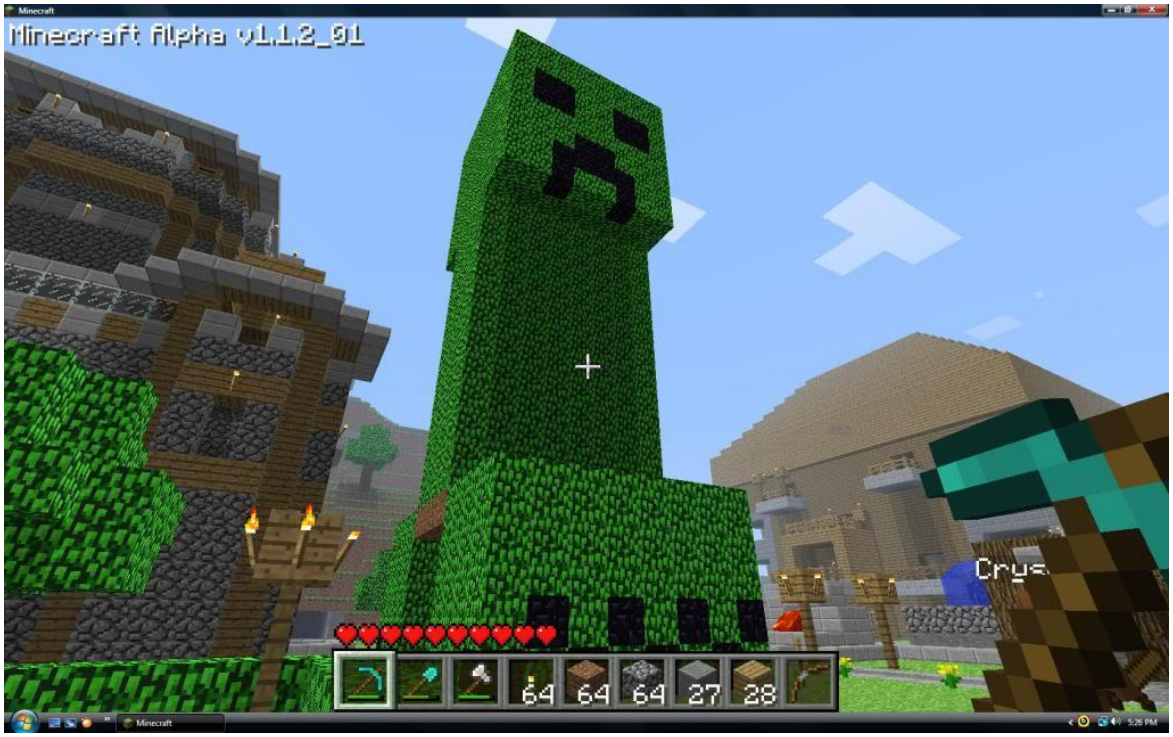
Curvas e superfícies

Computação Gráfica

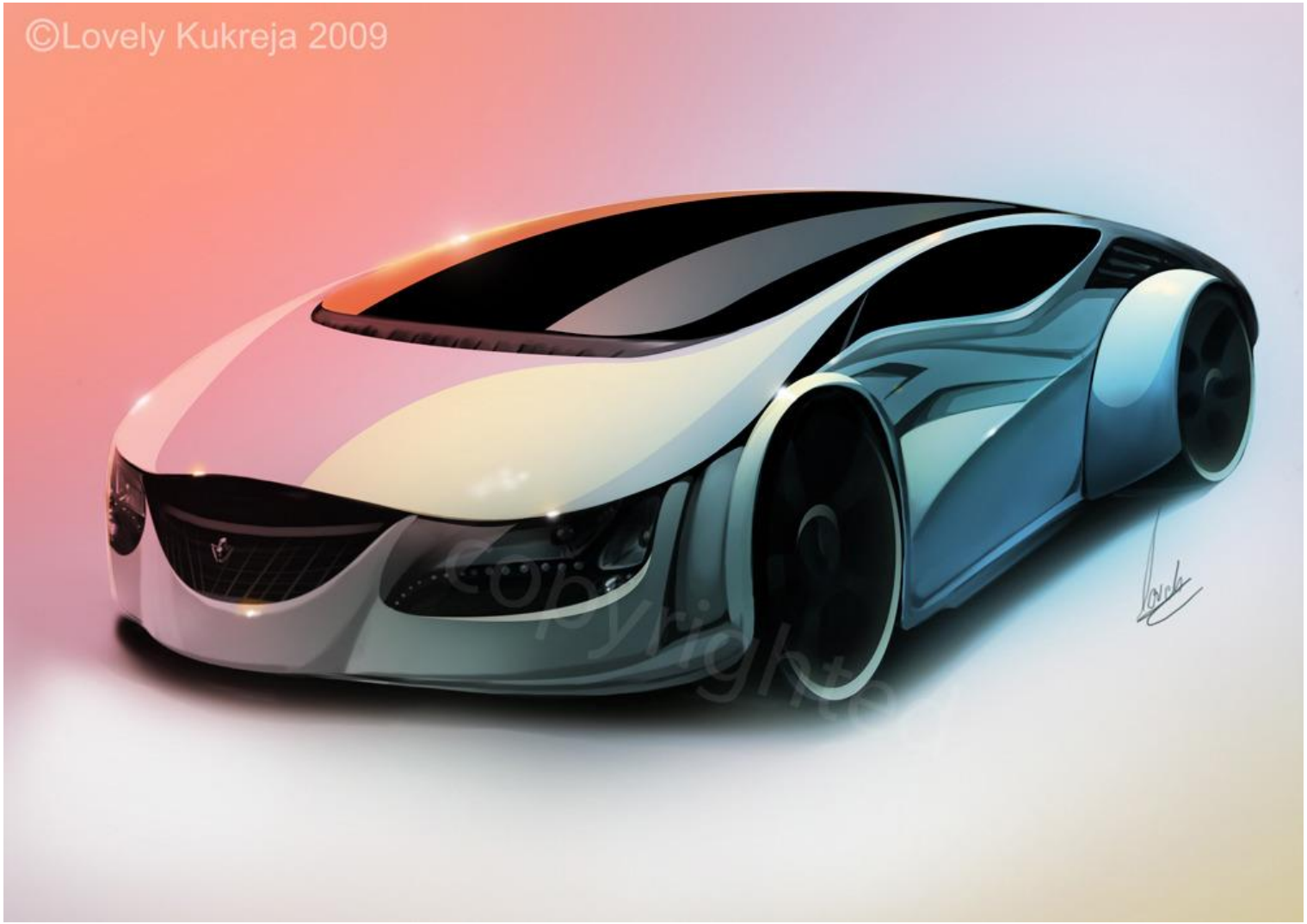
Curvas - realismo



Curvas - realismo

Curvas - realismo

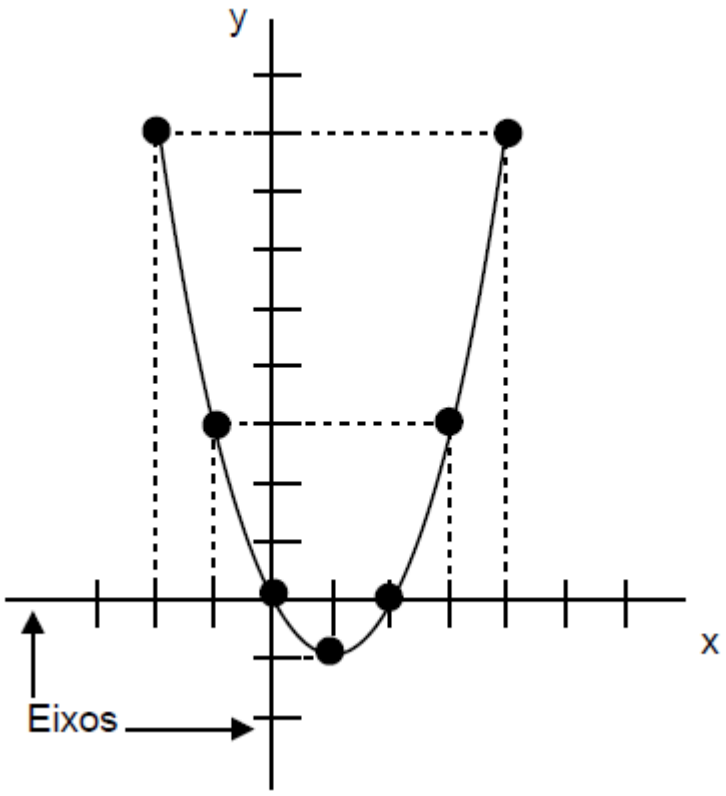
Curvas - realismo



Tópicos

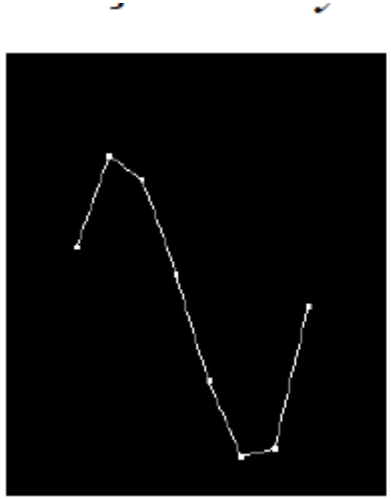
- Representações
  - Pontos, Analíticas, paramétricas, não-paramétricas
- Curvas
  - Hermite
  - Bézier
  - Splines
  - Curvas racionais
- Superfícies
  - Revolução, Deslocamento
  - Hermite, Bézier, B-Spline
  - NURBS e NURMS

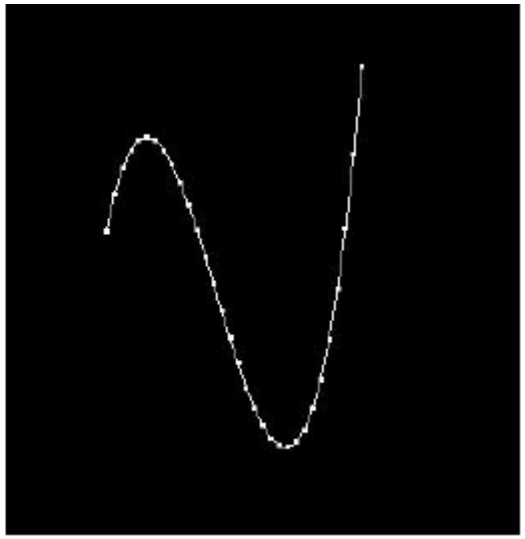
Representação por pontos



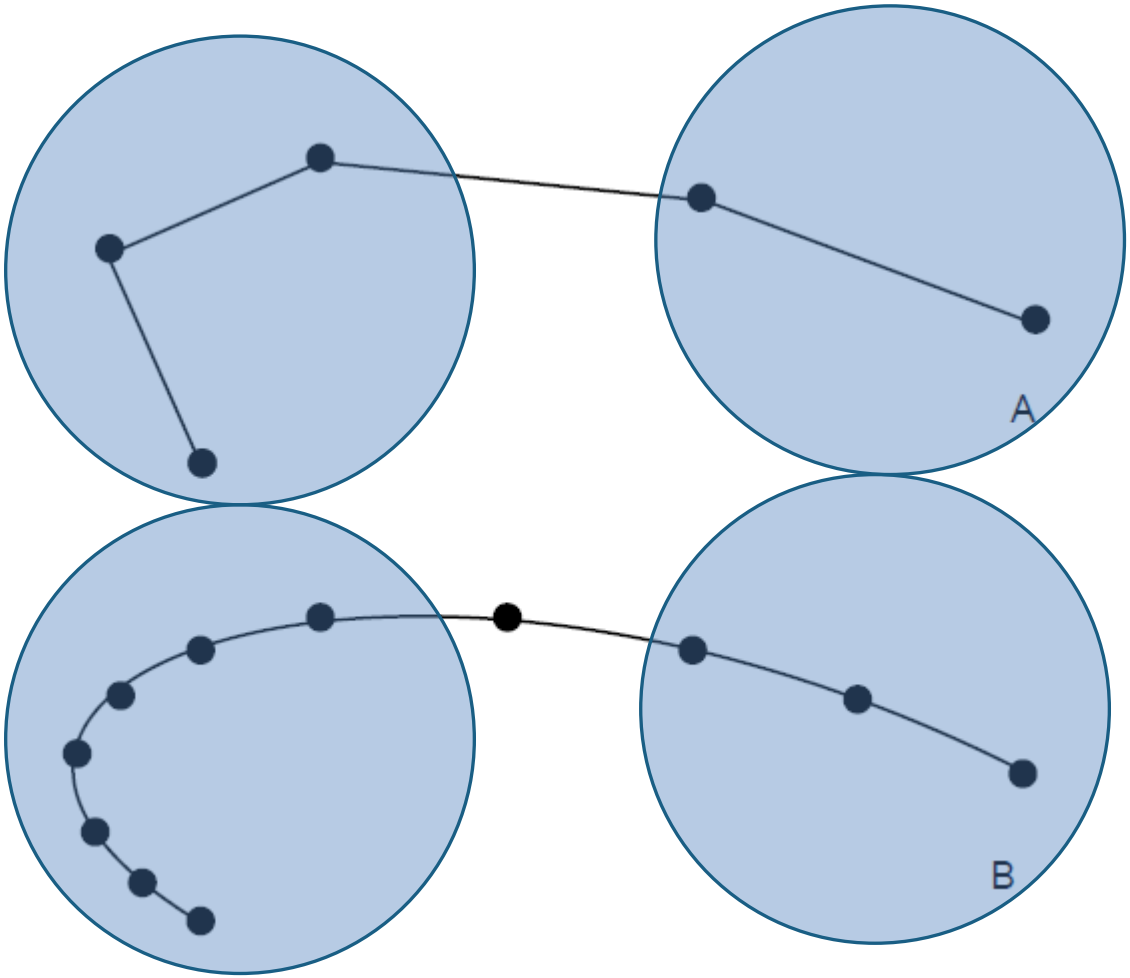
Representação por pontos

*Mais ou menos pontos dependendo da precisão desejada





Representação por pontos



Representação analítica

- Uso de fórmulas matemáticas
- Mais precisa
- Compacta
- Facilita o cálculo de novos pontos e de propriedades como área e inclinação
- Não-paramétricas
  - Explícitas
  - implícitas
- Paramétricas

Não-paramétricas

*X em função de Y e vice-versa

$$y = f_x(x) \text{ ou } x = f_y(y)$$

*Semicírculo

$$y = \sqrt{10^2 - x^2} \text{ ou } x = \sqrt{10^2 - y^2}$$

*Uma reta

$$y = 2x - 1 \text{ ou } x = \tfrac{1}{2}(y+1)$$

Não-paramétrica

*Explícita

$$y = ax^2 + bx + c$$

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots a_2 x^2 + a_1 x^1 + a_0$$

*Fáceis de combinar, derivar, integrar e calcular o valor em um ponto.

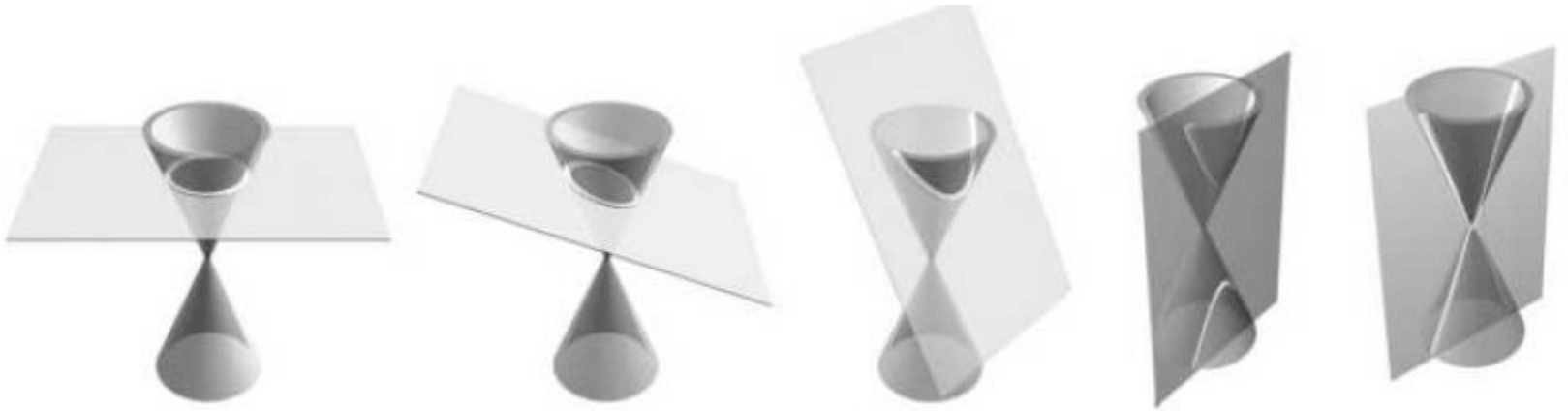
*Implícita

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

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

Não-paramétrica implícita

$$\begin{cases} F_1(x, y, z) = 0 \\ F_2(x, y, z) = 0 \end{cases}$$



Representação paramétrica

- y = y(t)
- x = x(t)
- P(t) = (x(t), y(t))
- círculo

$$x = 10\ cos\ \theta = f_x(\theta)$$
$$y = 10\ sen\ \theta = f_y(\theta)$$

- reta

$$x = t+1 = f_x(t)$$
$$y = 2t+1 = f_y(t)$$

Representação paramétrica

- P(t) = (x(t), y(t))
- Tangente:

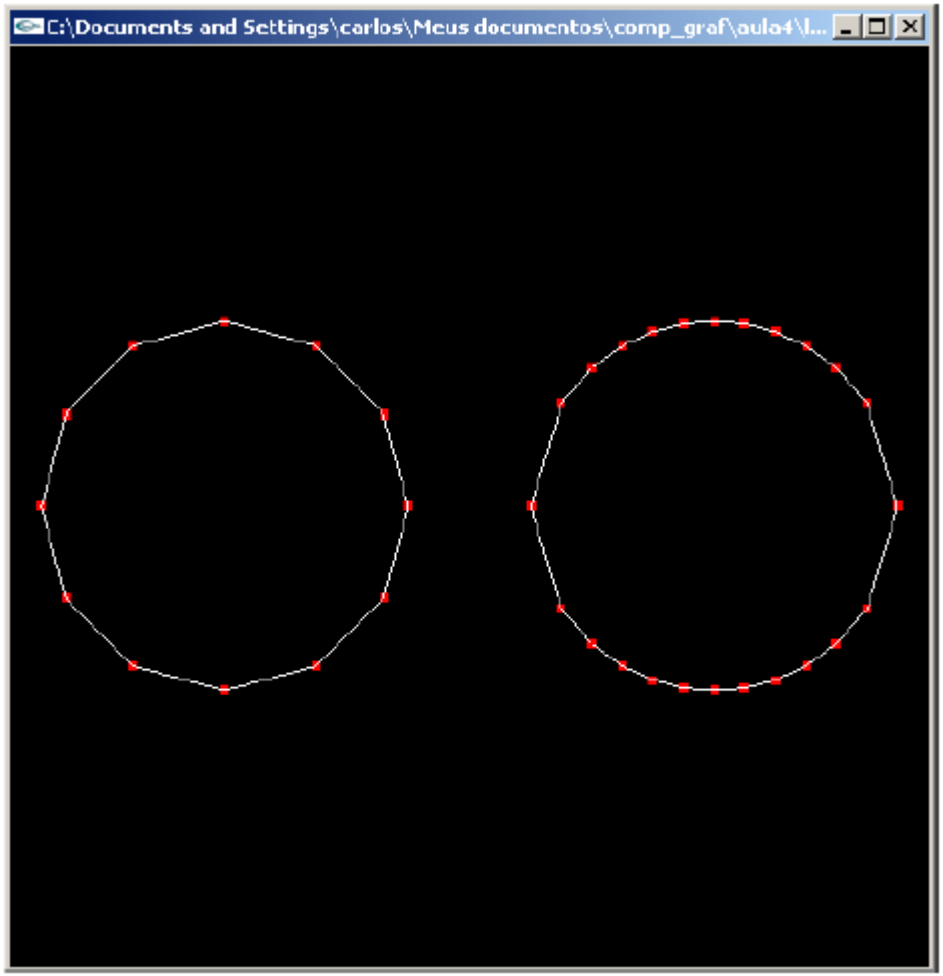
$$P'(t) = (x'(t), y'(t))$$

- Inclinação:

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{y'(t)}{x'(t)}$$

Cônica	**Forma Paramétrica**	**Forma Implícita**
Elipse	$x = a\ cos\ \theta$ $y = b\ sen\ \theta$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$
Parábola	$x = at^2,\ y = 2at$	$y^2 - 4ax = 0$
Hipérbole	$x = a\ cosh\ \theta$ $y = b\ senh\ \theta$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$

Paramétricas e não para métricas



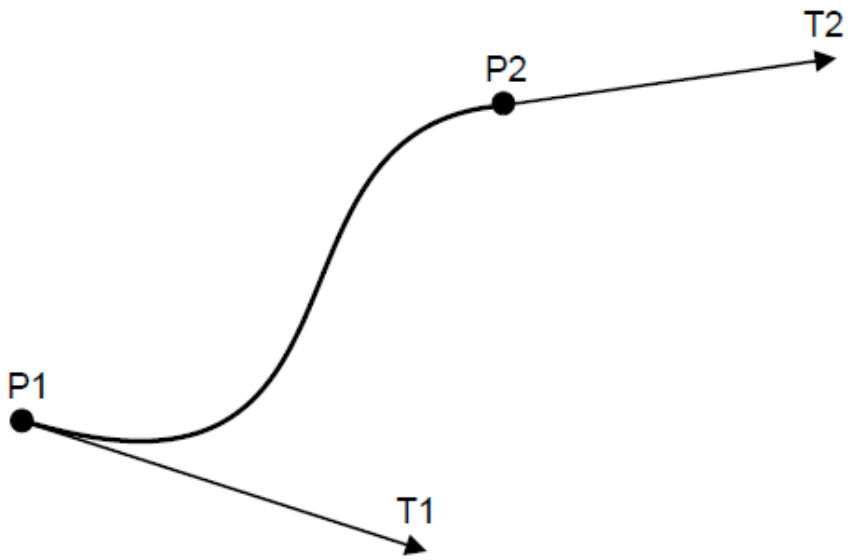
Hermite

*Charles Hermite (1822 - 1901)
*Polinômios de terceira ordem

$$x(t) = P_x = a_x t^3 + b_x t^2 + c_x t + d_x$$
$$y(t) = P_y = a_y t^3 + b_y t^2 + c_y t + d_y$$
$$z(t) = P_z = a_z t^3 + b_z t^2 + c_z t + d_z$$

Hermite

- Dois pontos (início e fim) P1 e P2
- Dois vetores tangentes (Saída e chegada) T1 e T2
  - Direção, sentido e magnitude



Hermite

$$x(t) = P_x = a_x t^3 + b_x t^2 + c_x t + d_x$$
$$y(t) = P_y = a_y t^3 + b_y t^2 + c_y t + d_y$$
$$z(t) = P_z = a_z t^3 + b_z t^2 + c_z t + d_z$$

$$x(t) = [t^3\ t^2\ t^1\ 1] \begin{vmatrix} a_x \\ b_x \\ c_x \\ d_x \end{vmatrix} = T\ C_x$$

$$y(t) = [t^3\ t^2\ t^1\ 1] \begin{vmatrix} a_y \\ b_y \\ c_y \\ d_y \end{vmatrix} = T\ C_y$$

$$z(t) = [t^3\ t^2\ t^1\ 1] \begin{vmatrix} a_z \\ b_z \\ c_z \\ d_z \end{vmatrix} = T\ C_z$$

Hermite

*Para P1, t=0

$$P1_x = [\,0\ \ 0\ \ 0\ \ 1\,]\ C_x$$
$$P1_y = [\,0\ \ 0\ \ 0\ \ 1\,]\ C_y$$
$$P1_z = [\,0\ \ 0\ \ 0\ \ 1\,]\ C_z$$

*$(P1_x, P1_y, P1_z) = (d_x, d_y, d_z)$

Hermite

*Para P2, t=1

$$x(1) = [1 \;\; 1 \;\; 1 \;\; 1]\, C_x$$
$$y(1) = [1 \;\; 1 \;\; 1 \;\; 1]\, C_y$$
$$z(1) = [1 \;\; 1 \;\; 1 \;\; 1]\, Cz$$

$$*(P2_x, P2_y, P2_z) = (a_x + b_x + c_x + d_x,$$
$$* \qquad a_y + b_y + c_y + d_y,$$
$$* \qquad a_z + b_z + c_z + d_z)$$

Hermite

*Para T1,

$$x'(t) = P'_x = 3a_x t^2 + 2b_x t + c_x$$
$$y'(t) = P'_y = 3a_y t^2 + 2b_y t + c_y$$
$$z'(t) = P'_z = 3a_z t^2 + 2b_z t + c_z$$

$$x'(t) = P'_x = [3t^2 \;\; 2t \;\; 1 \;\; 0]C_x$$
$$y'(t) = P'_y = [3t^2 \;\; 2t \;\; 1 \;\; 0]C_y$$
$$z'(t) = P'_z = [3t^2 \;\; 2t \;\; 1 \;\; 0]C_z$$

$$x'(0) = T1_x = [0 \;\; 0 \;\; 1 \;\; 0]C_x = c_x$$
$$y'(0) = T1_y = [0 \;\; 0 \;\; 1 \;\; 0]C_y = c_y$$
$$z'(0) = T1_z = [0 \;\; 0 \;\; 1 \;\; 0]C_z = c_z$$

*$(T1_x, T1_y, T1_z) = (c_x, c_y, c_z)$

Hermite

*Para T2,

$$x'(1) = T2_x = [3 \;\; 2 \;\; 1 \;\; 0]C_x$$
$$y'(1) = T2_y = [3 \;\; 2 \;\; 1 \;\; 0]C_y$$
$$z'(1) = T2_z = [3 \;\; 2 \;\; 1 \;\; 0]C_z$$

*$(T2_x, T2_y, T2_z) = (3a_x + 2b_x + c_x,$
* $3a_y + 2b_y + c_y,$
* $3a_z + 2b_z + c_z)$

Hermite

$$P1_x = [\,0\ \ 0\ \ 0\ \ 1\,]\,C_x$$
$$P1_y = [\,0\ \ 0\ \ 0\ \ 1\,]\,C_y$$
$$P1_z = [\,0\ \ 0\ \ 0\ \ 1\,]\,C_z$$

$$x(1) = [1\ \ 1\ \ 1\ \ 1]\,C_x$$
$$y(1) = [1\ \ 1\ \ 1\ \ 1]\,C_y$$
$$z(1) = [1\ \ 1\ \ 1\ \ 1]\,Cz$$

$$x'(0) = T1_x = [0\ \ 0\ \ 1\ \ 0]C_x = c_x$$
$$y'(0) = T1_y = [0\ \ 0\ \ 1\ \ 0]C_y = c_y$$
$$z'(0) = T1_z = [0\ \ 0\ \ 1\ \ 0]C_z = c_z$$

$$x'(1) = T2_x = [3\ \ 2\ \ 1\ \ 0]C_x$$
$$y'(1) = T2_y = [3\ \ 2\ \ 1\ \ 0]C_y$$
$$z'(1) = T2_z = [3\ \ 2\ \ 1\ \ 0]C_z$$

$$\begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_x = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} C_x = H^{-1}C_x$$

Hermite

$$\begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_x = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} C_x = H^{-1} C_x$$

$$C_X = HH^{-1}C_X = H \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_x$$

$$H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Hermite

$$x(t) = TC_x = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_x = THG_x$$

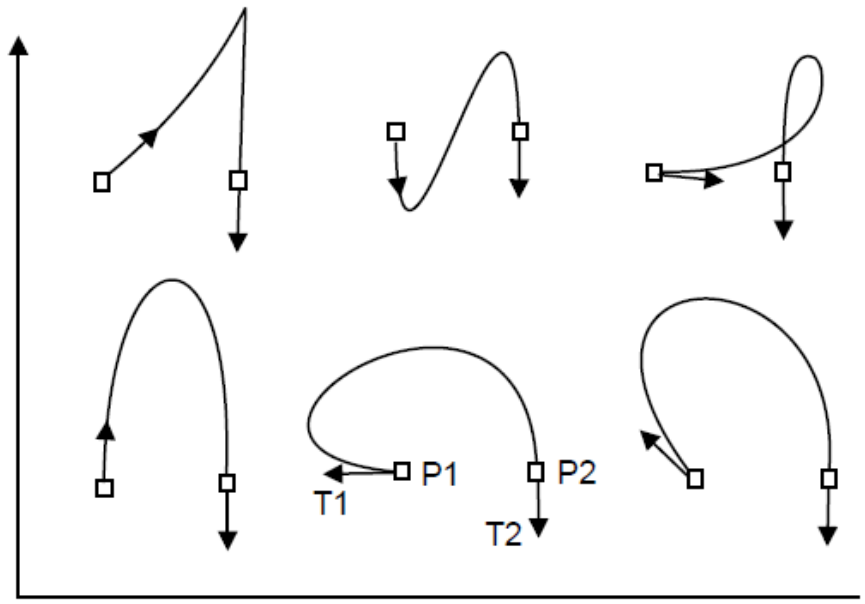
$$y(t) = TC_y = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_y = THG_y$$

$$z(t) = TC_z = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_z = THG_z$$

$$G_h = \begin{bmatrix} P1_x & P1_y & P1_z \\ P2_x & P2_y & P2_z \\ T1_x & T1_y & T1_z \\ T2_x & T2_y & T2_z \end{bmatrix}$$

Hermite

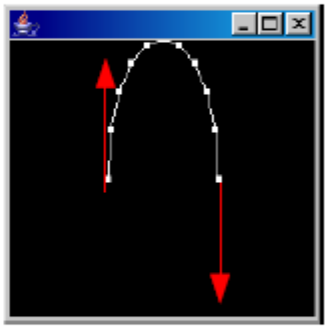
$$P(t) = \left((2t^3 - 3t^2 + 1), (-2t^3 + 3t^2), (t^3 - 2t^2 + t), (t^3 - t^2) \right) \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}$$



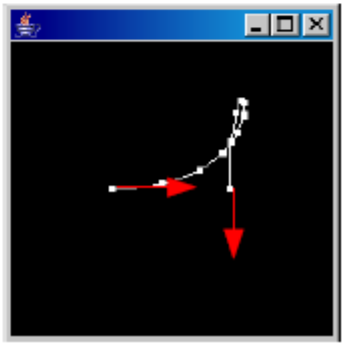
Hermite

P1=(-1,0), P1=(1,0), T1=(0,2), T2=(0,-2)

P1=(-1,0), P1=(1,0), T1=(0,10), T2=(0,-10)

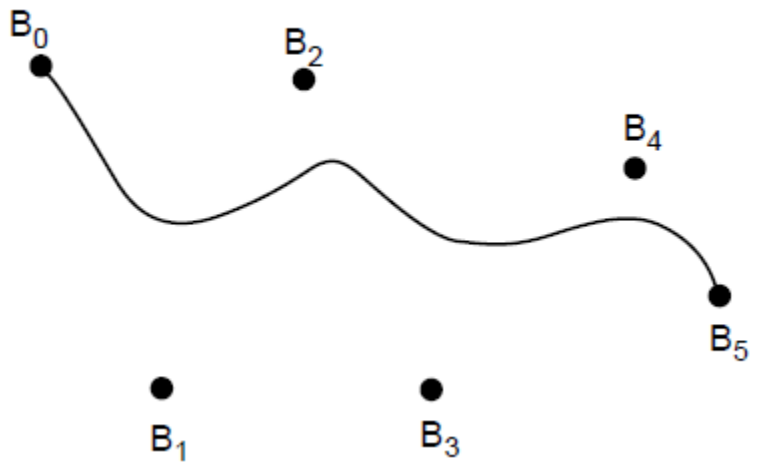


P1=(-1,0), P1=(1,0), T1=(10,0), T2=(0,-10)



Bézier

*Pierre Bézier
    *Renault
*As tangentes das curvas são determinadas por pontos e não vetores
*A curva fica completamente dentro do polígono convexo determinado pelos pontos de controle



Bézier

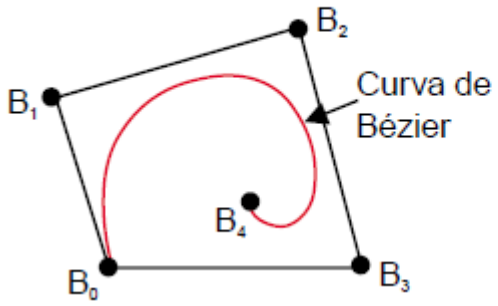
- Todos os ***B*** pontos influenciam toda a curva

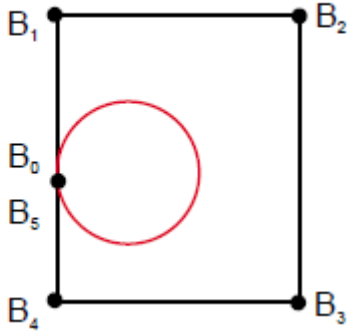
$$P(t) = \sum_{i=0}^{n} B_i \, J_{n,i}(t) \qquad 0 \le t \le 1$$

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

- Propriedade normalizante

$$\sum_{i=0}^{n} J_{n,i}(t) = 1, \qquad 0 \le t \le 1$$





Bézier

*Usando três pontos de controle (n=2)

$$P(t) = B_0 J_{2,0}(t) + B_1 J_{2,1}(t) + B_2 J_{2,2}(t)$$

$$J_{2,0} = \frac{2!}{0!2!} t^0 (1-t)^2 = (1-t)^2 = 1 - 2t + t^2$$

$$J_{2,1} = \frac{2!}{1!1!} t^1 (1-t)^1 = 2t(1-t) = 2t - 2t^2$$

$$J_{2,2} = \frac{2!}{2!0!} t^2 (1-t)^0 = t^2$$

Bézier

*Usando três pontos de controle (n=2)

$$P(t) = B_0 J_{2,0}(t) + B_1 J_{2,1}(t) + B_2 J_{2,2}(t)$$

$$P(t) = (1-t)^2 B_0 + 2t(1-t) B_1 + t^2 B_2$$

$$P(t) = \begin{bmatrix} (1-t)^2 & 2t(1-t) & t^2 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix}$$

Bézier

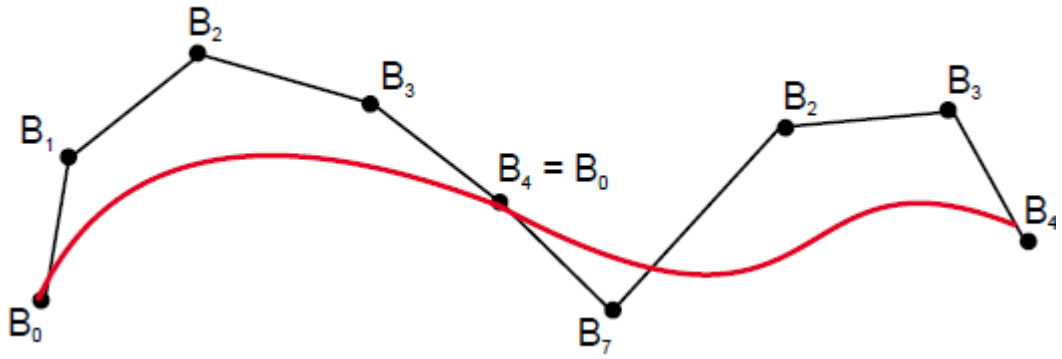
*Usando quatro pontos de controle (n=3)

$$P(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t) B_2 + t^3 B_3$$

$$P(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

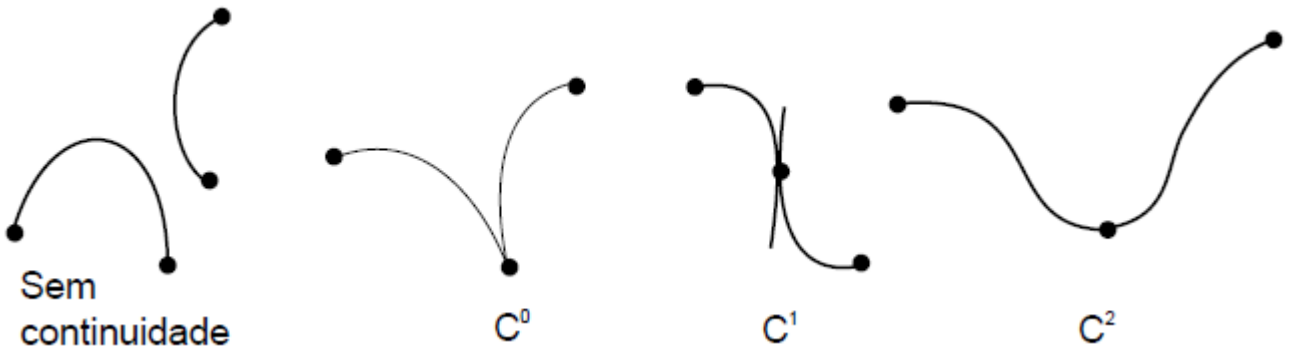
Bézier

- Poucos pontos – difícil ajuste fino
- Muitos pontos – funções muito complexas
- Solução:
  - Combinar curvas simples

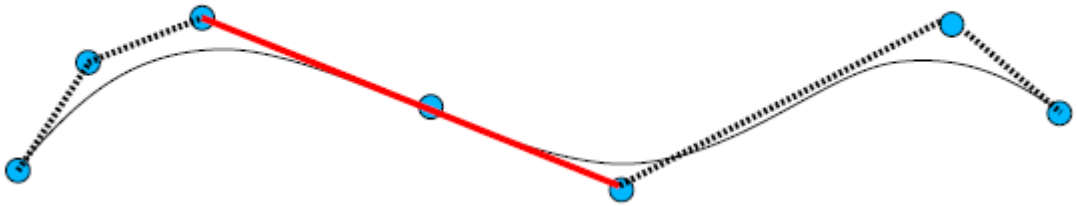


Continuidade de curvas

*C0: Curvas se encontram
*C1: Curvas se encontram e a primeira derivada é igual nesse ponto (mesma inclinação)
*C2: Curvas se encontram e a segunda derivada é igual nesse ponto (mesma curvatura)



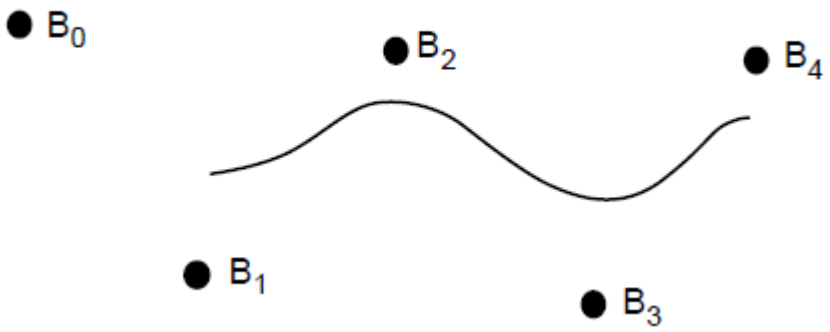
Continuidade de curvas



B-Spline

*Curva (geralmente) não passa pelos pontos de controle

    *A não ser que seja uma reta



B-Spline

- O parâmetro k altera a ordem do polinômio
  - k gera funções de ordem (k-1) e continuidade (k-2)

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t)$$

$$N_{i,1}(t) = \begin{cases} 1 & para\ t_i \leq t < t_{i+1} \\ 0 & nos\ demais\ intervalos \end{cases}$$

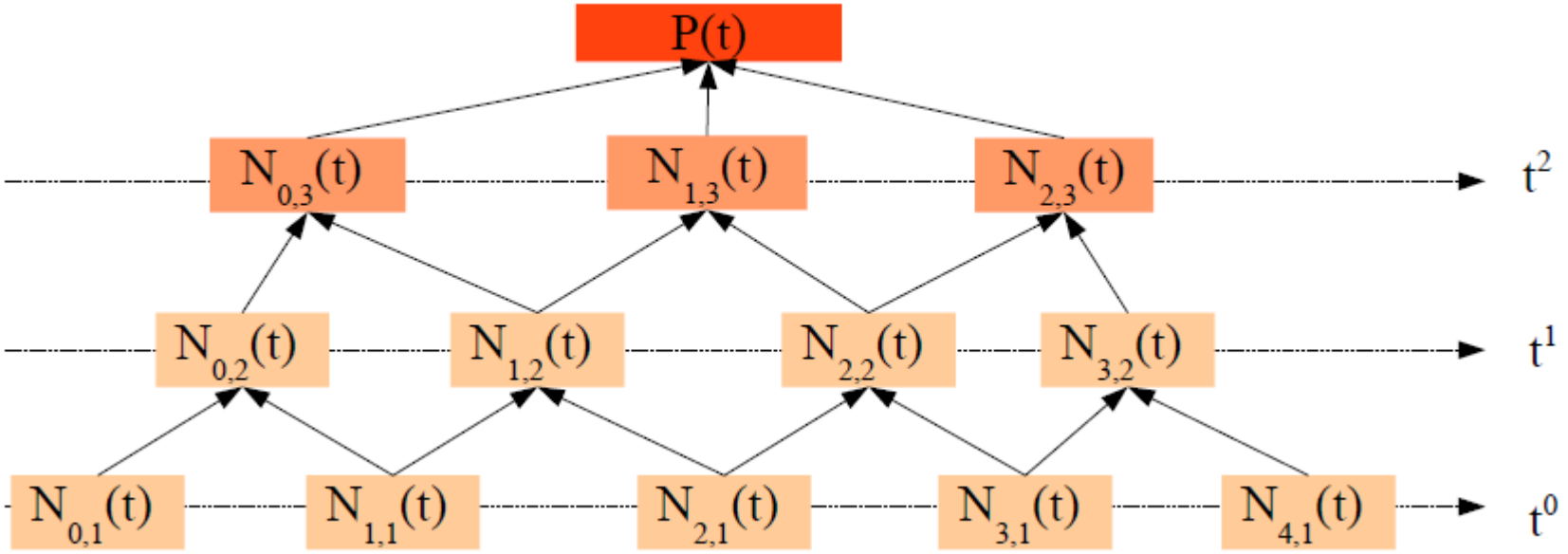
$$N_{i,k}(t) = \left(\frac{t - t_i}{t_{i+k-1} - t_i}\right) N_{i,k-1}(t) + \left(\frac{t_{i+k} - t}{t_{i+k} - t_{i+1}}\right) N_{i+1,k-1}(t)$$

B-Spline

- 0/0=0
- Os valores de ***t***, chamados de nós obedecem às seguintes condições:
  - ***t*** deve estar em ordem ascendente
  - Um mesmo valor de ***t*** não pode aparecer mais que ***k*** vezes.
- Uma B-Spline tem
  - (***n***+1) pontos de controle ***B***
  - Grau (***k***-1)
  - E ***m*** nós onde ***m=n+k***

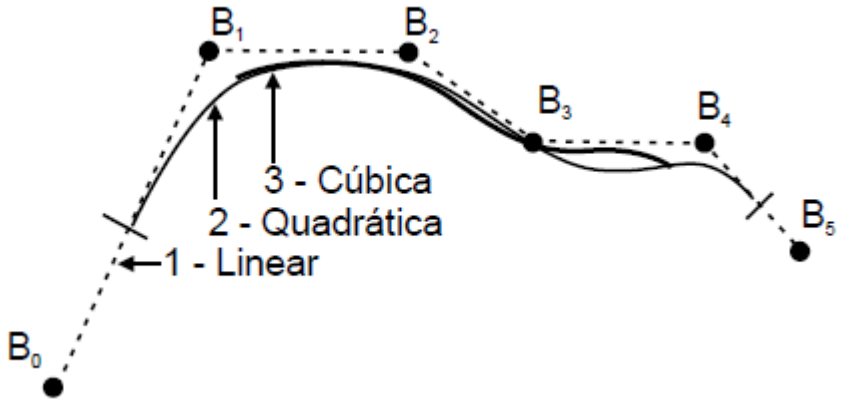
B-Splines

*Exemplo de interpolação de segunda ordem (k-1=2) com três pontos (n+1=3)



B-Spline uniforme e periódica

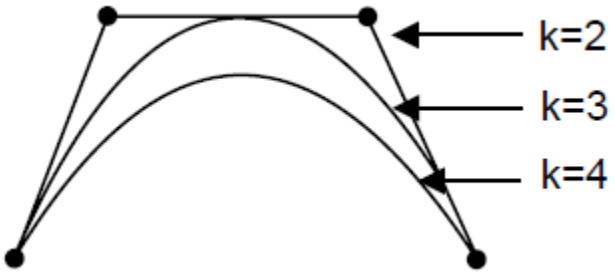
*t = [-2 0 2 4 6]
*t = [0 ¼ ½ ¾ 1]



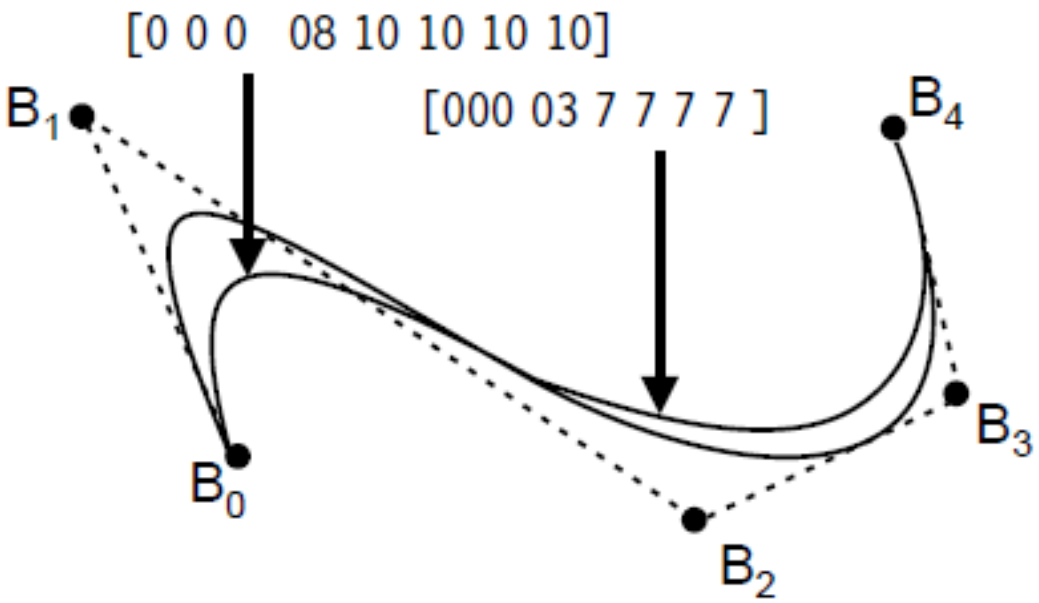
Splines não periódicas

- Valores repetidos nos extremos com multiplicidade k
- Nós internos igualmente espaçados

Ordem(k)	Nº de nós(m)	Vetor de nós não-periódicos
2	6	0 0 1 2 3 3
3	7	0 0 0 1 2 2 2
4	8	0 0 0 0 1 1 1 1

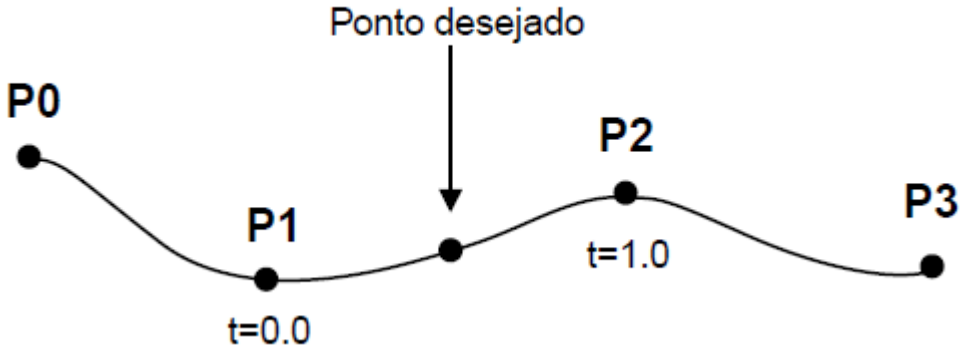


Splines não uniformes



Catmull-Rom Splines

- Passa por todos os pontos de controle
- Continuidade C1

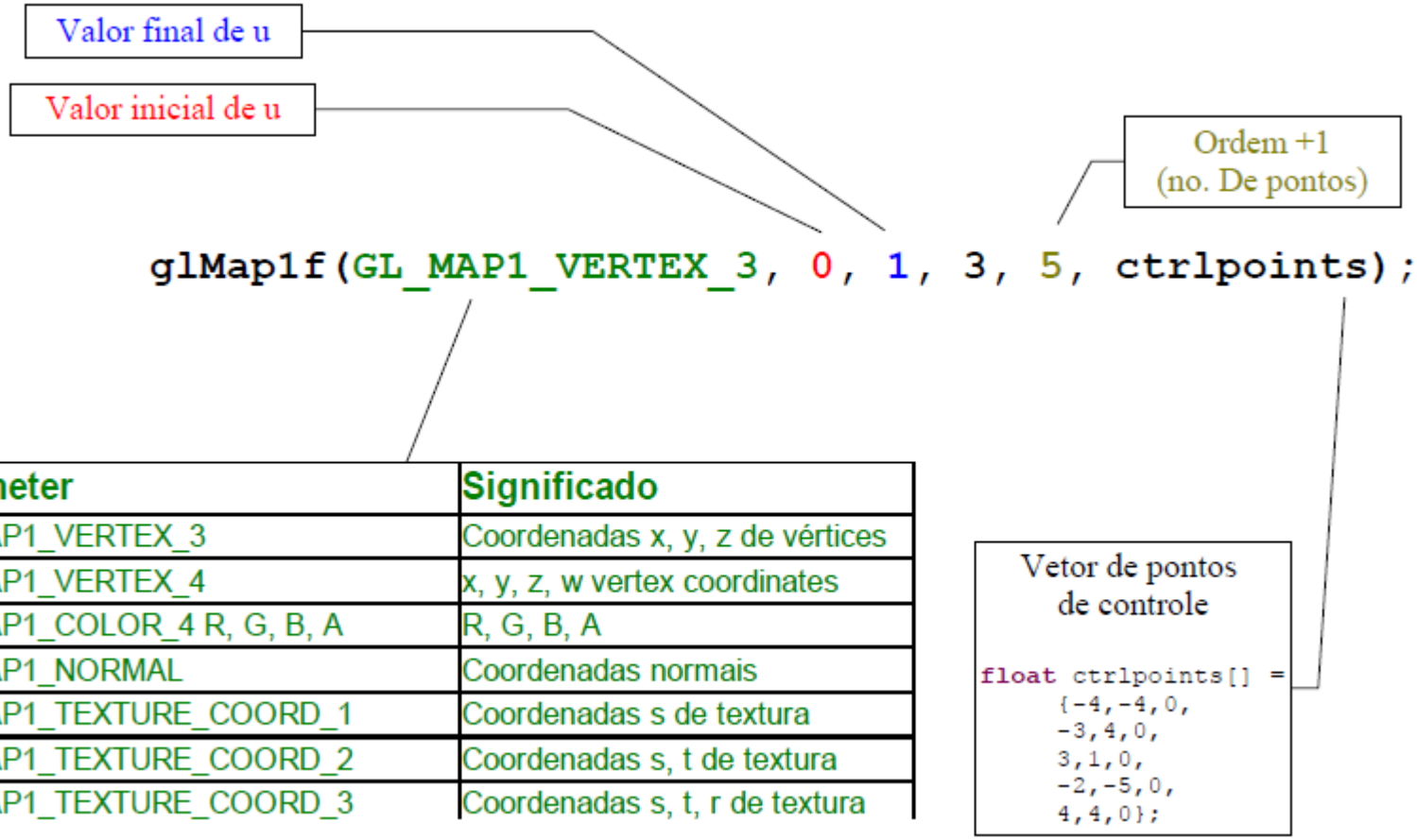


$$q(t) = \frac{1}{2}[t^3 \quad t^2 \quad t \quad 1] * \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} * \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Curvas em OpenGL

*Curvas de Bézier
*glEnable(GL_MAP1_VERTEX_3);
*glMap1f(GL_MAP1_VERTEX_3, 0, 1, 3, 5, ctrlpoints);
*glEvalCoord1f(i);

Curvas em OpenGL

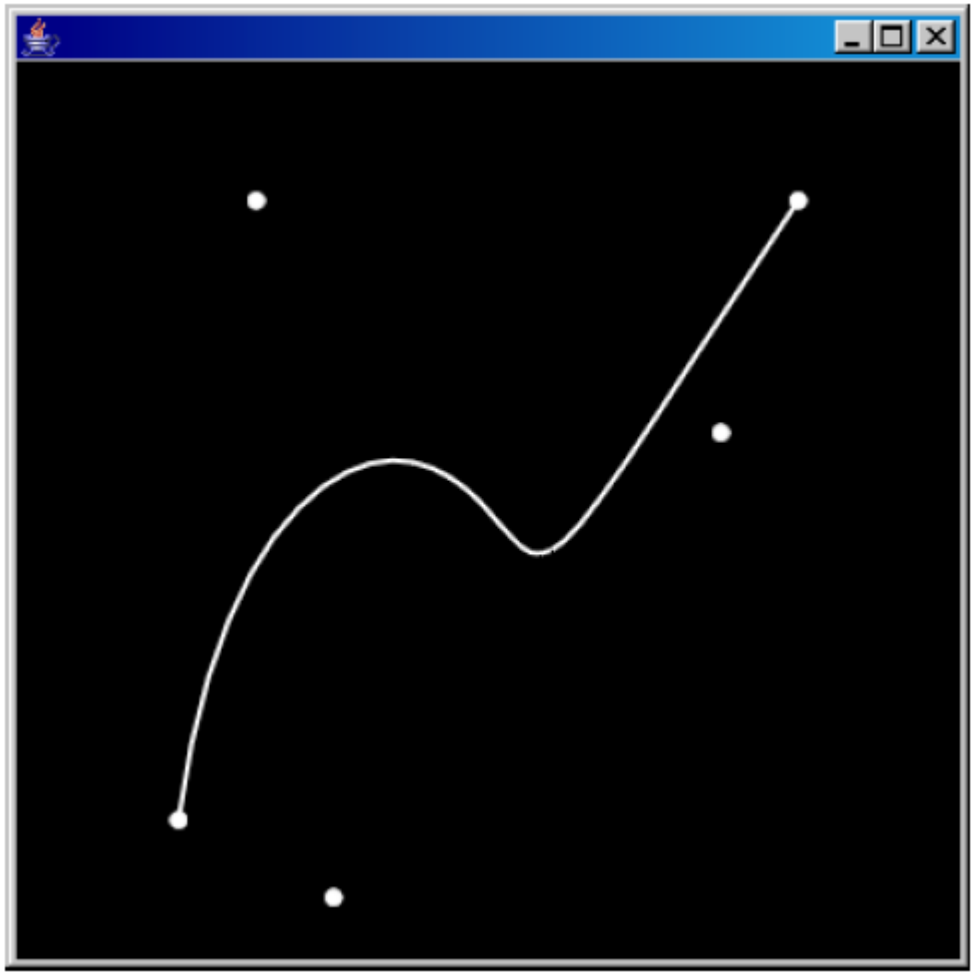


Parameter	Significado
GL_MAP1_VERTEX_3	Coordenadas x, y, z de vértices
GL_MAP1_VERTEX_4	x, y, z, w vertex coordinates
GL_MAP1_COLOR_4 R, G, B, A	R, G, B, A
GL_MAP1_NORMAL	Coordenadas normais
GL_MAP1_TEXTURE_COORD_1	Coordenadas s de textura
GL_MAP1_TEXTURE_COORD_2	Coordenadas s, t de textura
GL_MAP1_TEXTURE_COORD_3	Coordenadas s, t, r de textura

Curvas em OpenGL

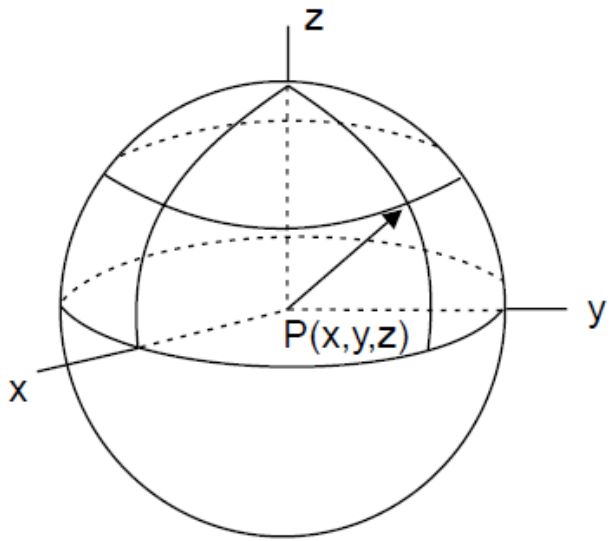
```
glBegin(GL_LINE_STRIP);
for(i=0;i<=30;i++)
        glEvalCoord1f(i/30);
glEnd();
```

Curvas em OpenGL



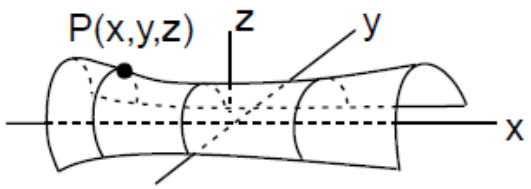
Superfícies

- Representações como em curvas
  - Por pontos, paramétricas, etc.



Esfera com centro (x_0,y_0,z_0)

Equação: $(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2=r^2$

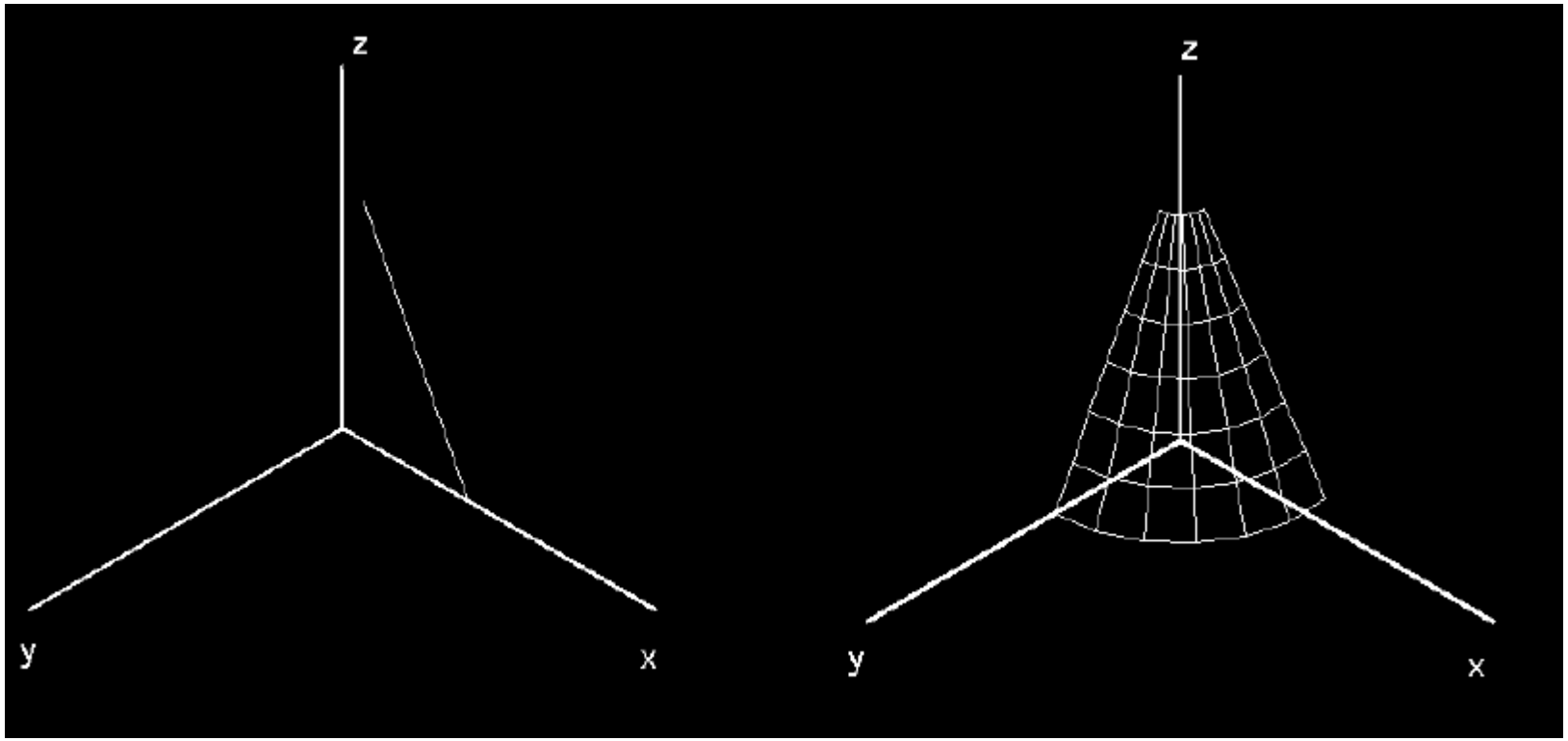


Parabolóide Hiperbólico

Equação: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = \frac{z}{c}$

Superfícies por revolução

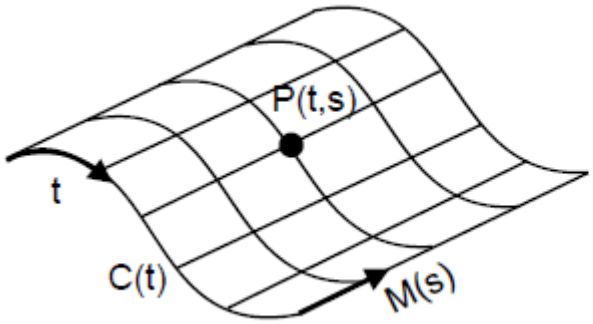
*O objeto que gera a superfície pode ser uma reta, ou curva qualquer.

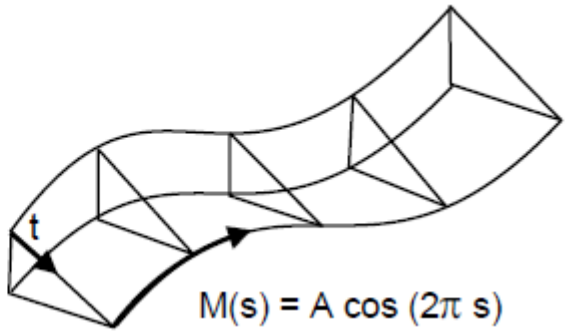


Superfícies geradas por deslocamento

- Curva a ser deslocada: C(t)
- Caminho a ser percorrido: M(s)

$$P(t, s) = C(t) \times M(s)$$

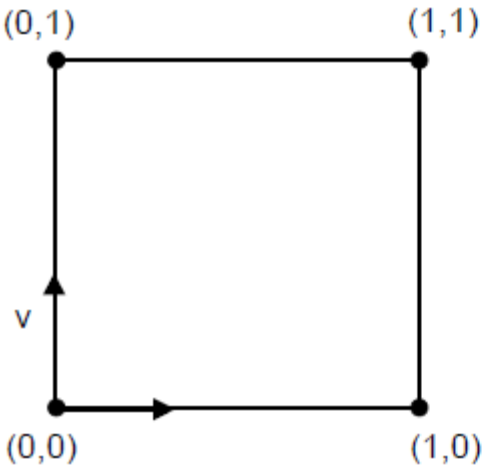


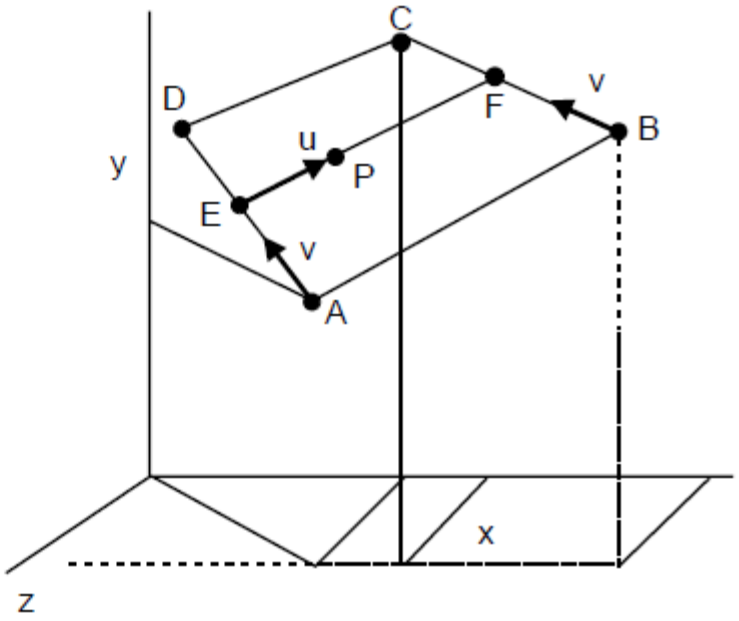


Interpolação bilinear

$$E = (1 - v)\,A + v\,D$$

$$F = (1 - v)\,B + v\,C$$

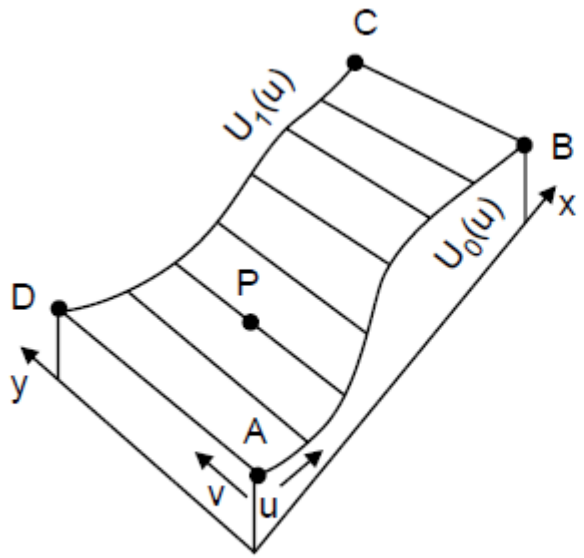




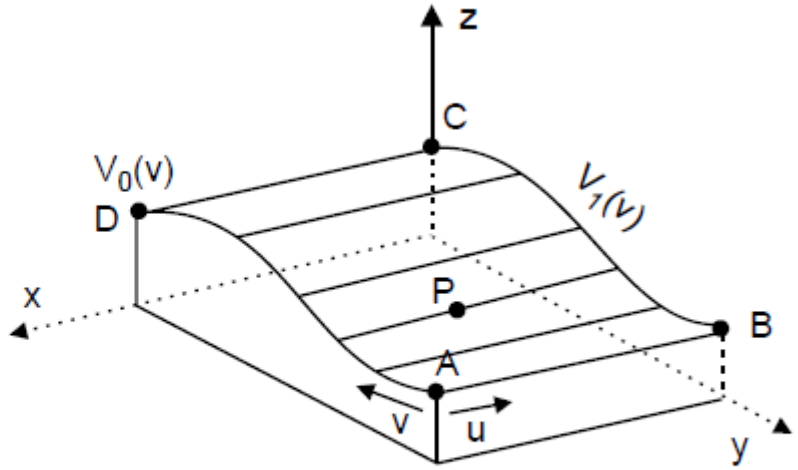
Interpolação bilinear

$$P(u,v) = (1 - u)\, E + u\, F$$

$$P(u,v) = (1 - u)\,(1 - v)\, A + (1 - u)\, v\ D + u\,(1 - v)\, B + u\, v\ C$$



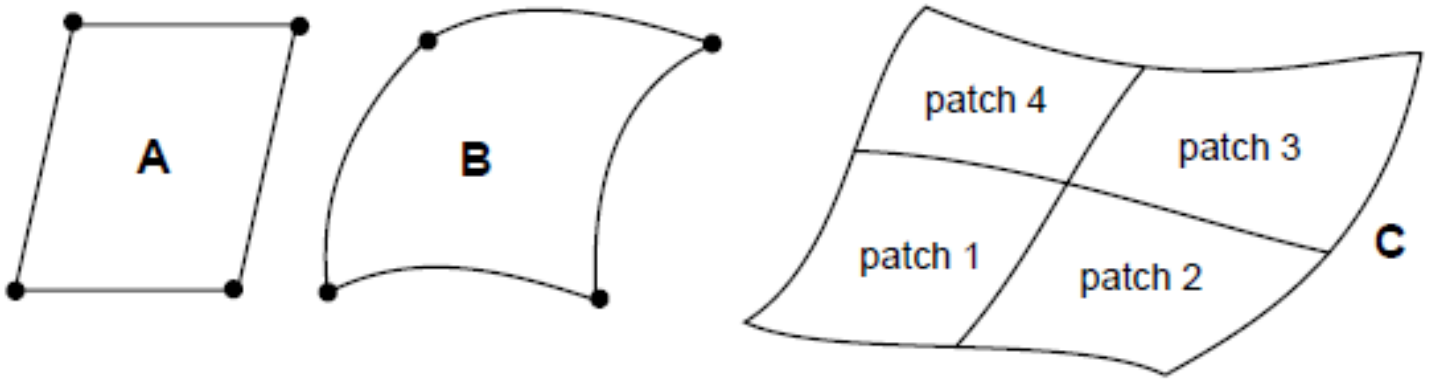
$P(u,v)=(1-v)\ U_0(u) + v\ U_1(u)$



$P(u,v)=(1-u)\ V_0(v) + u\ V_1(v)$

Superfícies Paramétricas Bicúbicas

$$P(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{ij} B_i(u) B_j(v)$$



Relembrando

*Curva Hermite
*Depende dos pontos e suas tangentes

$$x(t) = TC_x = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_x = THG_x$$

$$y(t) = TC_y = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_y = THG_y$$

$$z(t) = TC_z = TH \begin{bmatrix} P1 \\ P2 \\ T1 \\ T2 \end{bmatrix}_z = THG_z$$

Superfícies de Hermite

$$P(s,t) = S\ H\ G_h\ H^T\ T^T$$

$$G_h = \begin{bmatrix} P(0,0) & P(0,1) & \frac{\partial P}{\partial t}(0,0) & \frac{\partial P}{\partial t}(0,1) \\ P(1,0) & P(1,1) & \frac{\partial P}{\partial t}(1,0) & \frac{\partial P}{\partial t}(1,1) \\ \frac{\partial P}{\partial s}(0,0) & \frac{\partial P}{\partial s}(0,1) & \frac{\partial^2 P}{\partial s \partial t}(0,0) & \frac{\partial^2 P}{\partial s \partial t}(0,1) \\ \frac{\partial P}{\partial s}(1,0) & \frac{\partial P}{\partial s}(1,1) & \frac{\partial^2 P}{\partial s \partial t}(1,0) & \frac{\partial^2 P}{\partial s \partial t}(1,1) \end{bmatrix}$$

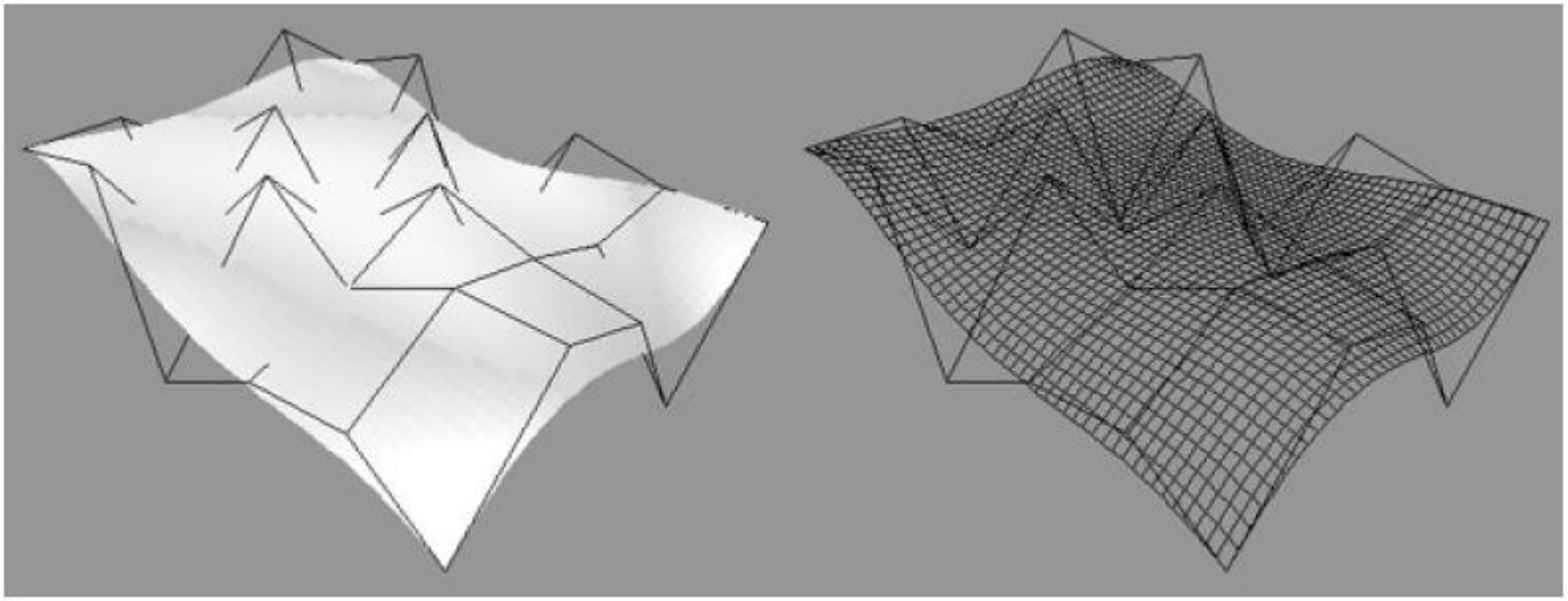
Superfícies de Bézier

$$P(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J_{i,n}(s)\, J_{j,m}(t) \quad 0 \le s,t \le 1$$

$$P(s,t) = S\, M_B G_b M_B^T T^T,$$

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad G_B = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$$

Superfícies de Bézier



Superfícies B-Spline

$$P(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,j} \, N_{i,k}(s) \, N_{j,l}(t)$$

$$P(s,t) = S \, M_s \, G_s \, M_s^T \, T^T$$

$$M_s = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

Superfícies Racionais

P (s, t) =	Forma Inteira	Forma Racional
Bézier	$\sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,j} \, J_{i,n}(s) \, J_{j,m}(t)$	$\dfrac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} \, B_{i,j} \, J_{i,n}(s) \, J_{j,m}(t)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} \, J_{i,n}(s) \, J_{j,m}(t)}$
B-Spline	$\sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,j} \, N_{i,k}(s) \, N_{j,l}(t)$	$\dfrac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} \, N_{i,k}(s) \, N_{j,l}(t) \, B_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} \, N_{i,k}(s) \, N_{j,l}(t)}$

NURBS

- Non-Uniform Rational B-Splines Surfaces
- Exclusivo para modelagem em computador
- Geralmente grau 3
  - Permite C2
- Versátil pois pode representar superfícies diferentes
  - Ex: $w_{i,j} = 1$

NURBS

NURBS em OpenGL

```
*Knots[8] = {0,0,0,0,1,1,1,1};
*float ctlPoints[4][4][3]={
*       {{-4, 4,0},{-2, 4, 0},{2, 4, 0},{4, 4, 0}},
*       {{-4, 2,0},{-2, 2,-H},{2, 2,H},{4, 2, 0}},
*       {{-4,-2,0},{-2,-2,-H},{2,-2,H},{4,-2, 0}},
*       {{-4,-4,0},{-2,-4, 0},{2,-4, 0},{4,-4, 0}}
*    };

* gluNurbsSurface(nurbSurface,
*                 8,Knots,8,Knots,
*                 4*3,3,&ctlPoints[0][0][0],
*                 4,4,GL_MAP2_VERTEX_3);
*  gluEndSurface(nurbSurface);
```

NURBS em OpenGL

```
* GLUnurbsObj *nurbSurface;

*nurbSurface = gluNewNurbsRenderer();
* gluNurbsProperty(nurbSurface,GLU_DISPLAY_MODE,GLU_FILL);

* gluBeginSurface(nurbSurface);
* gluNurbsSurface(nurbSurface,
*                 8,Knots,8,Knots,
*                 4*3,3,&ctlPoints[0][0][0],
*                 4,4,GL_MAP2_VERTEX_3);
*  gluEndSurface(nurbSurface);
```

Superfície Bézier em OpenGL

```
*glEnable(GL_MAP2_VERTEX_3);
*
*float ctlPoints[4][4][3]={
*       {{-4, 4,0},{-2, 4, 0},{2, 4, 0},{4, 4, 0}},
*       {{-4, 2,0},{-2, 2,-H},{2, 2,H},{4, 2, 0}},
*       {{-4,-2,0},{-2,-2,-H},{2,-2,H},{4,-2, 0}},
*       {{-4,-4,0},{-2,-4, 0},{2,-4, 0},{4,-4, 0}}
*    };

* glMap2f(GL_MAP2_VERTEX_3,
              0, 1, 3, 4,
              0, 1, 12, 4,
              &ctlPoints[0][0][0]);
*   glMapGrid2f(40, 0.0, 1.0, 40, 0.0, 1.0);
*   glEvalMesh2(GL_FILL,0,40,0,40);
```