http://computacaografica.ic.uff.br/conteudocap3.html

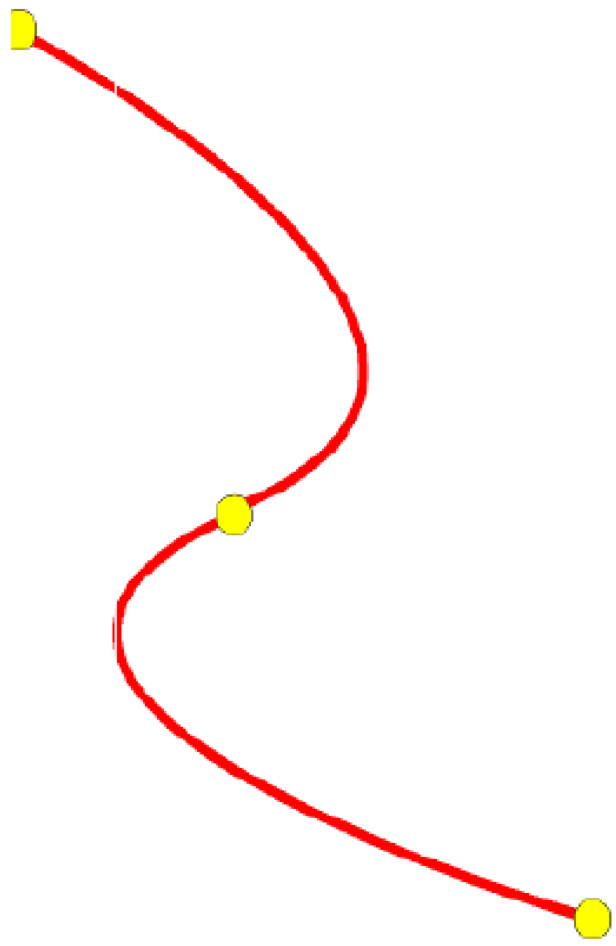
aula 16

IC/UFF – 2018 - 2

Curvas

Curvas são Elementos 1D

- Comprimento
- Distância ao inicio define a posição na curva
- Mas ela pode ser 2D e 3D



Curvas

- Formas de representação:
  - Procedural (não tem equação apenas algoritmo de geração:
    - exemplo *curvas fractais*)
  - Conjunto de pontos (digitalizados: x_i , y_i)
  - Por equações (analíticas):
    - Explicita : $y = f(x)$
    - Implícita : $x+y=0$
    - Paramétrica : $x= f(t)$, $y = f(t)$

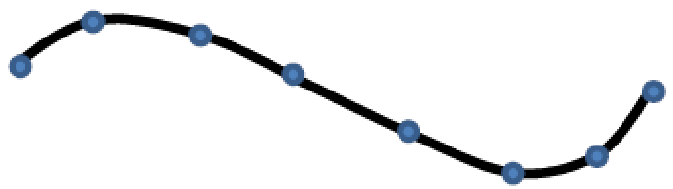
Também podem ser

Classificadas de acordo com seus termos: linear (grau 1), quadrática (grau 2), cúbica (grau 3), transcendental (sin, cos, log, ...)

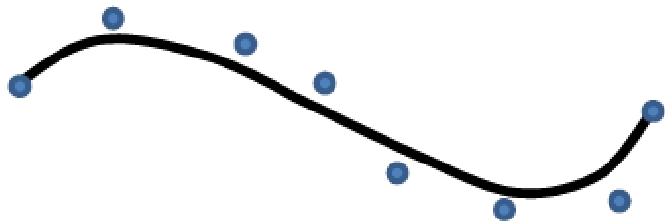
Representação analítica

- Não paramétrica e paramétrica
- Precisão sobre representação de ponto
- Armazenamento compacto
  - Centro do círculo e raio vs. pontos
- Ponto intermediário
  - Quaisquer pontos sobre a curva podem ser calculados
- Mais fácil gerar desenhos
- Mais fácil mudar a curvatura

Interpolação X Aproximação



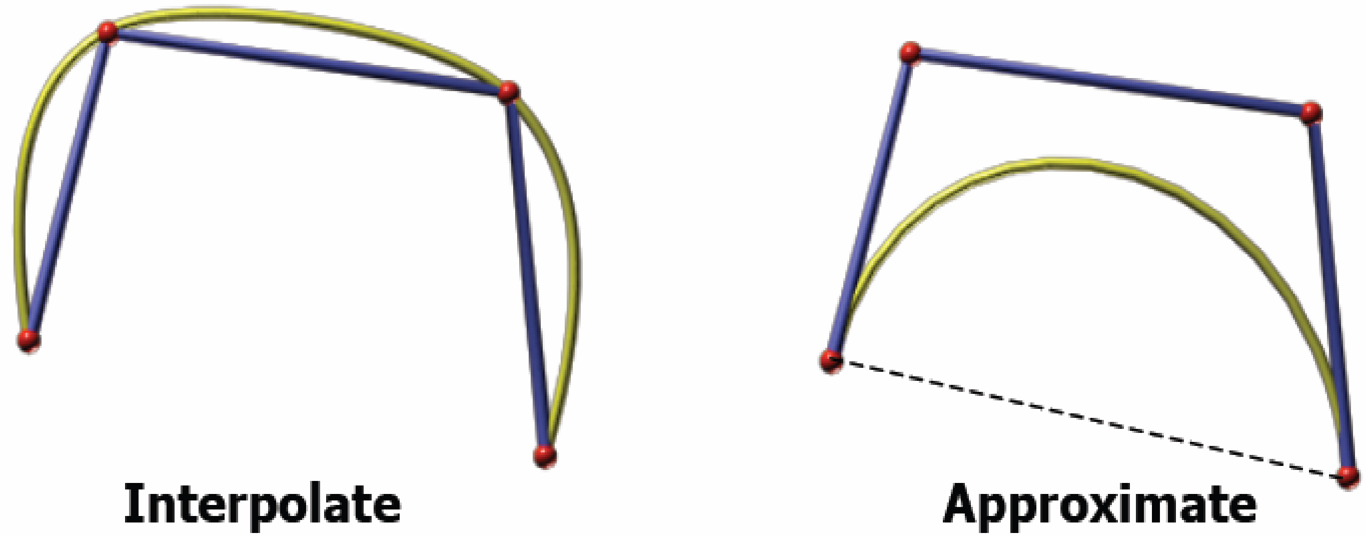
Na interpolação, a curva passa sobre todos os pontos definidos.



Na aproximação, a curva começa sobre o ponto inicial e termina sobre o final. Os demais pontos são aproximados.

Dado um número *n* de pontos para traçar uma curva:

- ***interpolar*** os pontos (curva passa *necessariamente* por todos os pontos)
- ***aproximar*** os pontos (pontos definem cobertura convexa (*convex hull)* da curva)

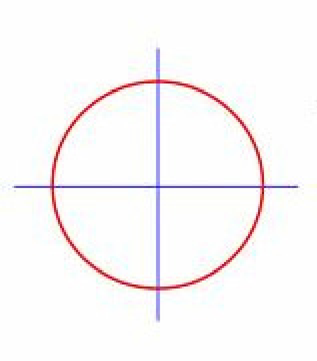


Não paramétrica vs paramétrica

- Não paramétrica
  - Explícita y = f(x)
  - Implícita f(x, y) = 0
- Equação implícita de segundo grau geral

$$ax^2 + b2xy + cy^2 + 2dx + 2cy + f = 0$$

Exemplo circunferência representações não paramétricas



Explícita y = f(x)

$$y = \sqrt{1 - x^2}$$

$$y = -\sqrt{1 - x^2}$$

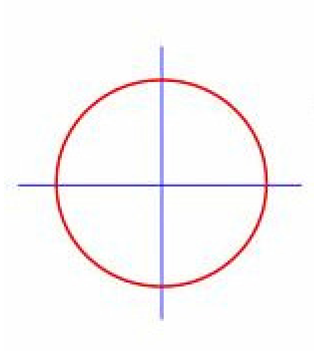
$$x^2 + y^2 - 1 = 0 \text{ or } x^2 + y^2 = 1$$

Implícita f(x, y) = 0

Exemplo : circunferência em representações paramétricas

$$f : [0, 2\pi] \rightarrow \mathbb{R}^2,\ f(\theta) = (x(\theta), y(\theta))$$

$$\begin{cases} x = \sin\ \theta \\ y = \cos \theta \end{cases} \qquad \text{where } \theta \in [0, 2\pi]$$



$$\begin{cases} x = \dfrac{2t}{1+t^2} \\ y = \dfrac{1-t^2}{1+t^2} \end{cases} \qquad \text{where } t \in [0, 1]$$

Representação implícita

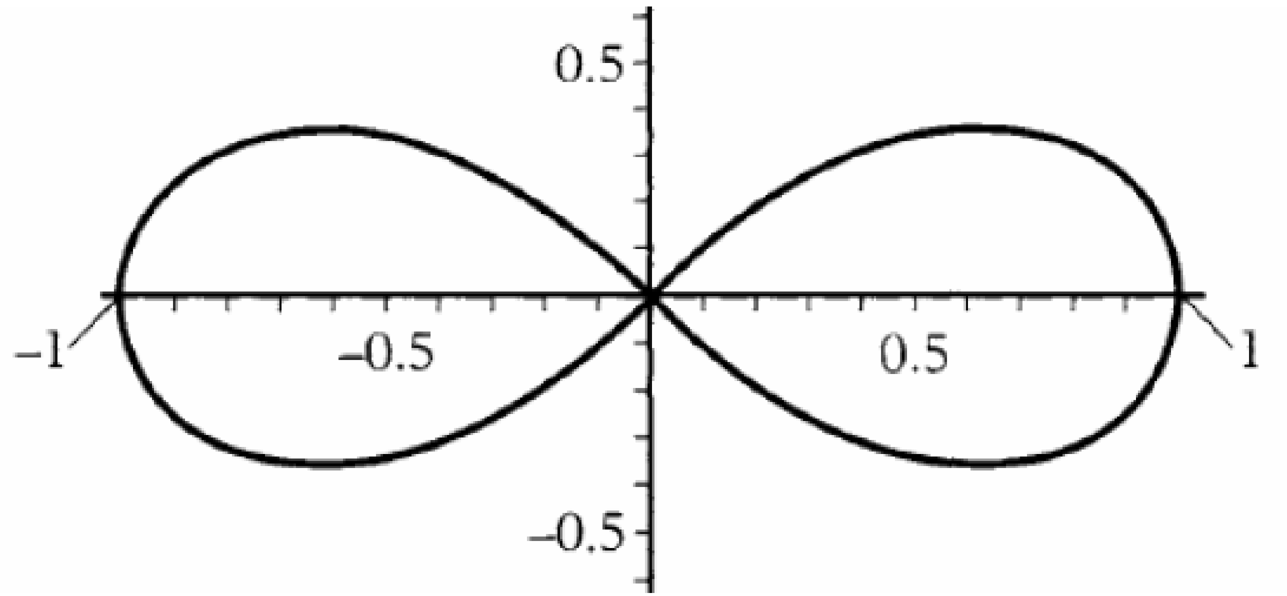
- Curva em 2D: $f(x,y) = 0$
  - Linha: $ax + by + c = 0$
  - Círculo: $x^2 + y^2 - r^2 = 0$
- Superfície em 3D: $f(x,y,z) = 0$
  - Plano: $ax + by + cz + d = 0$
  - Esfera: $x^2 + y^2 + z^2 - r^2 = 0$

Outros exemplos:

- Lemniscata de Bernoulli => símbolo infinito
- Quarto grau!

$(x^2+y^2)^2 - (x^2-y^2)^2 = 0$

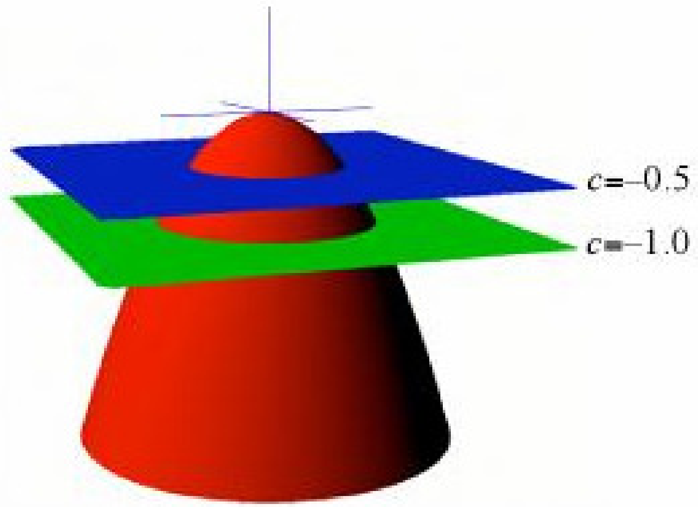
Implícita $f(x, y) = 0$



Curvas não paramétricas

- Linha
- Círculo
- Parábola
- Elipse
- Hipérbole

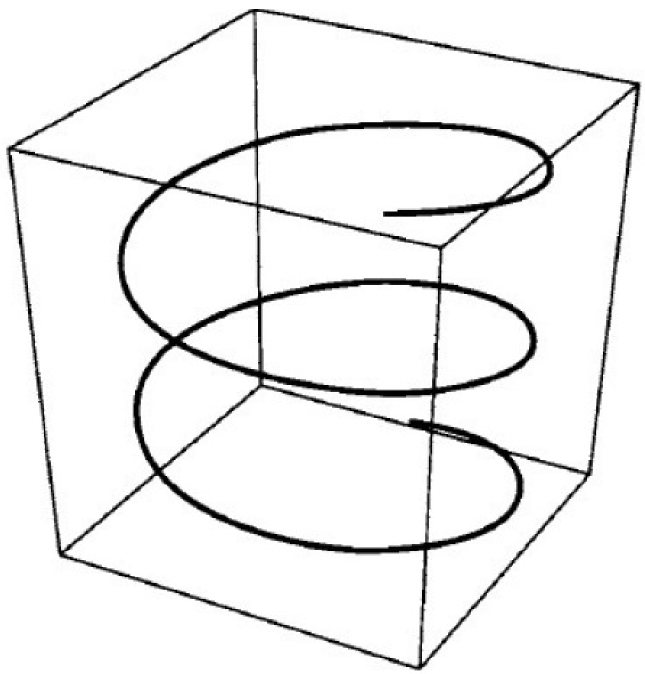
ı. The equation $-z = x^2 + y^2$ *explicitly* defines the paraboloid in $\mathbb{R}^3$.



Curvas paramétricas

– Pontos sobre uma curva são representados com uma função de um único parâmetro
  - $x = f(u)$, $y = g(u)$, $z = h(u)$
    – u : variável paramétrica

$x = \cos(t)$, $y = \sin(t)$, $z = t/5$

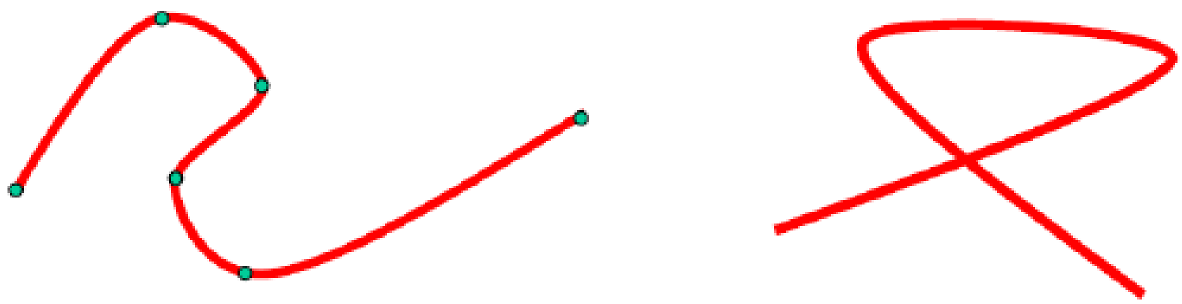


Peculiaridades das curvas em CG

Principais desvantagens das representações **não-paramétrica** em CG

- É difícil definir a equação não paramétrica de uma curva que passe por um conjunto de pontos pré-definidos.

- Não permite a representação de curvas com laços



Peculiaridades das curvas em CG

Para CG, representações paramétricas costumam ser as mais convenientes

Assim, genericamente, uma curva 3D é

– $Q(t)=[x(t)\ \ y(t)\ \ z(t)]$

$$P(u) = (X(u), Y(u), Z(u))$$
$$0 \leq u \leq 1$$

- As expressões paramétricas suportam declives infinitos, curvas fechadas ou multi-valor.

dy/dx = (dy/du) / (dx/du)

dy/dx = infinito => dx/du = 0

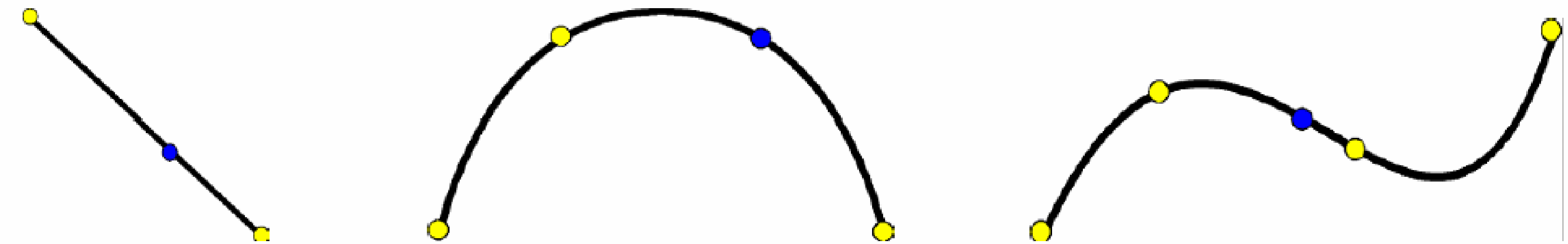
Reta na forma paramétrica

$$P(t) = P_0 + at$$

- $P_x = P_{x0} + at$
- $P_y = P_{y0} + at$
- $P_z = P_{z0} + at$

Parametrizando polinômios

$f(t) = at + b$ $\qquad$ $f(t) = at^2 + bt + c$ $\qquad$ $f(t) = at^3 + bt^2 + ct + d$



Linear $\qquad$ Quadrático $\qquad$ Cúbico

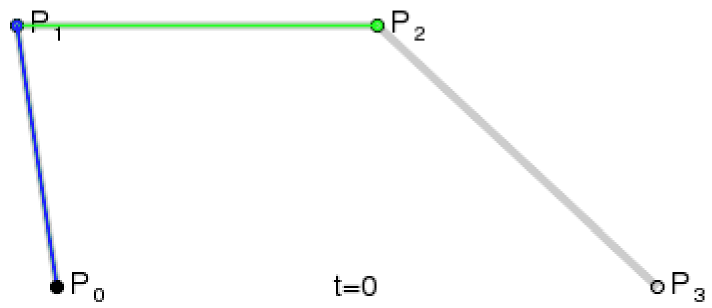
- Elementos geométricos definidos parametricamente são inerentemente limitados (0 <= u <= 1).

- As expressões paramétricas são facilmente traduzidas na forma de vectores e matrizes.

- Utilização de um só modelo matemático para representar qualquer curva ou superfície.

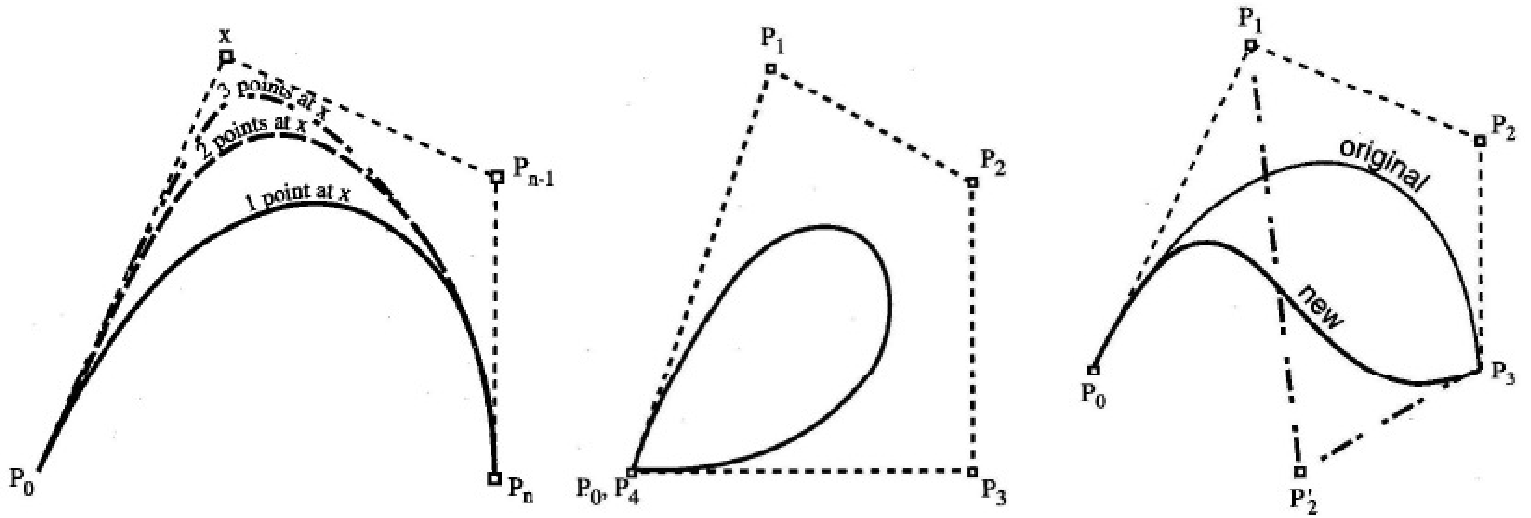
https://pt.wikipedia.org/wiki/Curva_de_Bezier

Curvas de Bezier



Curvas 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

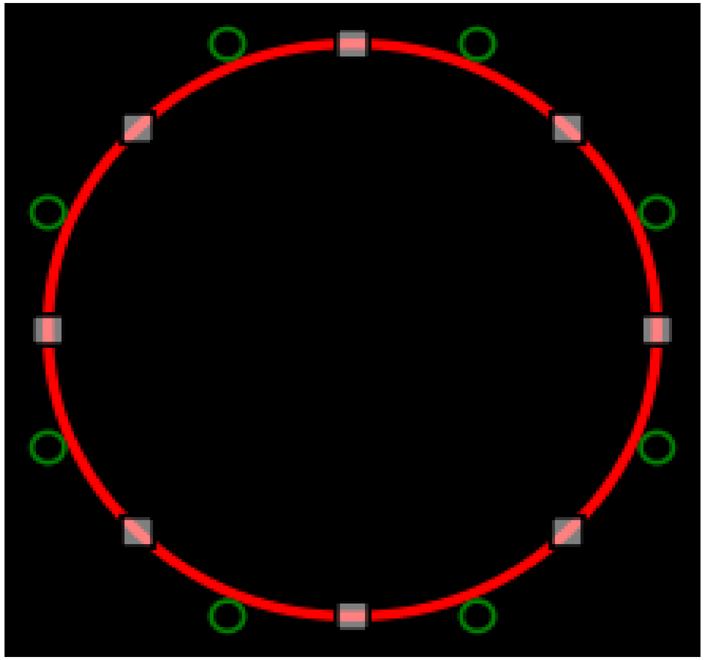


pontos de controle = P_i

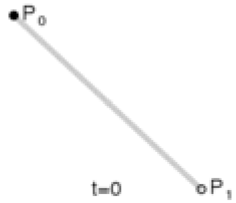
Onde se usa: Qualquer representação de curvas

Até onde você nem imaginar!
Por exemplo:
Os contornos dos caracteres em fontes TrueType são feitas de segmentos de retas e curvas Bézier quadrática.

O circulo ao lado é formado por 8 segmentos. Os quadrados são os pontos de controle da extremidade e os anéis os pontos de controle internos.

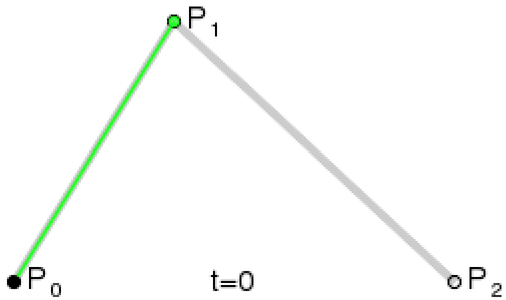


Um segmento linear pode ser definido como curva:



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

Curva polinomial desenvolvida em 1962 por Pierre Bézier.

Utilizada no projeto de automóveis (Renault).

Baseada no algoritmo de De Casteljau em 1957.

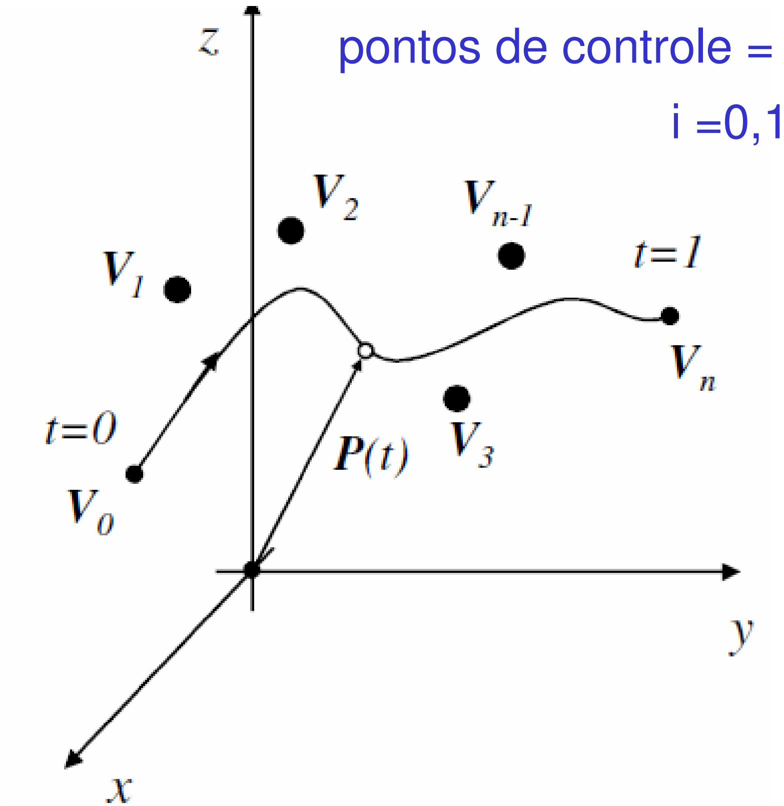
Curva de aproximação.

Controle global.

P. de Casteljau, 1959 (Citroën)
P. de Bézier, 1962 (Renault) - UNISURF
Forest 1970: Polinômios de Bernstein

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

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



pontos de controle = *Vi*

i =0,1,2 ,

onde:

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

pol. Bernstein

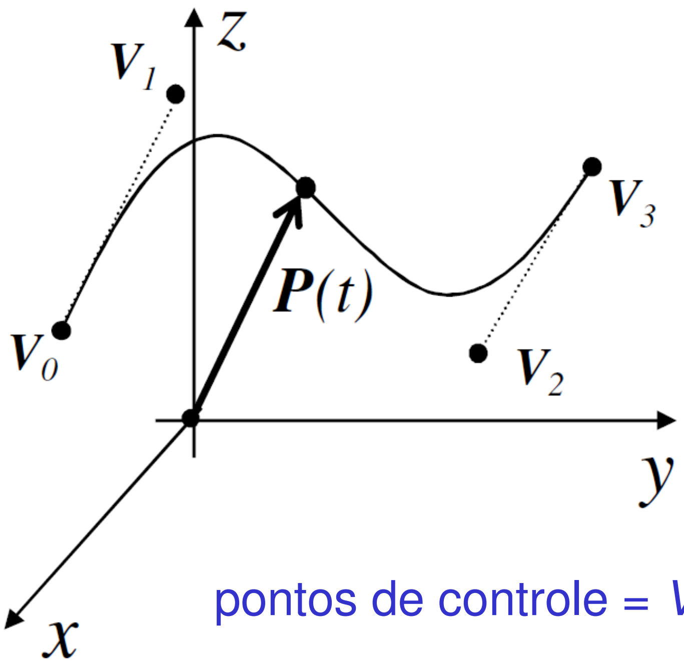
$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

coef. binomial

Fatorial de um numero = ***n ! = n (n-1) 1***

Bezier cúbica, n=3:

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



pontos de controle = *Vi*

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

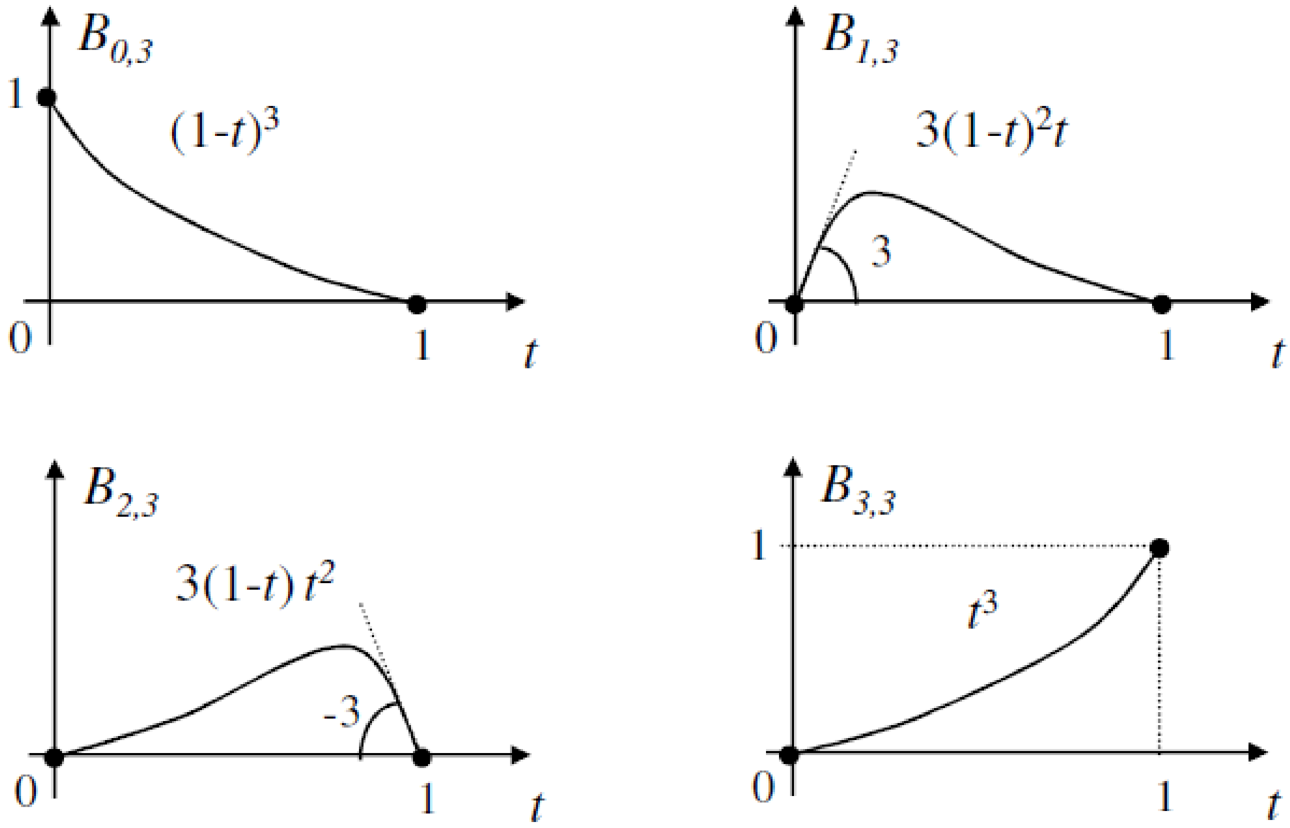
$$B_{1,3}(t) = \binom{3}{1}(1-t)^{3-1}\,t^1 = 3(1-t)^2 t$$

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

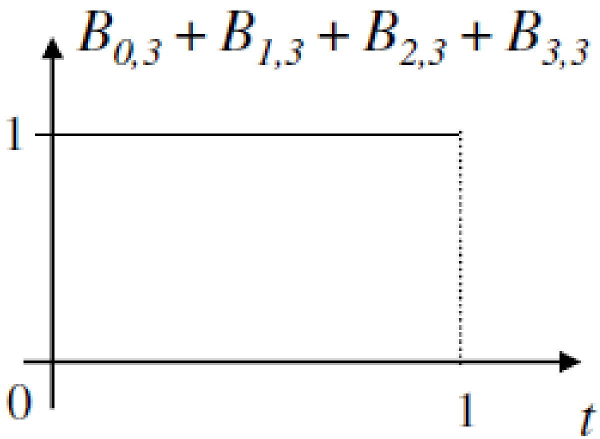
$$B_{3,3}(t) = \binom{3}{3}(1-t)^{3-3}\,t^3 = t^3$$

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

Polinômios cúbicos de **Bernstein**

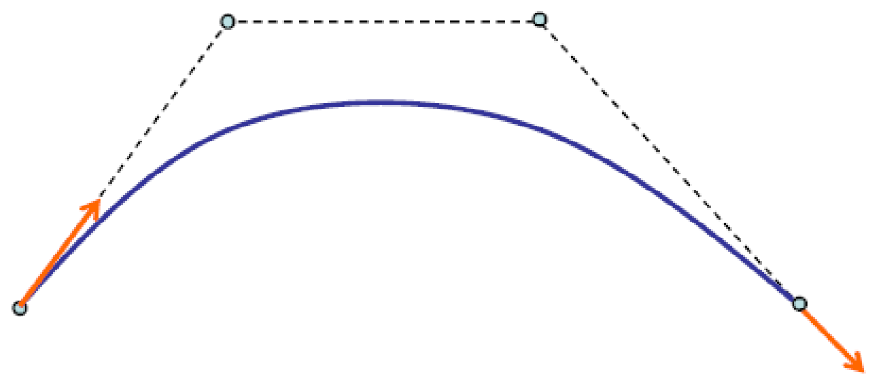


A soma dos **Polinômios Cúbicos de Bernstein** resulta:



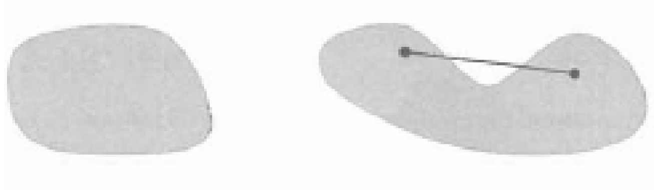
Propriedades de Curva de Bézier

- 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



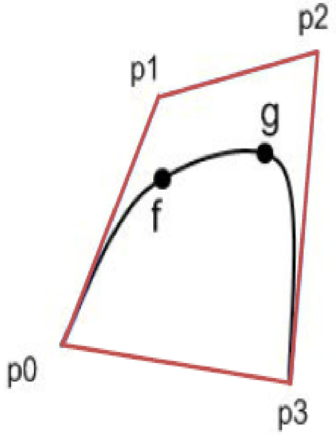
Fecho convexo?

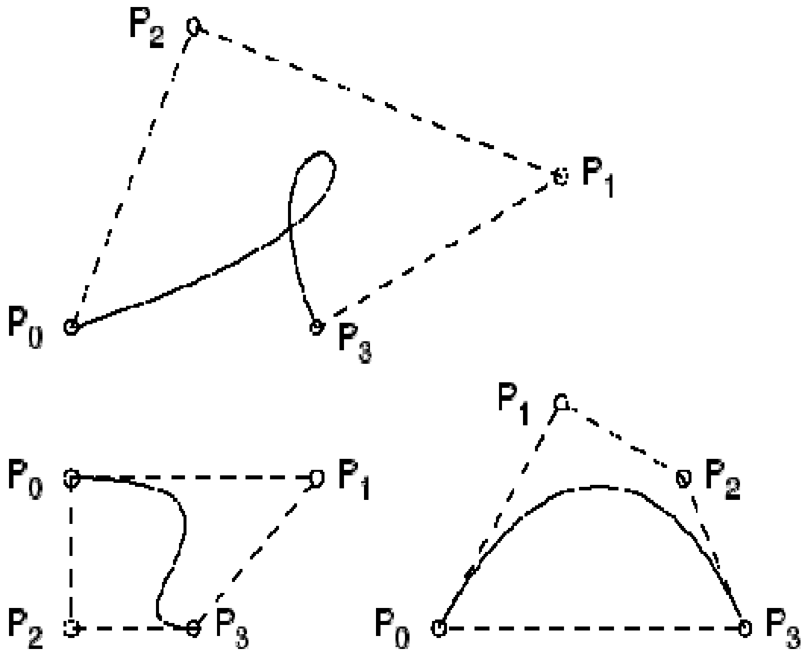
- Já falamos de objetos convexos em aulas passadas, não?
- O que você acha então que seria um fecho convexo??



Propriedade: *Convex Hull*

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





pontos de controle = *Pi*

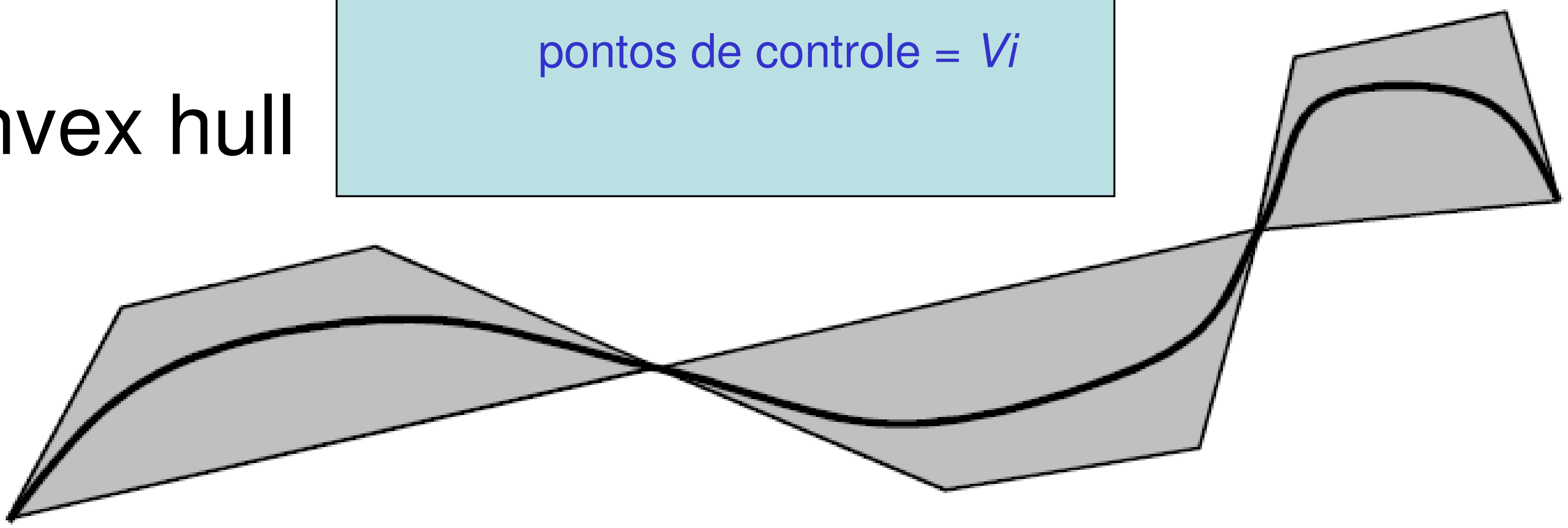
Cont.

- 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

E para uma curva única formada por muitas curvas como fica o Fecho convexo?

pontos de controle = *Vi*

Convex hull



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

Se fosse pedido para re-parametrizar de forma especial

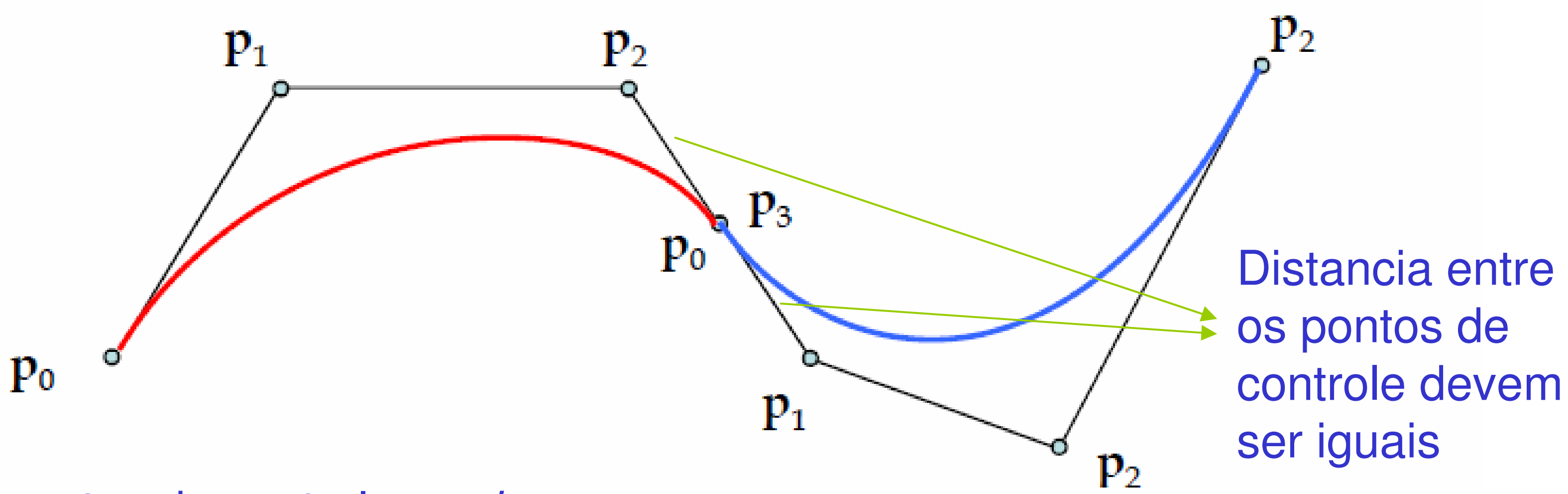
- por exemplo com mais pontos onde a derivada da curva maior sempre suave ?

- Com as expressões do slide anterior isso poderia ser feito fazendo com que os valores do parâmetro continuassem entre as curvas e o primeiro iniciasse em 0 e o ultimo finalizasse em 1!

- (simples não??)

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 que o segmento $\mathbf{p}_0\mathbf{p}_1$ da segunda
- Continuidade C^2: C^1 e + restrições sobre pontos $\mathbf{p}_1$ da segunda e $\mathbf{p}_2$ da primeira

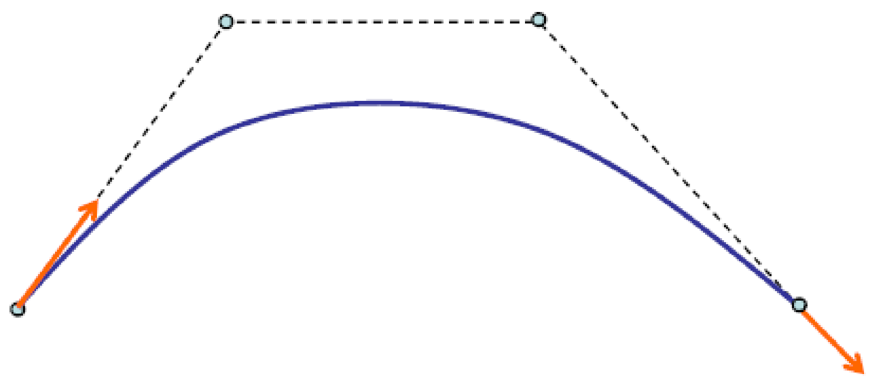


pontos de controle = *pi*

Cont.

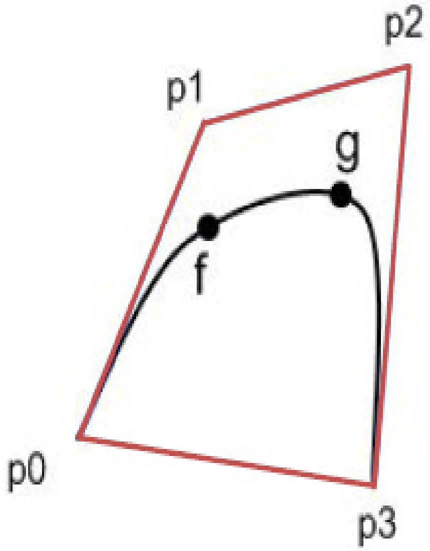
pontos de controle = *Pi*

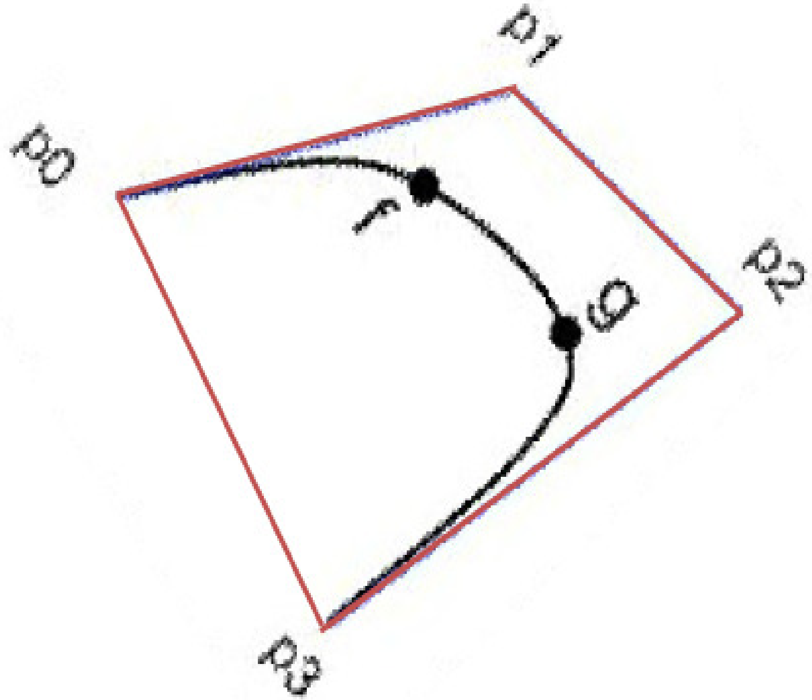
- As tangentes à curva em $\mathbf{p}_0$ e $\mathbf{p}_n$ têm a direção dos segmentos de reta $\mathbf{p}_0\mathbf{p}_1$ e $\mathbf{p}_{n-1}\mathbf{p}_n$, respectivamente
  - Para cúbicas, as derivadas são $3(\mathbf{p}_1 - \mathbf{p}_0)$ e $3(\mathbf{p}_2 - \mathbf{p}_3)$
- Transformar os pontos de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



Transformações

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





pontos de controle = *pi*

Demonstrando essas propriedades para uma Bezier cúbica:

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

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

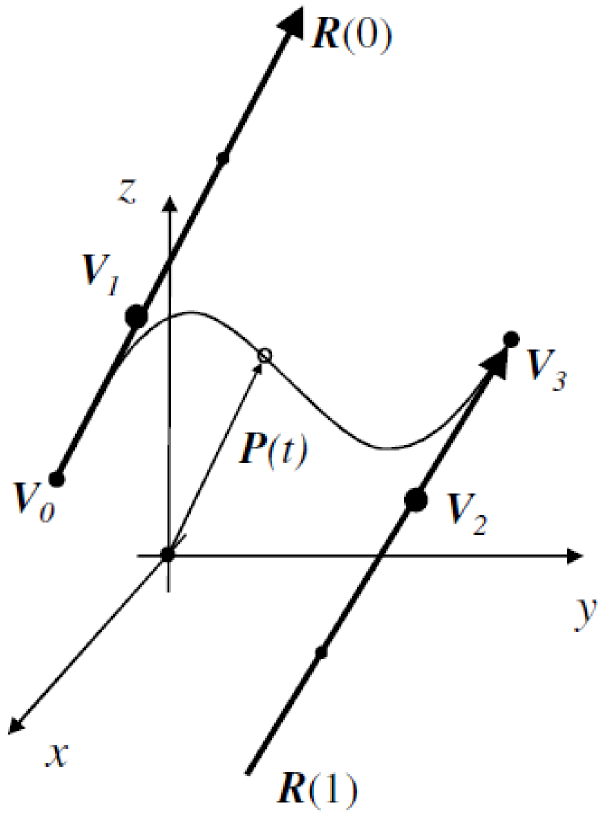
pontos de controle = *Vi*

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

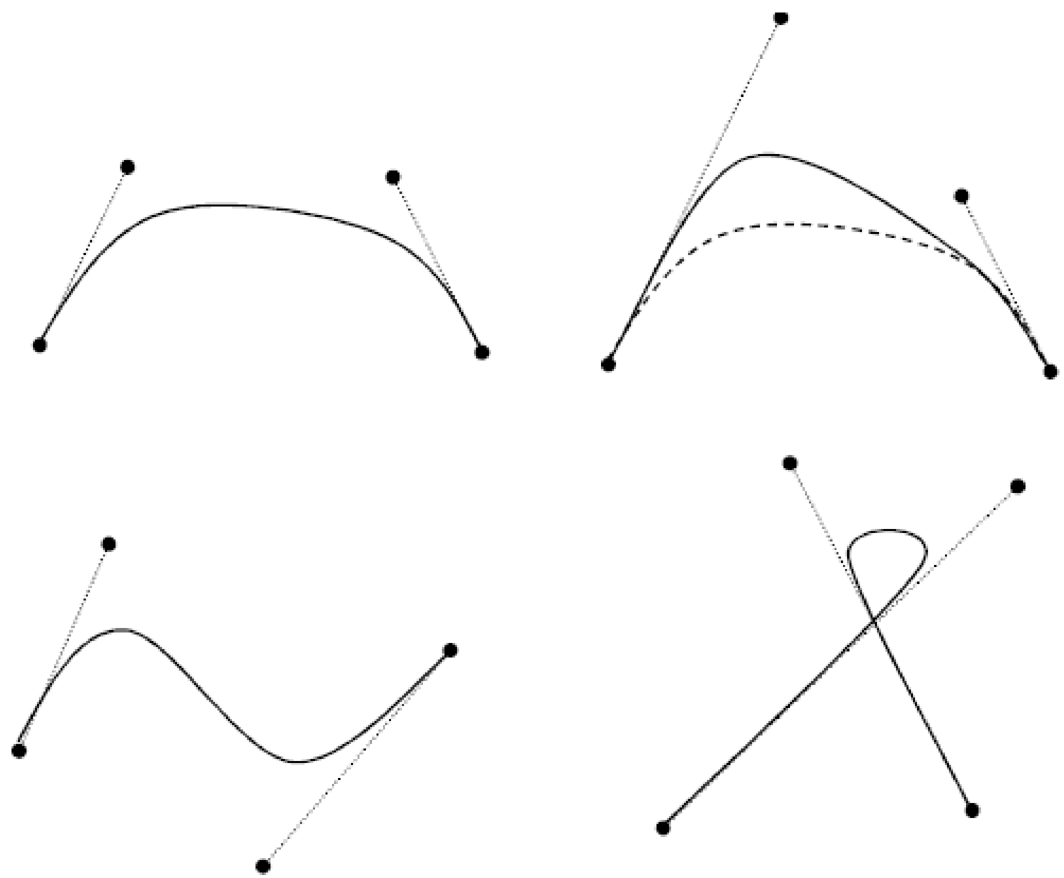
$$\vec{P}(1) = \vec{V}_3$$

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

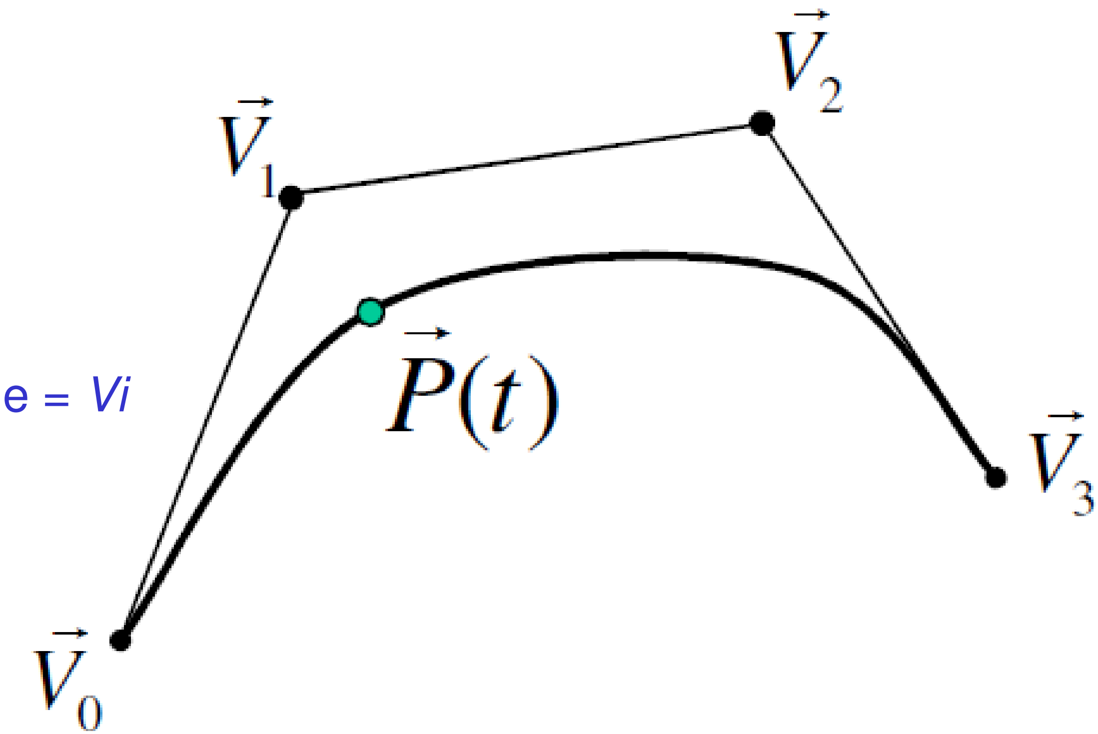
$$\frac{d}{dt}\vec{P}(1) = -3\,\vec{V}_2 + 3\,\vec{V}_3$$



A ordem e posição dos pontos controla a forma final da curva!



Representações matriciais :



pontos de controle = *Vi*

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

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

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

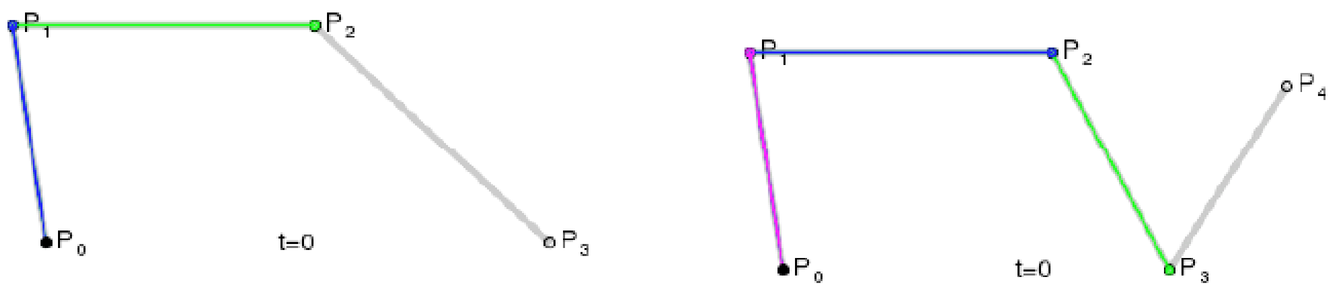
$$Q(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1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

$$Q(t) = TMG$$

pontos de controle = *Gi*

Outras formas de Bezier

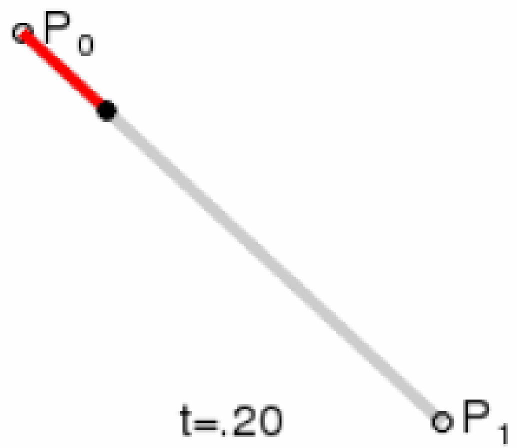
pontos de controle = *Pi*



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

Outras formas de Bezier

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

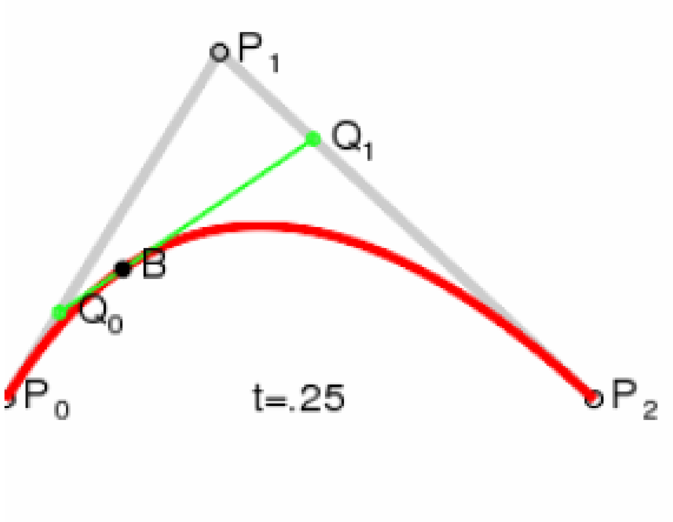


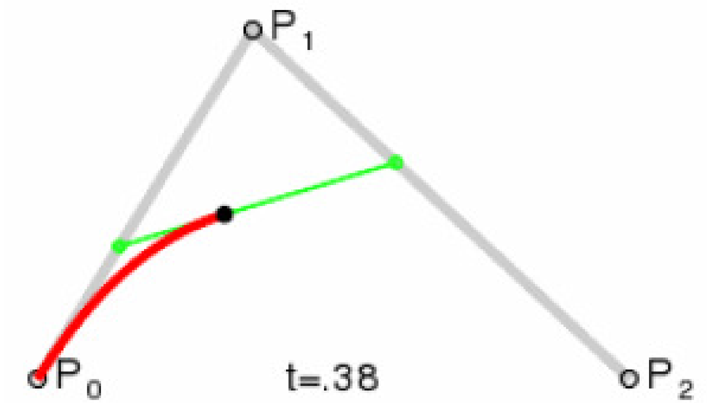
pontos de controle = *Pi*

Outras formas de Bezier

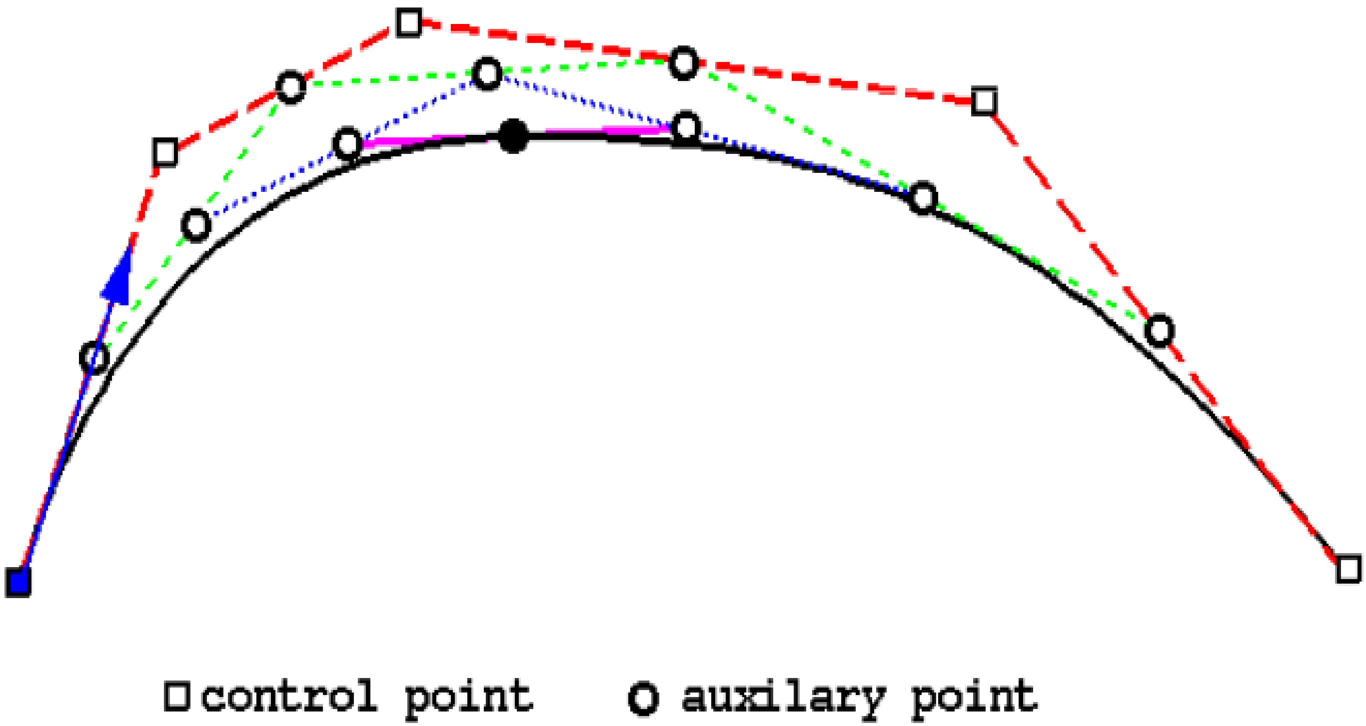
$$P(t) = (1-t)^2 \mathbf{P}_0 + 2\,t\,(1-t)\,\mathbf{P}_1 + t^2\,\mathbf{P}_2$$

pontos de controle = *Pi*





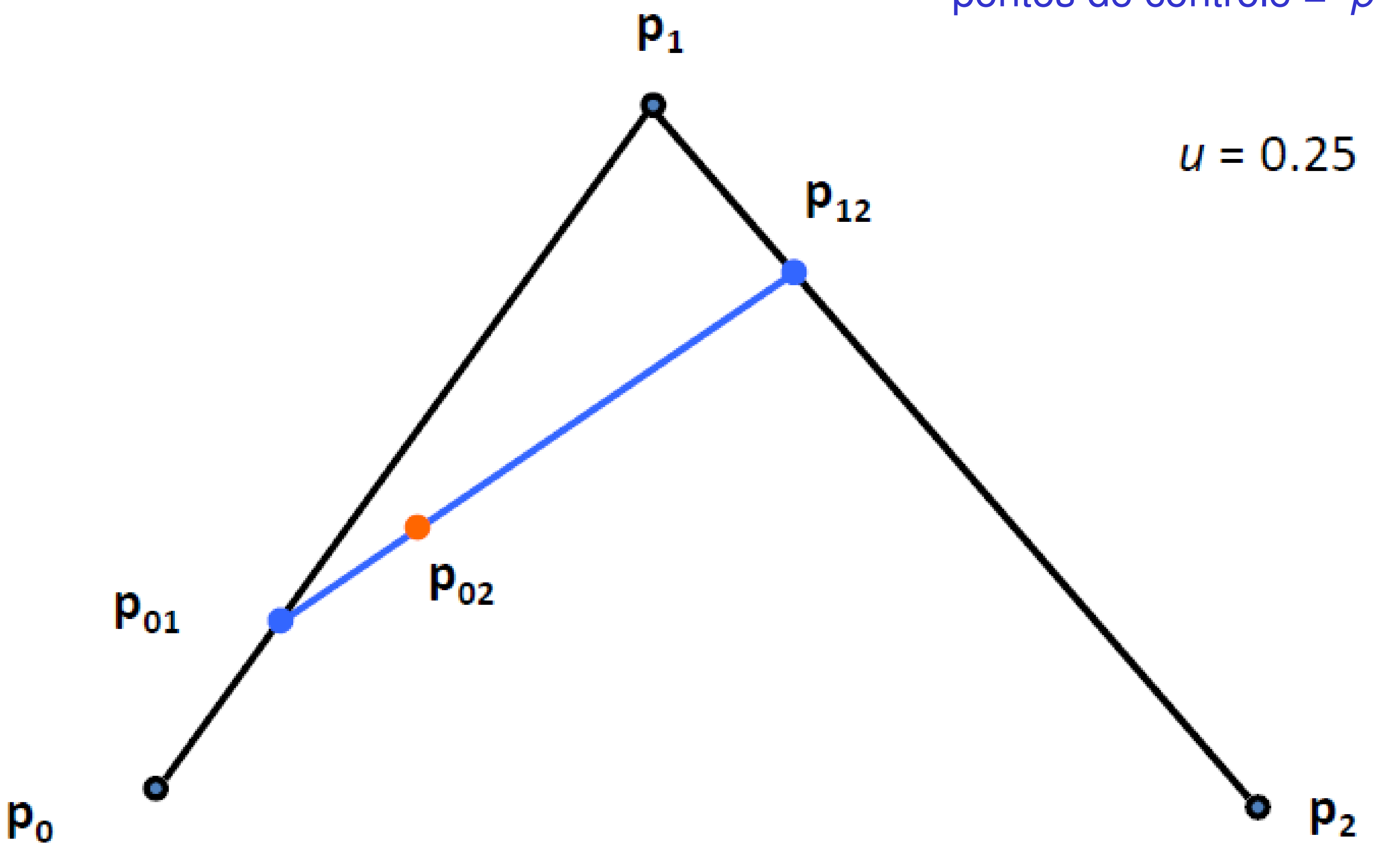
algoritmo geométrico para construção de curvas Bézier.

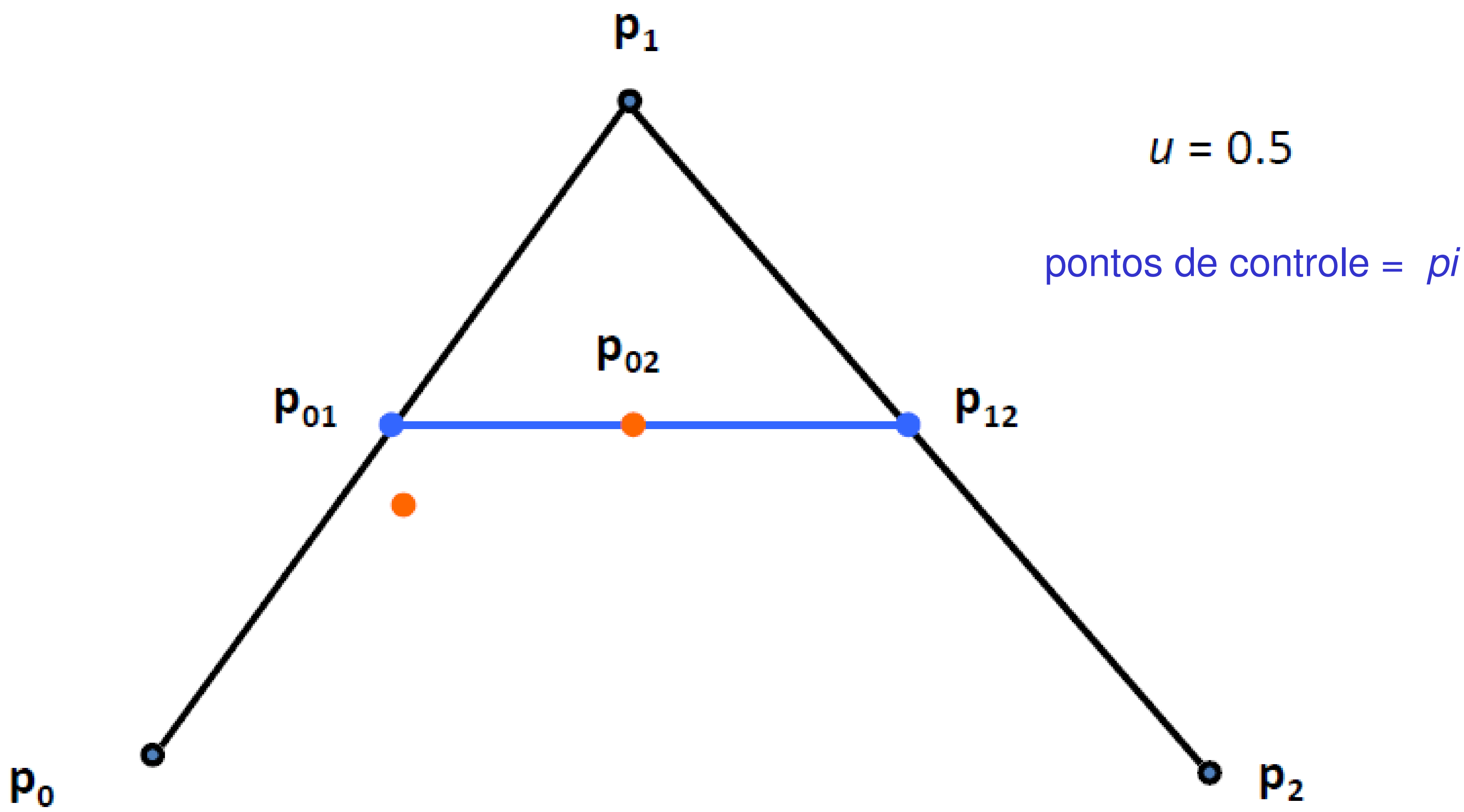


Algoritmo geométrico

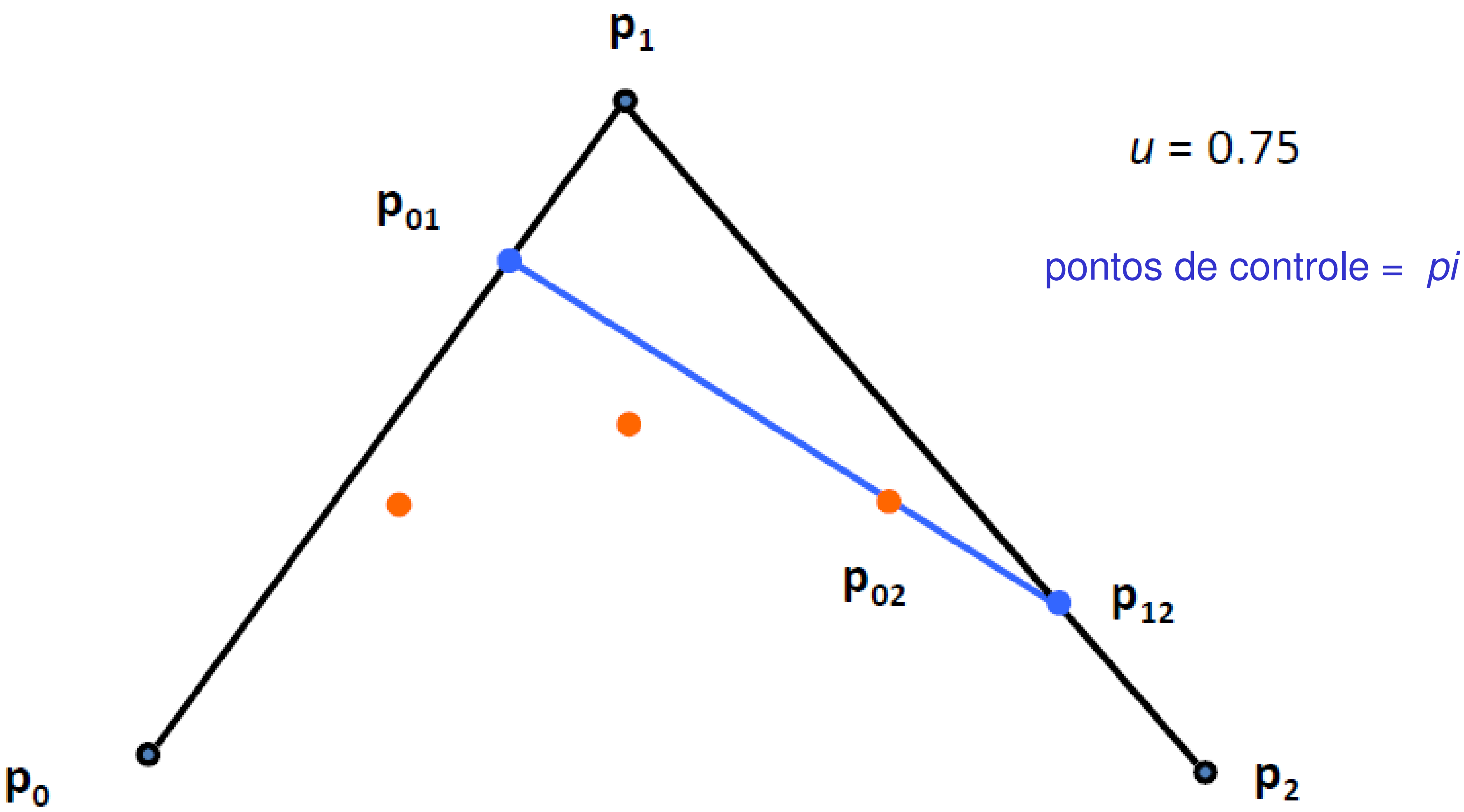
pontos de controle = *pi*

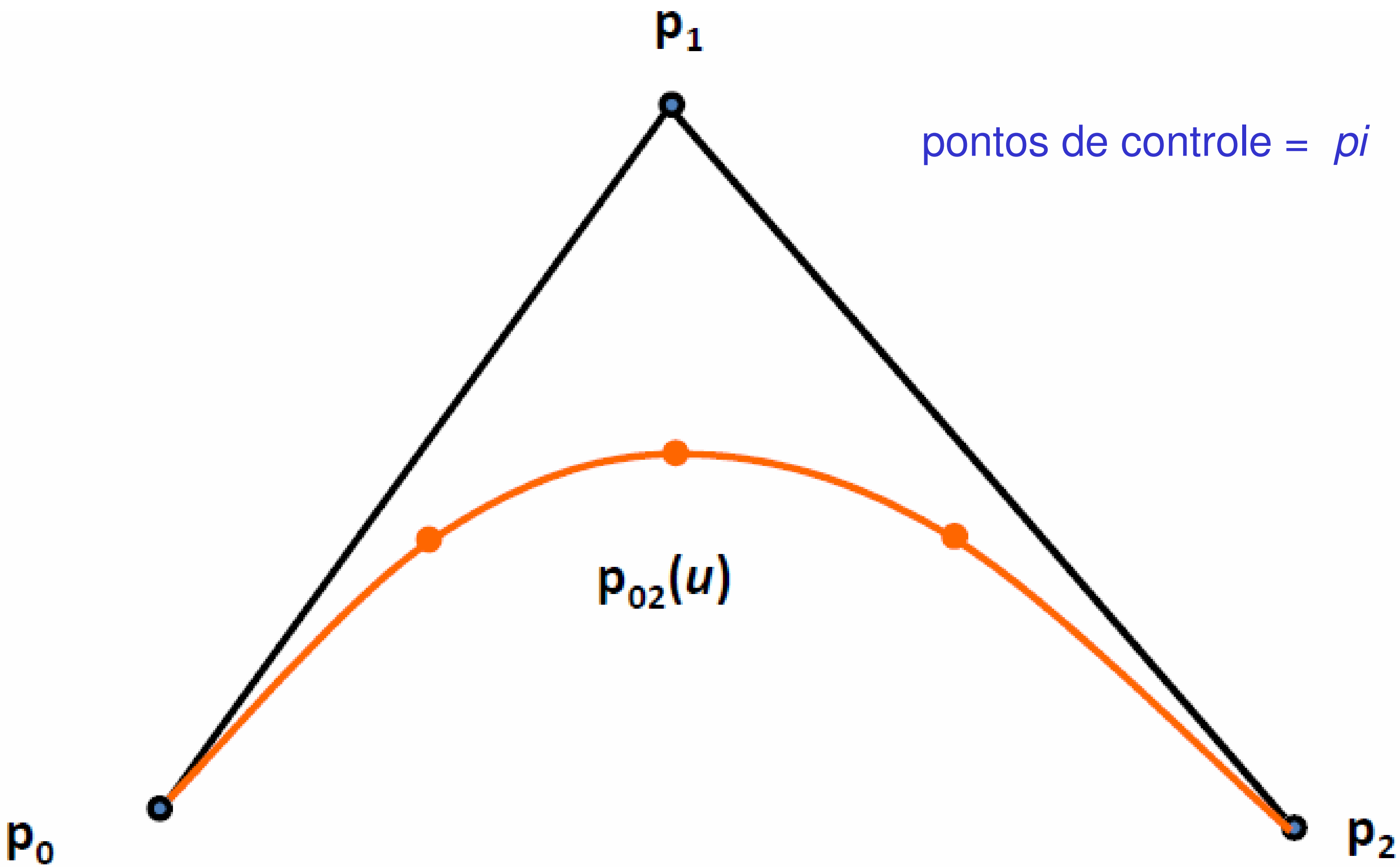
$u = 0.25$





p_1
$u = 0.5$
pontos de controle = *pi*
p_{02}
p_{01}
p_{12}
p_0
p_2

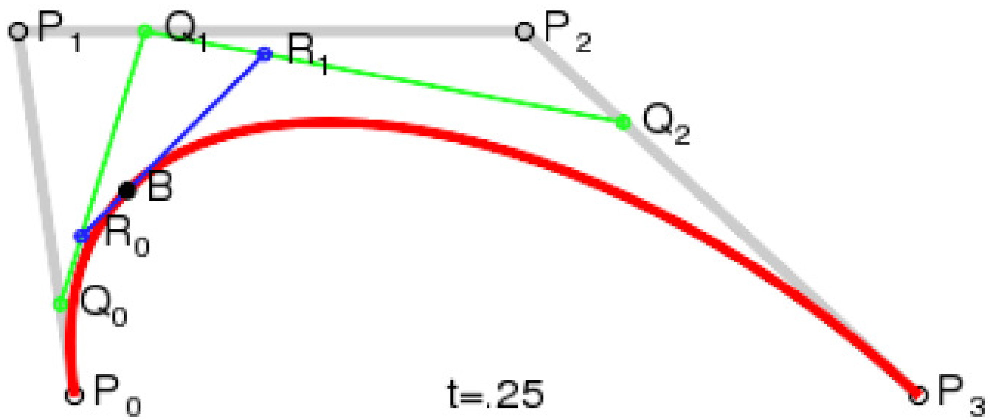

p1
u = 0.75
p01
pontos de controle = pi
p02
p12
p0
p2



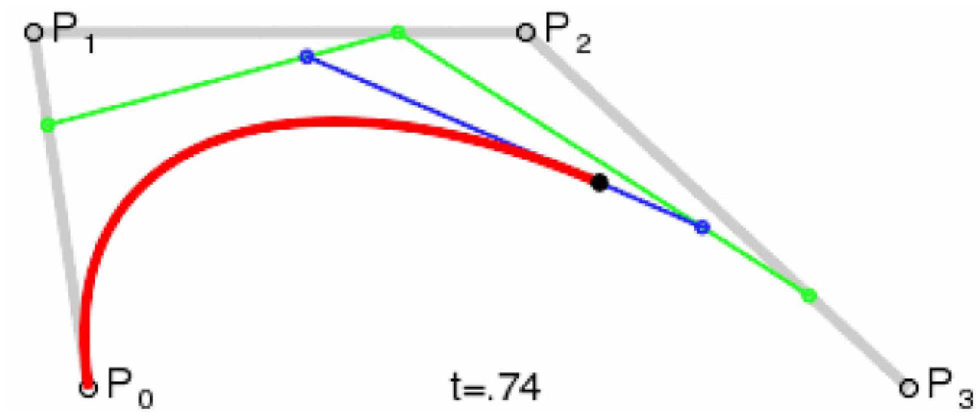
pontos de controle = *pi*

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 = *Pi*



O usuário

- Definirá os pontos iniciais finais e os intermediários nas coordenadas dele
- A curva de Bezier pode ser desenhada!!

Socorro!!

Como faço isso???

De muitas maneiras!!!

Por exemplo:

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

Ou as matrizes já descritas.