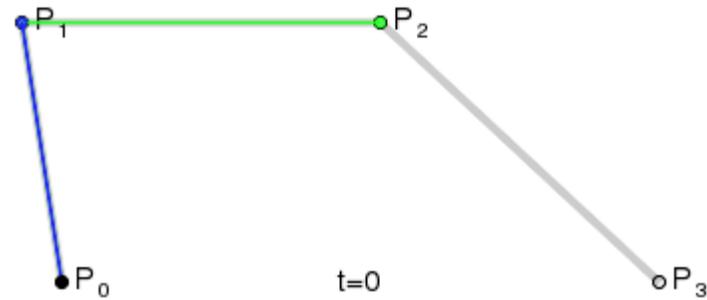


https://pt.wikipedia.org/wiki/Curva_de_Bezier

aula 17



Curvas de Bezier
2017/2 – IC / UFF

Curva polinomial desenvolvida em 1962 por Pierre Bézier.

Utilizada no projeto de automóveis (Renault).

Baseada no algoritmo de De Casteljaou em 1957.

Curva de aproximação.

Controle global.

P. de Casteljaou, 1959 (Citroën)

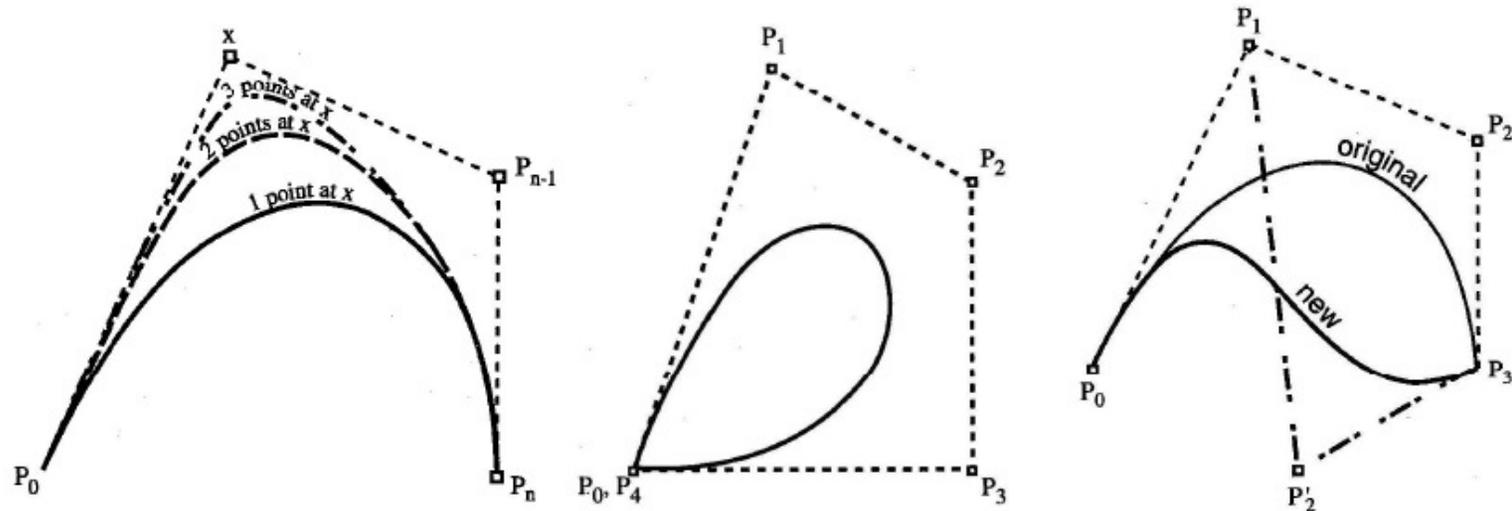
P. de Bézier, 1962 (Renault) - UNISURF

Forest 1970: Polinômios de Bernstein



Curva de Bezier

- Desenvolvidas por Pierre Bezier para descrever o desenho de curvas e superfícies de forma livre
- Polígono definidor
- Primeiro e último ponto
- Vetor tangente



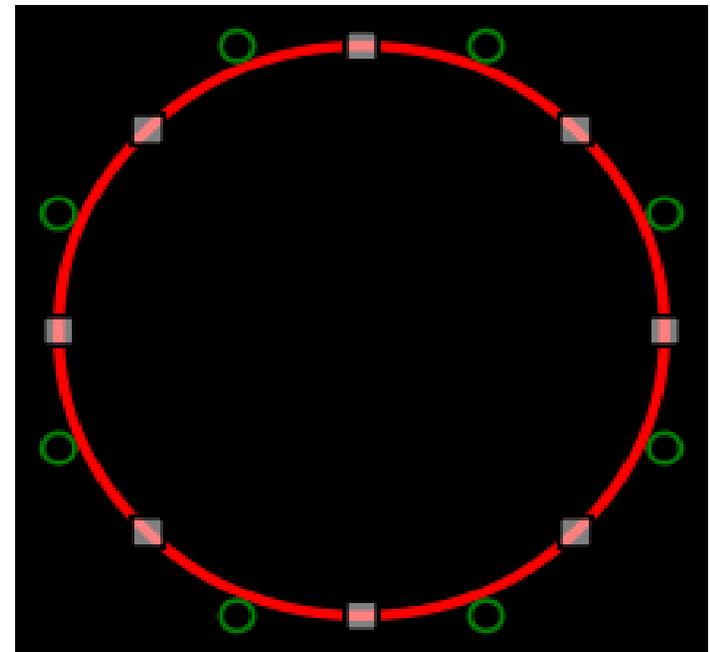
pontos de controle = P_i

Onde se usa: Qualquer representação de curvas

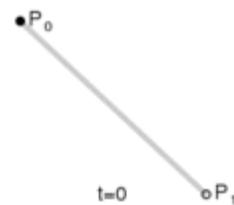
Por exemplo:

Os contornos dos caracteres em fontes TrueType são feitas de segmentos de retas e curvas Bézier quadrática.

O círculo ao lado é formado por 8 segmentos. Os quadrados são os **pontos de controle de extremidade** dos segmentos e os anéis os **de controle do interior** destes segmentos

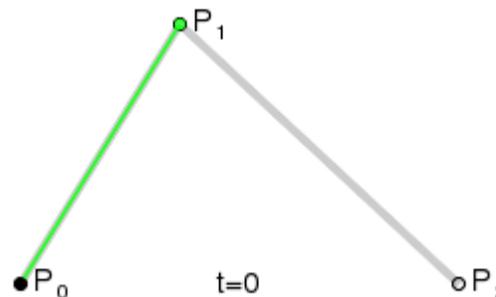


Mesmo segmentos lineares podem ser definido por Bezier:



pontos de controle = P_0 , P_1

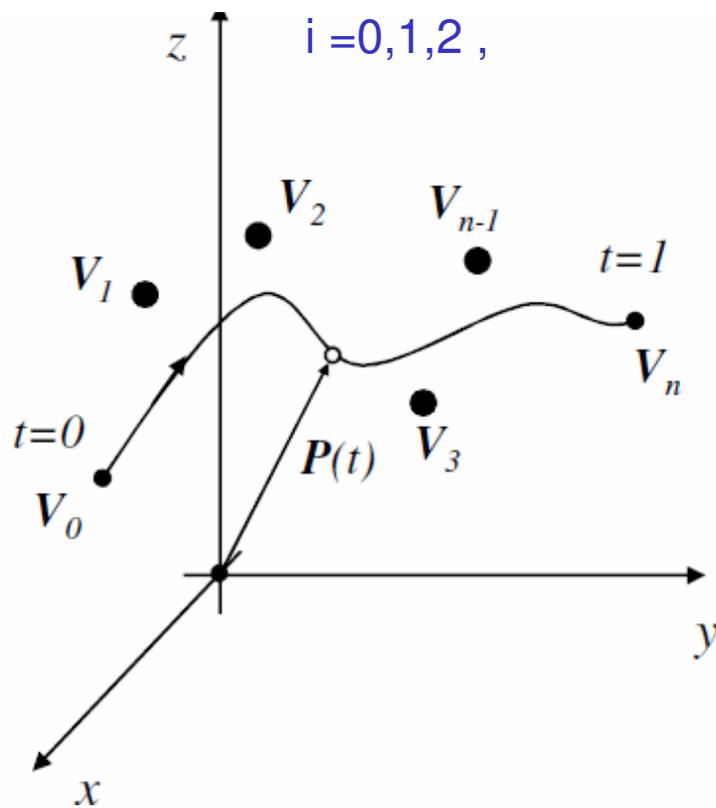
Um segmento de curva quadrática de Bézier é definido por 2 pontos extremos e 1 de controle.



pontos de controle, $i = 0, 1, 2$, P_i

Forma geral pode ter **$n+1$** pontos de controle, vamos chamar esses agora de V_i e $P(t)$ os pontos da curva:

pontos de controle = V_i



$$\vec{P}(t) = \sum_{i=0}^n B_{i,n}(t) \vec{V}_i$$

onde:

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

pol. Bernstein

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

coef. binomial

Lembre que o Fatorial de um número = **$n! = n(n-1) \dots 1$**

Como vimos na aula anterior

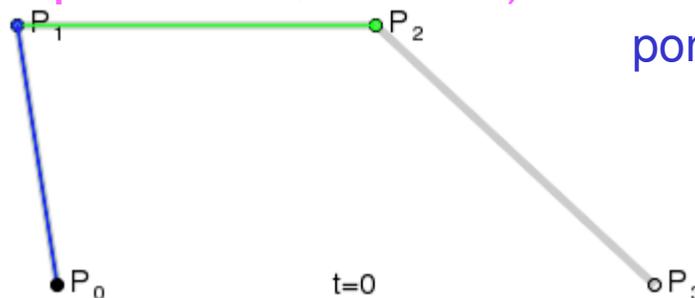
- As **curvas** cúbicas são especialmente úteis

(porque mesmo ????)

(esqueceu?)

(Vai lá, na aula passada, ver....)

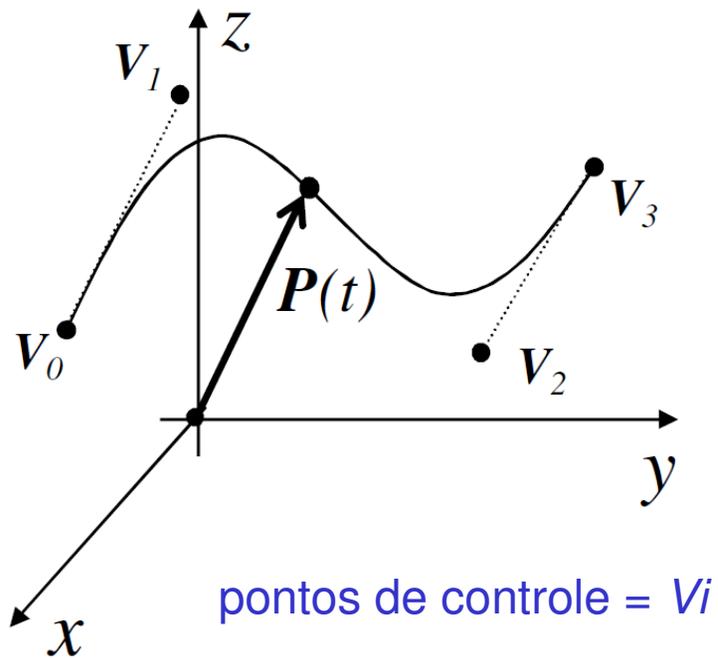
Assim vê



pontos de controle = P_i

Entrar na...

Bezier cúbica:



$$\vec{P}(t) = \sum_{i=0}^3 B_{i,3}(t) \vec{V}_i$$

$$B_{0,3}(t) = \binom{3}{0} (1-t)^{3-0} t^0 = (1-t)^3$$

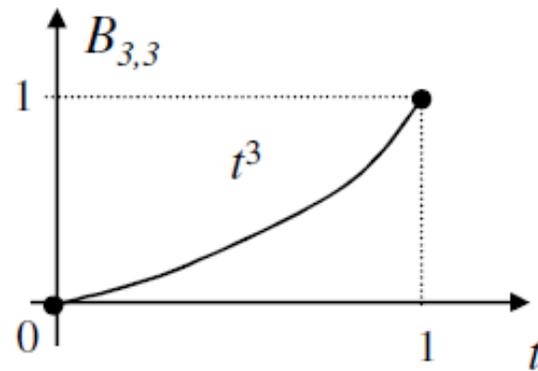
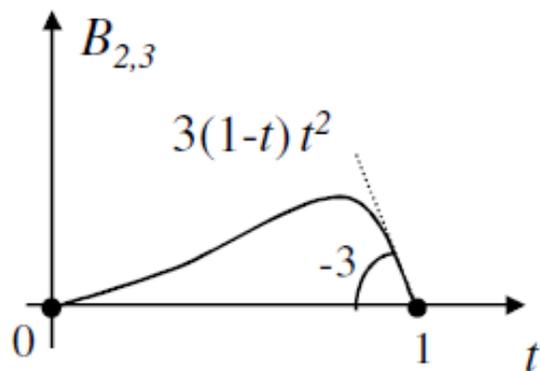
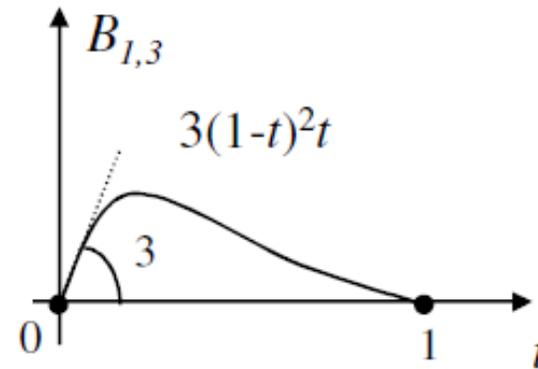
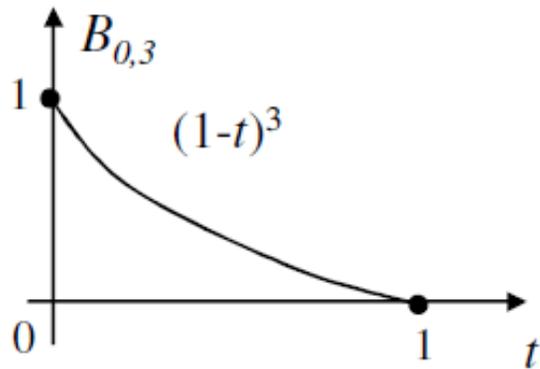
$$B_{1,3}(t) = \binom{3}{1} (1-t)^{3-1} t^1 = 3(1-t)^2 t$$

$$B_{2,3}(t) = \binom{3}{2} (1-t)^{3-2} t^2 = 3(1-t) t^2$$

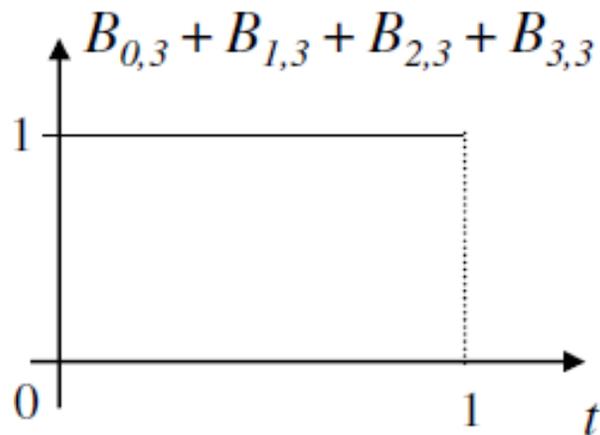
$$B_{3,3}(t) = \binom{3}{3} (1-t)^{3-3} t^3 = t^3$$

$$\vec{P}(t) = (1-t)^3 \vec{V}_0 + 3(1-t)^2 t \vec{V}_1 + 3(1-t) t^2 \vec{V}_2 + t^3 \vec{V}_3$$

Os interpoladores usados são chamados de Polinômios cúbicos de **Bernstein**



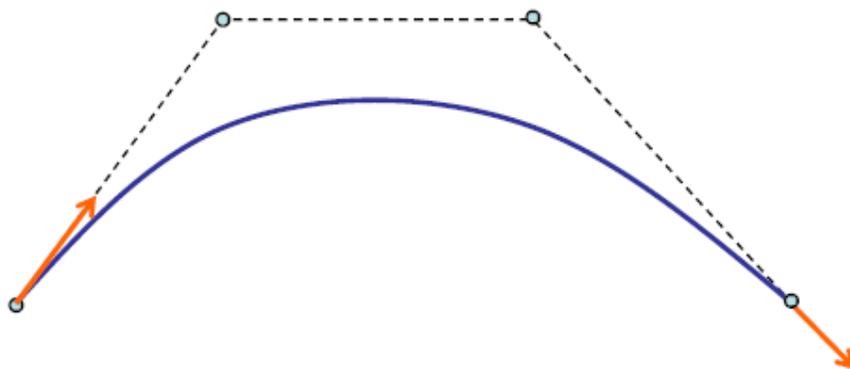
A soma dos Polinômios Cúbicos de Bernstein resulta:



Sempre 1 !!!!!

Propriedades de Curva de Bézier

- Continuidade infinita (todas as derivadas são contínuas)
- O grau da curva (do polinômio) é dado pelo número de pontos do polígono de controle menos 1
- A curva de Bézier está contida no fecho convexo do polígono de controle
- A curva interpola o primeiro e último ponto do polígono de controle



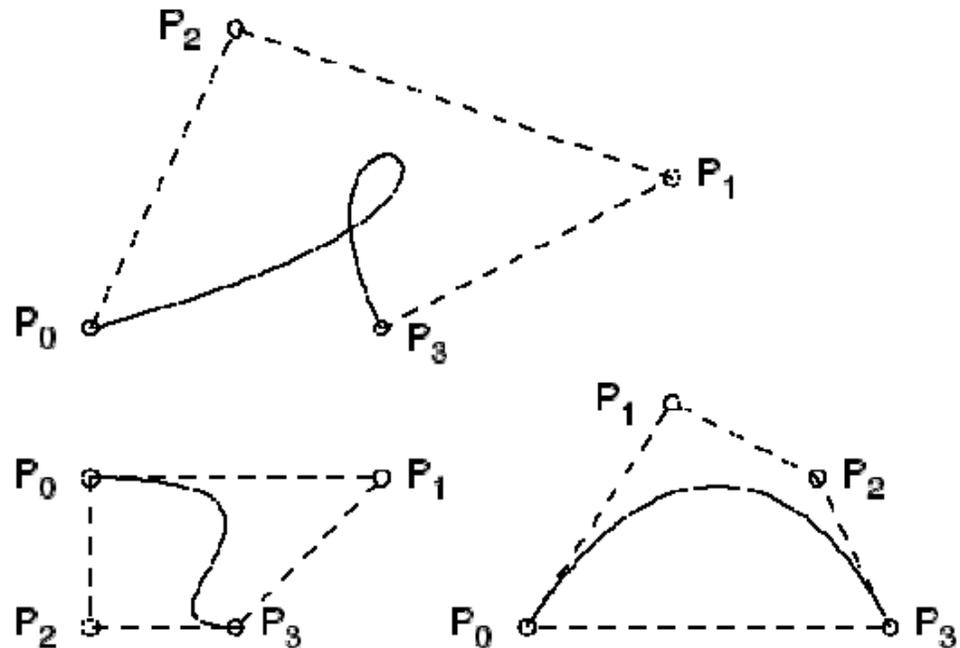
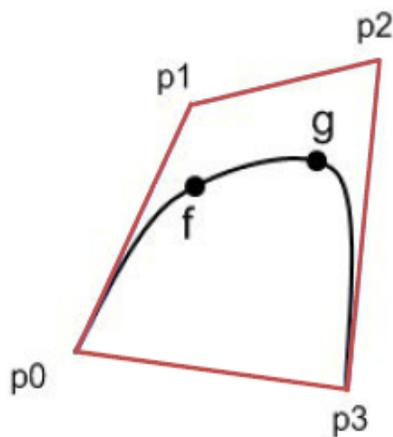
Fecho convexo?

- Falamos isso na aula passada ! ! ! !



Propriedade: *Convex Hull*

- Uma curva de Bézier está completamente dentro do maior polígono convexo, formado pelos pontos de controle.

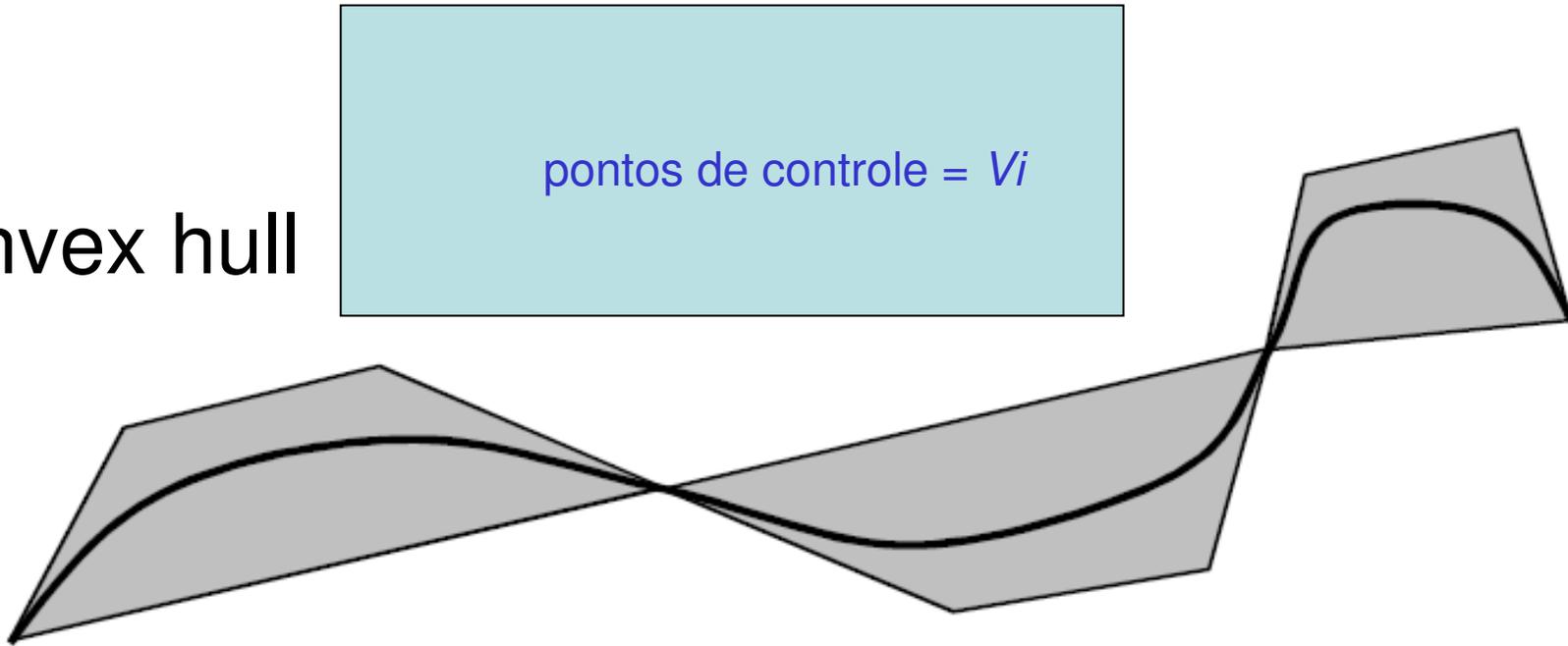


pontos de controle = P_i

Quando se tem muitas curvas para formar uma única como fica o Fecho convexo?

Convex hull

pontos de controle = V_i

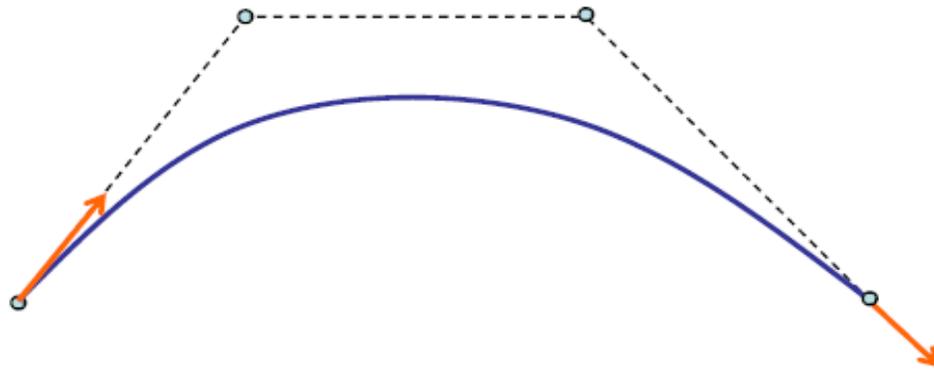


$$\vec{P}(t) = \sum_{i=0}^n \alpha_i \vec{V}_i \quad \text{com} \quad \sum_{i=0}^n \alpha_i = 1$$

Cont.

pontos de controle = P_i

- As tangentes à curva em \mathbf{p}_0 e \mathbf{p}_n têm a direção dos segmentos de reta $\mathbf{p}_0\mathbf{p}_1$ e $\mathbf{p}_{n-1}\mathbf{p}_n$, respectivamente
 - Para cúbicas, as derivadas são $3(\mathbf{p}_1 - \mathbf{p}_0)$ e $3(\mathbf{p}_2 - \mathbf{p}_3)$
- Transformar os pontos de controle (transf. afim) e desenhar a curva é equivalente a desenhar a curva transformada



Demonstrando essas propriedades para uma Bezier cúbica:

$$\vec{P}(t) = (1-t)^3 \vec{V}_0 + 3(1-t)^2 t \vec{V}_1 + 3(1-t)t^2 \vec{V}_2 + t^3 \vec{V}_3$$

$$\frac{d}{dt} \vec{P}(t) = -3(1-t)^2 \vec{V}_0 + [-6(1-t)t + 3(1-t)^2] \vec{V}_1 + [-3t^2 + 6(1-t)t] \vec{V}_2 + t^3 \vec{V}_3$$

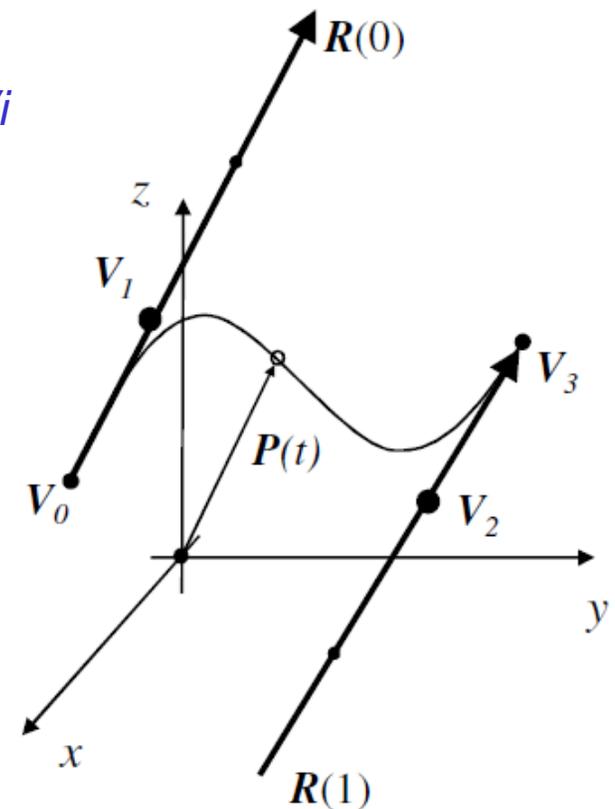
$$\vec{P}(0) = \vec{V}_0$$

pontos de controle = V_i

$$\vec{P}(1) = \vec{V}_3$$

$$\frac{d}{dt} \vec{P}(0) = -3 \vec{V}_0 + 3 \vec{V}_1$$

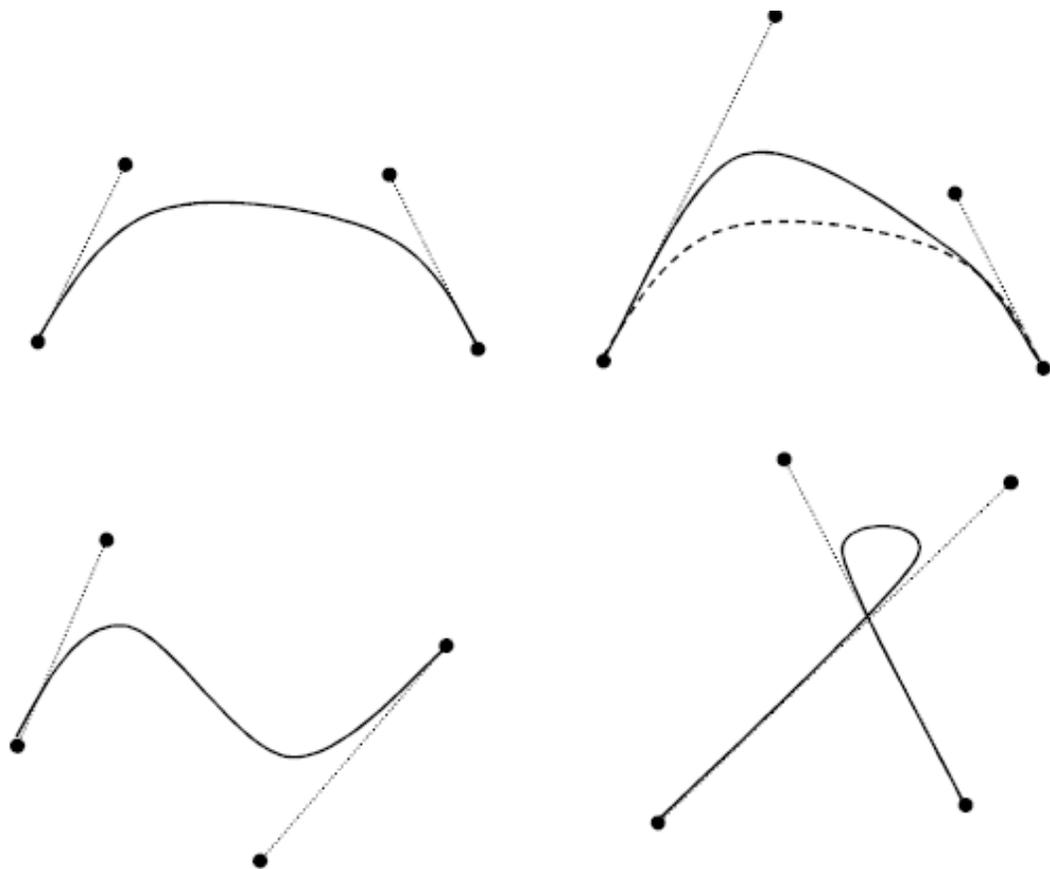
$$\frac{d}{dt} \vec{P}(1) = -3 \vec{V}_2 + 3 \vec{V}_3$$



Se fosse pedido para reparametrizar de forma especial

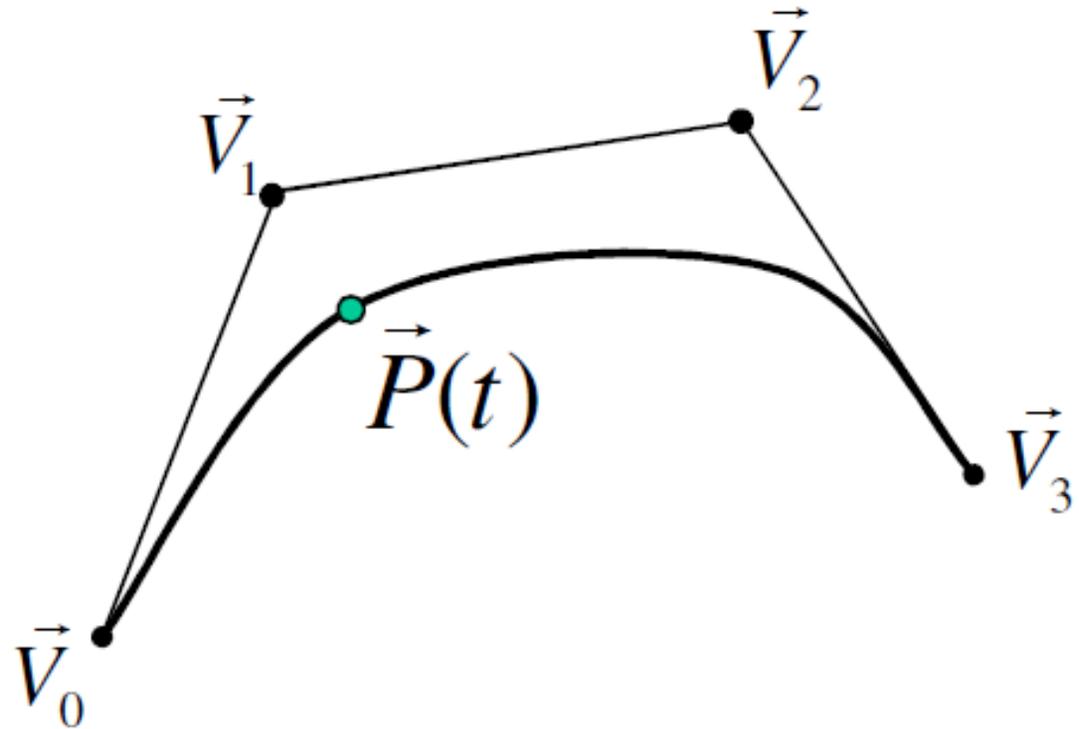
- por exemplo com mais pontos onde a derivada da curva fosse maior, ou ela tivesse maior curvatura ?
- Com as expressões do slide anterior isso poderia ser feito!
- (simples não??)

A ordem e posição dos pontos controla a curva!



Representação matricial :

pontos de controle = V_i



$$\vec{P}(t) = (1-t)^3 \vec{V}_0 + 3(1-t)^2 t \vec{V}_1 + 3(1-t) t^2 \vec{V}_2 + t^3 \vec{V}_3$$

$$\vec{P}(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & -3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{0x} & V_{0y} & V_{0z} \\ V_{1x} & V_{1y} & V_{1z} \\ V_{2x} & V_{2y} & V_{2z} \\ V_{3x} & V_{3y} & V_{3z} \end{bmatrix}$$

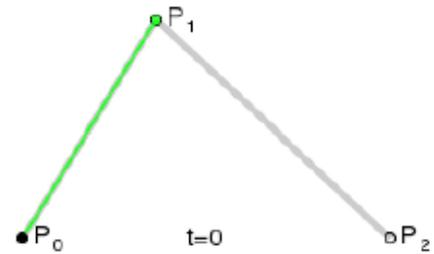
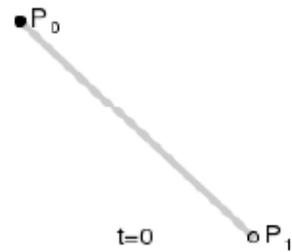
■ Matriz de Geometria (**G**) e Matriz Base (**M**)

$$Q(t) = [x(t) \quad y(t) \quad z(t)] = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

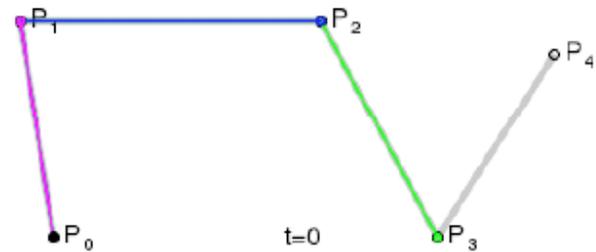
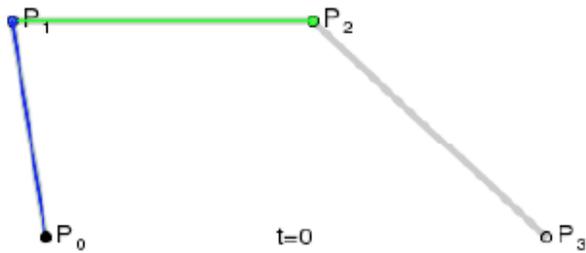
$Q(t) = TMG$

pontos de controle = G_i

Outras formas de Bezier



pontos de controle = P_i

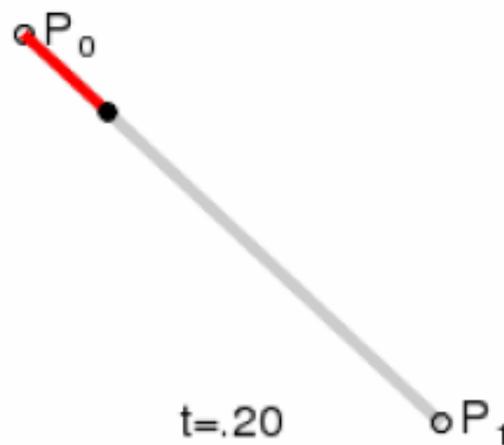


Curvas de Bézier: 1. linear; 2. quadrática; 3. cúbica; 4. quártica.

Outras formas de Bezier

$$P(t) = (1-t) \cdot P_0 + t \cdot P_1$$

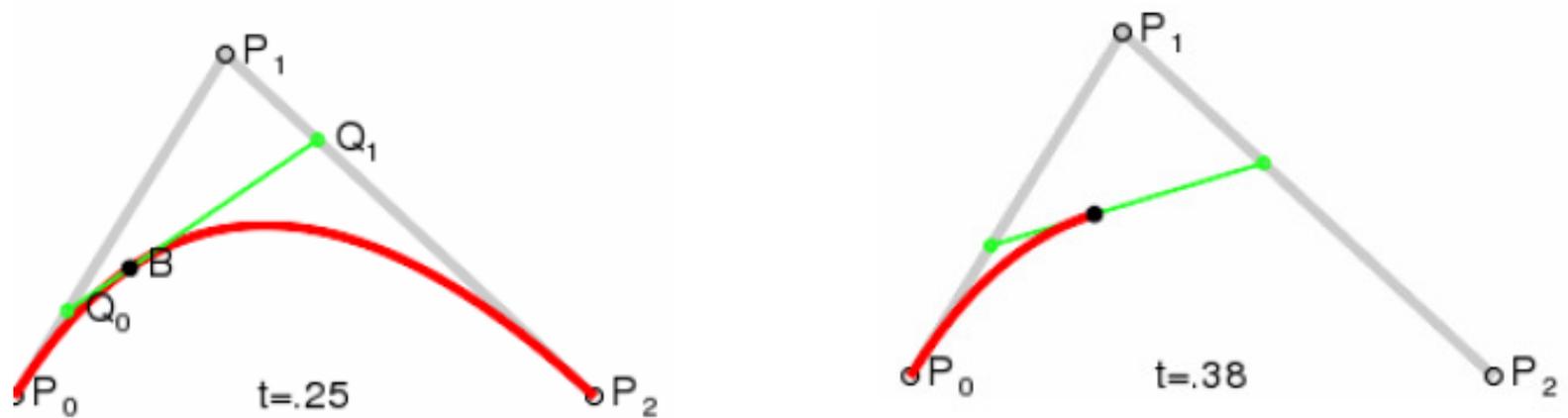
pontos de controle = P_i



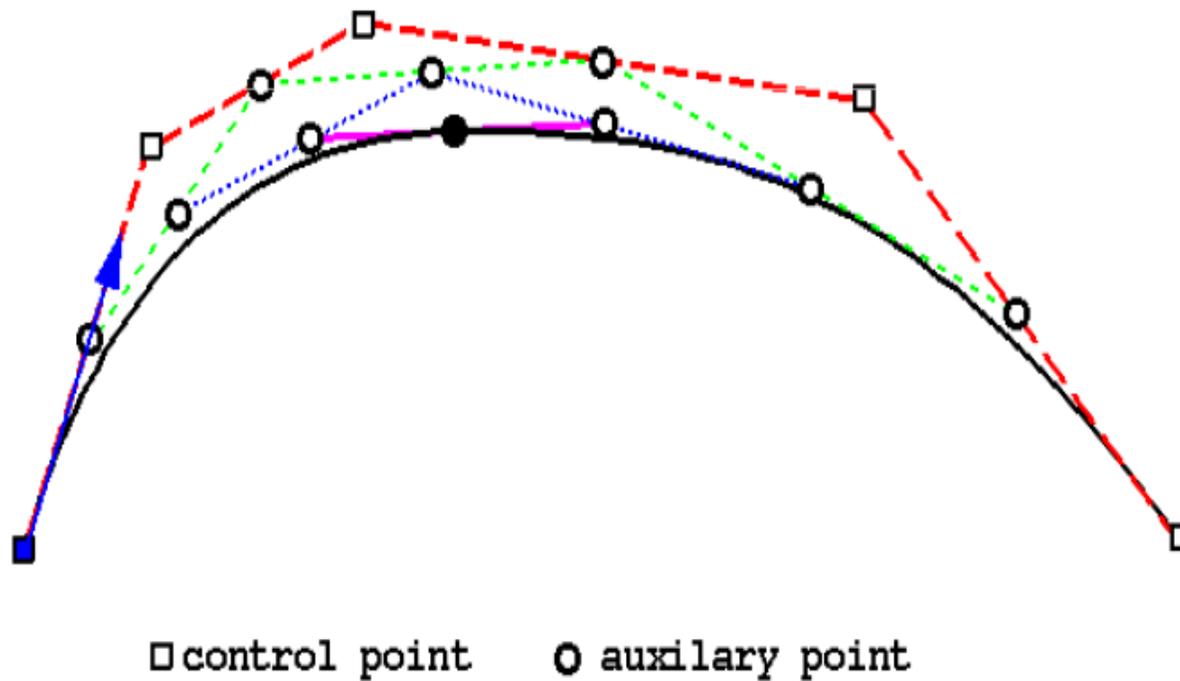
Outras formas de Bezier

$$P(t) = (1 - t)^2 \mathbf{P}_0 + 2 t (1 - t) \mathbf{P}_1 + t^2 \mathbf{P}_2$$

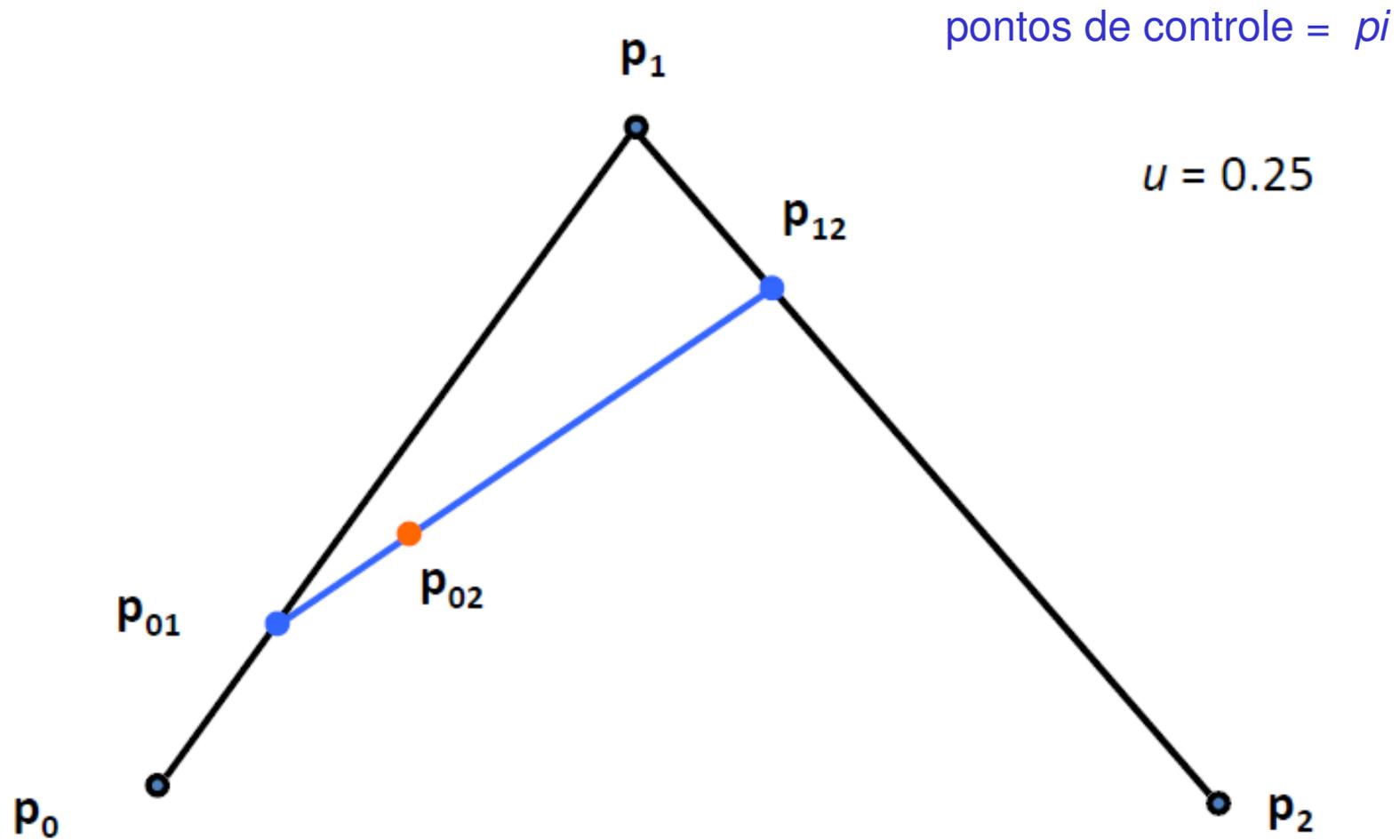
pontos de controle = P_i

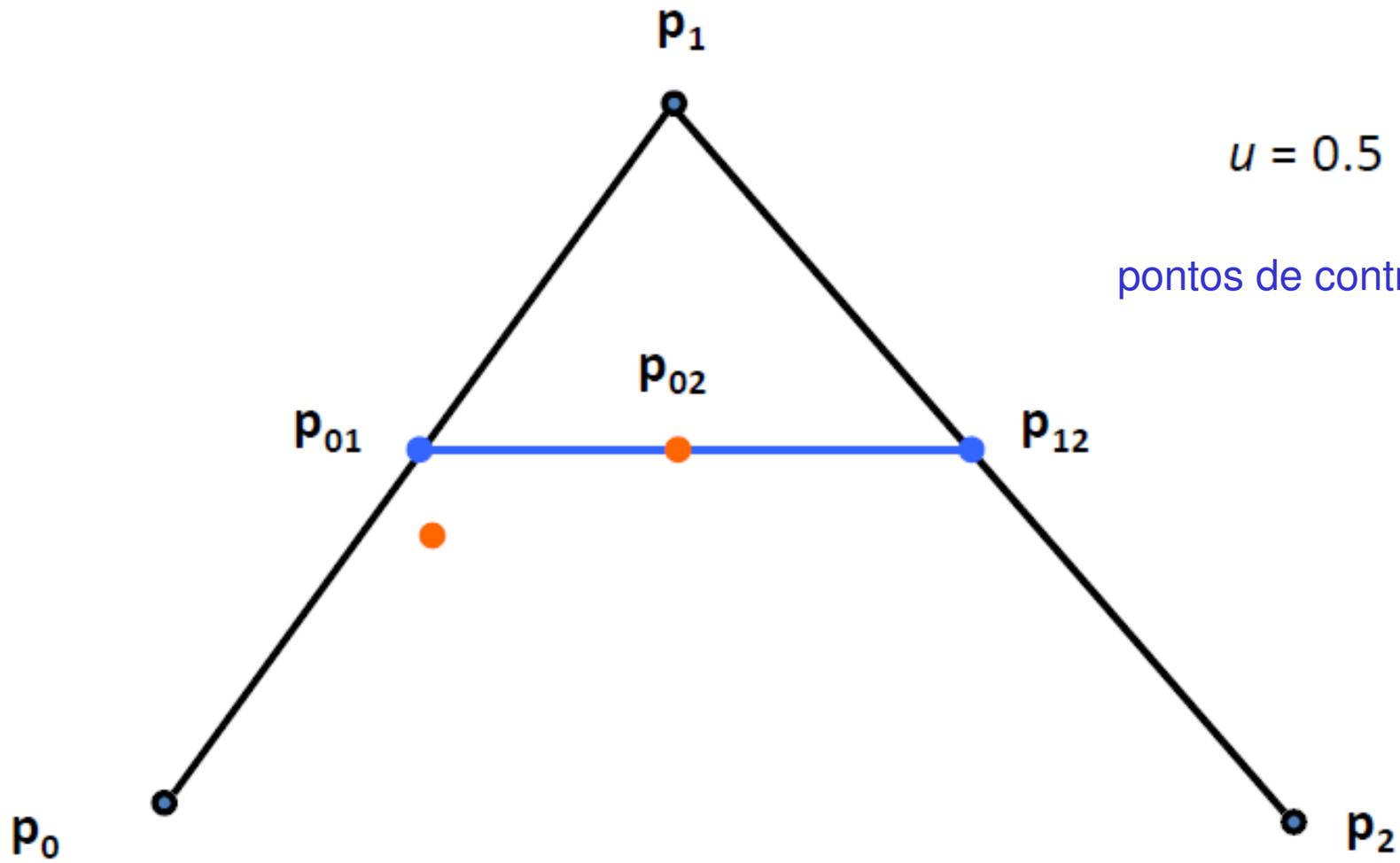


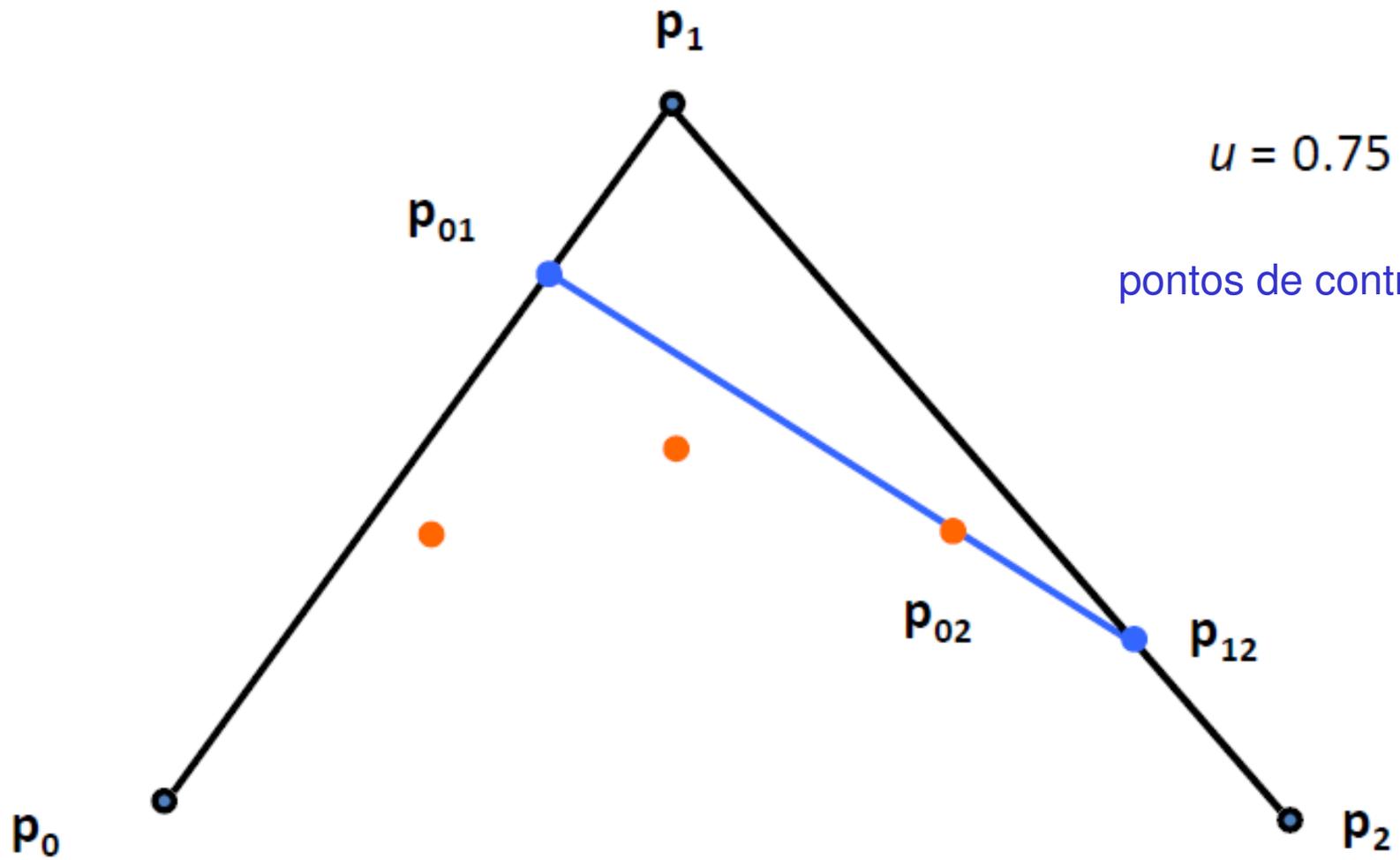
De Casteljau: algoritmo geométrico para construção de curvas Bézier.

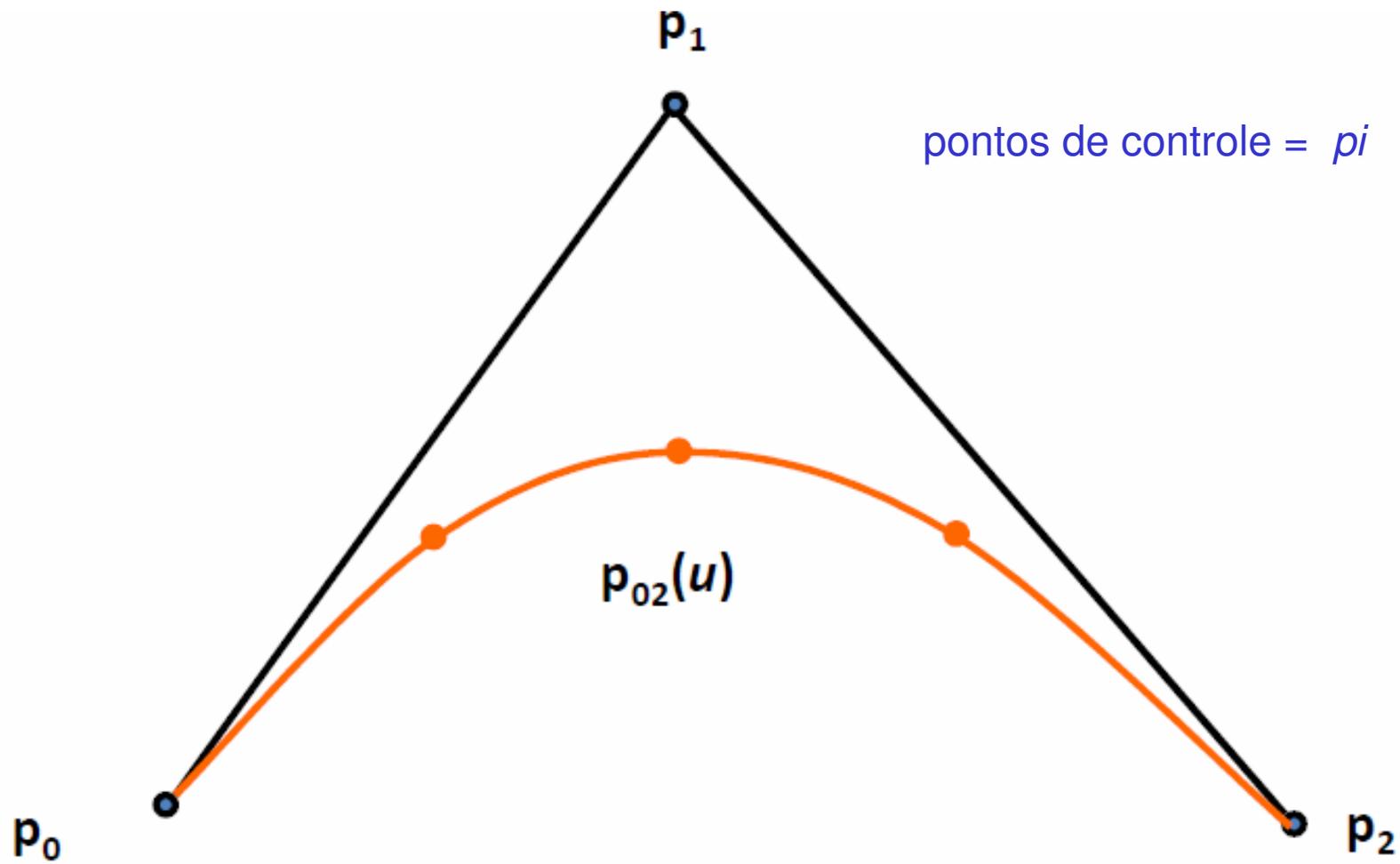


Algoritmo geométrico



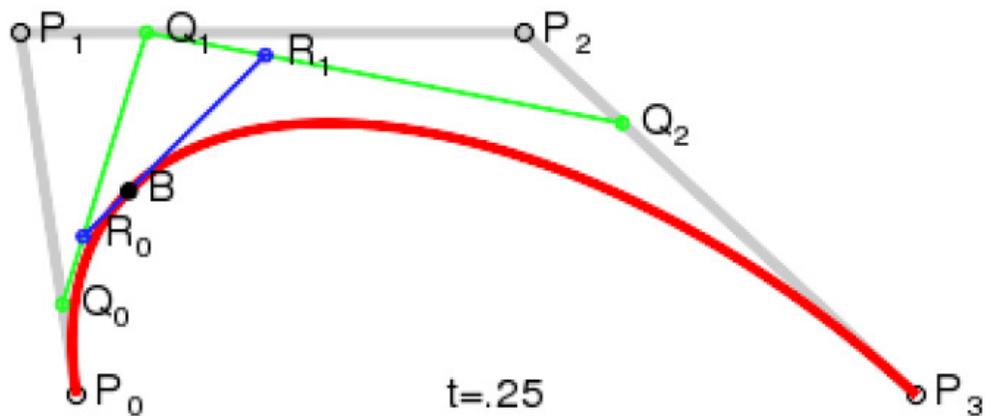




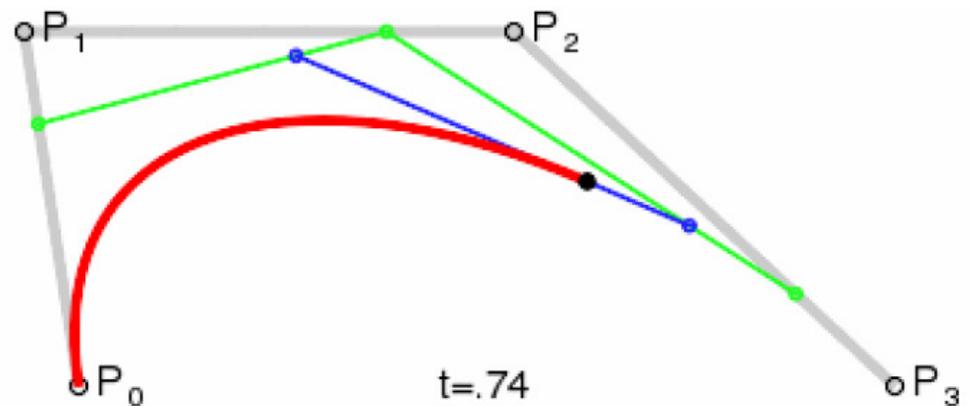


Outras formas de Bezier

$$P(t) = (1 - t)^3 \mathbf{P}_0 + 3 t (1 - t)^2 \mathbf{P}_1 + 3 t^2 (1 - t) \mathbf{P}_2 + t^3 \mathbf{P}_3$$



pontos de controle = P_i

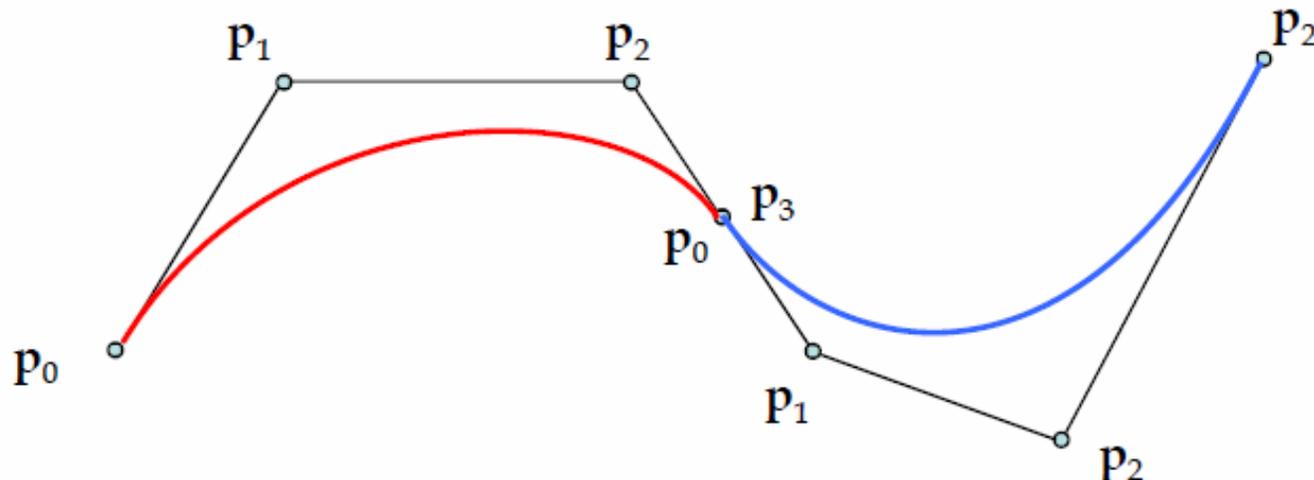


Cont.

- Curvas Bézier com k pontos de controle são de grau $k - 1$
- Curvas de grau alto são difíceis de desenhar
 - Complexas
 - Sujeitas a erros de precisão
- Normalmente, queremos que pontos de controle tenham efeito *local*
 - Em curvas Bézier, todos os pontos de controle têm efeito *global*
- Solução:
 - Emendar curvas polinomiais de grau baixo
 - Relaxar condições de continuidade

Emendando Curvas Bézier

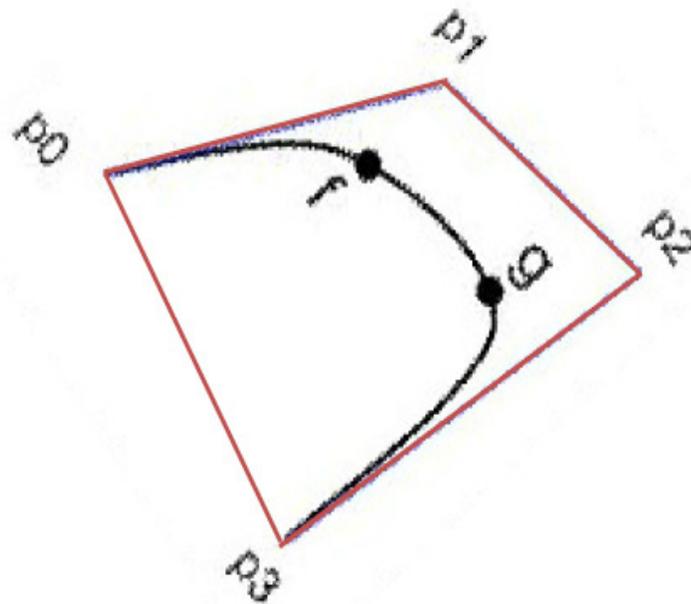
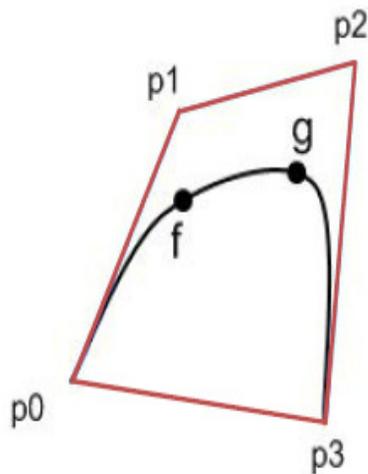
- Continuidade C^0 : Último ponto da primeira = primeiro ponto da segunda
- Continuidade C^1 : C^0 e segmento $\mathbf{p}_2\mathbf{p}_3$ da primeira com mesma direção e comprimento que o segmento $\mathbf{p}_0\mathbf{p}_1$ da segunda
- Continuidade C^2 : C^1 e + restrições sobre pontos \mathbf{p}_1 da primeira e \mathbf{p}_2 da segunda



pontos de controle = p_i

Transformações

- Executar as transformações (S,R,T) na curva é equivalente a realizar as transformações nos pontos de controle.



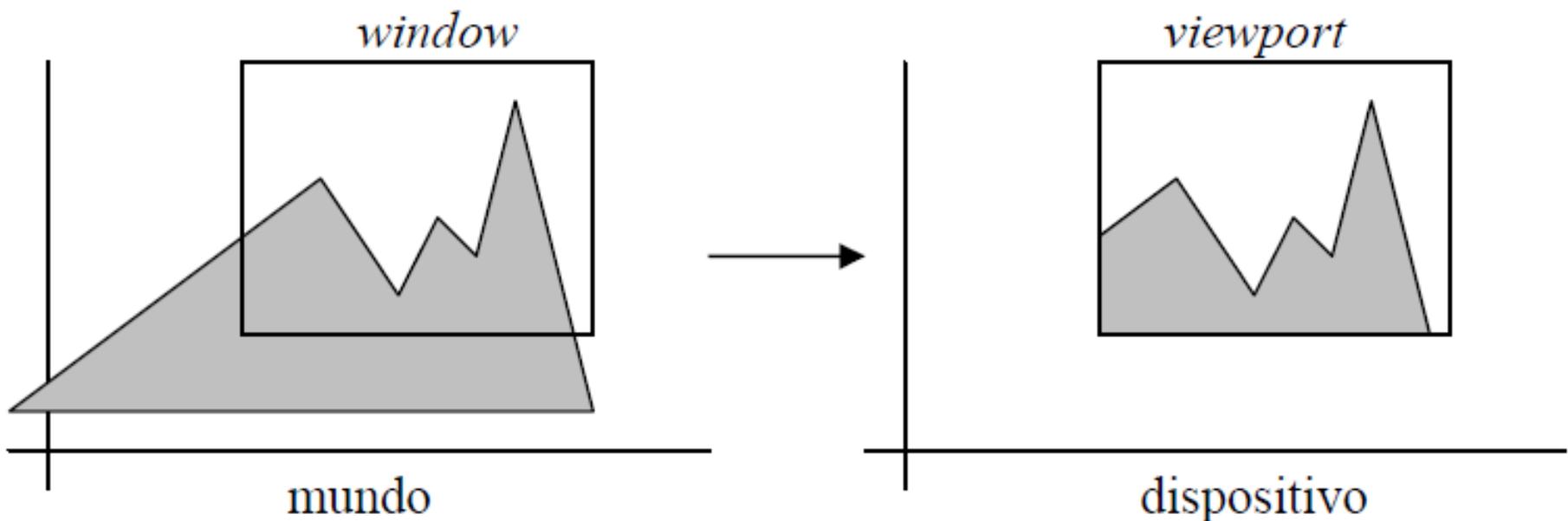
pontos de controle = p_i

Vamos falar em coordenadas:

Window → área selecionada do sistema de coordenadas do mundo para ser exibida no display (área que será vista). Deve-se observar que nem toda área de um ambiente pode ser exibida em uma única *window*. Esta área é determinada pela configuração e posicionamento da câmera sintética.

Viewport → área do display (monitor) na qual a *window* é mapeada. Corresponde a posição na tela onde mostrar. Ex: janela gráfica de uma aplicação gráfica.

Viewing transformation (ou *windowing transformation*) → mapeamento de uma parte da cena em coordenadas do mundo para coordenadas do dispositivo.



O usuário

- Definirá os pontos iniciais finais e os intermediários nas **coordenadas dele**.
- A curva de Bezier pode ser desenhada no seu trabalho, agora!!



???

Como faço isso

De muitas maneiras!!!

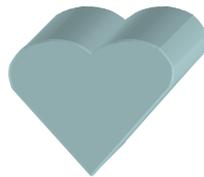
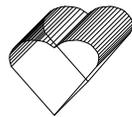
Mas vamos ficar com as de graus 3 :

$$\vec{P}(t) = (1-t)^3 \vec{V}_0 + 3(1-t)^2 t \vec{V}_1 + 3(1-t)t^2 \vec{V}_2 + t^3 \vec{V}_3$$

e iniciar com seus pontos de vértices limitando o *convex hull*,
mas pode ir incluindo novos pontos de controle a vontade .

Até seu objeto 2D ficar como você achar adequado!

Quando isso tiver legal transforme ele em 3D usando a
mesma técnica que transformou seu objeto anterior de 2D
para 3D.



Finalmente, renderize ele usando um shading suave....