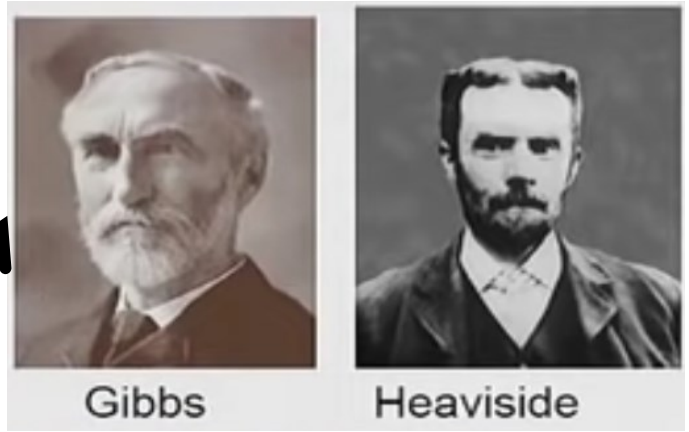
Vektor háború

Szirmay-Kalos László

Vektor háború

Gauss:
Komplex szám
1825

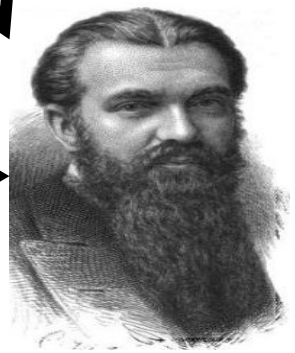
Hamilton:
Kvaternió
1843



$[x, y, z]$

Descartes
Koordináták
1637

Grassman:
Grassmann algebra
1844



Clifford:
Geometriai
Algebra
1879

2D geometria = vektor algebra

- Koordinátákkal számolható, de szerkeszthető is (tenzor)
- Pont: $\boldsymbol{p} = [x, y, 1]$
- Vektor: $\boldsymbol{v} = [x, y, 0]$
- Eltolás: $\boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{v}$ Invertálható, van kivonás
- (Eltolás), forgatás, skálázás:

$$[x', y', 1] = [x, y, 1] \begin{bmatrix} a & e & 0 \\ b & f & 0 \\ c & d & 1 \end{bmatrix}$$

Nincs rendes szorzás! Vektor osztás?

- Skaláris szorzás:
  - Kivezet a vektorok köréből
  - Nem asszociatív
  - Nem invertálható, független egyenletek száma kisebb, mint a vektor koordinátáinak száma:

$$\boldsymbol{v} \cdot \boldsymbol{a} = b \Rightarrow |\boldsymbol{v}||\boldsymbol{a}| \cos(\alpha) = b$$

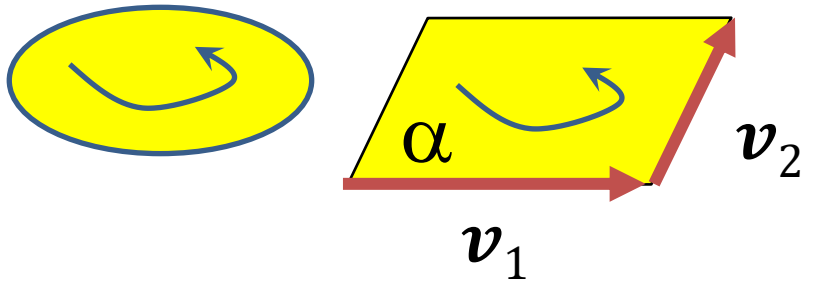
- Vektoriális szorzás:
  - Csak 3D-ben kétváltozós
  - Nem asszociatív
  - Nem invertálható

$$\boldsymbol{v} \times \boldsymbol{a} = \boldsymbol{b} \Rightarrow \boldsymbol{v} \text{ síkja ismert és } |\boldsymbol{v}||\boldsymbol{a}| \sin(\alpha) = |\boldsymbol{b}|$$

- A (skaláris + külső) invertálható volna

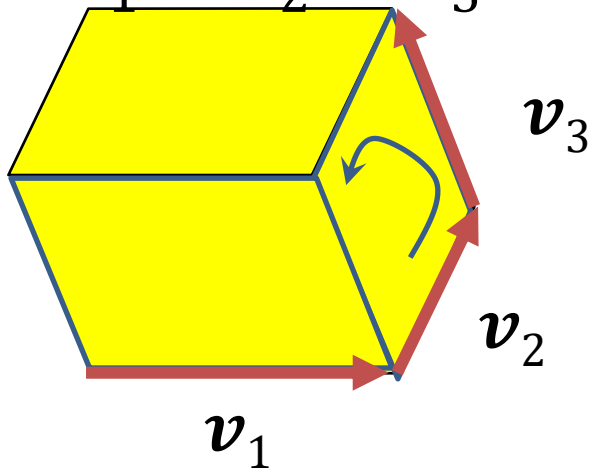
Külső (wedge) szorzat

$$\boldsymbol{B} = \boldsymbol{v}_1 \wedge \boldsymbol{v}_2$$



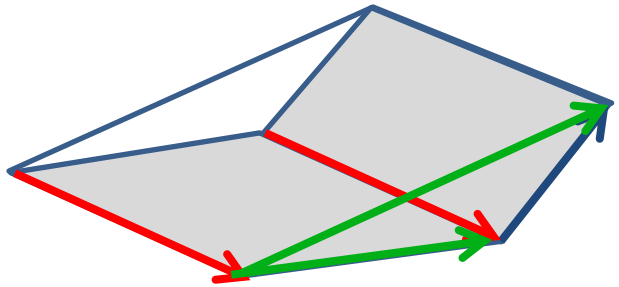
- **Definíció:** $|\boldsymbol{v}_1 \wedge \boldsymbol{v}_2| = |\boldsymbol{v}_1||\boldsymbol{v}_2|\sin(\alpha)$
- **Jelentés:** **<u>Irányított terület/térfogat</u>**
- **Multivektor:** $\mathbf{V} = s + \boldsymbol{v} + \boldsymbol{B}$
  - Összeadás, skálázás a szokásos módon
- **Tulajdonságok:**
  - <u>Asszociatív</u>
  - Antiszimmetrikus: $\boldsymbol{v}_1 \wedge \boldsymbol{v}_2 = -\boldsymbol{v}_2 \wedge \boldsymbol{v}_1$
  - Disztributív: $\boldsymbol{v}_1 \wedge (\boldsymbol{v}_2 + \boldsymbol{v}_3) = \boldsymbol{v}_1 \wedge \boldsymbol{v}_2 + \boldsymbol{v}_1 \wedge \boldsymbol{v}_3$

$$\boldsymbol{v}_1 \wedge \boldsymbol{v}_2 \wedge \boldsymbol{v}_3$$

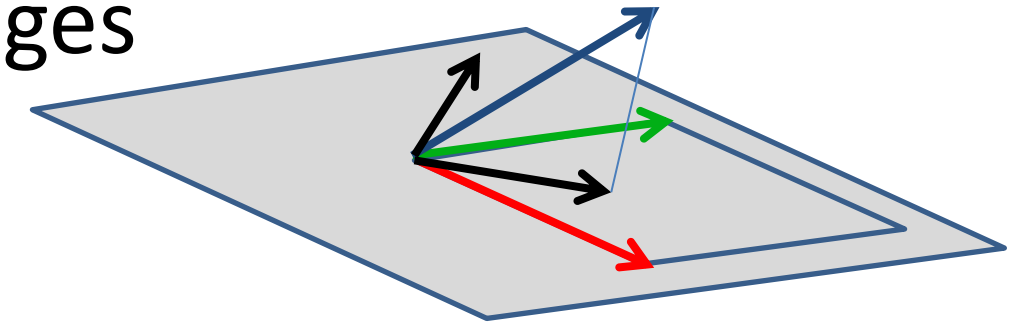


Műveletek bivektorokkal

- Számmal szorzás: értelemszerű
- Összeadás:

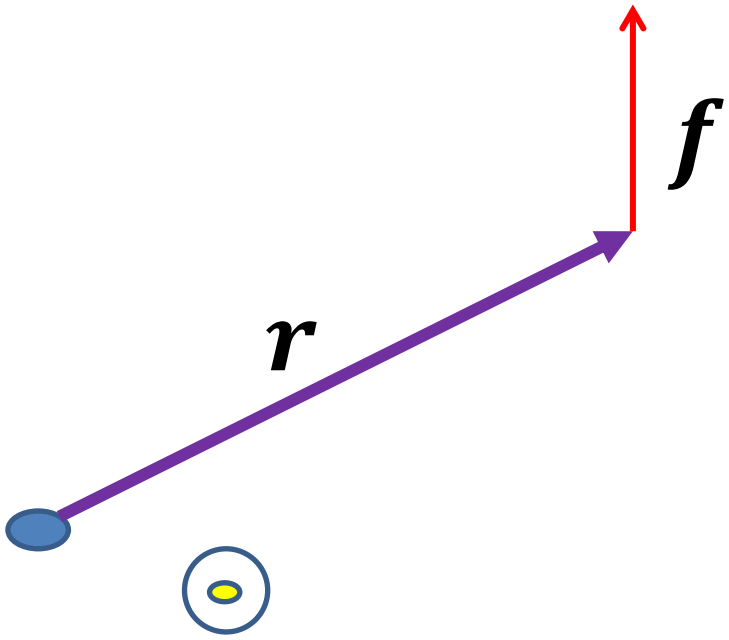


- Belső szorzat: A bivektor síkjában, a vektorra merőleges



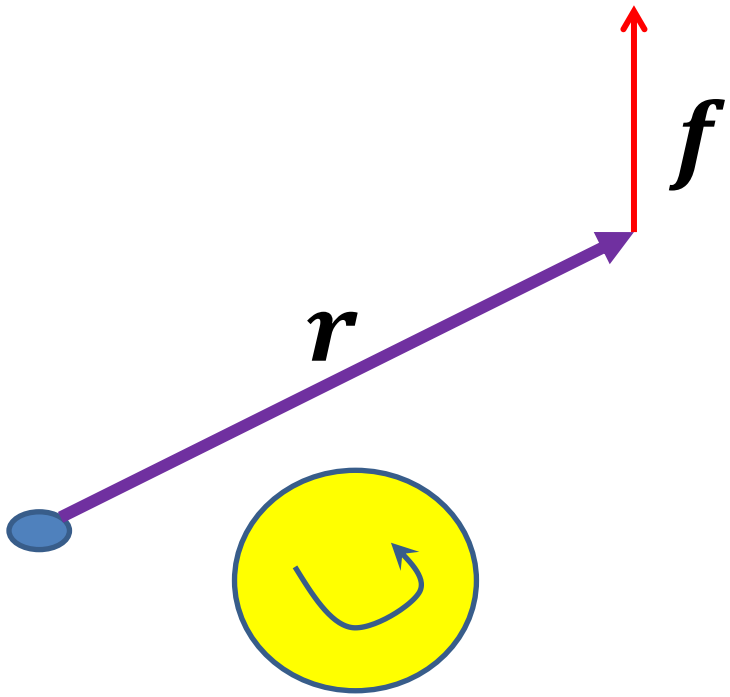
Külső szorzat példa: Forgatónyomaték

Vektoriális szorzat



$$\boldsymbol{M} = \boldsymbol{r} \times \boldsymbol{f}$$

Külső szorzat



$$\boldsymbol{M} = \boldsymbol{r} \wedge \boldsymbol{f}$$

2D geometria = komplex szám

- Pont:

$$z_p = x_p + y_p \boldsymbol{i} = Re^{i\alpha} = R\cos\alpha + \boldsymbol{i}R\sin\alpha$$

- Eltolás: $z_t = x_t + y_t \boldsymbol{i}$

$$\boxed{z_p{}' = z_p + z_t}$$

- Irányfüggetlen skálázás: $z_s = s$

$$\boxed{z_p{}' = z_p \cdot z_s}$$

- Forgatva nyújtás: $z_r = x_r + y_r \boldsymbol{i} = se^{i\varphi}$

$$\boxed{z_p{}' = z_p \cdot z_r} = Rs \cdot e^{\boldsymbol{i}(\alpha+\varphi)}$$

- Forgatás = egység abszolút értékű komplex szám

Komplex számok algebrája

```
struct Complex {
   float x, y;
   Complex(float x0, float y0) { x = x0, y = y0; }
   Complex operator+(Complex r) { return Complex(x + r.x, y + r.y); }
   Complex operator-(Complex r) { return Complex(x - r.x, y - r.y); }
   Complex operator*(Complex r) {
      return Complex(x * r.x - y * r.y, x * r.y + y * r.x);
   }
   Complex operator/(Complex r) {
      float l = r.x * r.x + r.y * r.y;
      return (*this) * Complex(r.x / l, -r.y / l); // conjugate
   }
};

Complex Polar(float r, float phi) { // Constructor
    return Complex(r * cosf(phi), r * sinf(phi));
}
```

2D transzformációk komplex számokkal

A **p** pontot az (1,-1) pivot pont körül nyújtsuk 2-szeresére és forgassuk el t-vel, majd toljuk el a (2, 3) vektorral és végül nyújtsuk az origó körül 0.8-szorosára és forgassuk –t/2-radiánnal:

```
Complex p, tp;
Complex pivot(1,-1);
tp = (((p - pivot) * Polar(2,t) + pivot) + Complex(2,3)) * Polar(0.8,-t/2);
```

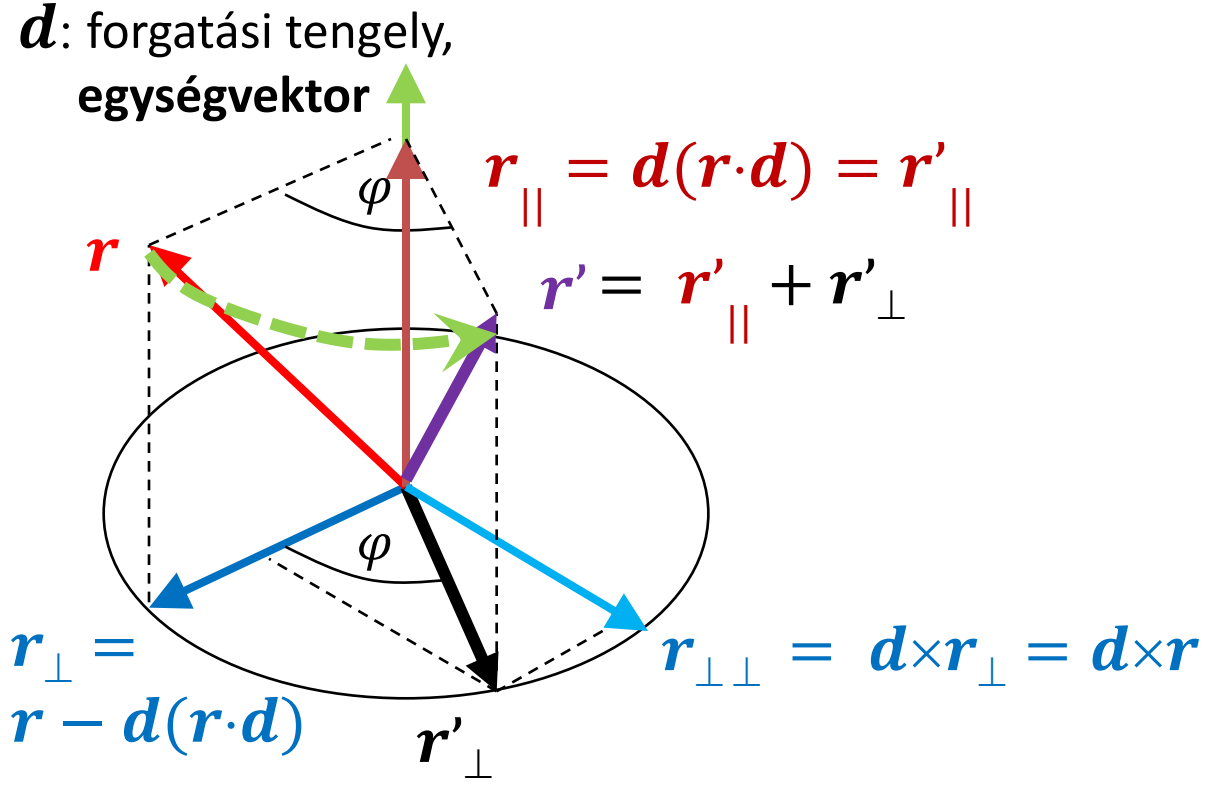
És 3D-ben?

- Eltolás, skálázás (összeadás, skalárral szorzás) tetszőleges dimenzióban általánosítható.
- Forgatás?
  - Ha lineáris művelet, akkor mátrixszal megadható: $\boldsymbol{r'} = \boldsymbol{R}(\boldsymbol{r})$

$$x'\boldsymbol{i} + y'\boldsymbol{j} + z'\boldsymbol{k} = \boldsymbol{R}(x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}) = x\boldsymbol{R}(\boldsymbol{i}) + y\boldsymbol{R}(\boldsymbol{i}) + z\boldsymbol{R}(\boldsymbol{k})$$

$$[x', y', z'] = [x, y, z]\begin{bmatrix} \boldsymbol{R}(\boldsymbol{i}).x & \boldsymbol{R}(\boldsymbol{i}).y & \boldsymbol{R}(\boldsymbol{i}).z \\ \boldsymbol{R}(\boldsymbol{j}).x & \boldsymbol{R}(\boldsymbol{j}).y & \boldsymbol{R}(\boldsymbol{j}).z \\ \boldsymbol{R}(\boldsymbol{k}).x & \boldsymbol{R}(\boldsymbol{k}).x & \boldsymbol{R}(\boldsymbol{j}).z \end{bmatrix}$$

Origón átmenő $\boldsymbol{d}$ tengely körüli forgatás:
(Olinde) Rodrigues formula

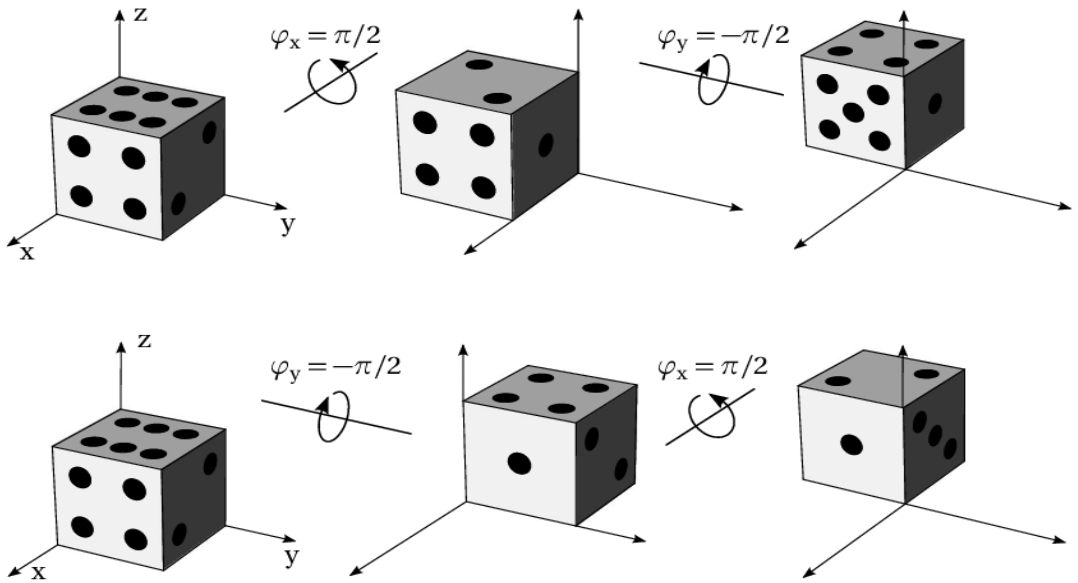


$\boldsymbol{r}'_{||}$ $\boldsymbol{r}_{\perp}$ $\boldsymbol{r}_{\perp\perp}$

$$\boldsymbol{r}' = \boldsymbol{d}(\boldsymbol{r}\cdot\boldsymbol{d}) + (\boldsymbol{r} - \boldsymbol{d}(\boldsymbol{r}\cdot\boldsymbol{d}))\cos(\varphi) + \boldsymbol{d}\times\boldsymbol{r}\sin(\varphi)$$
$$= \boldsymbol{r}\cos(\varphi) + \boldsymbol{d}(\boldsymbol{r}\cdot\boldsymbol{d})(1 - \cos(\varphi)) + \boldsymbol{d}\times\boldsymbol{r}\sin(\varphi)$$

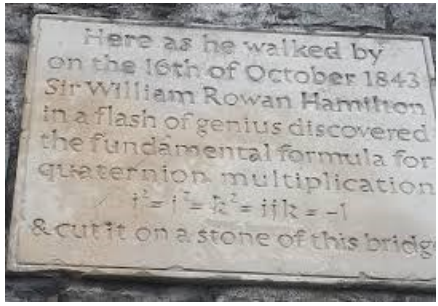
Komplex szám működik 3D-ben?

- $z = x + y\boldsymbol{i} + z\boldsymbol{j}$
- Összeadás és irányfüggetlen skálázás OK
- **Forgatás mint szorzás?** Tulajdonságok:
  - Asszociatív, összeadásra disztributív (biz: mátrix)
  - Nem kommutatív
  - Invertálható



- $\boldsymbol{i}^2 = ?, \quad \boldsymbol{j}^2 = ?, \quad \boldsymbol{ij} = ?, \quad \boldsymbol{ji} = ?$

(Sir William Rowan) Hamilton
Kvaternió: 4D komplex szám



- $\boldsymbol{q} = [s, x, y, z] = [s, \boldsymbol{d}] = s + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$
- $\boldsymbol{q}_1 + \boldsymbol{q}_2 = [s_1 + s_2, x_1 + x_2, y_1 + y_2, z_1 + z_2]$
- $a\boldsymbol{q} = \boldsymbol{q}a = [as, ax, ay, az]$
- $|\boldsymbol{q}| = \sqrt{s^2 + x^2 + y^2 + z^2}$
- **Szorzás:**
  - $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$

$$\boldsymbol{ij} = \boldsymbol{k},\ \ \boldsymbol{ji} = -\boldsymbol{k},\ \ \boldsymbol{jk} = \boldsymbol{i},\ \ \boldsymbol{kj} = -\boldsymbol{i},\ \ \boldsymbol{ki} = \boldsymbol{j},\ \ \boldsymbol{ik} = -\boldsymbol{j}$$

$$[s_1, \boldsymbol{d}_1]\cdot[s_2, \boldsymbol{d}_2] = [s_1 s_2 - \boldsymbol{d}_1\cdot\boldsymbol{d}_2, s_1\boldsymbol{d}_2 + s_2\boldsymbol{d}_1 + \boldsymbol{d}_1 \times \boldsymbol{d}_2]$$

  - Szorzás asszociatív, **de nem kommutatív**,
  - Összeadásra disztributív
  - Van egységelem: $[1,0,0,0]$
  - Van inverz: $\boldsymbol{q}^{-1} = [s, -\boldsymbol{d}]/|\boldsymbol{q}|^2$, $\boldsymbol{q}^{-1}\cdot\boldsymbol{q} = \boldsymbol{q}\cdot\boldsymbol{q}^{-1} = [1,0,0,0]$

	1	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{k}$
1	1	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{k}$
$\boldsymbol{i}$	$\boldsymbol{i}$	-1	$\boldsymbol{k}$	$-\boldsymbol{j}$
$\boldsymbol{j}$	$\boldsymbol{j}$	$-\boldsymbol{k}$	-1	$\boldsymbol{i}$
$\boldsymbol{k}$	$\boldsymbol{k}$	$\boldsymbol{j}$	$-\boldsymbol{i}$	-1

Kvaternió = forgatás α szöggel az origón átmenő *d* irányú tengely körül

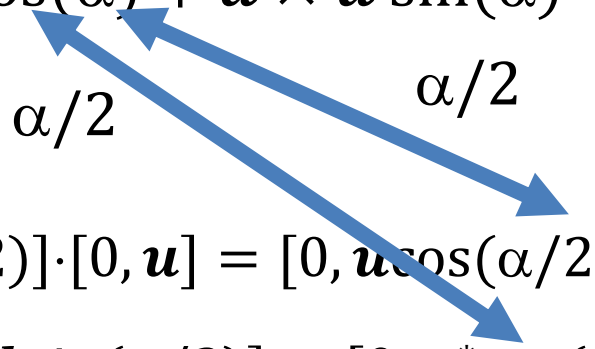
$$\boldsymbol{q} = [\cos(\alpha/2), \boldsymbol{d}\sin(\alpha/2)], \qquad |\boldsymbol{d}| = 1$$

$\boldsymbol{q}\cdot[0,\boldsymbol{u}]\cdot\boldsymbol{q}^{-1} = [0,\boldsymbol{v}]$, $\boldsymbol{v}$ az $\boldsymbol{u}$ elforgatottja a $\boldsymbol{d}$ körül α-val

Rodriguez: $\boldsymbol{v} = \boldsymbol{u}\cos(\alpha) + \boldsymbol{d}(\boldsymbol{u}\cdot\boldsymbol{d})(1-\cos(\alpha)) + \boldsymbol{d}\times\boldsymbol{u}\sin(\alpha)$

Bizonyítás $\boldsymbol{d}$ merőleges $\boldsymbol{u}$ esetre (párhuzamos→HF):

Rodriguez: $\boldsymbol{v} = \boldsymbol{u}\cos(\alpha) + \boldsymbol{d}\times\boldsymbol{u}\sin(\alpha)$



Kvaternió:

$$[\cos(\alpha/2), \boldsymbol{d}\sin(\alpha/2)]\cdot[0,\boldsymbol{u}] = [0, \boldsymbol{u}\cos(\alpha/2) + \boldsymbol{d}\times\boldsymbol{u}\sin(\alpha/2)] = [0,\boldsymbol{u}^*]$$

$$[0,\boldsymbol{u}^*]\cdot[\cos(\alpha/2), -\boldsymbol{d}\sin(\alpha/2)] = [0, \boldsymbol{u}^*\cos(\alpha/2) - \boldsymbol{u}^*\times\boldsymbol{d}\sin(\alpha/2)]$$

$$[s_1,\boldsymbol{d}_1]\cdot[s_2,\boldsymbol{d}_2] = [s_1s_2 - \boldsymbol{d}_1\cdot\boldsymbol{d}_2, s_1\boldsymbol{d}_2 + s_2\boldsymbol{d}_1 + \boldsymbol{d}_1\times\boldsymbol{d}_2]$$

Implementáció: $q = s+xi+yj+zk$ = vec4

```
struct vec4 {
    float x, y, z, w;
    …
};
```

$$[s_1, \boldsymbol{d}_1]\cdot[s_2, \boldsymbol{d}_2] = [s_1 s_2 - \boldsymbol{d}_1\cdot\boldsymbol{d}_2, s_1\boldsymbol{d}_2 + s_2\boldsymbol{d}_1 + \boldsymbol{d}_1 \times \boldsymbol{d}_2]$$

```
vec4 qmul(vec4 q1, vec4 q2) { // kvaternió szorzás
    vec3 d1(q1.x, q1.y, q1.z), d2(q2.x, q2.y, q2.z);
    return vec4(d2 * q1.w + d1 * q2.w + cross(d1, d2),
                q1.w * q2.w - dot(d1, d2));
}

vec4 quaternion(float ang, vec3 axis) { // konstruálás
    vec3 d = normalize(axis) * sinf(ang / 2);
    return vec4(d.x, d.y, d.z, cosf(ang / 2));
}
```

```
vec3 Rotate(vec3 u, vec4 q) {
    vec4 qinv(-q.x, -q.y, -q.z, q.w);  // conjugate
    vec4 qr = qmul(qmul(q, vec4(u.x, u.y, u.z, 0)), qinv);
    return vec3(qr.x, qr.y, qr.z);
}
```

GPU shader programozás GLSL nyelven

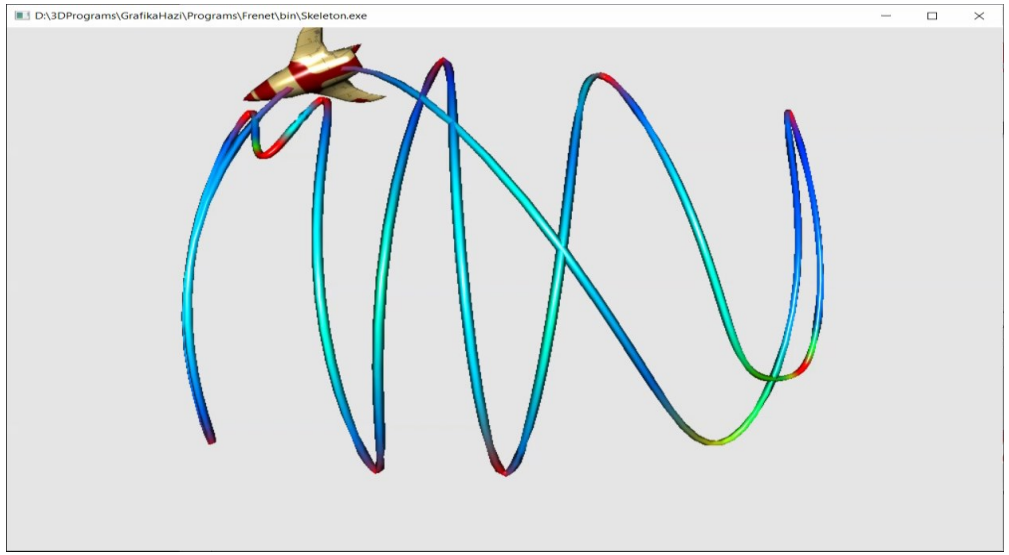
$$[s_1, \boldsymbol{d}_1]\cdot[s_2, \boldsymbol{d}_2] = [s_1 s_2 - \boldsymbol{d}_1\cdot\boldsymbol{d}_2, s_1\boldsymbol{d}_2 + s_2\boldsymbol{d}_1 + \boldsymbol{d}_1 \times \boldsymbol{d}_2]$$

```
uniform vec4 q;    // quaternion as uniform variable
in vec3 u;         // Varying input: vertex

vec4 qmul(vec4 q1, vec4 q2) {
   vec3 d1 = q1.xyz, d2 = q2.xyz;
   return vec4(d2 * q1.w + d1 * q2.w + cross(d1, d2),
               q1.w * q2.w - dot(d1, d2));
}

void main() { // vertex shader program
  vec4 qinv = vec4(-q.xyz, q.w);     // conjugate
  vec3 v = qmul(qmul(q, vec4(u, 0)), qinv).xyz;
  gl_Position = vec4(v, 1);
}
```

“Navigare necesse est,
vivere non est necesse.”
Cnaeus Pompeius Magnus

Automatikus deriválás

Szirmay-Kalos László



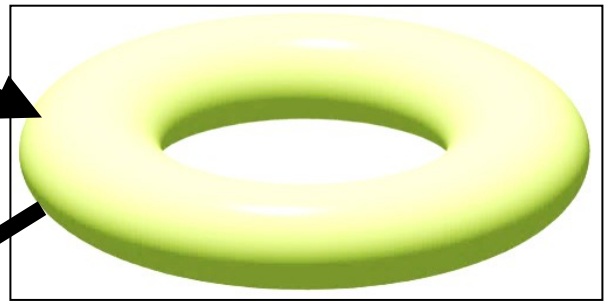
Inverz feladatok

Virtuális világ modell

Tórusz R, ρ sugarakkal

$$\left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 - \rho^2 = 0$$

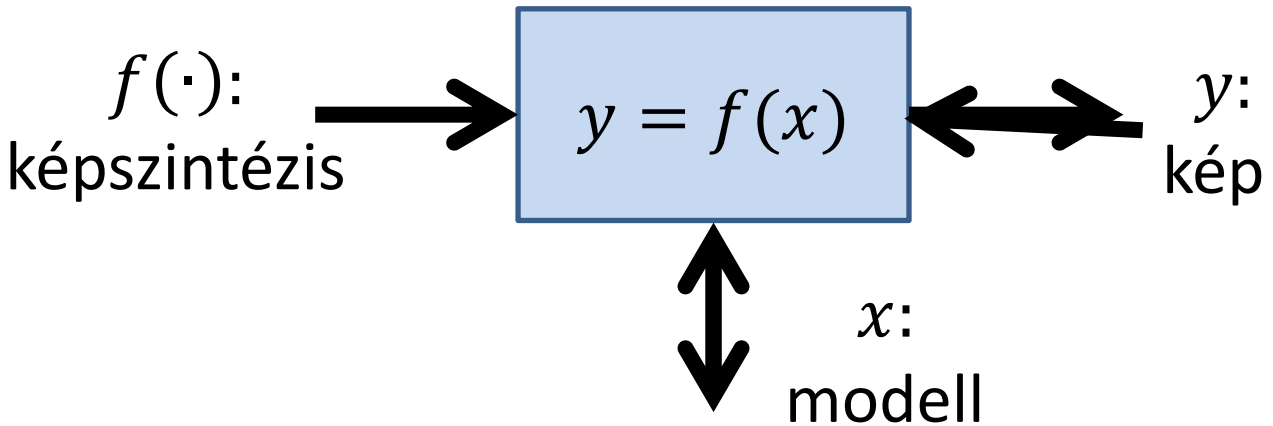
Kép

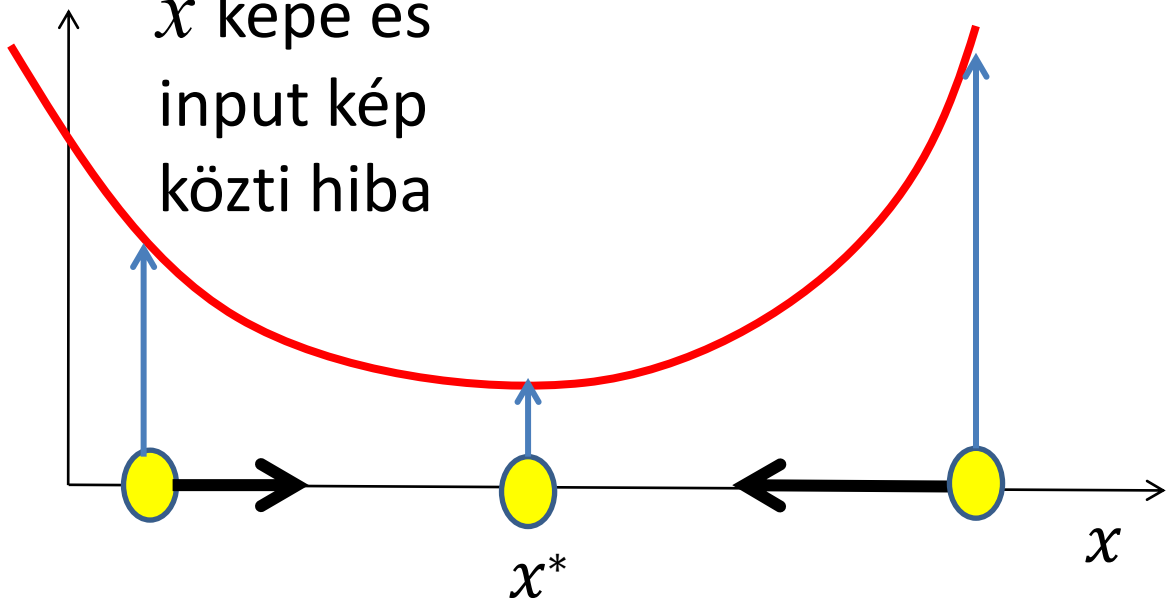


képszintézis

gépi látás

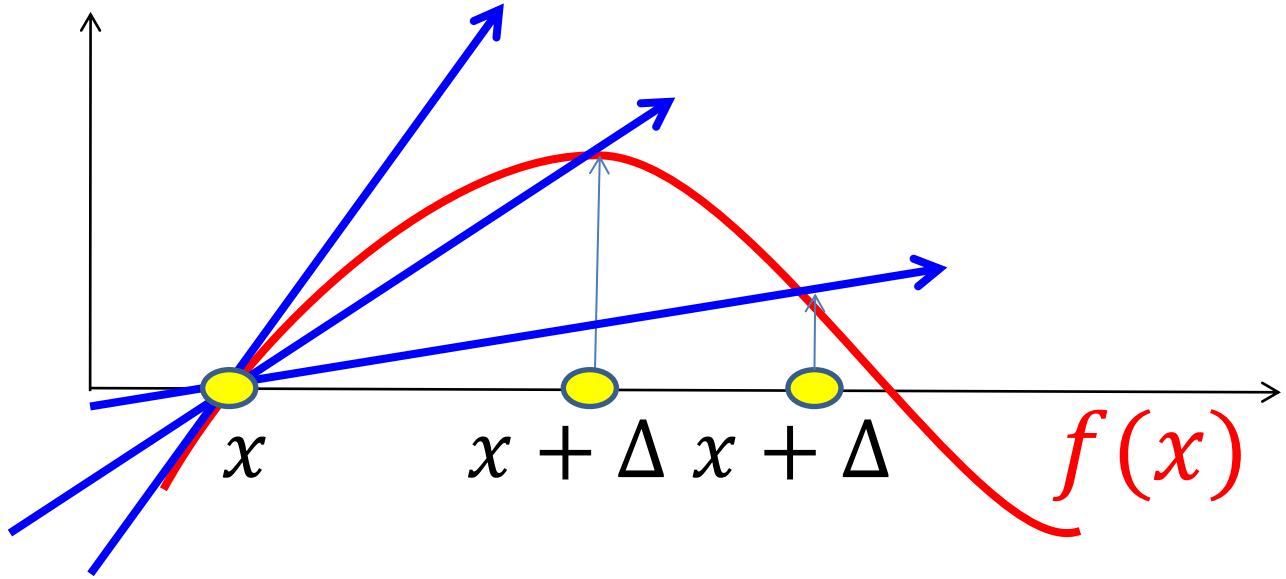
Függvény inverz





Hogyan NE deriváljunk!

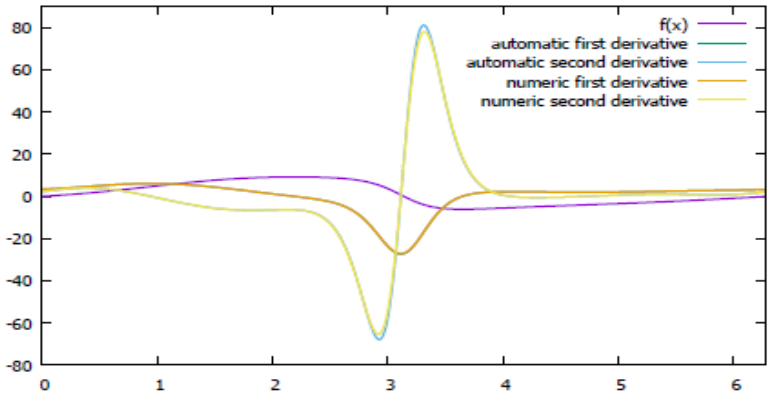
$$f'(x) \approx \frac{f(x+\Delta) - f(x)}{\Delta}$$



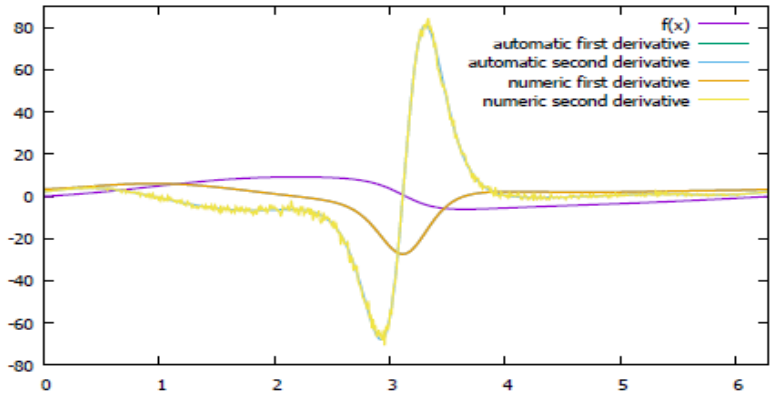
Kivonás: 💣☠✝

1234.56**9**?????	7 értékes jegy
−1234.56**7**?????	7 értékes jegy
0.00**2**?????	**1** értékes jegy

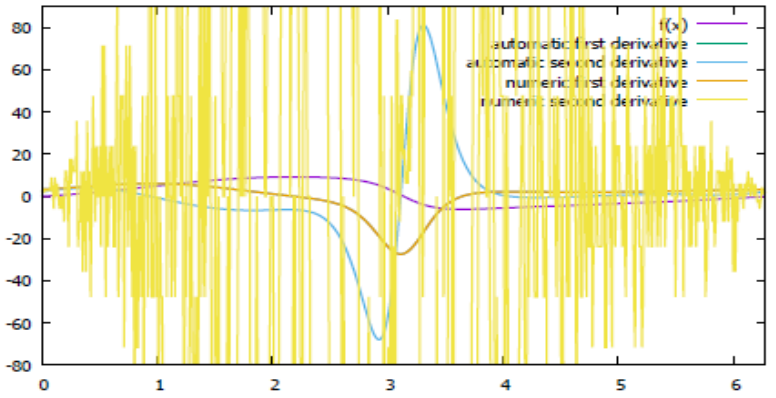
$$\frac{\sin(t)\,(\sin(t)+3)4}{\tan(\cos(t)+2)}$$



$\Delta = 0.1$



$\Delta = 0.001$



$\Delta = 0.0001$

(William) Clifford algebra: Hiperszám

- Tanítsuk meg a C++-t deriválni (is)!

függvény | derivált

- Hiperszám: $z = x + y\boldsymbol{i}$, ahol
  - $\boldsymbol{i}^2 = -1$: komplex szám; $\boldsymbol{i}^2 = 1$: hiperbolikus szám;
  - $\boldsymbol{i}^2 = 0$: **duális szám,** a deriváláshoz ez kell

össz/különb | függvény össz/kül | össz/kül deriváltja

$$(x_1+y_1\boldsymbol{i}) \pm (x_2+y_2\boldsymbol{i}) = (x_1 \pm x_2) + (y_1 \pm y_2)\boldsymbol{i}$$

szorzat | függvény szorzat | szorzat deriváltja

$$(x_1+y_1\boldsymbol{i})\cdot(x_2+y_2\boldsymbol{i}) = (x_1x_2) + (x_1y_2+y_1x_2)\boldsymbol{i} + (y_1y_2)\boldsymbol{i}^2$$

hányados | függvény hányados | hányados deriváltja

$$\frac{x_1+y_1\boldsymbol{i}}{x_2+y_2\boldsymbol{i}} = \frac{(x_1+y_1\boldsymbol{i})(x_2-y_2\boldsymbol{i})}{(x_2+y_2\boldsymbol{i})(x_2-y_2\boldsymbol{i})} = \frac{x_1x_2+(y_1x_2-x_1y_2)\boldsymbol{i}-(y_1y_2)\boldsymbol{i}^2}{x_2^2-y_2^2\boldsymbol{i}^2} = \frac{x_1}{x_2} + \frac{y_1x_2-x_1y_2}{x_2^2}\boldsymbol{i}$$

Miért működik?

Taylor sor:

$$f(x+\varepsilon) = f(x) + f'(x)\varepsilon + f^r(x)\varepsilon^2$$
$$g(x+\varepsilon) = g(x) + g'(x)\varepsilon + g^r(x)\varepsilon^2$$

$$(fg)(x+\varepsilon) = (fg)(x) + (fg)'(x)\varepsilon + (fg)^r(x)\varepsilon^2$$
$$f(x+\varepsilon)g(x+\varepsilon) = f(x)g(x) + \big(f(x)g'(x) + f'(x)g(x)\big)\varepsilon + (fg)^r(x)\varepsilon^2$$

Elsőrendű közelítés:

$$f(x+\varepsilon) \approx f(x) + f'(x)\varepsilon, \qquad \varepsilon^2 = 0$$

Duális szám osztály

```
struct Dnum {
    float f, d; // function and derivative values

    Dnum(float f0, float d0 = 0) { // constant' = 0
       f = f0, d = d0;
    }

    Dnum operator+(Dnum r) { return Dnum(f + r.f, d + r.d); }

    Dnum operator-(Dnum r) { return Dnum(f - r.f, d - r.d); }

    Dnum operator*(Dnum r) { return Dnum(f * r.f, f * r.d + d * r.f); }

    Dnum operator/(Dnum r) {
       return Dnum(f / r.f, (d * r.f - f * r.d) / r.f / r.f);
    }
};
```

Duális szám alkalmazása

- Deriválás nélkül:

```
float t = value;
float F = t * a / (t * t + b);
```

- Deriválással együtt:

```
Dnum F = Dnum(t,1) * Dnum(a,0) /
         (Dnum(t,1) * Dnum(t,1) + Dnum(b,0));
```

- Deriválással együtt szebben, kihasználva a default paraméterezést a konstruktorban:

```
Dnum t(value, 1);
Dnum F = t * a / (t * t + b);
```

Elemi függvény

```
float F, x, y, a;
…
F = 3 * t + a * sin(t);
```

```
struct Dnum {
    float f, d; // function and derivative values
    Dnum(float f0, float d0 = 0) { f = f0, d = d0; }
    …
};

Dnum Sin(float t) { return Dnum(sinf(t), cosf(t)); }
Dnum Cos(float t) { return Dnum(cosf(t), -sinf(t)); }
…
```

Összetett függvény

$$\frac{\mathrm{d}f(g(t))}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}t}$$

```
float F, x, y, a;
…
F = 3 * t + a * sin(t) + cos(y * log(t) + 2);
```

```
struct Dnum {
   float f, d; // function and derivative values
   Dnum(float f0, float d0 = 0) { f = f0, d = d0; }
   …
};

Dnum Sin(Dnum g) { return Dnum(sinf(g.f), cosf(g.f) * g.d); }
Dnum Cos(Dnum g) { return Dnum(cosf(g.f), -sinf(g.f) * g.d); }
Dnum Tan(Dnum g) { return Sin(g)/Cos(g); }
Dnum Log(Dnum g) { return Dnum(logf(g.f), 1/g.f * g.d); }
Dnum Exp(Dnum g) { return Dnum(expf(g.f), expf(g.f) * g.d); }
Dnum Pow(Dnum g, float n) {
   return Dnum(powf(g.f, n), n * powf(g.f, n - 1) * g.d);
}
```

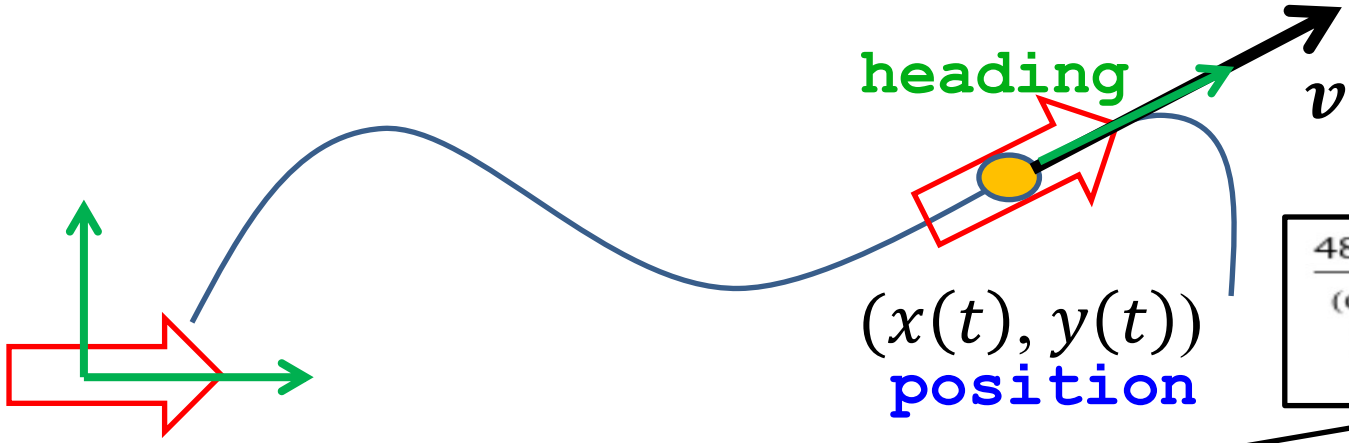
Többváltozós függvények

```
template<class T> struct Dnum {
   float f; // function value
   T d;  // derivatives
   Dnum(float f0, T d0 = T(0)) { f = f0, d = d0; }
   Dnum operator+(Dnum r) { return Dnum(f+r.f, d+r.d); }
   Dnum operator*(Dnum r) { return Dnum(f*r.f, f*r.d + d*r.f); }
   Dnum operator/(Dnum r) { return Dnum(f/r.f, (d*r.f-f*r.d)/r.f/r.f); }
};

template<class T> Dnum<T> Exp(Dnum<T> g) {
   return Dnum<T>(expf(g.f), expf(g.f) * g.d);
}
```

F(x,y,z) gradiense:

```
float x, y, z;
Dnum<vec3> X(x,vec3(1,0,0)), Y(y,vec3(0,1,0)), Z(z,vec3(0,0,1));
Dnum<vec3> F = X*X/a + Y*Y/b + Z*Z/c - 1;
vec3 grad = F.d;
```

Mire jó? Pálya animáció



$$\frac{48\sin^2(x)\sin^2(\cos(x)+2)}{(\cos(2(\cos(x)+2))-1)^2} + \frac{16\sin^3(x)\sin^2(\cos(x)+2)}{(\cos(2(\cos(x)+2))-1)^2} - \frac{8\sin(x)\cos(x)\sin(2(\cos(x)+2))}{\cos(2(\cos(x)+2))-1} - \frac{12\cos(x)\sin(2(\cos(x)+2))}{\cos(2(\cos(x)+2))-1}$$

Pálya: $x(t) = \frac{\sin(t)(\sin(t)+3)4}{\tan(\cos(t)+2)}$, $y(t) = \frac{(\cos(\sin(t))8+1)12+2}{(\sin(t)\,\sin(t))^3+2}$

```
void Animate(float tt) {
  Dnum t(tt, 1);
  Dnum x = Sin(t)*(Sin(t)+3)*4 / (Tan(Cos(t))+2);
  Dnum y = (Cos(Sin(t)*8+1)*12+2)/(Pow(Sin(t)*Sin(t),3)+2);
  vec2 position(x.f, y.f), velocity(x.d, y.d);
  vec2 heading = normalize(velocity);
  Draw(position, heading);
}
```

“For geometry, you know, is the gate of science, and the gate is so low and small that we can only enter it as a little child.”

William Kingdon Clifford

Geometriai (Clifford) algebra

Szirmay-Kalos László

Geometriához használt algebrák állatkertje

Alakzatok

2D

- Pont: $[x, y]$, $[x, y, w]$, komplex szám
- Egyenes: $ax + by + c = 0$
- Metrika: skaláris szorzás, külső szorzás

Mozgatások (motorok)

- Eltolás: vektor, komplex szám
- **Forgatás, eltolás**, skálázás, tükröz, nyírás, vetítés, …: 3x3-as mátrix
- 2D forgatva nyújtás: komplex sz.
- Deriválás: duális szám

3D

- Pont: $[x, y, z]$, $[x, y, z, w]$
- Sík: $ax + by + cz + d = 0$
- Metrika: skaláris szorzás, külső szorzás, vektoriális szorzás

- Eltolás: vektor
- Forgatás, skálázás, tükrözés, nyírás, vetítés, perspektíva, …: 4x4-es mátrix
- Origón átmentő tengely körül 3D forgatás: kvaternió

Geometriai számok és szorzat

- Vektor: $\boldsymbol{v} = x\boldsymbol{e}_1 + y\boldsymbol{e}_2$ Ortogonális (skaláris szorzás): $\boldsymbol{e}_1 \cdot \boldsymbol{e}_2 = 0$
- Geometriai szorzás: asszociatív, disztributív, invertálható
- Kapcsolat a valós számokkal: $\boldsymbol{vv}$ legyen valós

  $\boldsymbol{vv} = (x\boldsymbol{e}_1 + y\boldsymbol{e}_2)(x\boldsymbol{e}_1 + y\boldsymbol{e}_2) = x^2\boldsymbol{e}_1\boldsymbol{e}_1 + y^2\boldsymbol{e}_2\boldsymbol{e}_2 + xy(\boldsymbol{e}_1\boldsymbol{e}_2 + \boldsymbol{e}_2\boldsymbol{e}_1)$

  - Bázis (geometriai számok): $\boldsymbol{e}_k\boldsymbol{e}_k = 1, \quad 0, \quad -1$?
  - Antiszimmetrikus: $\boldsymbol{e}_{12} = \boldsymbol{e}_1\boldsymbol{e}_2 = -\boldsymbol{e}_2\boldsymbol{e}_1 = -\boldsymbol{e}_{21}$
- Geometriai (Clifford) szorzat:

  $\boldsymbol{v}_1\boldsymbol{v}_2 = (x_1\boldsymbol{e}_1 + y_1\boldsymbol{e}_2)(x_2\boldsymbol{e}_1 + y_2\boldsymbol{e}_2) = x_1x_2 + y_1y_2 + (x_1y_2 - x_2y_1)\boldsymbol{e}_1\boldsymbol{e}_2$

$\boldsymbol{v}_1\boldsymbol{v}_2 = \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 + \boldsymbol{v}_1 \wedge \boldsymbol{v}_2$ Inverz: $\boldsymbol{v}^{-1} = \frac{\boldsymbol{v}}{\boldsymbol{vv}}$

2D geometriai algebra

0 dim 1 dim 2 dim

- Multivektor: $\mathbf{V} = s + \overbrace{x\boldsymbol{e}_1 + y\boldsymbol{e}_2} + B\boldsymbol{e}_{12}$
- Összeadás, skálázás a szokásos módon
- Szorzás (asszociatív, disztributív, nemkommutatív):
  - skalár és bármi: a szokásos
  - Pszeudó-skalár és pszeudó-skalár:

$$\boldsymbol{e}_{12}\boldsymbol{e}_{12} = \boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_1\boldsymbol{e}_2 = -\boldsymbol{e}_1\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_2 = -1$$

(−1 a felcserélésnél; 1 és 1 a négyzetre emelésnél)

Ha $x = y = 0$, akkor a komplex számokat kapjuk: $\boldsymbol{e}_{12} = \boldsymbol{I}$

  - Pszeudó-skalárral jobbról/balról: $\pm 90°$-os forgatás

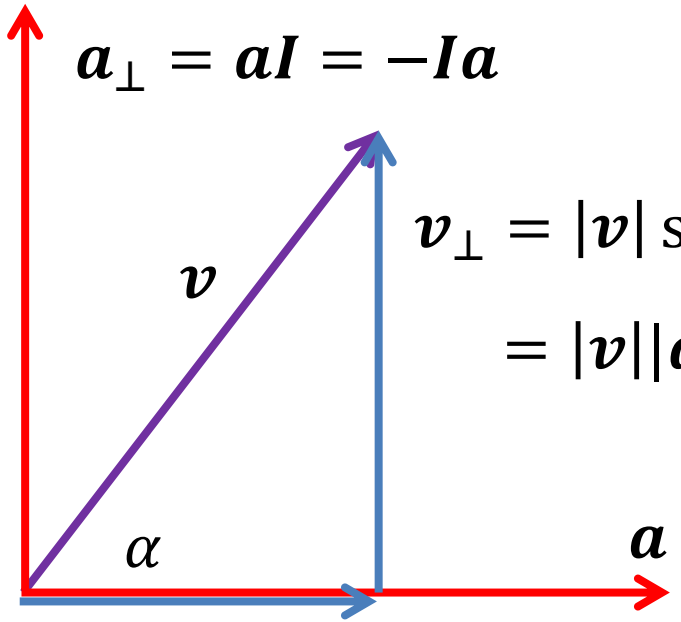
$$(x\boldsymbol{e}_1 + y\boldsymbol{e}_2)\boldsymbol{e}_{12} = x\boldsymbol{e}_1\boldsymbol{e}_1\boldsymbol{e}_2 + y\boldsymbol{e}_2\boldsymbol{e}_1\boldsymbol{e}_2 = -y\boldsymbol{e}_1 + x\boldsymbol{e}_2$$

Szorzótábla

	1	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{I}$
1	1	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{I}$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1$	1	$\boldsymbol{I}$	$\boldsymbol{e}_2$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2$	$-\boldsymbol{I}$	1	$-\boldsymbol{e}_1$
$\boldsymbol{I}$	$\boldsymbol{I}$	$-\boldsymbol{e}_2$	$\boldsymbol{e}_1$	-1

$$\mathbf{V} = s + x\boldsymbol{e}_1 + y\boldsymbol{e}_2 + B\boldsymbol{I}$$

Projection és Rejection



$$\boldsymbol{v}_\perp = |\boldsymbol{v}|\sin(\alpha)\boldsymbol{a}_\perp/|\boldsymbol{a}| = |\boldsymbol{v}||\boldsymbol{a}|\sin(\alpha)\boldsymbol{a}_\perp/|\boldsymbol{a}|^2$$

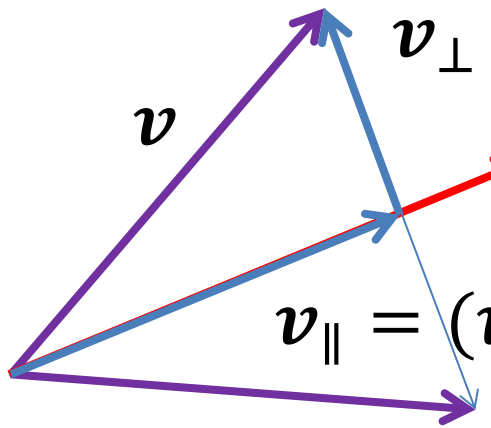
$$= |\boldsymbol{v}||\boldsymbol{a}|\sin(\alpha)(-\boldsymbol{I})\boldsymbol{a}/|\boldsymbol{a}|^2 = \boxed{(\boldsymbol{v}\wedge\boldsymbol{a})\boldsymbol{a}^{-1}}$$

$$\boldsymbol{v}_\parallel = |\boldsymbol{v}|\cos(\alpha)\,\boldsymbol{a}/|\boldsymbol{a}| = |\boldsymbol{v}||\boldsymbol{a}|\cos(\alpha)\,\boldsymbol{a}/|\boldsymbol{a}|^2 = \boxed{(\boldsymbol{v}\cdot\boldsymbol{a})\boldsymbol{a}^{-1}}$$

Vektorokra:

$$\boldsymbol{va} = \boldsymbol{v}\cdot\boldsymbol{a} + \boldsymbol{v}\wedge\boldsymbol{a} \quad \Rightarrow \quad \begin{aligned} \boldsymbol{v}\cdot\boldsymbol{a} &= (\boldsymbol{va}+\boldsymbol{av})/2 \\ \boldsymbol{v}\wedge\boldsymbol{a} &= (\boldsymbol{va}-\boldsymbol{av})/2 \end{aligned}$$

Tükrözés és szendvics

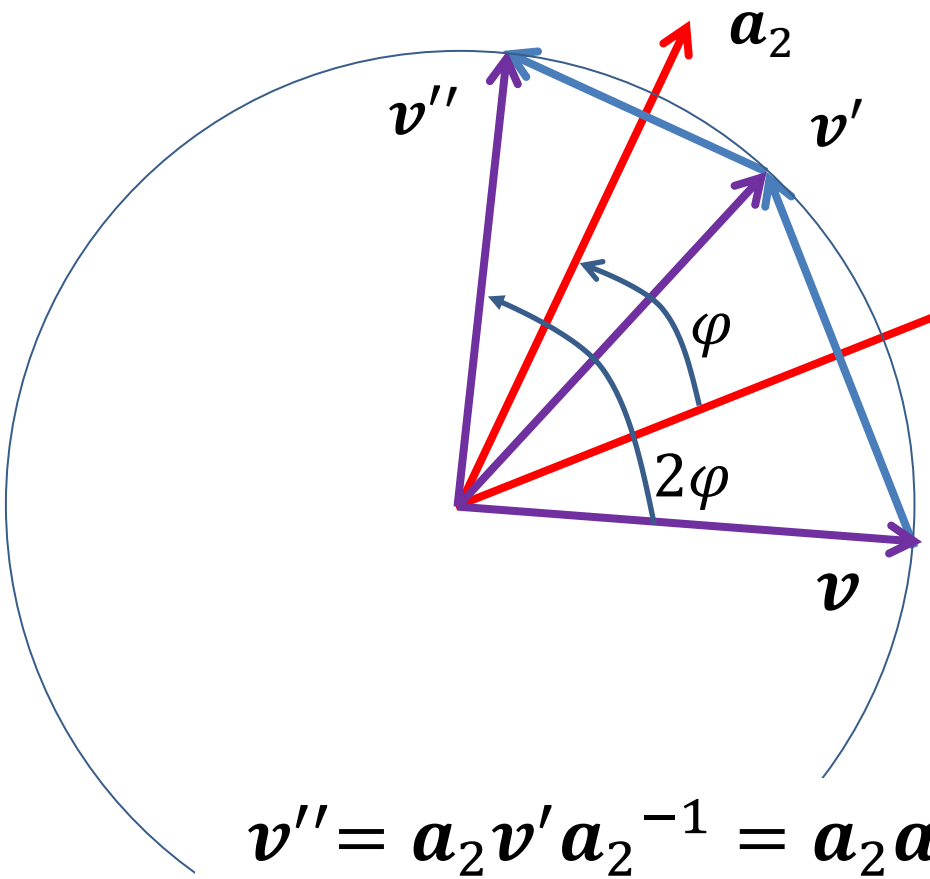


$$v' = v_\parallel - v_\perp = (v \cdot a)a^{-1} - (v \wedge a)a^{-1}$$

$$= (v \cdot a - v \wedge a)a^{-1} = (a \cdot v + a \wedge v)a^{-1}$$

$$\boxed{v' = ava^{-1}}$$

Forgatás = 2 tükrözés



$$\boldsymbol{R} = \boldsymbol{a}_1\boldsymbol{a}_2 = |\boldsymbol{a}_1||\boldsymbol{a}_2|(\cos(\varphi) + \sin(\varphi)\boldsymbol{I}) = |\boldsymbol{a}_1||\boldsymbol{a}_2|\exp(\varphi\boldsymbol{I})$$

$$\boldsymbol{v}'' = \boldsymbol{a}_2\boldsymbol{v}'\boldsymbol{a}_2^{-1} = \boldsymbol{a}_2\boldsymbol{a}_1\boldsymbol{v}'\boldsymbol{a}_1^{-1}\boldsymbol{a}_2^{-1} = (\boldsymbol{a}_2\boldsymbol{a}_1)\boldsymbol{v}(\boldsymbol{a}_2\boldsymbol{a}_1)^{-1}$$

$$\boldsymbol{v}'' = \boldsymbol{R}\boldsymbol{v}\boldsymbol{R}^{-1} = (\cos(\varphi) - \sin(\varphi)\boldsymbol{I})\boldsymbol{v}(\cos(\varphi) + \sin(\varphi)\boldsymbol{I})$$

Exponentiation

- A transzformációk szorzással működnek (multiplikatív csoport): $(\boldsymbol{R}_2\boldsymbol{R}_1)\boldsymbol{v}(\boldsymbol{R}_2\boldsymbol{R}_1)^{-1}$
- Rotor: $\boldsymbol{R}(\varphi_2 + \varphi_1) = \boldsymbol{R}(\varphi_2)\boldsymbol{R}(\varphi_2)$
- Hatványfüggvény: $\boldsymbol{R}(\varphi) = a^{\varphi} = \exp(b\varphi)$
- Ha $\varphi = \pi$, akkor $(-\boldsymbol{I})\boldsymbol{v}\boldsymbol{I} \Rightarrow \boldsymbol{R}(\varphi) = \exp(\boldsymbol{I}\varphi)$