldsymbol{p})|$

Kvadratikus görbék = kvadratikus alak

- Implicit függvény négyzetgyökök nélkül:

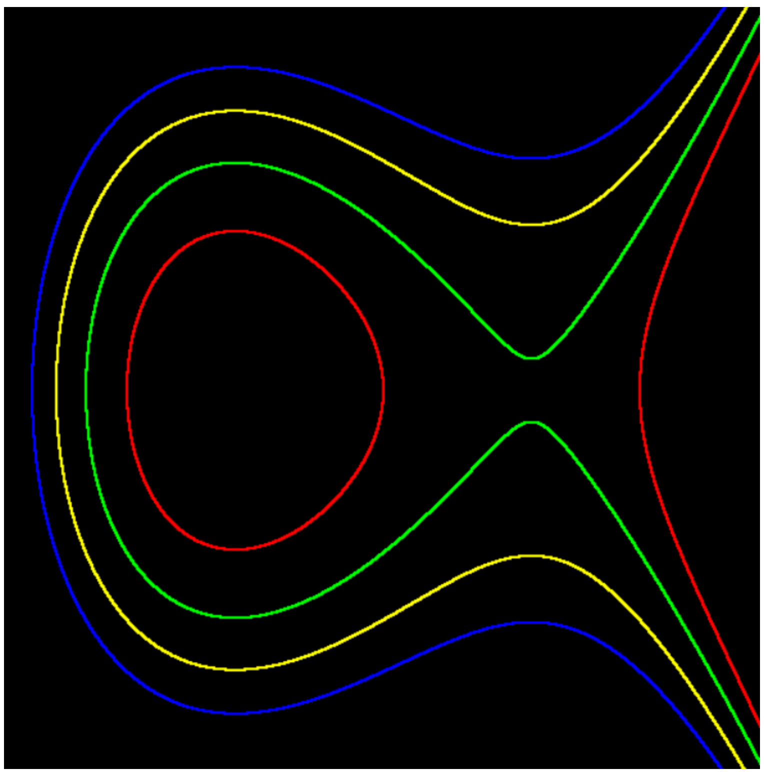
$$f(x, y) = a_{11}x^2 + a_{22}y^2 + 2a_{12}xy + 2a_{13}x + 2a_{23}y + a_{33} = 0$$

- Mátrixszal:

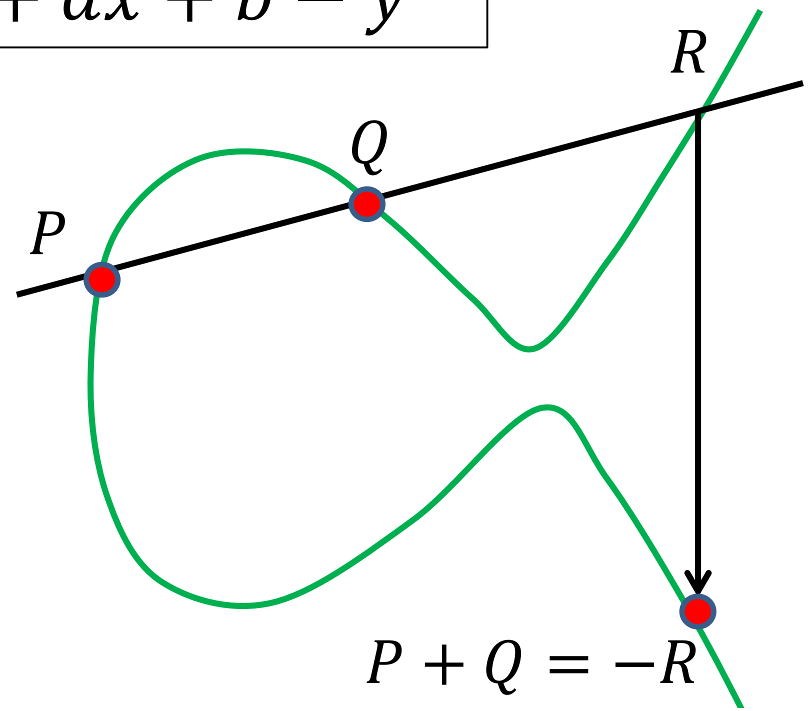
$$\boxed{f(x, y) = [x, y, 1] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0}$$

Elliptikus görbék

$$f(x,y) = x^3 + ax + b - y^2$$



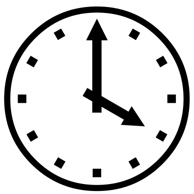
$a = -1, b = 0, \ldots, 1.6$



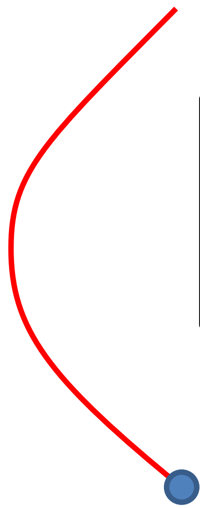
Ezzel az összeadással **csoport**,
Nem vezet ki a racionális számokból

Alkalmazás: Kriptográfia, számelmélet (Fermat tétel bizonyítás)

Görbe: Paraméteres egyenlet



t



$$x = x(t), y = y(t), z = z(t)$$
vagy $\boldsymbol{r} = \boldsymbol{r}(t)$

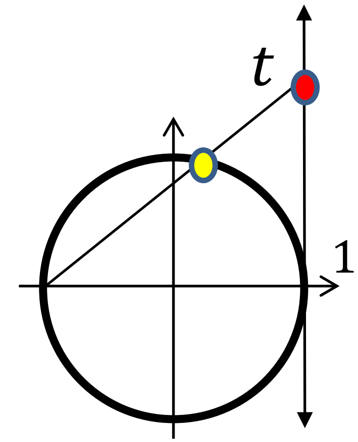
3D egyenes:

$$x(t) = x_0 + v_x t$$
$$y(t) = y_0 + v_y t$$
$$z(t) = z_0 + v_z t$$
$$t \in (-\infty, +\infty)$$

Kör:

$$x(t) = c_x + R\cos(t)$$
$$y(t) = c_y + R\sin(t)$$
$$t \in [0,2\pi)$$

Klasszikus görbék



- **Kör:**

$$x(t) = \frac{(4-t^2)}{(4+t^2)} \qquad t \in (-\infty, +\infty)$$

$$y(t) = \frac{4t}{(4+t^2)}$$

Pitagoraszi számhármasok

- **Cikloisz:**

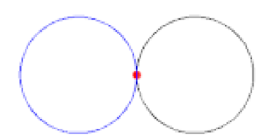
$$x(t) = t - \sin t$$
$$y(t) = 1 - \cos t$$

- **Tractrix:**

$$x(t) = \operatorname{sech} t$$
$$y(t) = t - \tanh t$$

- **Kardioid:**

$$x(t) = (1 - \cos t)\cos t$$
$$y(t) = (1 - \cos t)\sin t$$

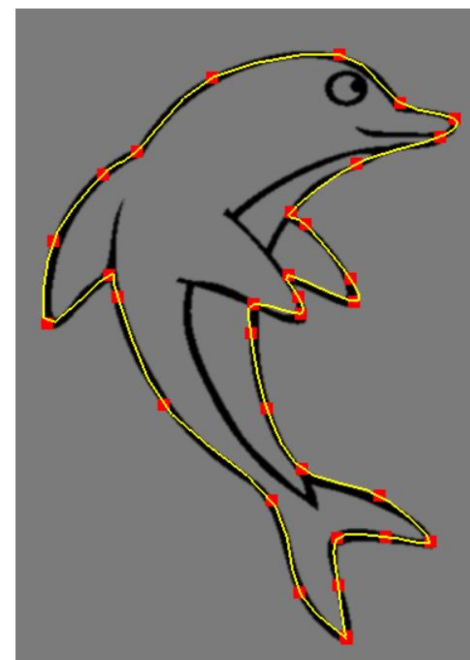


Pontok és klasszikus görbék

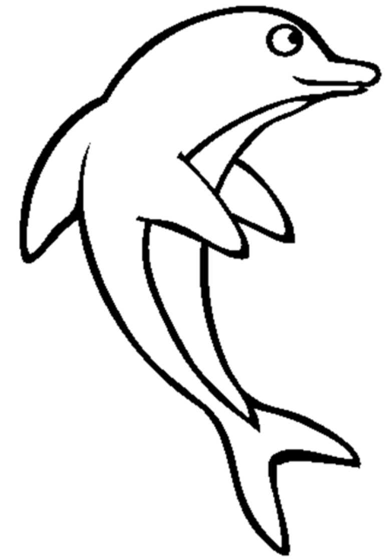
- Ponthoz koordinátarendszer kell
  - Descartes, Baricentrikus
- Görbéhez egyenlet kell
  - Az explicit ritkán használható
  - Az implicit feltételeket fogalmaz meg a pontokra
    - Azon pontok halmaza, amelyekben … pontok távolságra ($|\boldsymbol{p}-\boldsymbol{q}|$), … merőleges ($\boldsymbol{d}\cdot\boldsymbol{v}=0$), … párhuzamos ($\boldsymbol{d}\times\boldsymbol{v}=0$), … vetülete ($\boldsymbol{v}^0\cdot\boldsymbol{r}$).
  - A paraméteres mozgásként fogalmazza meg a görbét

Geometriai modellezés
2. Szabadformájú görbék

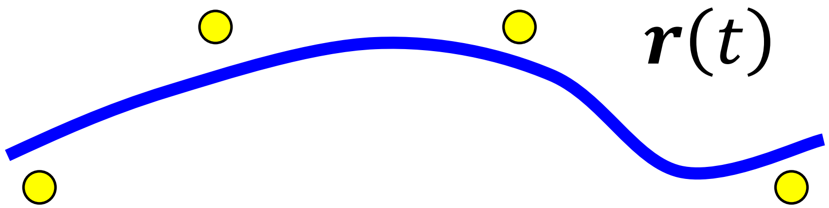
Szirmay-Kalos László



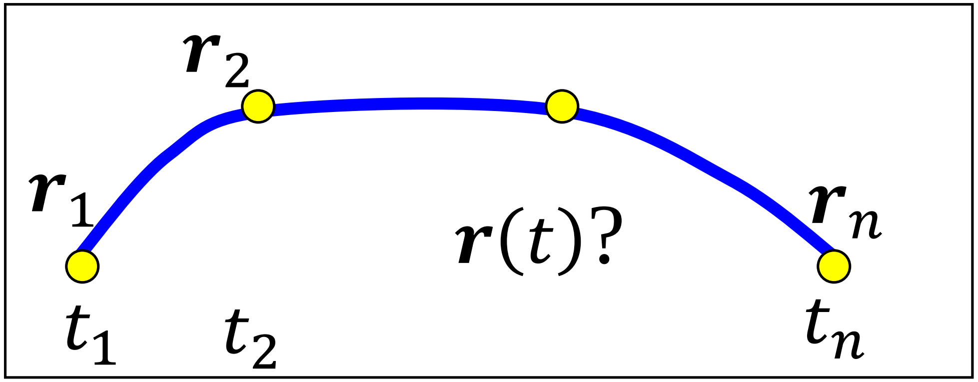
Szabadformájú görbék



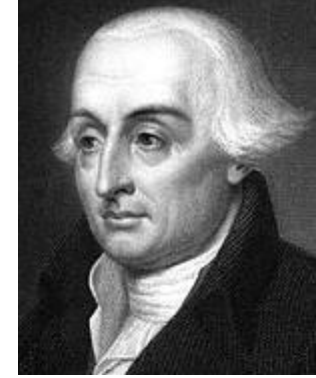
- Definíció vezérlőpontokkal



- Polinom: $\boxed{x(t) = \sum_i a_i t^i, y(t) = \sum_i b_i t^i, z(t) = \cdots}$
- Polinom együtthatók:
  - Kövesse a vezérlőpontokat: **Interpoláció/Approximáció**
  - **Természetesség:** C^2 folytonosság
  - **Szépség**: kis görbületváltozás indokolatlan hullámzás nélkül
  - Független legyen a koordinátarendszertől (súlypont)
  - Lokális vezérelhetőség



(Giuseppe) Lagrange interpoláció



- Keresd: $\boldsymbol{r}(t) = \left(\sum_i a_i t^i,\ \sum_i b_i t^i,\ \sum_i c_i t^i\right)$, amelyre $\boldsymbol{r}(t_1) = \boldsymbol{r}_1,\ \boldsymbol{r}(t_2) = \boldsymbol{r}_2,\ \dots,\ \boldsymbol{r}(t_n) = \boldsymbol{r}_n$
- Hányad fokú a polinom? $n - 1$
- Megoldás:

$$\boldsymbol{r}(t) = \sum_i L_i(t)\boldsymbol{r}_i \qquad \boldsymbol{r}(t_k) = \sum_i L_i(t_k)\boldsymbol{r}_i = \boldsymbol{r}_k$$

$$\boxed{L_i(t) = \frac{\prod_{j \neq i}(t - t_j)}{\prod_{j \neq i}(t_i - t_j)}} \qquad \boxed{L_i(t_k) = \frac{\prod_{j \neq i}(t_k - t_j)}{\prod_{j \neq i}(t_i - t_j)}} \begin{cases} 1 \text{ ha } i = k \\ 0 \text{ ha } i \neq k \end{cases}$$

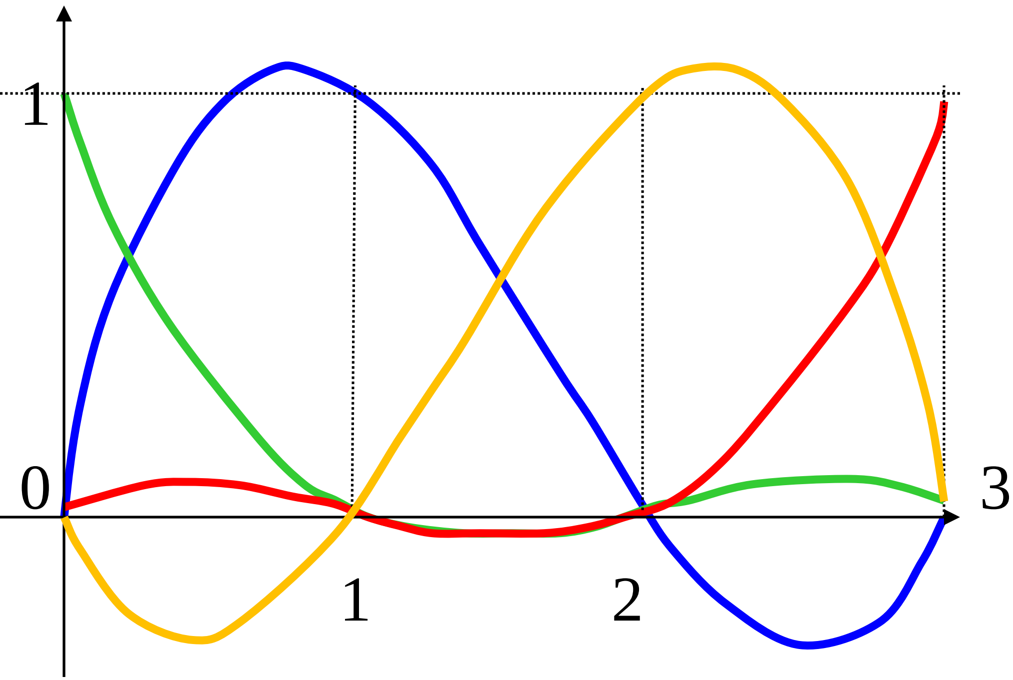
LagrangeCurve

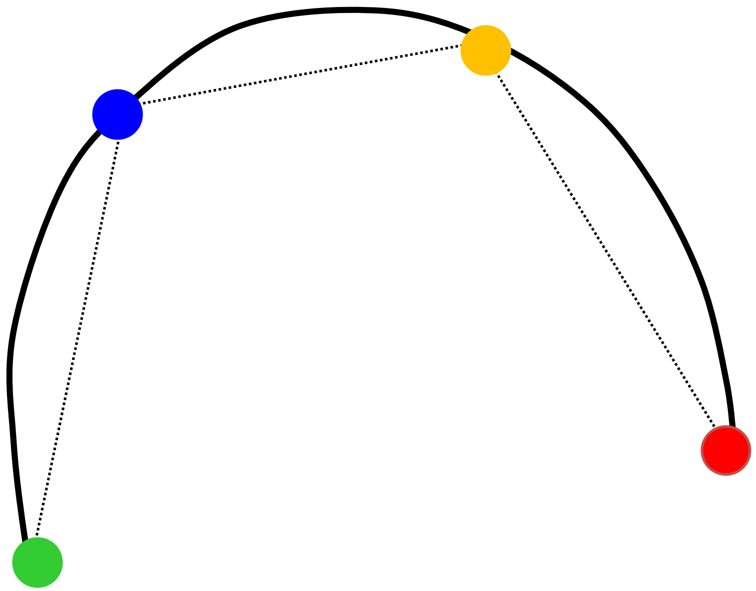
```
class LagrangeCurve {
    vector<vec3>  cps; // control pts
    vector<float> ts;  // knots

    float L(int i, float t) {
        float Li = 1.0f;
        for(int j = 0; j < cps.size(); j++)
            if (j != i) Li *= (t - ts[j])/(ts[i] - ts[j]);
        return Li;
    }
public:
    void AddControlPoint(vec3 cp) {
        float ti = cps.size();     // or something better
        cps.push_back(cp); ts.push_back(ti);
    }

    vec3 r(float t) {
        vec3 rt(0, 0, 0);
        for(int i=; i < cps.size(); i++) rt += cps[i] * L(i,t);
        return rt;
    }
};
```

$$L_i(t) = \frac{\prod_{j \neq i}(t - t_j)}{\prod_{j \neq i}(t_i - t_j)}$$

$$\boldsymbol{r}(t) = \sum_i L_i(t)\boldsymbol{r}_i$$

Lagrange interpolació bázisfüggvényei

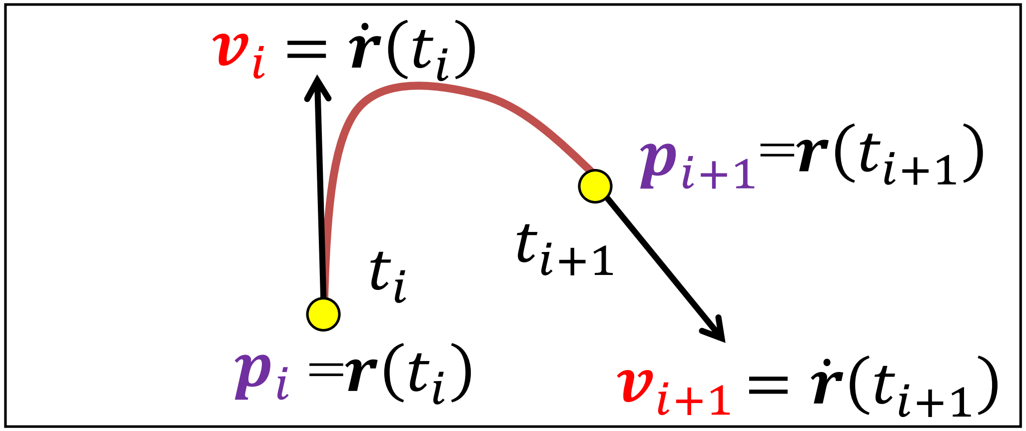




$$\boldsymbol{r}(t) = \sum_i L_i(t)\boldsymbol{r}_i$$

$$L_i(t) = \frac{\prod_{j\neq i}(t - t_j)}{\prod_{j\neq i}(t_i - t_j)}$$

(Charles) Hermite interpoláció



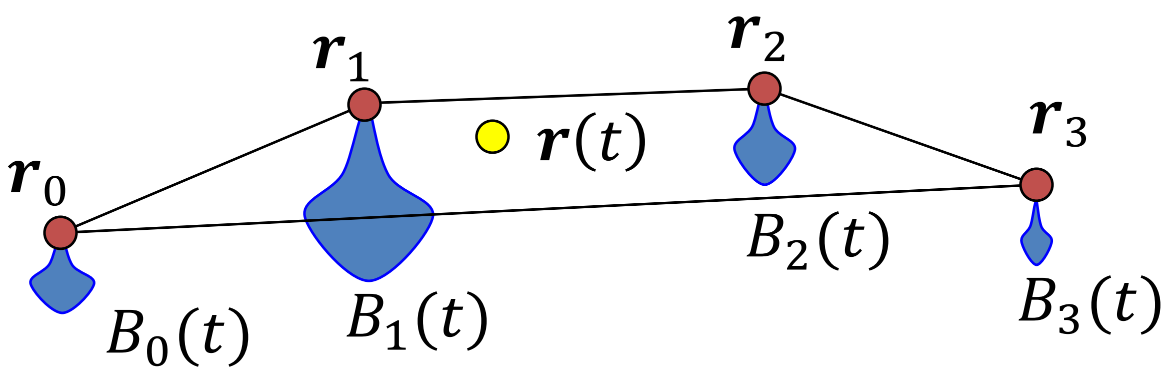
$$\boldsymbol{a}_0 = \boldsymbol{p}_i$$
$$\boldsymbol{a}_1 = \boldsymbol{v}_i$$
$$\boldsymbol{a}_2 = \frac{3(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)}{(t_{i+1} - t_i)^2} - \frac{(\boldsymbol{v}_{i+1} + 2\boldsymbol{v}_i)}{t_{i+1} - t_i}$$
$$\boldsymbol{a}_3 = \frac{2(\boldsymbol{p}_i - \boldsymbol{p}_{i+1})}{(t_{i+1} - t_i)^3} + \frac{(\boldsymbol{v}_{i+1} + \boldsymbol{v}_i)}{(t_{i+1} - t_i)^2}$$

- $\boldsymbol{r}(t) = \boldsymbol{a}_3(t - t_i)^3 + \boldsymbol{a}_2(t - t_i)^2 + \boldsymbol{a}_1(t - t_i) + \boldsymbol{a}_0$
- $\dot{\boldsymbol{r}}(t) = 3\boldsymbol{a}_3(t - t_i)^2 + 2\boldsymbol{a}_2(t - t_i) + \boldsymbol{a}_1$

$$\boldsymbol{r}(t_i) = \boldsymbol{a}_0 = \boldsymbol{p}_i$$
$$\boldsymbol{r}(t_{i+1}) = \boldsymbol{a}_3(t_{i+1} - t_i)^3 + \boldsymbol{a}_2(t_{i+1} - t_i)^2 + \boldsymbol{a}_1(t_{i+1} - t_i) + \boldsymbol{a}_0 = \boldsymbol{p}_{i+1}$$
$$\dot{\boldsymbol{r}}(t_i) = \boldsymbol{a}_1 = \boldsymbol{v}_i$$
$$\dot{\boldsymbol{r}}(t_{i+1}) = 3\boldsymbol{a}_3(t_{i+1} - t_i)^2 + 2\boldsymbol{a}_2(t_{i+1} - t_i) + \boldsymbol{a}_1 = \boldsymbol{v}_{i+1}$$

(Pierre) Bézier approximáció

- Keresd: $\boldsymbol{r}(t) = \sum_i B_i(t)\boldsymbol{r}_i$
  - $B_i(t)$: ne oszcilláljon
  - Konvex burok tulajdonság
  - $B_i(t) \geq 0, \quad \sum_i B_i(t) = 1$



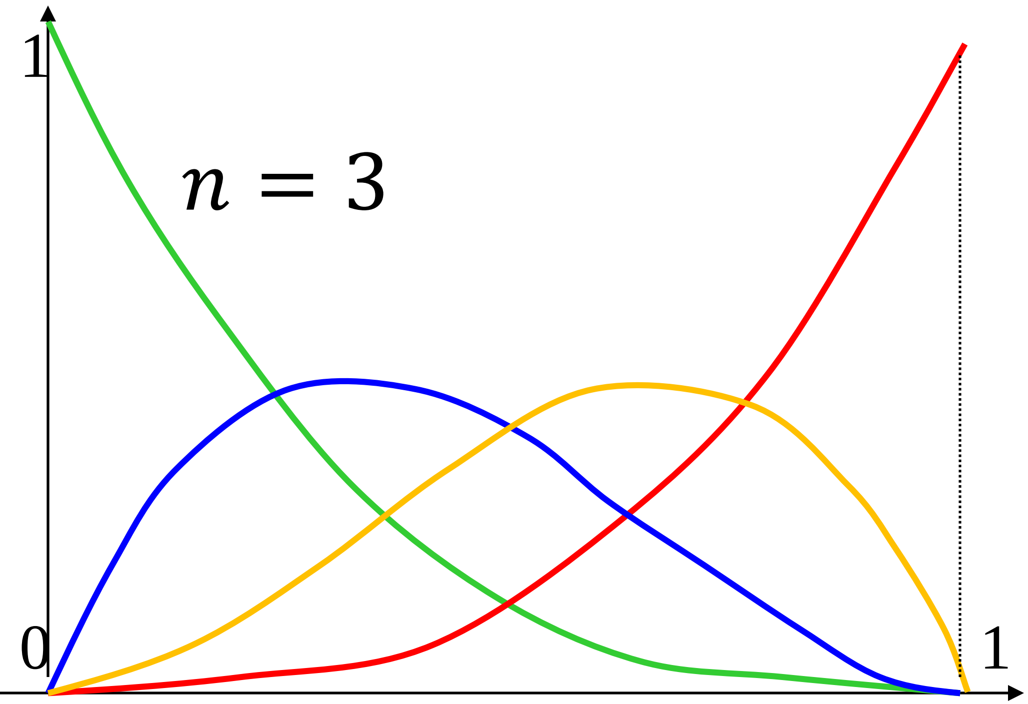
(Сергéй Натáнович) Bernstein polinomok

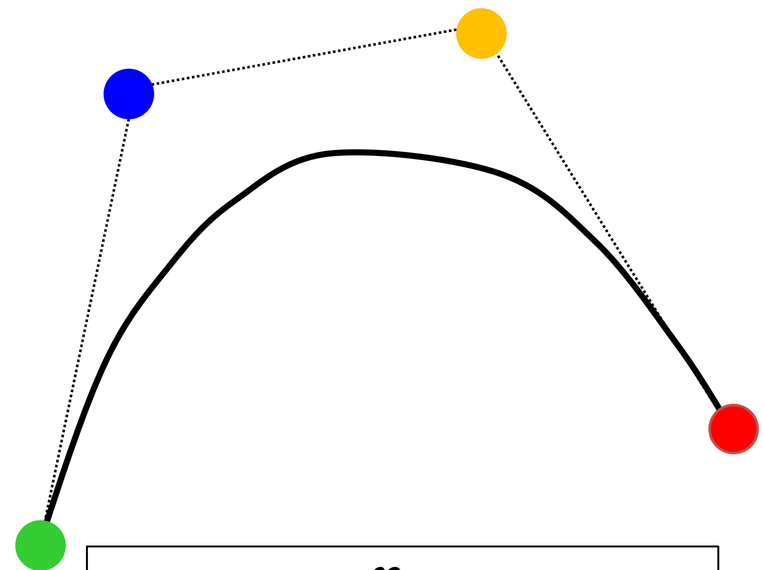
Newton binomiális tétel

$$1^n = (t + (1 - t))^n = \sum_{i=0}^{n} \underbrace{\binom{n}{i} t^i (1 - t)^{n-i}}_{B_i(t)}$$

$$B_i(t) \geq 0,\ \textstyle\sum_i B_i(t) = 1 \ :\ \text{OK}$$

Bézier approximáció





$$\boldsymbol{r}(t) = \sum_{i=0}^{n} B_i(t)\boldsymbol{r}_i$$

$$B_i(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

BezierCurve

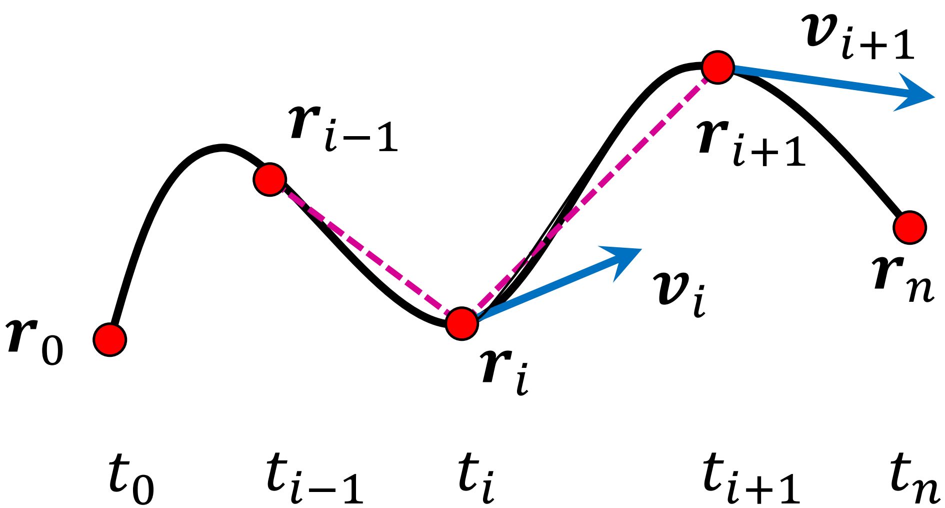
$$B_i(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

$$\boldsymbol{r}(t) = \sum_{i=0}^{n} B_i(t)\boldsymbol{r}_i$$

```
class BezierCurve {
   vector<vec3> cps;// control pts

   float B(int i, float t) {
       int n = cps.size()-1; // n deg polynomial = n+1 pts!
       float choose = 1;
       for(int j = 1; j <= i; j++) choose *= (float)(n-j+1)/j;
       return choose * pow(t, i) * pow(1-t, n-i);
   }
public:
   void AddControlPoint(vec3 cp) { cps.push_back(cp); }


    vec3 r(float t) {
       vec3 rt(0, 0, 0);
       for(int i=0; i < cps.size(); i++) rt += cps[i] * B(i,t);
       return rt;
   }
};
```

Catmull-Rom spline

- Minden két vezérlőpont közé egy Hermite
- C^1 simaság: a sebesség is legyen közös két egymás utánira
- Közelítő C^2 simaság: A közös sebességet úgy válaszd meg, hogy a gyorsulás is közelítőleg folytonos legyen



$$\boldsymbol{v}_i = \frac{1}{2}\left(\frac{\boldsymbol{r}_{i+1} - \boldsymbol{r}_i}{t_{i+1} - t_i} + \frac{\boldsymbol{r}_i - \boldsymbol{r}_{i-1}}{t_i - t_{i-1}}\right)$$

CatmullRom

```
class CatmullRom {
    vector<vec3>  cps; // control points
    vector<float> ts;  // parameter (knot) values

    vec3 Hermite( vec3 p0, vec3 v0, float t0,
                  vec3 p1, vec3 v1, float t1, float t ) {
    }

public:
    void AddControlPoint(vec3 cp, float t) { … }

    vec3 r(float t) {
        for(int i = 0; i < cps.size() - 1; i++)
            if (ts[i] <= t && t <= ts[i+1]) {
                vec3 v0 = …, v1 = …;
                return Hermite(cps[i], v0, ts[i],
                               cps[i+1], v1, ts[i+1], t);
            }
    }
};
```

$$\boldsymbol{r}(t) = \boldsymbol{a}_3(t-t_0)^3 + \boldsymbol{a}_2(t-t_0)^2 + \boldsymbol{a}_1(t-t_0) + \boldsymbol{a}_0$$

$$\boldsymbol{a}_0 = \boldsymbol{p}_i\,,\ \boldsymbol{a}_1 = \boldsymbol{v}_i$$

$$\boldsymbol{a}_2 = \frac{3(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)}{(t_{i+1} - t_i)^2} - \frac{(\boldsymbol{v}_{i+1} + 2\boldsymbol{v}_i)}{t_{i+1} - t_i}$$

$$\boldsymbol{a}_3 = \frac{2(\boldsymbol{p}_i - \boldsymbol{p}_{i+1})}{(t_{i+1} - t_i)^3} + \frac{(\boldsymbol{v}_{i+1} + \boldsymbol{v}_i)}{(t_{i+1} - t_i)^2}$$

$$\boldsymbol{v}_i = \frac{1}{2}\left(\frac{\boldsymbol{r}_{i+1} - \boldsymbol{r}_i}{t_{i+1} - t_i} + \frac{\boldsymbol{r}_i - \boldsymbol{r}_{i-1}}{t_i - t_{i-1}}\right)$$

CatmullRom, 2. verzió

```
class CatmullRom {
   vector<vec3> cps;        // control points
   vector<float>  ts;       // parameter (knot) values

   vec3 Seg(vec3 p_1, float t_1, vec3 p0, float t0,
            vec3 p1,  float t1, vec3 p2, float t2, float t){
      float c_1 = …, c0 = …, c1 = …, c2 = …; // t_i, t
      return p_1 * c_1 + p0 * c0 + p1 * c1 + p2 * c2;
   }

public:
   void AddControlPoint(vec3 cp, float t) { … }

   vec3 r(float t) {
      for(int i = 0; i < cps.size() - 1; i++)
         if (ts[i] <= t && t <= ts[i+1]) {
            // Túlcímzést lekezelni!
            return Seg(cps[i-1],ts[i-1], cps[i],ts[i],
                       cps[i+1],ts[i+1], cps[i+2],ts[i+2], t);
         }
   }
};
```

Szabadformájú görbék

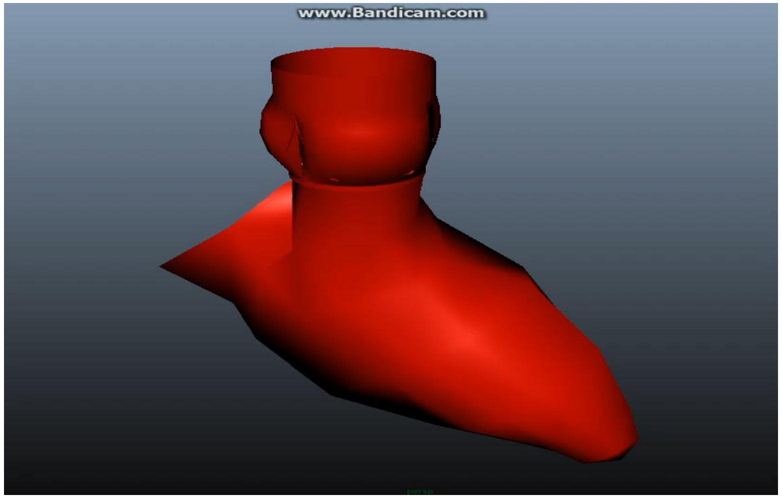
- Paraméteres egyenlet (mozgás), polinomok
- Kontrolpontokkal definiáljuk (approximációs, interpolációs)
- Görbe = kontrolpontok kombinációja (súlypont)
- Görbe tulajdonságait a súlyfüggvények határozzák meg
  - Folytonosság (C^0, C^1, C^2)
  - Konvex burok: súlyfüggvények nem negatívak
  - Lokális vezérelhetőség: súlyfüggvények a tartomány egy részében nem zérus értékűek

"La semplicità è la sofisticazione finale."

Leonardo da Vinci

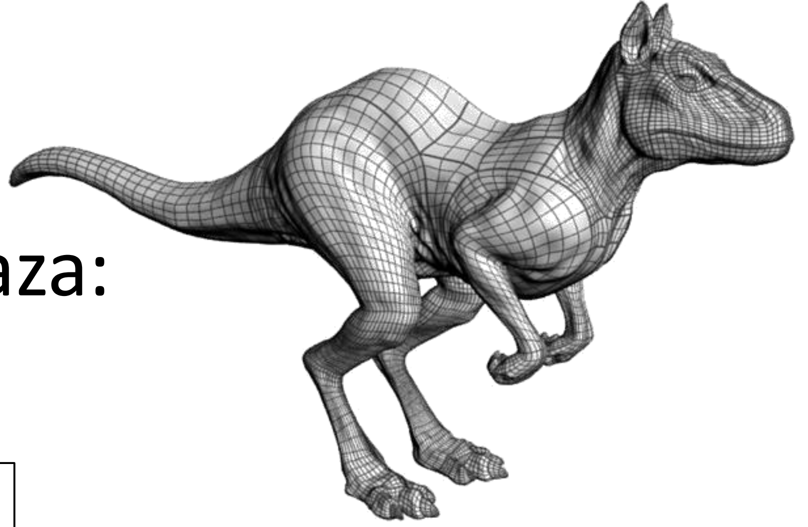
Geometriai modellezés
3. Felületek

Szirmay-Kalos László



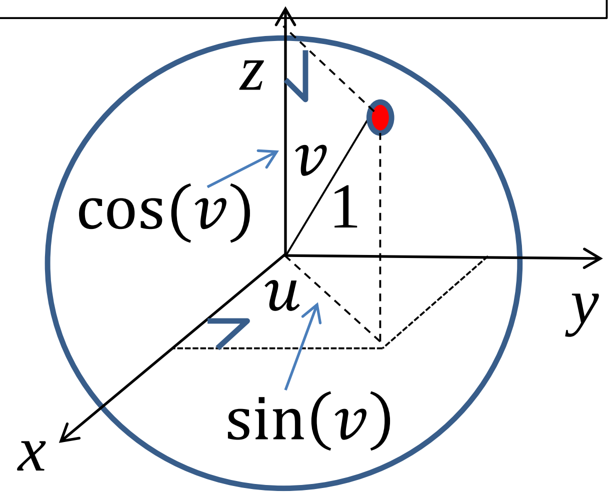
Felületek

Felület a 3D tér 2D részhalmaza:



- **Explicit:** $\boxed{z = h(x, y)}$
- **Implicit**: $\boxed{f(x, y, z) = 0}$
  - gömb: $(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - R^2 = 0$
  - sík: $ax + by + cz + d = 0$
- **Parametrikus**: $\boxed{x = x(u, v), y = y(u, v), z = z(u, v)}$
  - gömb:

$$\boxed{\begin{aligned} x(u,v) &= c_x + R\cos(u)\sin(v) \\ y(u,v) &= c_y + R\sin(u)\sin(v) \\ z(u,v) &= c_z + R\cos(v) \\ u \in [0,2\pi&), v \in [0,\pi) \end{aligned}}$$



Implicit felületek normálvektora

Normál vektor = grad f $= \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$

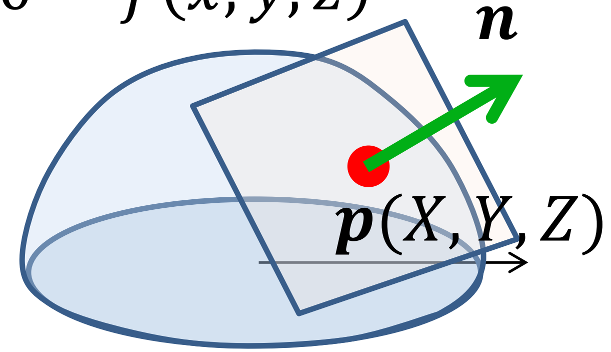
$$0 = f(x, y, z)$$

$$= f\big(X + (x - X), Y + (y - Y), Z + (z - Z)\big)$$

$$\approx \cancel{f(X, Y, Z)} + \frac{\partial f}{\partial x}(x - X) + \frac{\partial f}{\partial y}(y - Y) + \frac{\partial f}{\partial z}(z - Z)$$

$$0 = f(x, y, z)$$

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) \cdot (x - X, y - Y, z - Z) = 0$$

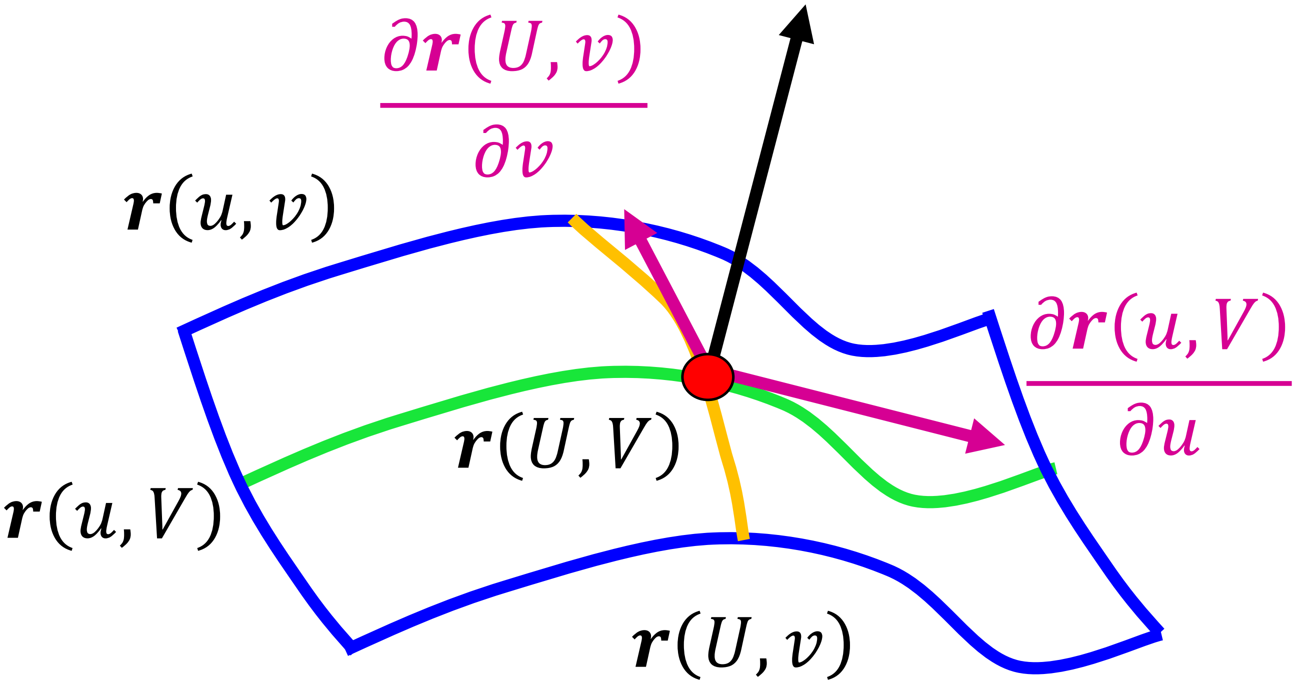


$$\boldsymbol{n} \cdot (\boldsymbol{r} - \boldsymbol{p}) = \boldsymbol{0}$$

Parametrikus felületek normálvektora

$$\boxed{\mathbf{N}(U,V) = \frac{\partial \mathbf{r}(u,v)}{\partial u} \times \frac{\partial \mathbf{r}(u,v)}{\partial v}}$$

$u = U$
$v = V$



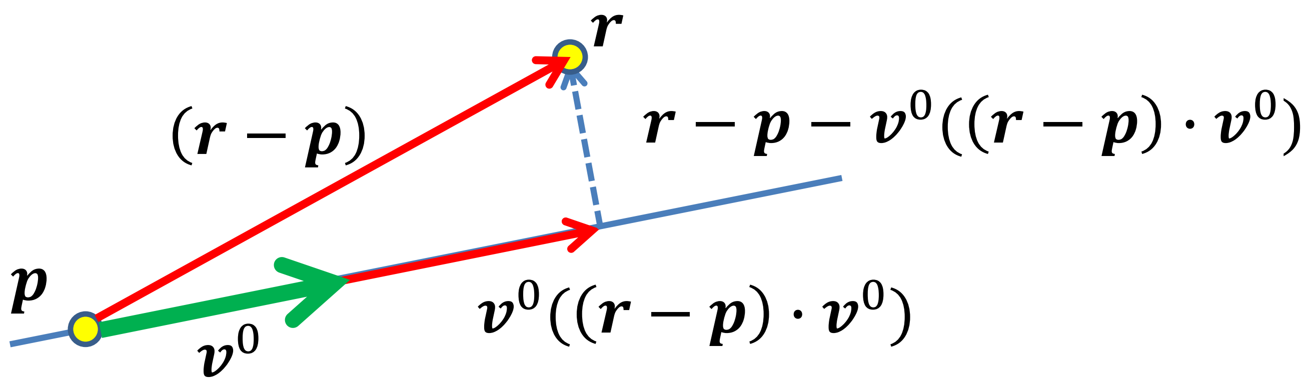
Implicit kvadratikus felületek

- **<u>Gömb:</u>** Azon $\boldsymbol{r}(x,y)$ pontok mértani helye, amelyek a $\boldsymbol{c}(c_x,c_y)$ középponttól R távolságra vannak: $|\boldsymbol{r}-\boldsymbol{c}| = R$
- **<u>Henger:</u>** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{v}^0$ irányvektorú és $\boldsymbol{p}$ helyvektorú egyenestől mért távolsága R: $|\boldsymbol{r}-\boldsymbol{p}-\boldsymbol{v}^0((\boldsymbol{r}-\boldsymbol{p})\cdot\boldsymbol{v}^0)\,| = R$

Implicit kvadratikus felületek

- **<u>Gömb:</u>** Azon $\boldsymbol{r}(x,y)$ pontok mértani helye, amelyek a $\boldsymbol{c}(c_x,c_y)$ középponttól R távolságra vannak: $|\boldsymbol{r}-\boldsymbol{c}| = R$

- **<u>Henger:</u>** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{v}^0$ irányvektorú és $\boldsymbol{p}$ helyvektorú egyenestől mért távolsága R: $|\boldsymbol{r}-\boldsymbol{p}-\boldsymbol{v}^0((\boldsymbol{r}-\boldsymbol{p})\cdot\boldsymbol{v}^0)\,| = R$



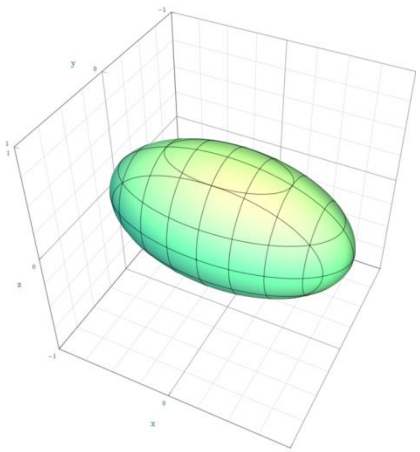
Implicit kvadratikus felületek

- **Gömb:** Azon $\boldsymbol{r}(x,y)$ pontok mértani helye, amelyek a $\boldsymbol{c}(c_x,c_y)$ középponttól R távolságra vannak: $|\boldsymbol{r}-\boldsymbol{c}| = R$
- **Henger:** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{v}^0$ irányvektorú és $\boldsymbol{p}$ helyvektorú egyenestől mért távolsága R: $|\boldsymbol{r}-\boldsymbol{p}-\boldsymbol{v}^0((\boldsymbol{r}-\boldsymbol{p})\cdot\boldsymbol{v}^0)\,| = R$
- **Ellipszoid:** Azon $\boldsymbol{r}$ pontok, amelyek a $\boldsymbol{f}_1$és $\boldsymbol{f}_2$ fókuszpontoktól mért távolság összege állandó C: $|\boldsymbol{r}-\boldsymbol{f}_1| + |\boldsymbol{r}-\boldsymbol{f}_2| = C$
- **Hiperboloid:** Azon $\boldsymbol{r}$ pontok, melyek a $\boldsymbol{f}_1$és $\boldsymbol{f}_2$ fókuszpontoktól mért távolság különbsége állandó C: $|\boldsymbol{r}-\boldsymbol{f}_1| - |\boldsymbol{r}-\boldsymbol{f}_2| = C$
- **Paraboloid:** Azon $\boldsymbol{r}$ pontok, amelyek az $\boldsymbol{f}$ fókuszponttól mért távolsága megegyezik az $\boldsymbol{n}$ normálvektorú és $\boldsymbol{p}$ helyvektorú síktól mért távolsággal: $|\boldsymbol{r}-\boldsymbol{f}| = |\boldsymbol{n}^0\cdot(\boldsymbol{r}-\boldsymbol{p})|$

Implicit kvadratikus felületek

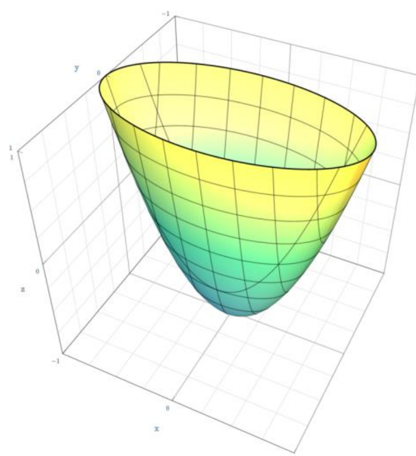
$$f(x,y,z) = [x,y,z,1]\mathbf{Q}\begin{bmatrix}x\\y\\z\\1\end{bmatrix} = 0$$

Mindig felírható úgy is, hogy **Q** szimmetrikus



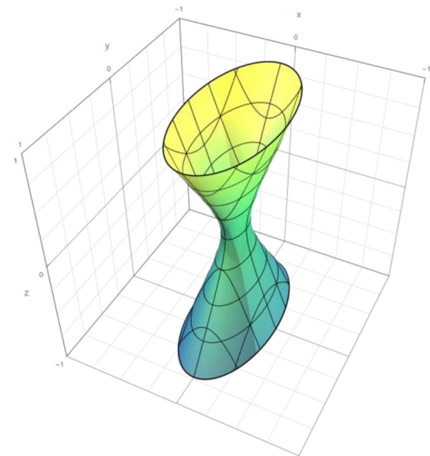
$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}-1=0$$

Ellipszoid



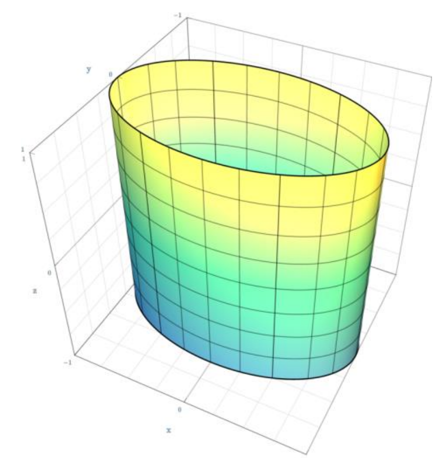
$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-z=0$$

Paraboloid



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}-1=0$$

Hiperboloid



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-1=0$$

Elliptikus henger

Kvadratikus objektum

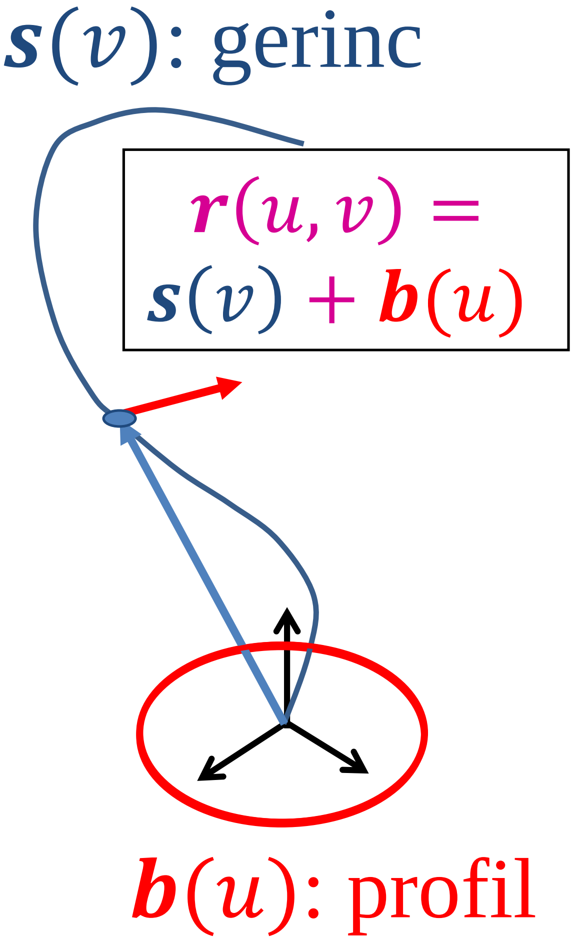
$$f(x,y,z) = [x,y,z,1]\mathbf{Q}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

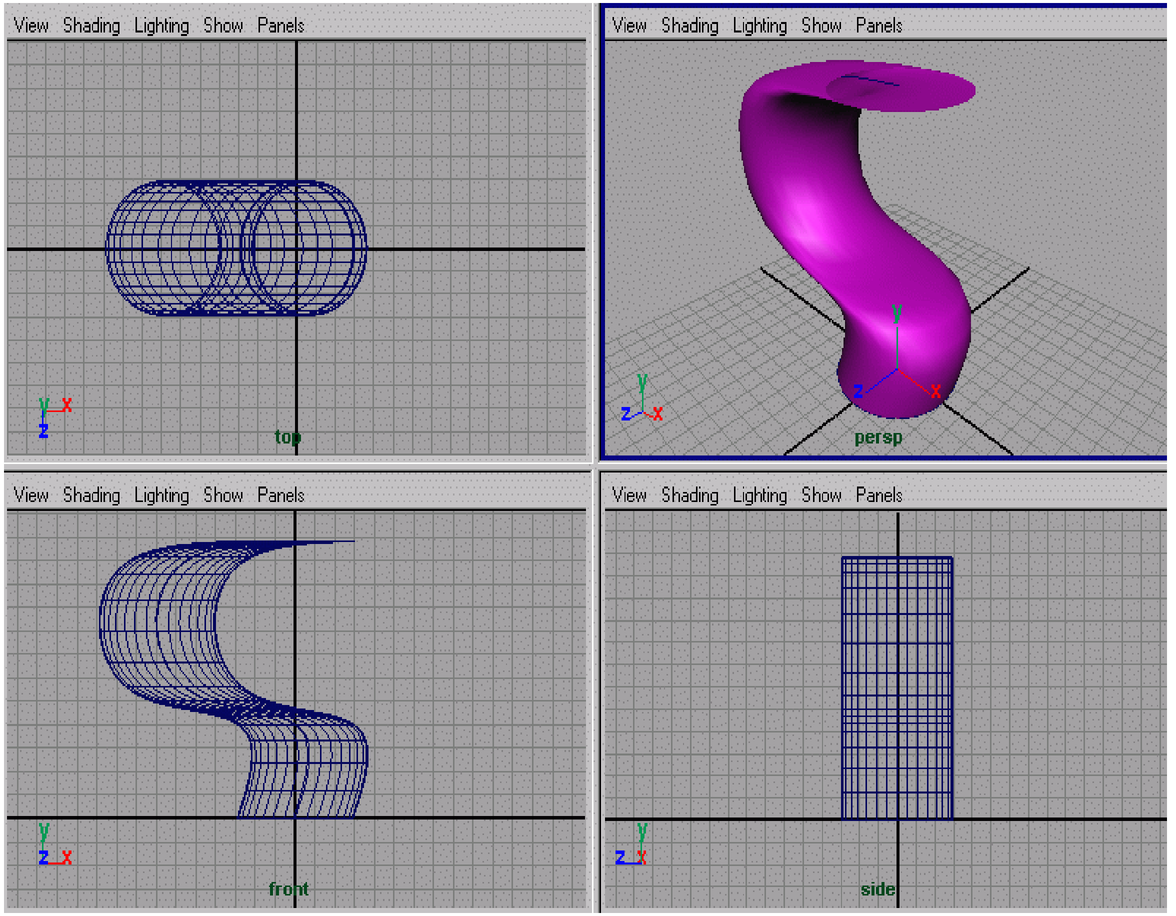
$$\frac{\partial f}{\partial x} = [1,0,0,0]\mathbf{Q}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + [x,y,z,1]\mathbf{Q}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= [x,y,z,1]\mathbf{Q}\begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
struct Quadrics {
   mat4 Q; // symmetric matrix

   float f(vec4 r) { // r.w = 1
      return dot(r * Q, r);
   }

   vec3 gradf(vec4 r) { // r.w = 1
      vec4 g = r * Q * 2;
      return vec3(g.x, g.y, g.z);
   }
};
```

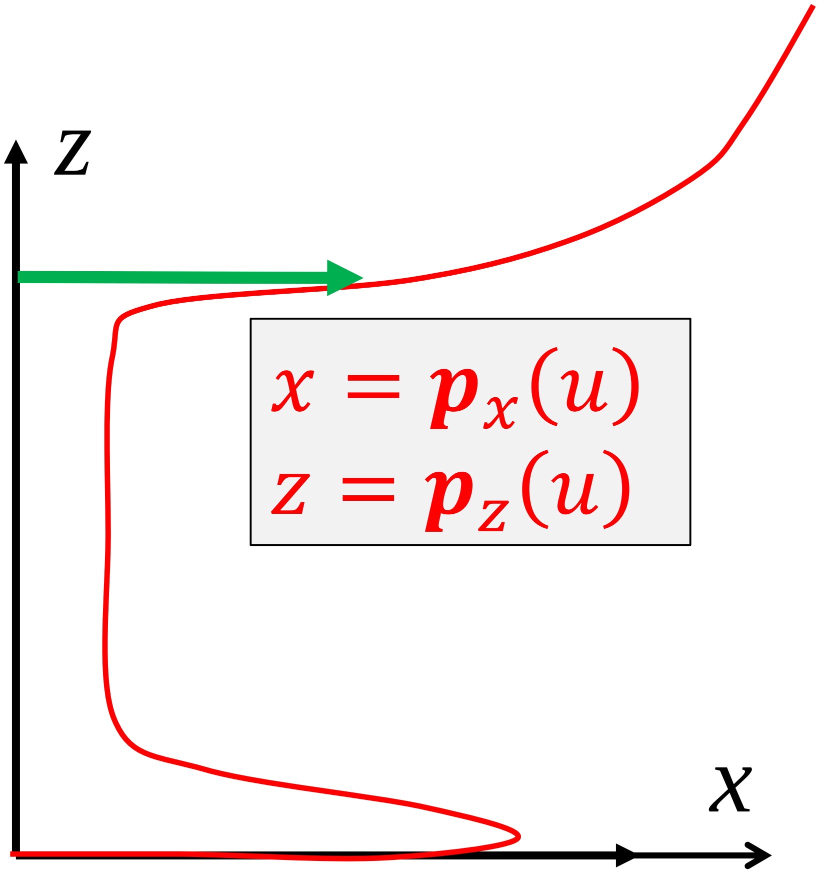
Parametrikus felületek: Kihúzás





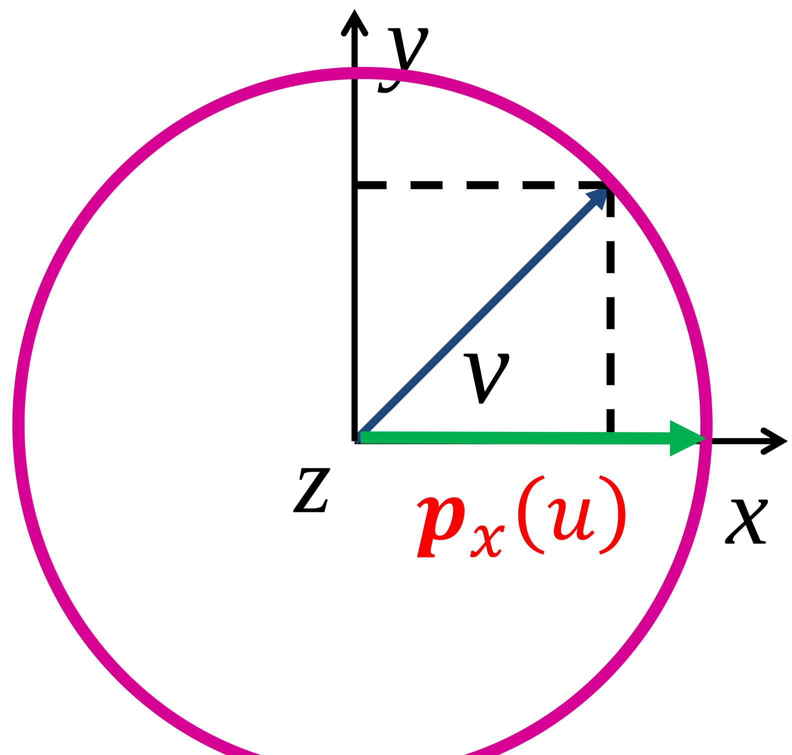
Parametrikus felületek: Forgatás

Oldalnézet



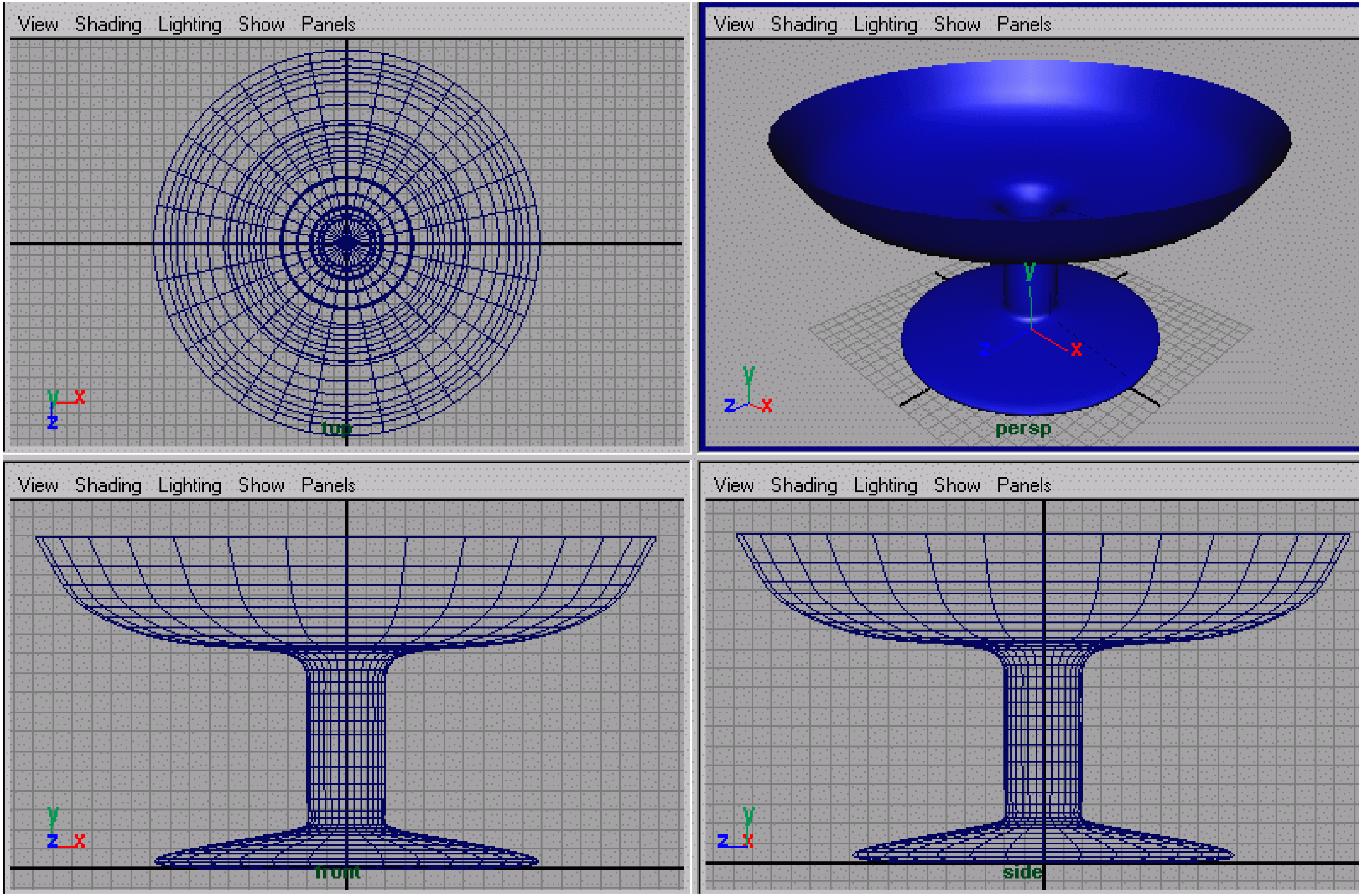
$$x = \boldsymbol{p}_x(u)$$
$$z = \boldsymbol{p}_z(u)$$

Felülnézet

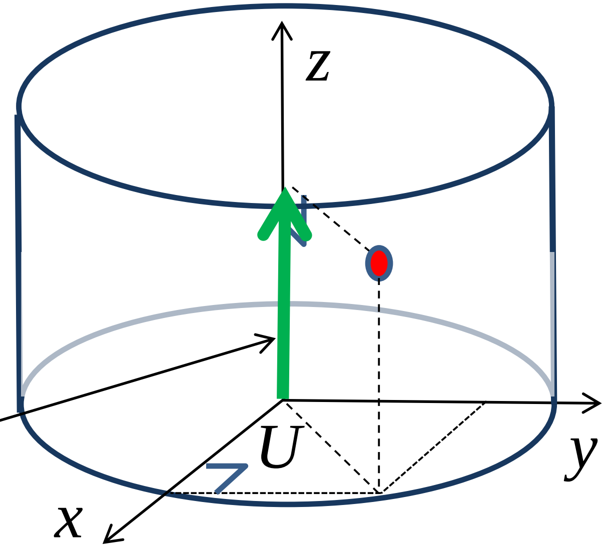


$$x = \boldsymbol{p}_x(u)\cos(v)$$
$$y = \boldsymbol{p}_x(u)\sin(v)$$
$$z = \boldsymbol{p}_z(u)$$

Forgatás


View Shading Lighting Show Panels
top
View Shading Lighting Show Panels
persp
View Shading Lighting Show Panels
front
View Shading Lighting Show Panels
side

Henger



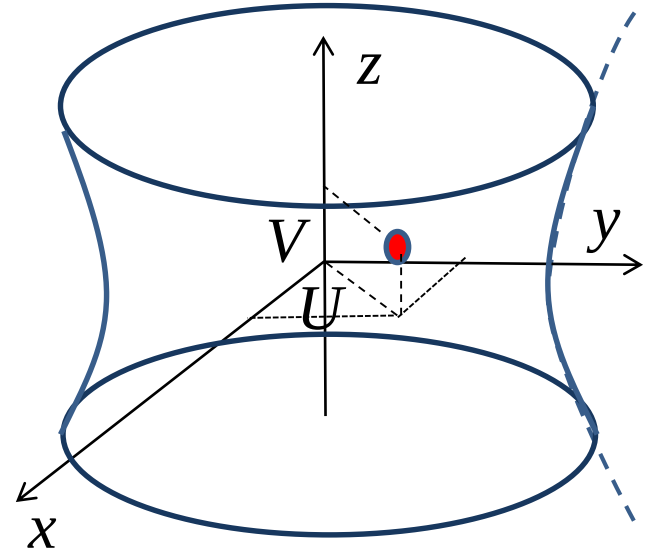
$$\boldsymbol{s}(V) = (0,0,V)$$

$$\boldsymbol{b}(U) = (r\cos(U), r\sin(U), 0)$$

$$x(U,V) = r\cos(U)$$
$$y(U,V) = r\sin(U)$$
$$z(U,V) = V$$
$$U \in [0,2\pi], \quad V \in [0,h]$$

```
void eval(float u, float v, vec3& point, vec3& normal) {
   float U = u * 2 * M_PI, V = v * height;
   vec3 base(cos(U) * r, sin(U) * r, 0), spine(0, 0, V);
   point = base + spine;
   normal = base;
}
```

Hiperboloid

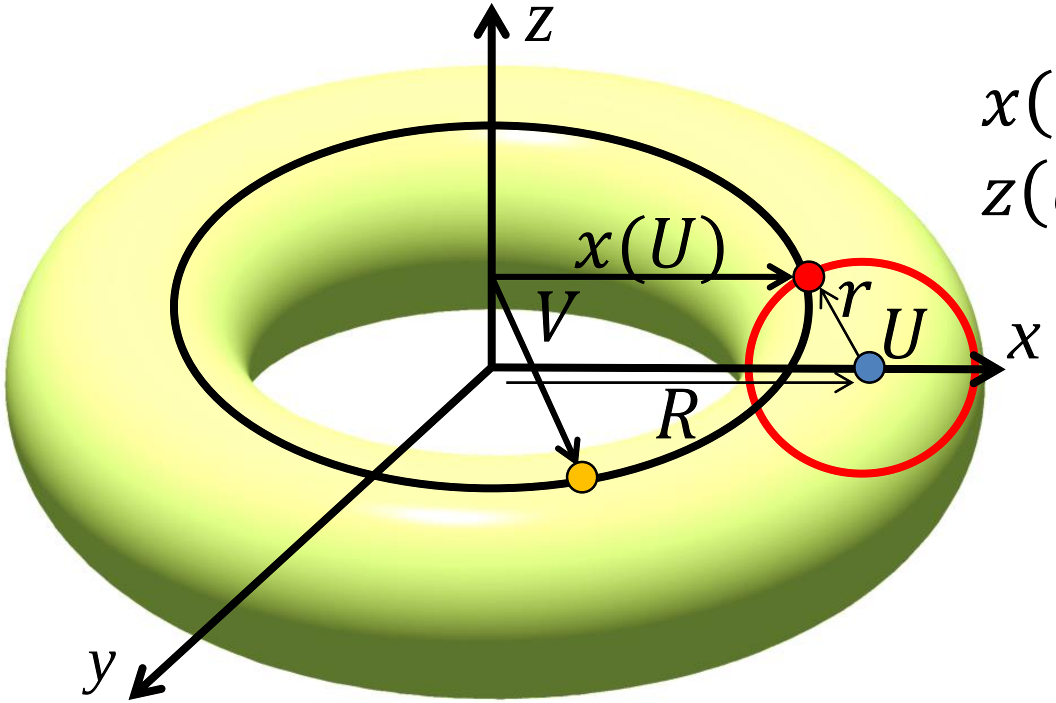


$$\boldsymbol{p}(U) = (r\cosh(U), 0, \sinh(U))$$

$$x(U,V) = r\cosh(U)\cos(V)$$
$$y(U,V) = r\cosh(U)\sin(V)$$
$$z(U,V) = \sinh(U)$$
$$V \in [0,2\pi], \quad U \in [-h/2, h/2]$$

```
void eval(float u, float v, vec3& point, vec3& normal){
   float U = (v - 0.5f) * h, V = u * 2 * M_PI;
   float shu=sinh(U), chu=cosh(U), cv=cos(V), sv=sin(V);
   point = vec3(r * chu * cv,     r * chu * sv, shu);
   vec3    drdU(r * shu * cv,     r * shu * sv, chu);
   vec3    drdV(r * chu * (-sv), r * chu * cv, 0);
   normal = cross(drdU, drdV);
}
```

Tórusz



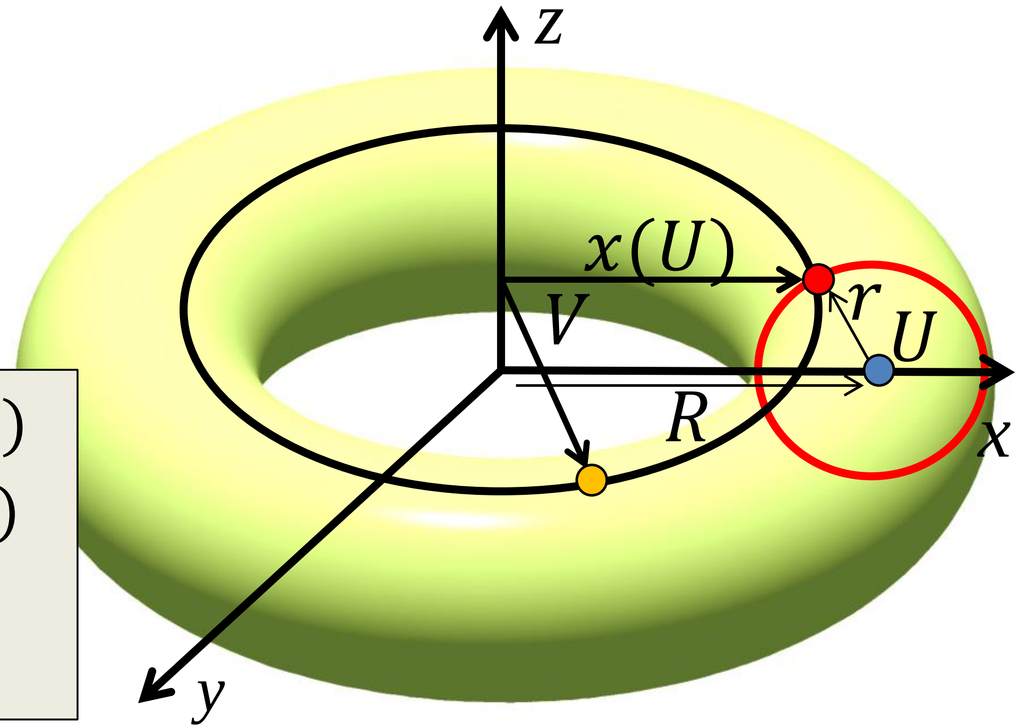
$$x(U) = R + r\cos(U)$$
$$z(U) = r\sin(U)$$

$$x(U,V) = \big(R + r\cos(U)\big)\cos(V)$$
$$y(U,V) = \big(R + r\cos(U)\big)\sin(V)$$
$$z(U,V) = r\sin(U)$$
$$U \in [0,2\pi],\quad V \in [0,2\pi]$$

Tórusz

$$\mathbf{D} = R + r\cos(U)$$

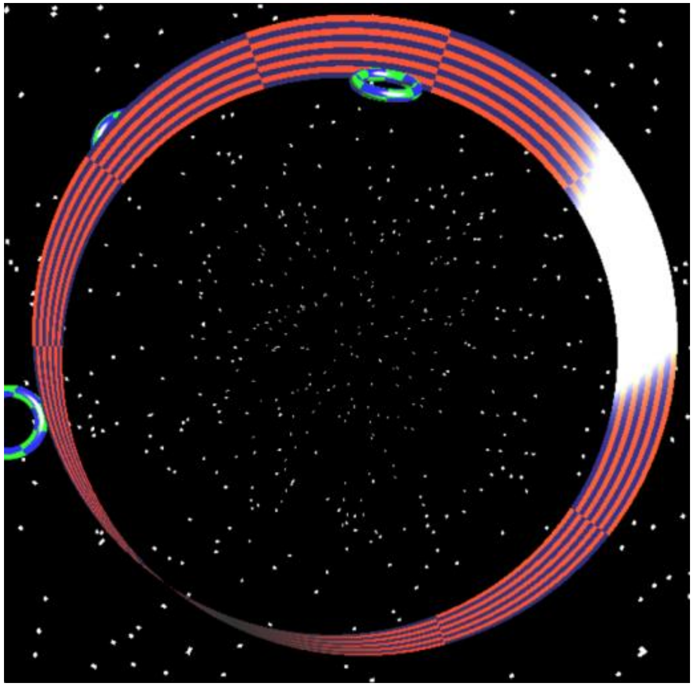
$$x(U,V) = \boxed{\big(R + r\cos(U)\big)}\cos(V)$$
$$y(U,V) = \big(R + r\cos(U)\big)\sin(V)$$
$$z(U,V) = r\sin(U)$$
$$U \in [0,2\pi], \quad V \in [0,2\pi]$$




```
typedef Dnum<vec2> Dnum2;

void eval(float u, float v, vec3& point, vec3& normal){
   Dnum2 U(u*2*M_PI, vec2(1,0)), V(v*2*M_PI, vec2(0,1));
   Dnum2 D = Cos(U) * r + R;
   Dnum2 X = D * Cos(V), Y = D * Sin(V), Z = Sin(U) * r;
   point = vec3(X.f, Y.f, Z.f);
   vec3 drdU(X.d.x,Y.d.x,Z.d.x), drdV(X.d.y,Y.d.y,Z.d.y);
   normal = cross(drdU, drdV);
}
```

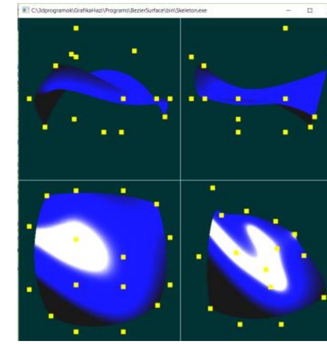
Möbius

$$x(U,V) = \big(R + V\cos(U)\big)\cos(2U)$$
$$y(U,V) = \big(R + V\cos(U)\big)\sin(2U)$$
$$z(U,V) = V\sin(U)$$
$$U \in [0,\pi], \quad V \in [-w/2, w/2]$$

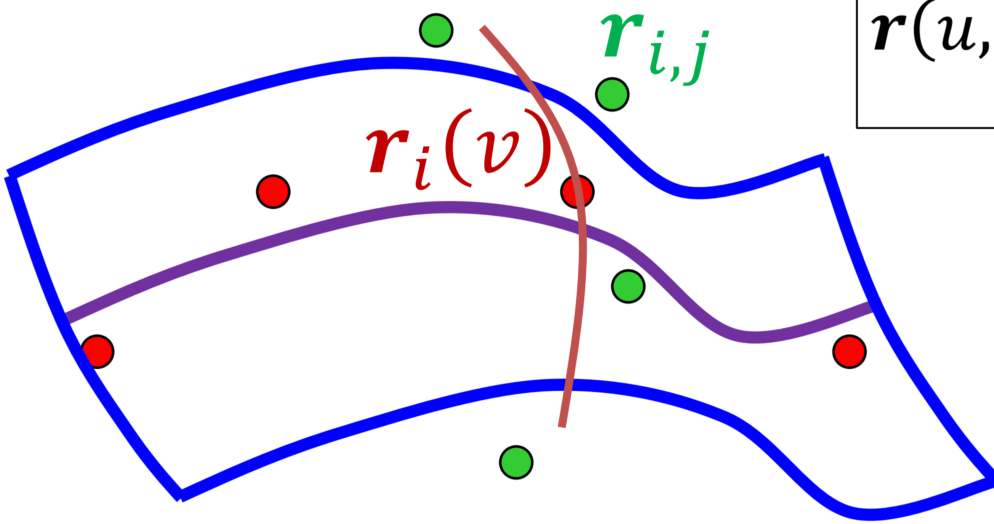


```
typedef Dnum<vec2> Dnum2;

void eval(float u, float v, vec3& point, vec3& normal){
   Dnum2 U(u*M_PI, vec2(1,0)), V((v-0.5)*w, vec2(0,1));
   Dnum2 X = (Cos(U) * V + R) * Cos(U * 2);
   Dnum2 Y = (Cos(U) * V + R) * Sin(U * 2);
   Dnum2 Z = Sin(U) * V;
   point = vec3(X.f, Y.f, Z.f);
   vec3 drdU(X.d.x,Y.d.x,Z.d.x), drdV(X.d.y,Y.d.y,Z.d.y);
   normal = cross(drdU, drdV);
}
```

Szabadformájú felület



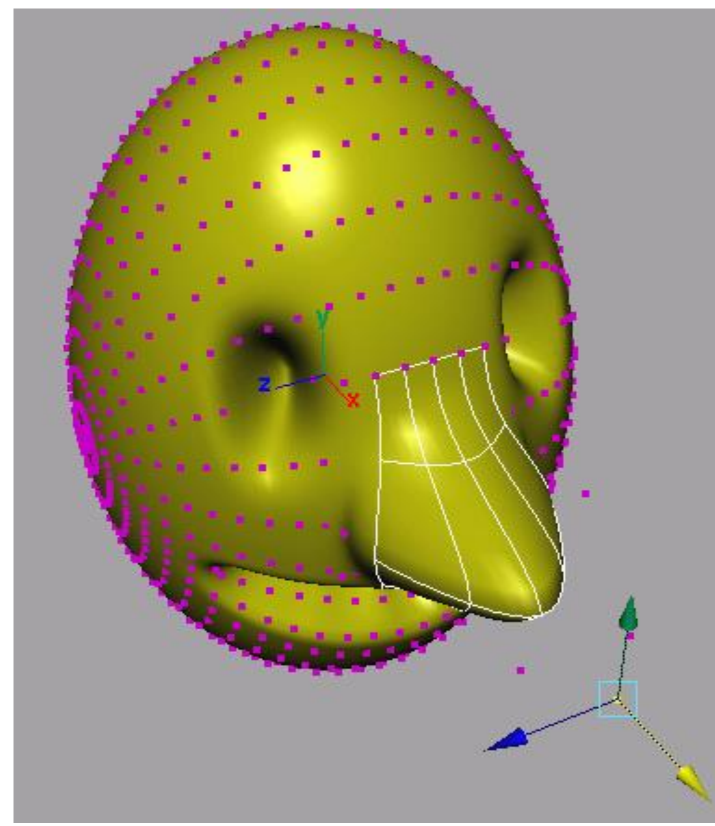
Definíció kontroll pontokkal:



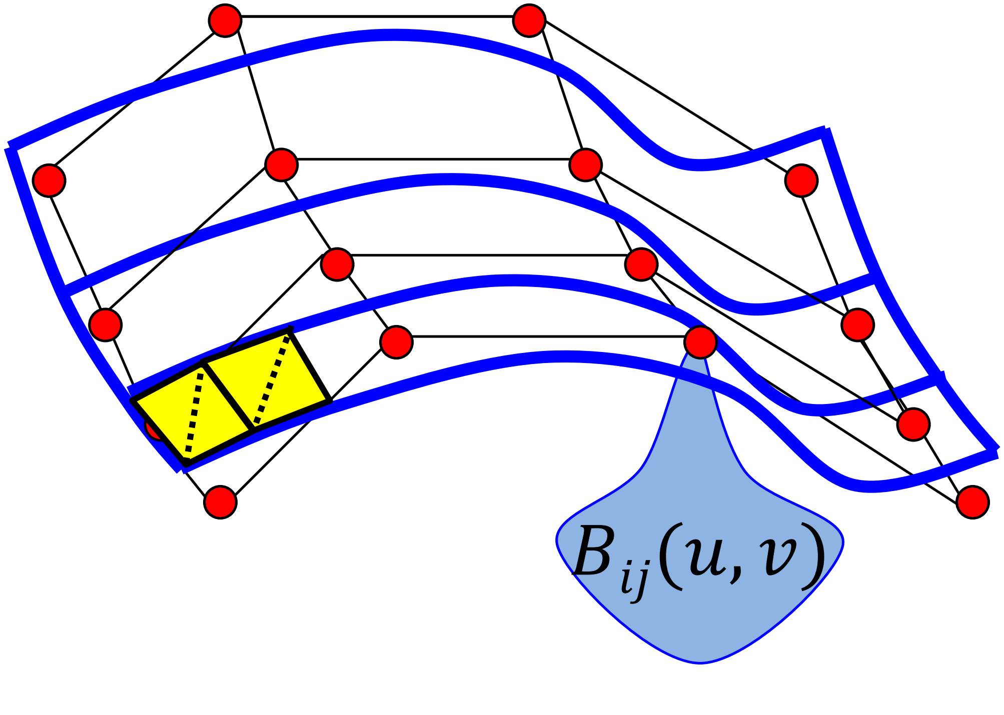
$$\boldsymbol{r}(u,v) = \sum_i \sum_j B_j(v)\, B_i(u)\boldsymbol{r}_{i,j}$$

$$\boldsymbol{r}(u,v) = \boldsymbol{r}_v(u) = \sum_i B_i(u)\, \boldsymbol{r}_i(v)$$

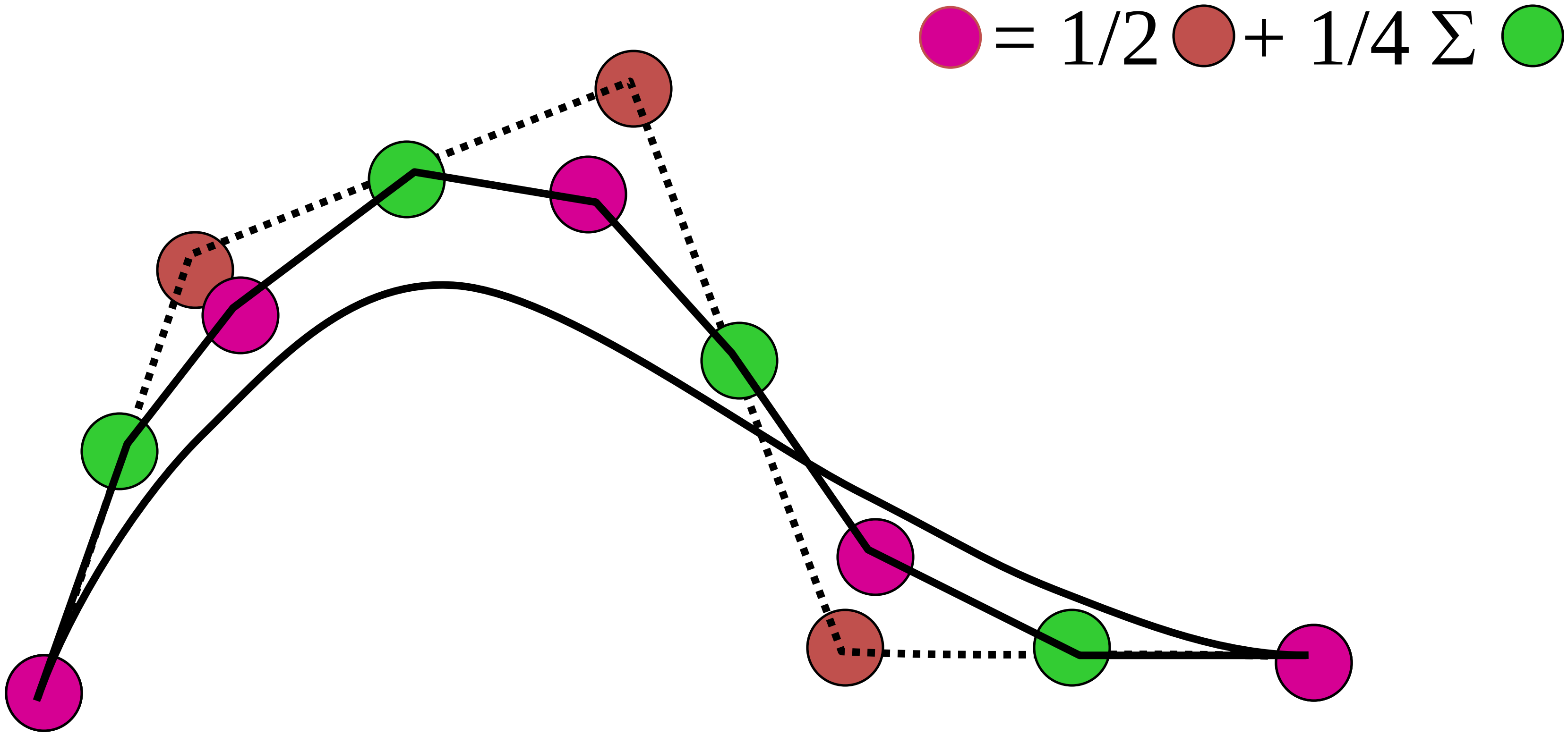
$$\boldsymbol{r}_i(v) = \sum_j B_j(v)\, \boldsymbol{r}_{i,j}$$

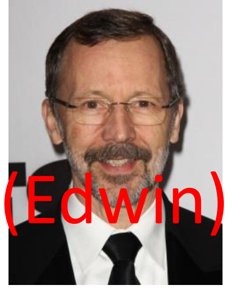


Poligonháló finomítása

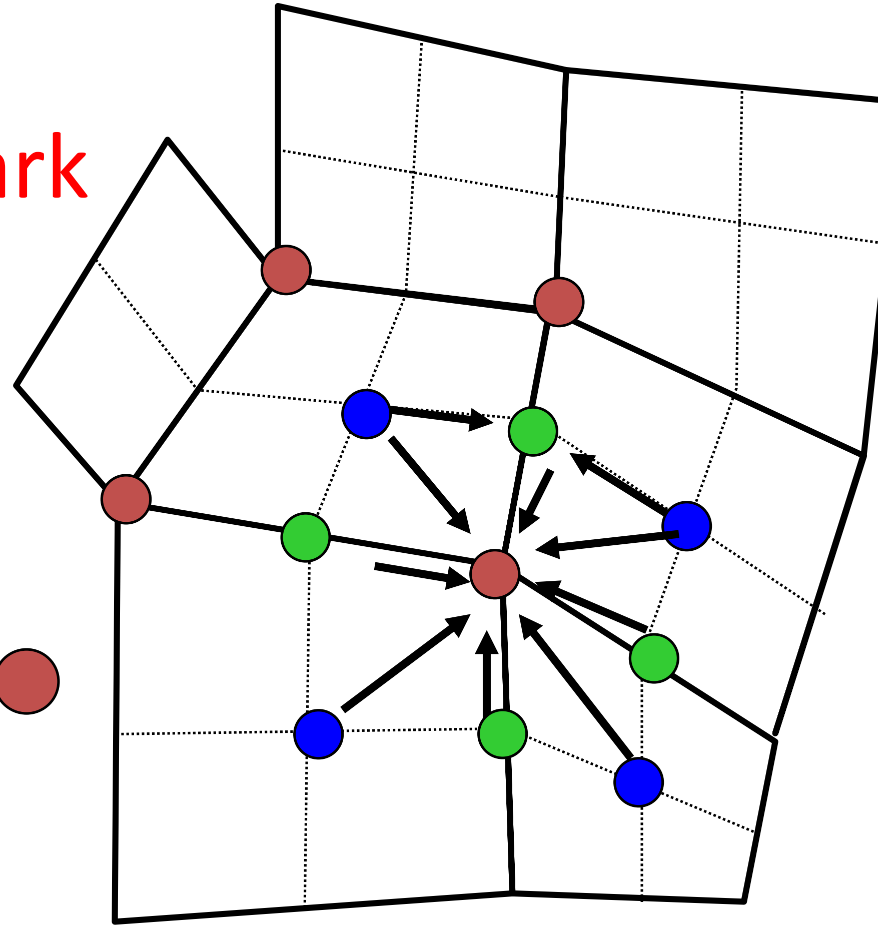


Subdivision görbék





(Edwin) Catmull- (James) Clark subdivision felület

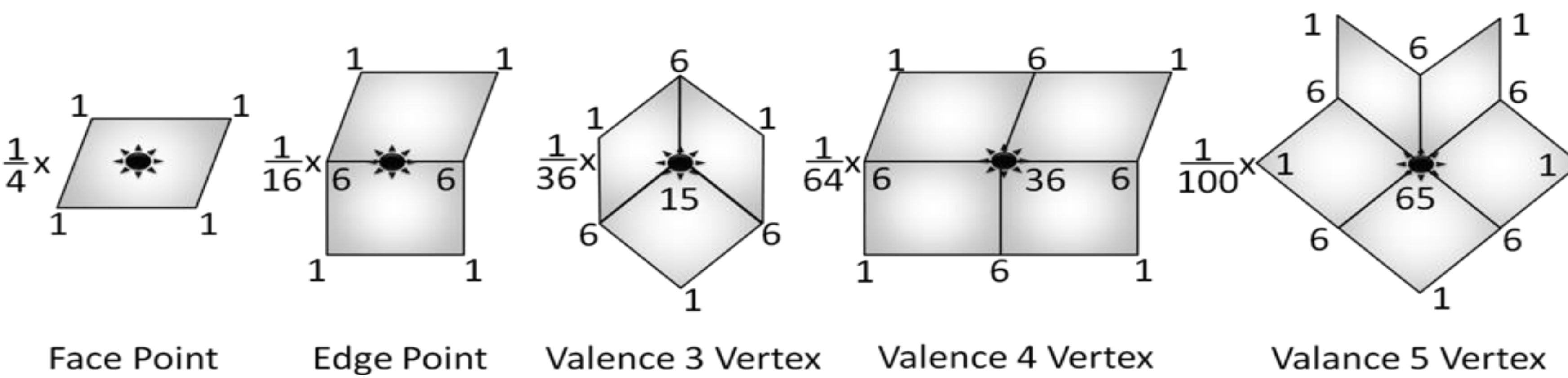


🔵 $= 1/4\Sigma$ 🔴

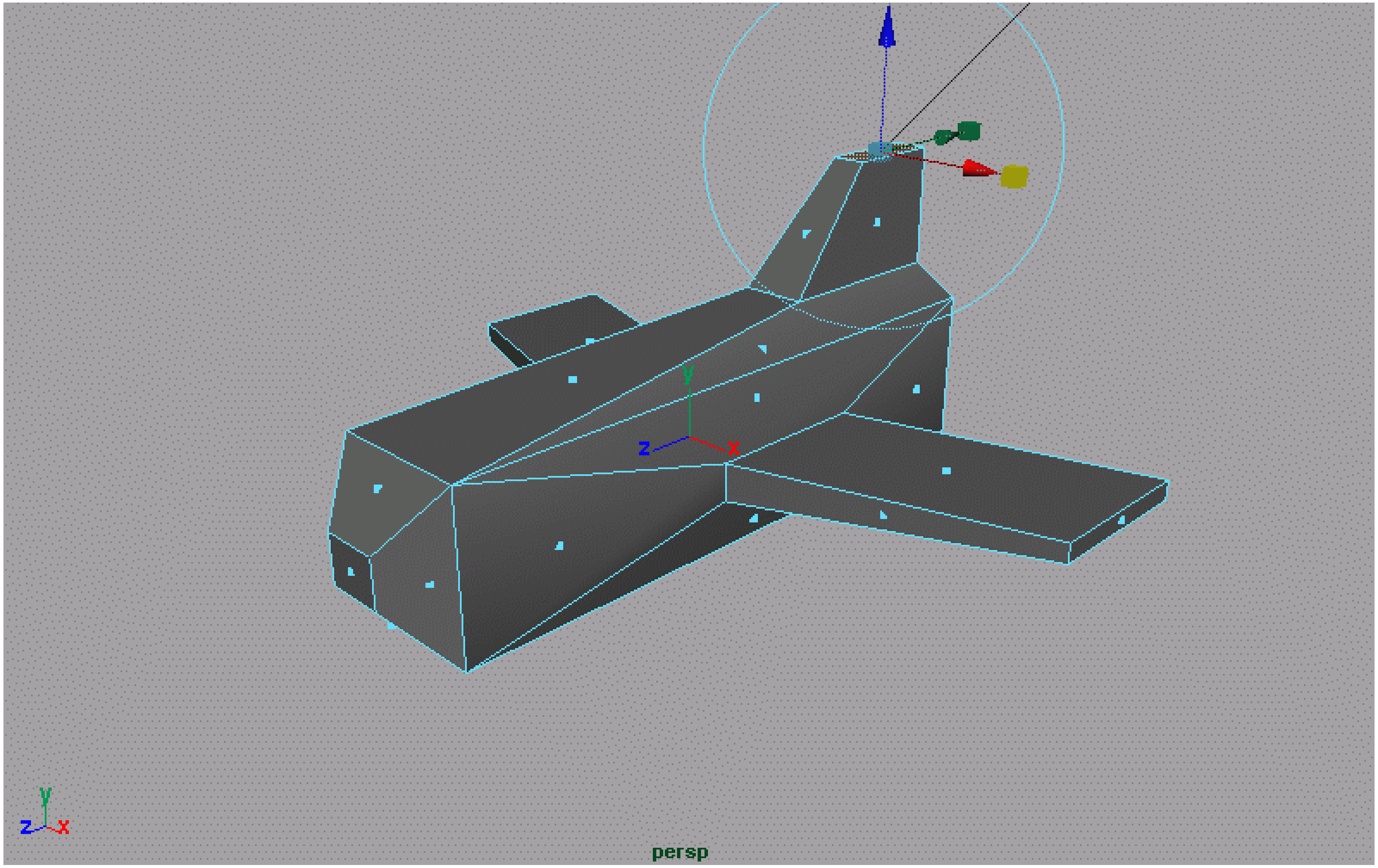
🟢i $= 1/2\Sigma$ 🔴

🔴 $= 1/v^2\Sigma$ 🔵 $+2/v^2\Sigma$ 🟢i $+(v-3)/v$ 🔴

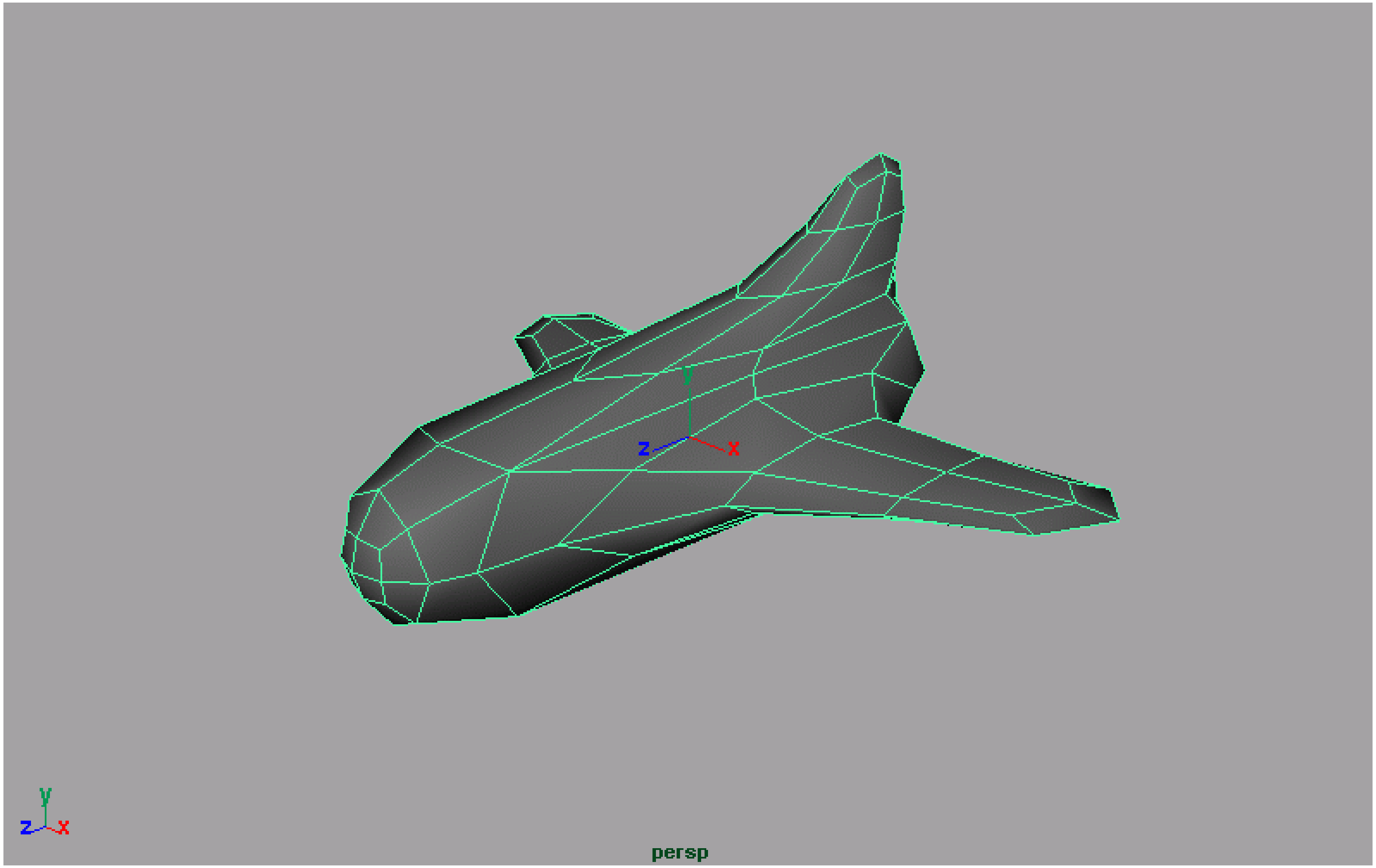
🟢 $= 1/4\Sigma$ 🔵 $+ 1/2$ 🟢i



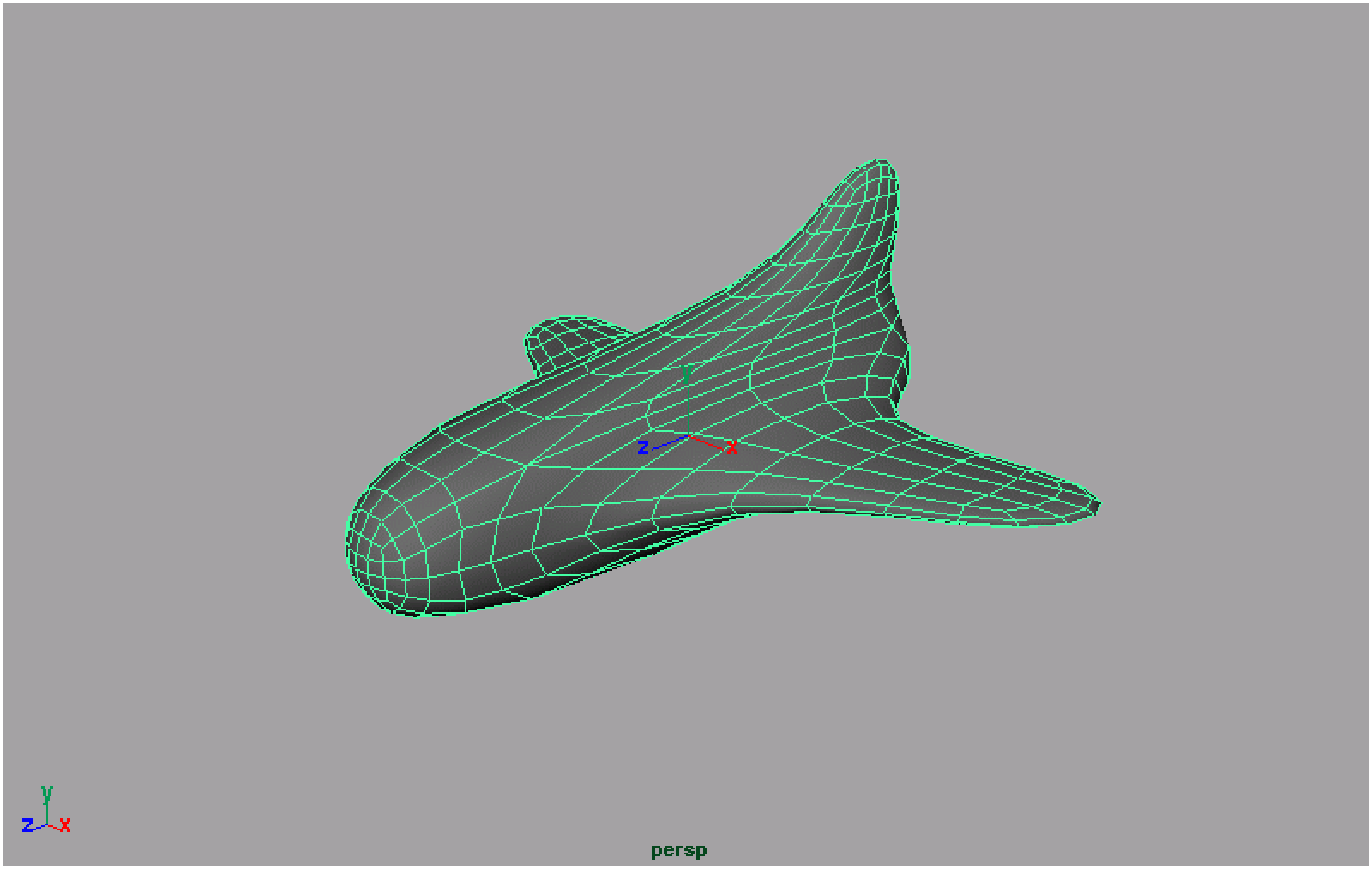
Durva poligon modell



Subdivision 1

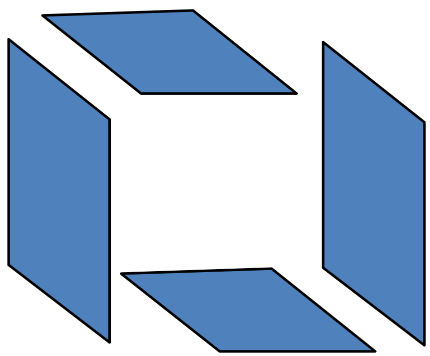


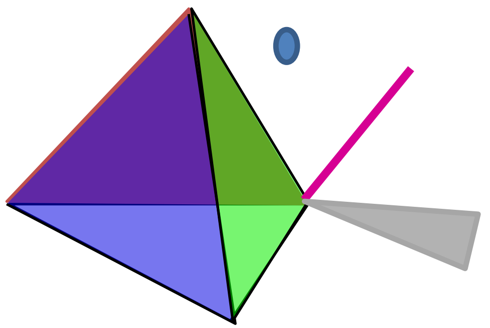
Subdivision 2



B-rep: határfelületek megadása

- Test = érvényes (létrehozható): ne legyenek alacsonyabb dimenziós elfajuló részek: minden határpont környezetében kell belső pontnak is lennie.



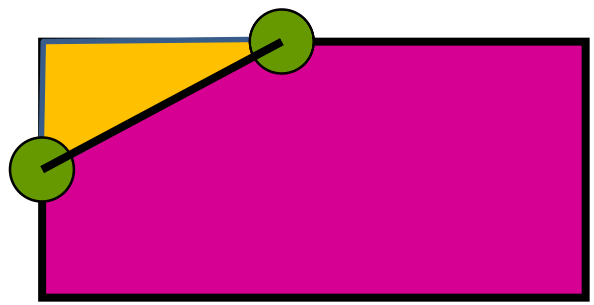


- Topológiai érvényesség:
  - Élek (2,3,…) csúcspontban találkoznak
  - Egy él két lapot választ el, és nem metsz élt
  - Egy lapot él és csúcs sorozat határol
  - A felület nem metszi saját magát
  - **Euler-Poincaré tétel**: $\boxed{\text{csúcs} - \text{él} + \text{lap} = 2(\text{db-lyuk})}$

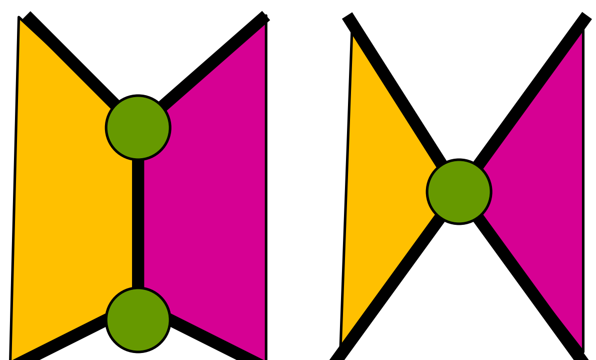
(Leonhard) Euler operátorok

- Lap kihúzás

- Lap felvágás

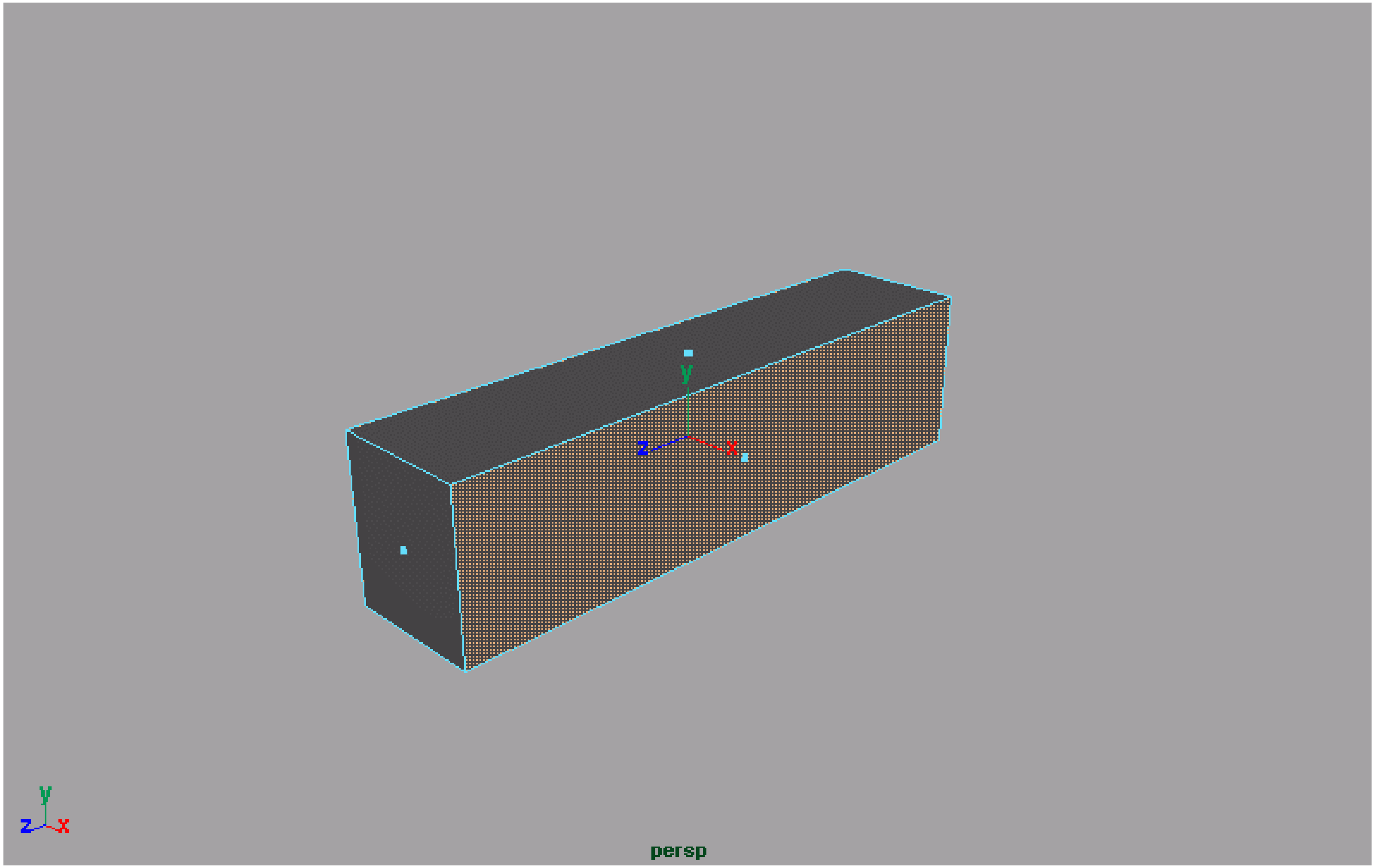


- Él törlés
- Csúcs szétvágás

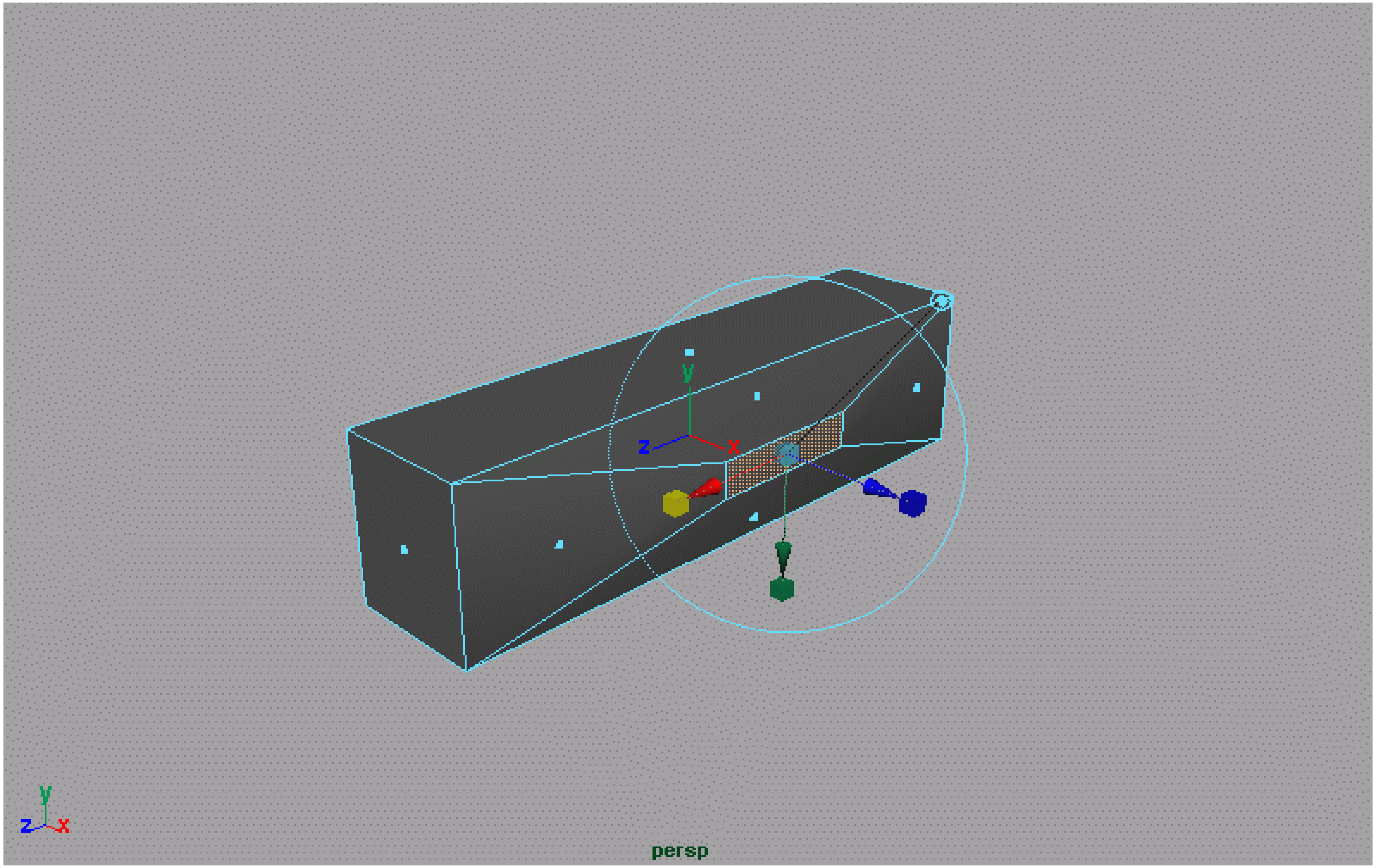


	Csúcsok	Lapok		Élek
Lap kihúzás	+4	+4	=	+8
Lap felvágás	+2	+1	=	+3
Él törlés	-1	0	=	-1
Csúcs szétvágás	+1	0	=	+1

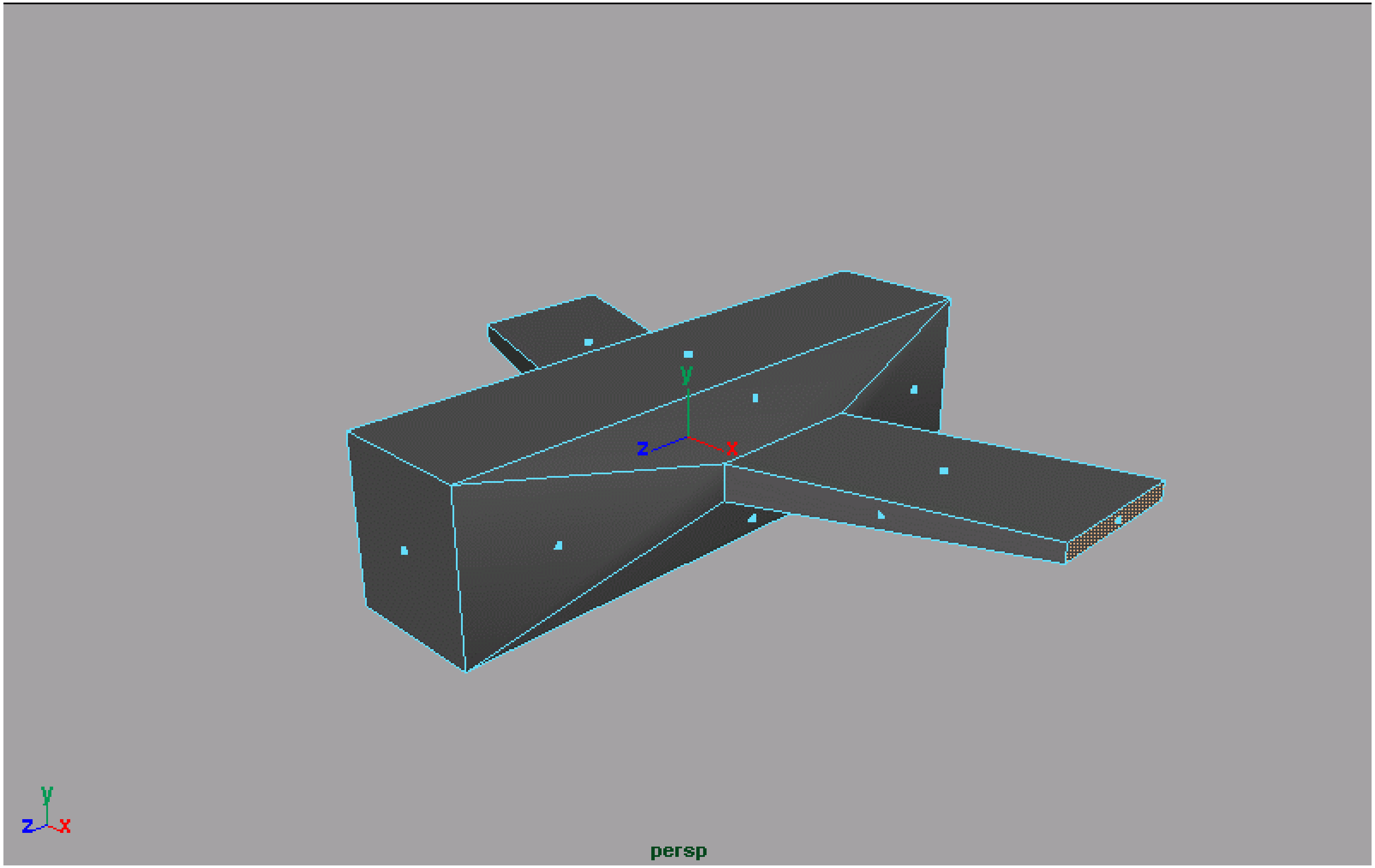
Kezdet: érvényes téglatest



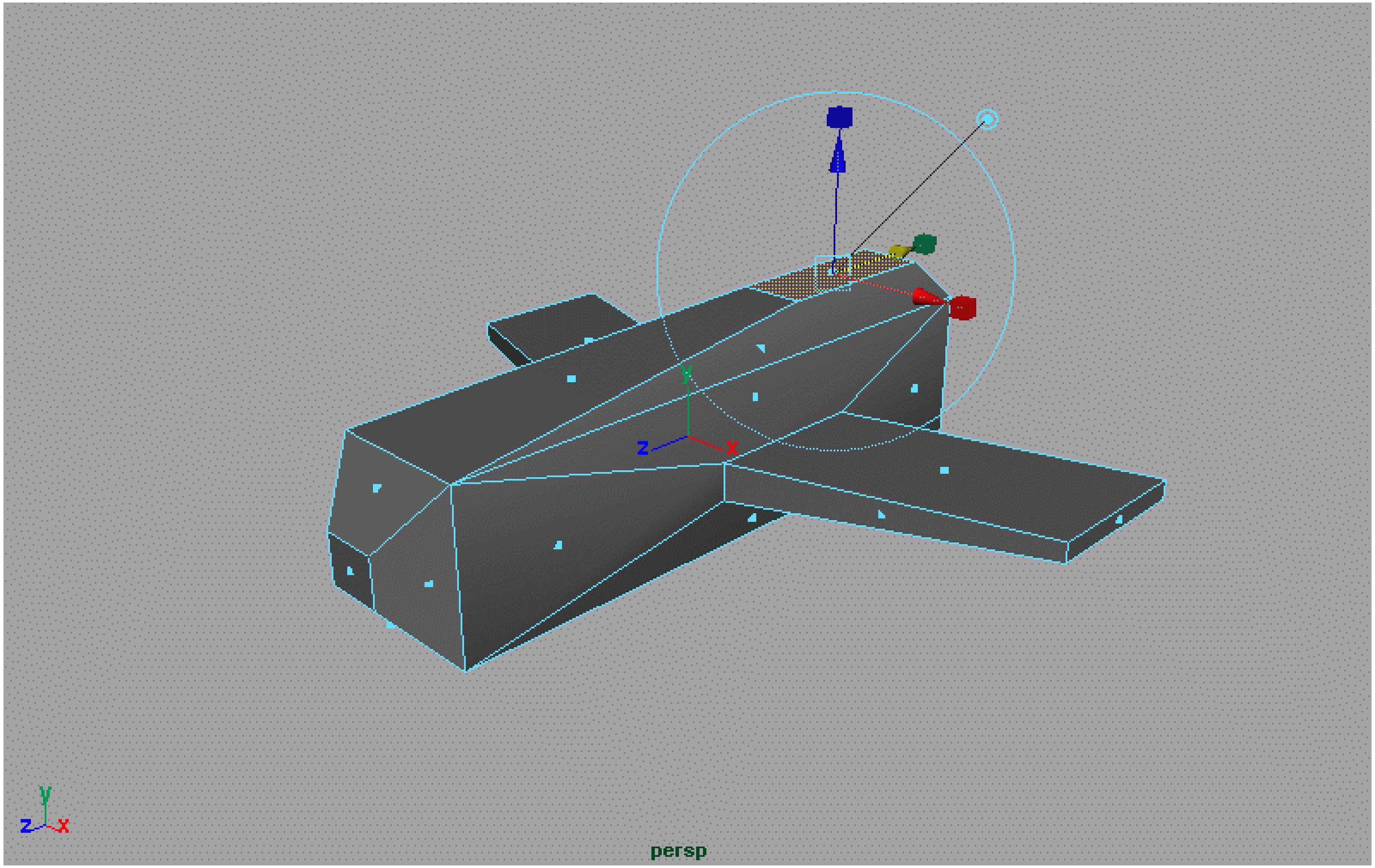
Lap kihúzás



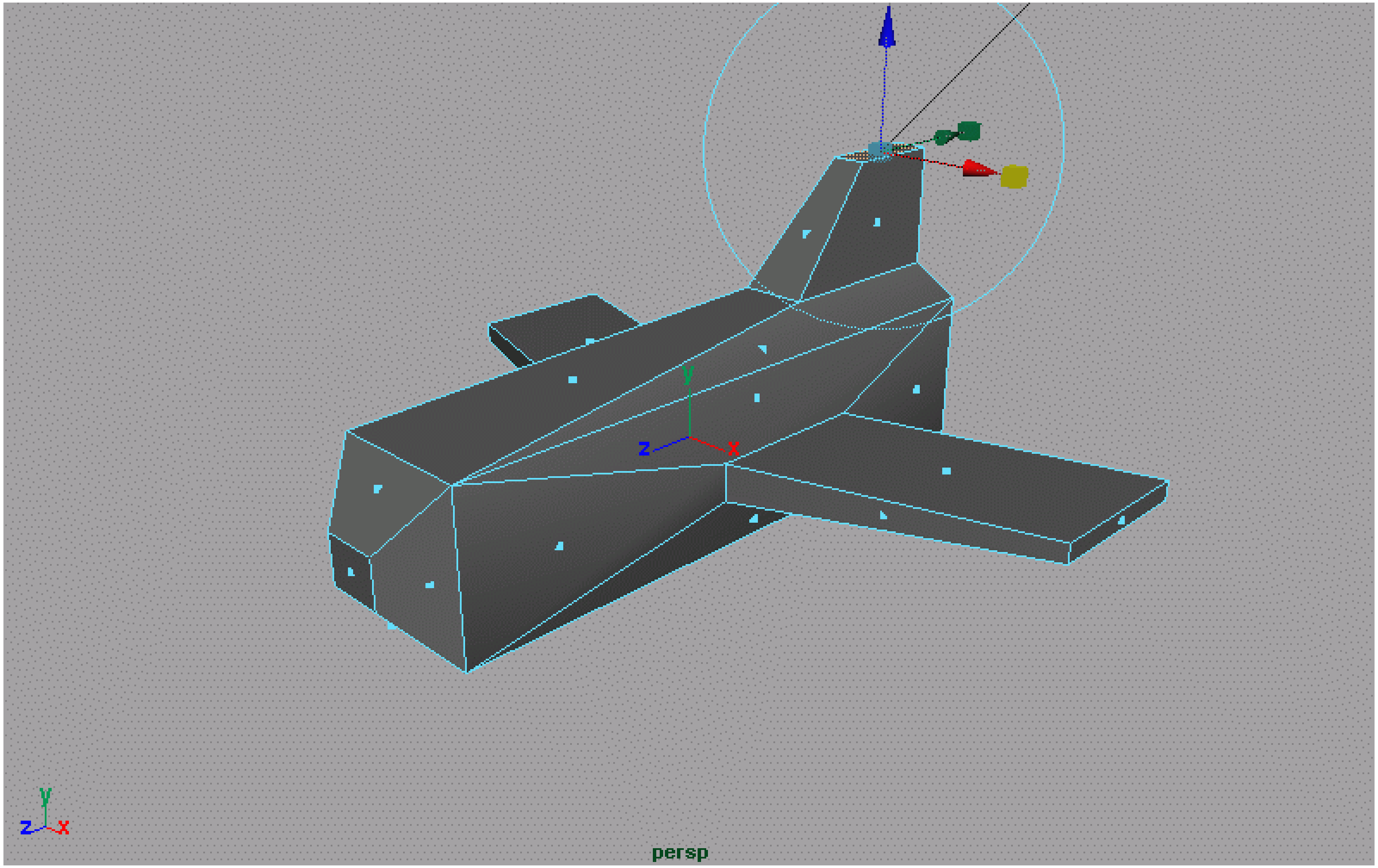
Lap kihúzás



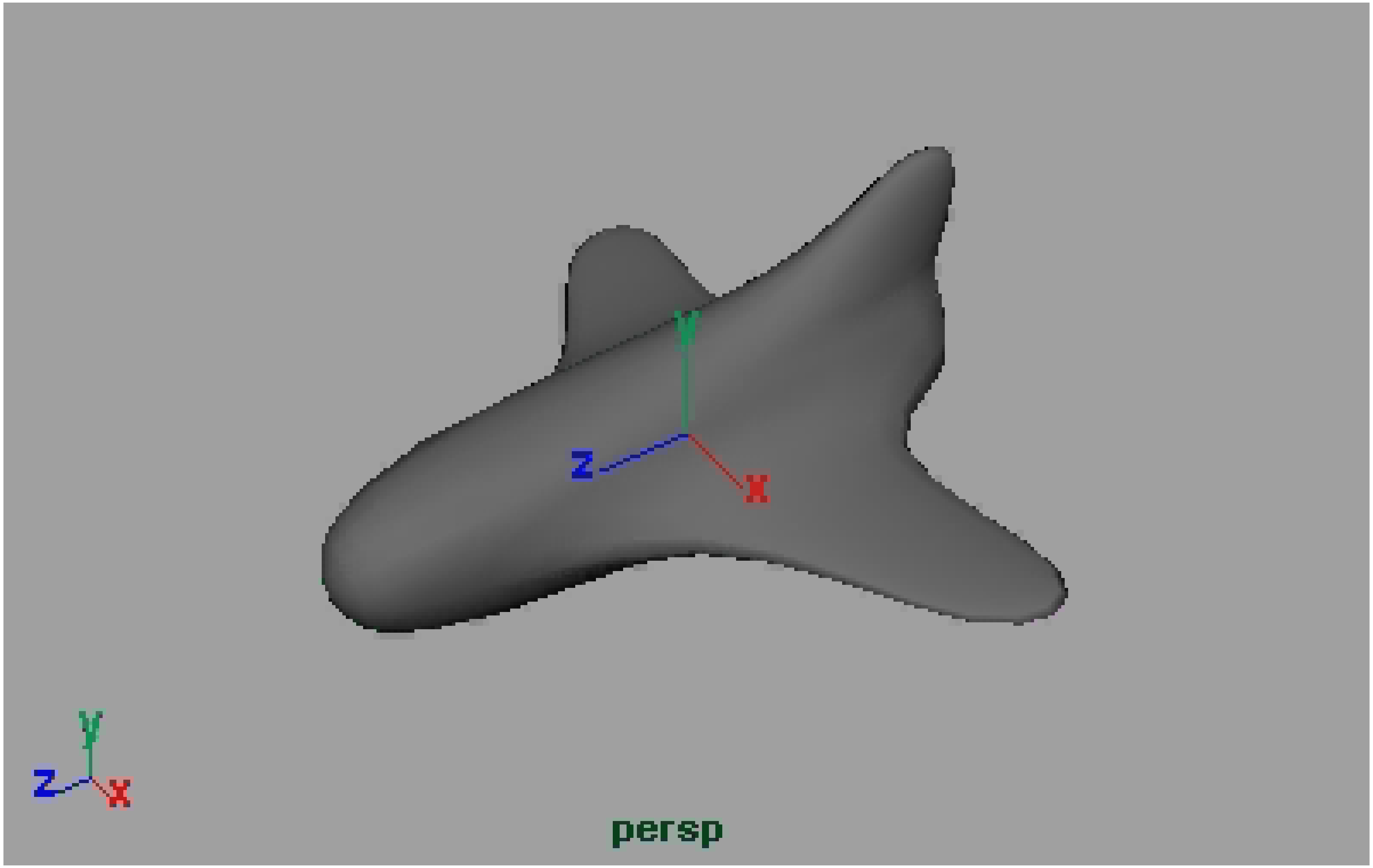
Lap kihúzás



Lap kihúzás



Subdivision simítás



Felület modellezés

- Explicit = magasságmező ritkán használható
- Implicit: geometriai definíció vektoralgebrai fordítása
  - Implicit felület normálvektor = gradiens
- Paraméteres: paraméteres görbékre vagy a súlypont analógiára vezetjük vissza
  - Paraméteres felület normálvektora = parciális deriváltak vektoriális szorzata
- Felosztott: háromszögháló

Ellenőrző kérdések

- Bizonyítsa be, hogy a Lagrange bázisfüggvények összege 1 (pl. teljes indukció)
- Bizonyítsa be, hogy a görbék koordinátarendszer függetlensége megköveteli, hogy a bázisfüggvények összege 1 legyen.
- Írja fel az egyenletes Catmull-Rom spline bázisfüggvényeit!
- Írja fel a tórusz paraméteres és implicit egyenleteit.
- Implementáljon PowerPoint-szerű szabadformájú görbét!
- Soroljon fel minél több Euler operátort!
- *Tervezzen adatstruktúrát egy poliéderhez! Hogyan implementálható azon a Catmull-Clark Subdivision?*
- Adja meg az egyenes másodfokú egyenletét!
- Általánosítsa az Euler tételt több darabból álló és lyukas objektumokra.
- Írjon óraprogramot: számjegyek animált Catmull-Rom spline-ok.
- Írjon programot, amely pontokkal adott függvényt interpolál, integrál és differenciál.
- Animálja végig a sebesség és gyorsulás vektorokat a Bézier, Lagrange és Catmull-Rom görbéken.