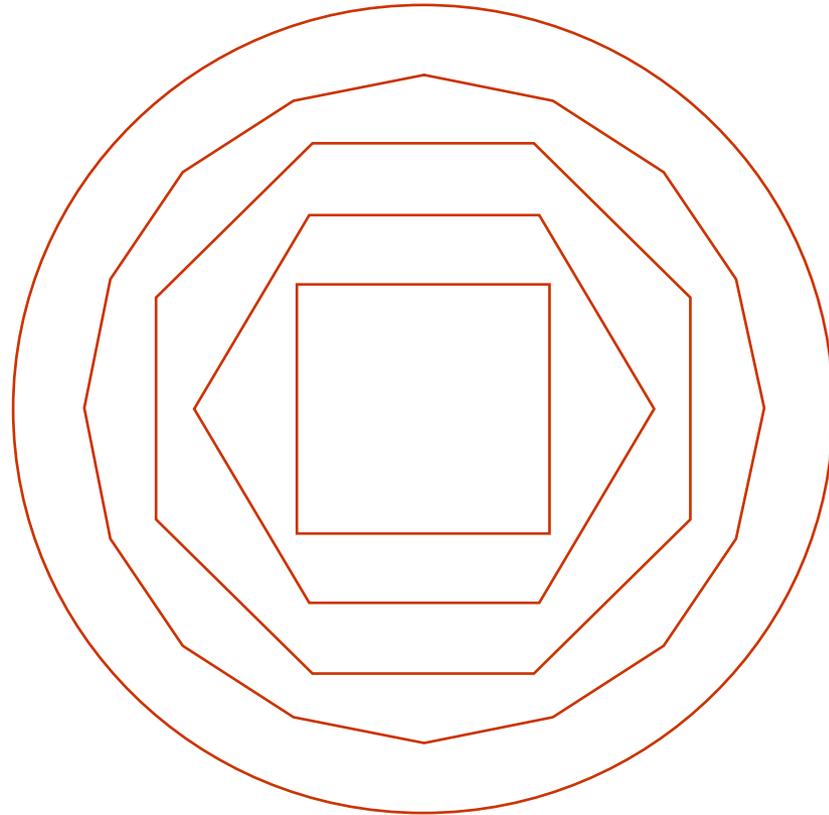


Rasterização de linhas curvas genéricas



Embora haja algoritmos para casos particulares, de uma forma geral bastará encontrar uma aproximação razoável por segmentos de recta e executar o algoritmo de rasterização destes para cada um.

Curva

Especificada por uma ou mais equações com **uma** só variável independente.

◆ Descrição não-paramétrica

- Forma explícita

$$Q = (x , y=f(x) , z=g(x))$$

- Forma implícita

$$Q = (F(x,y,z)=0 , G(x,y,z)=0)$$

◆ Descrição paramétrica

$$Q = (x=f(t) , y=g(t) , z=h(t))$$

$$a \leq t \leq b$$

Superfície

Especificada por uma ou mais equações com **duas** variáveis independentes.

◆ Descrição não-paramétrica

- Forma explícita

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

- Forma implícita

$$Q = (F(x,y,z)=0)$$

◆ Descrição paramétrica

$$Q = (x=f(s,t) , y=g(s,t) , z=h(s,t))$$

$$a \leq t \leq b$$

$$c \leq s \leq d$$

Curvas e Superfícies

◆ Descrição não-paramétrica

- Forma explícita

- 😊 Facilidade de cálculo

- 😞 Não pode representar uma correspondência que não seja função

- 😞 Não se podem aplicar directamente transformações por operadores matriciais

- ◆ Exemplo:

$$y = \sqrt{4 - x^2} \quad \text{e} \quad y = -\sqrt{4 - x^2}$$

$$\text{com } -2 \leq x \leq 2$$

Curvas e Superfícies

◆ Descrição não-paramétrica

● Forma implícita

- 😊 Representação de correspondências que não sejam funções
- 😞 Pode ser difícil a determinação das raízes
- 😞 Não se podem aplicar directamente transformações por operadores matriciais

◆ Exemplo:

$$x^2 + y^2 - 4 = 0$$

com $-2 \leq x \leq 2$

Curvas e Superfícies

◆ Descrição paramétrica

- 😊 Representação de correspondências que não sejam funções
- 😊 Podem aplicar-se directamente transformações por operadores matriciais

◆ Exemplo:

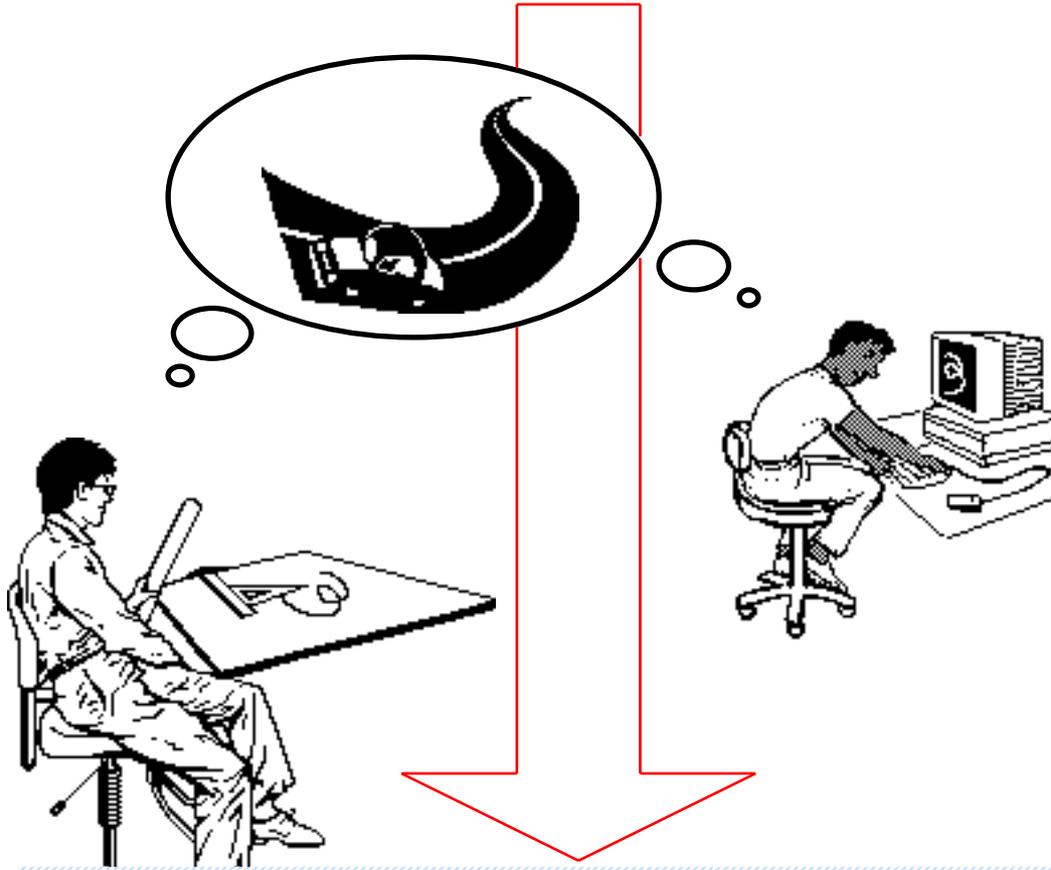
$$\begin{aligned}x &= 2 \cos t \\y &= 2 \sin t\end{aligned}\quad \text{com } 0 \leq t \leq 2\pi$$

ou

$$\begin{aligned}x &= 2 \cos (2\pi t) \\y &= 2 \sin (2\pi t)\end{aligned}\quad \text{com } 0 \leq t \leq 1$$

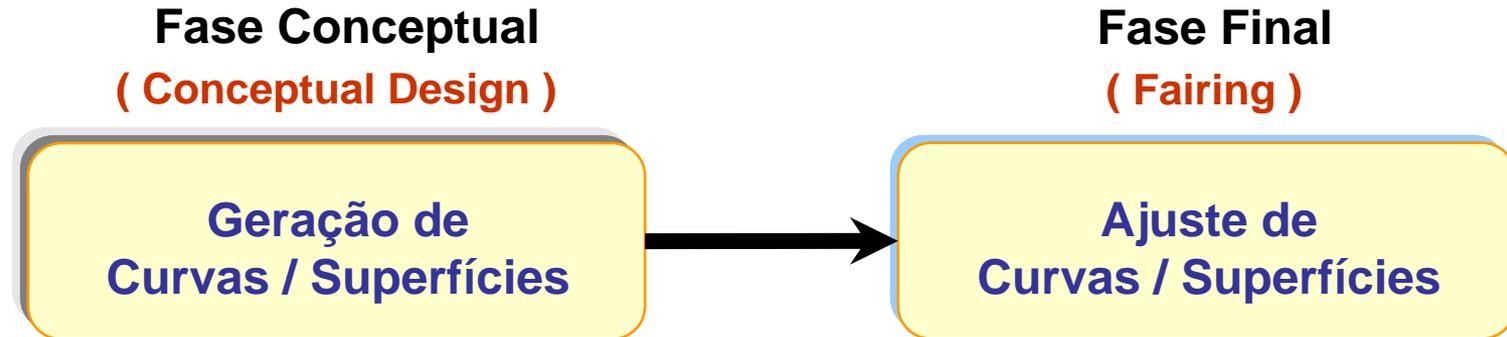
O processo de Design

Requisitos Técnicos, Requisitos Físicos, Requisitos Estéticos, etc.



Requisitos Geométricos

Fases no processo de Design

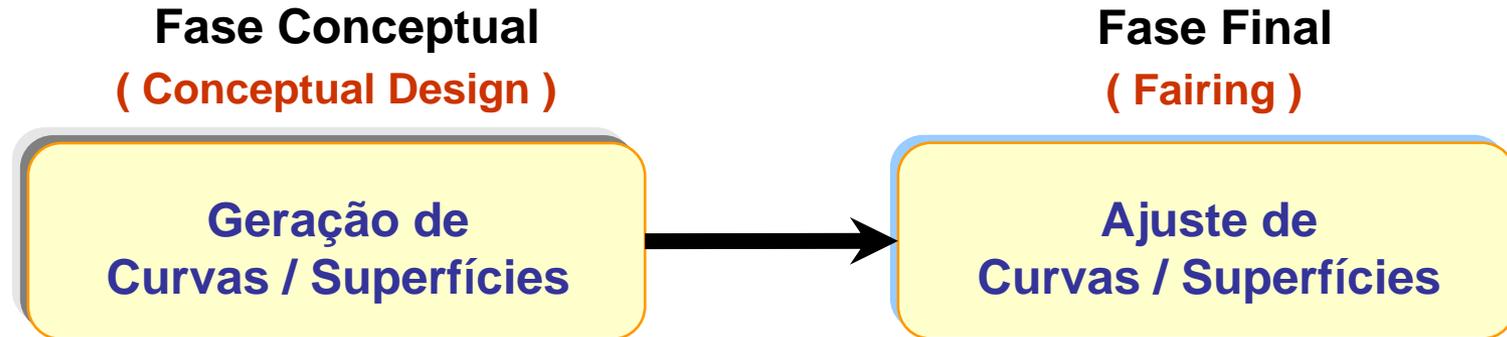


The Design Problem

Given: A set of geometric requirements [such as offsets, slopes, second derivatives, areas, volumes, centroids, ...] sufficient to characterize the intended shape.

Find: A curve (surface) meeting these requirements.

Fases no processo de Design



The Fairing Problem

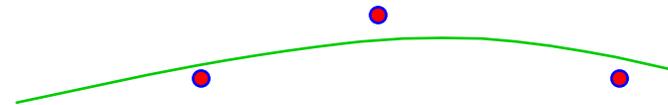
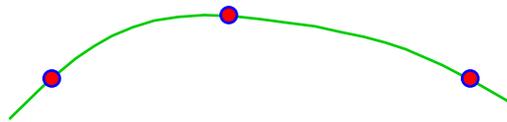
- Given:** A curve (surface), or a corresponding set of offset points, approximating an intended shape.
- Find:** A fairer curve (surface) according to some fairness criterion without deviating too far from the given shape.

What does fair mean?

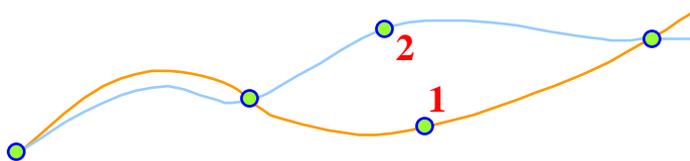
- Pleasing to the eye
- Related to the absence of unwanted deficiencies, particularly oscillations
- Spline fairness

Alguns requisitos no design de Curvas

- ◆ Interpolar ou aproximar um certo número de pontos conhecidos, obtendo-se a equação da curva



- ◆ Controlar através de pontos conhecidos e de forma previsível: **local** (preferível) ou **globalmente**

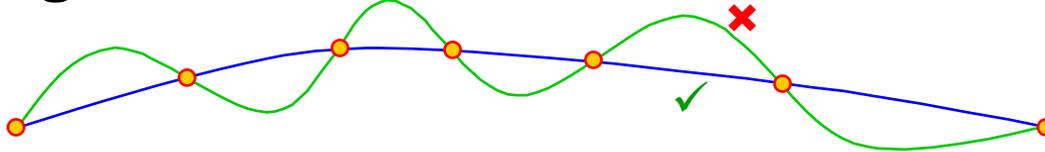


- ◆ Haver independência da forma da curva em relação ao sistema de eixos usado para representar os pontos
- ◆ Permitir correspondências que não sejam funções

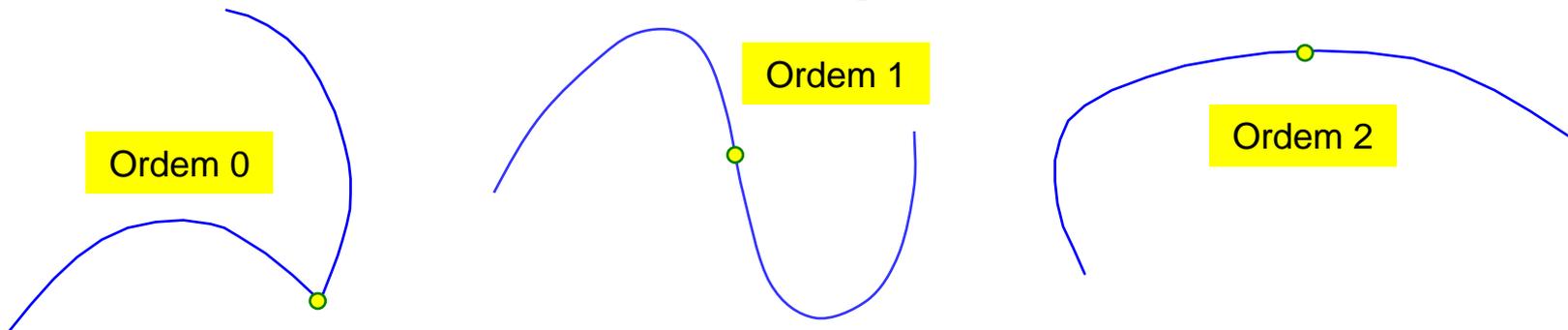


Alguns requisitos no design de Curvas

- ◆ **Existir tendência para suavizar pequenas irregularidades**



- ◆ **Disponibilizar grande versatilidade de formas**
- ◆ **Permitir a continuidade entre os troços que constituam uma curva complexa**

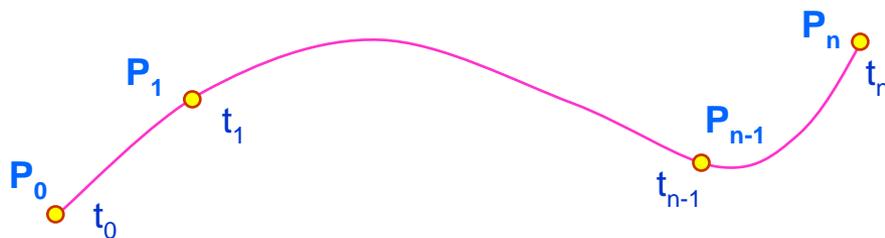


1.^a conclusão: Usar descrições paramétricas.

Curvas de Interpolação

◆ Problema:

Encontrar uma curva $Q(t)$ que passe por $n+1$ pontos P_i



$t_i \equiv$ nó (*knot*) associado a P_i

Como consequência da descrição paramétrica, a escolha dos valores dos nós influencia o andamento da curva.

● Condições do problema:

- $Q(t_i) = P_i$, $i = 0..n$
- Continuidade das funções e suas derivadas

● Resolução:

- Usar polinómios interpoladores

Interpolação de Lagrange

Existe um e um só polinómio de grau n que resolve o problema:

$$Q(t) = \sum_{i=0}^n P_i L_{in}(t)$$

em que
$$L_{in}(t) = \frac{(t - t_0) \dots (t - t_{i-1}) (t - t_{i+1}) \dots (t - t_n)}{(t_i - t_0) \dots (t_i - t_{i-1}) (t_i - t_{i+1}) \dots (t_i - t_n)}$$

Mas ...

... um polinómio de grau n tem até $n-1$ extremos relativos e $n-2$ pontos de inflexão !

Em medições precisas, quando se usam muitos pontos, o resultado poderá ser uma oscilação indesejada da curva.

Este efeito negativo poderá ser evitado ?

Não, pois basta que haja um só ponto incompatível com uma curva suave.

A maioria das aplicações não requer continuidade das derivadas de elevada ordem

A solução é a interpolação da curva por troços contínuos !



Polinómios de baixo grau



Interpolação por Splines (naturais)

A teoria dos splines trata da interpolação polinomial, na generalidade e por troços.

DEFINIÇÃO DE SPLINE :

Uma função $\mathbf{S}(t)$, escalar ou vectorial, definida no intervalo $[t_0, t_n]$, é um **SPLINE** de **ordem k** (ou **grau k-1**) se :

- $\mathbf{S}(t)$ é um polinómio de grau **k-1** em cada intervalo $[t_i, t_{i+1}]$, com $t_0 < \dots < t_i < t_{i+1} < \dots < t_n$
- $\mathbf{S}(t)$ e as suas derivadas de ordem **1..k-2** são contínuas em todo o intervalo onde é definida (isto é, $\mathbf{S}(t)$ pertence à **classe C^{k-2}**)

$k = 4$  **SPLINE CÚBICO**

DADOS: $n+1$ pontos P_i ($i=0..n$) e nós $t_i \in [t_0, t_n]$ associados

OBJECTIVO:

Encontrar os polinómios cúbicos interpoladores por troços e que definem a função $\mathbf{Q}(t)$ em $[t_0, t_n]$ tal que pertence a C^2 nos nós (*knots*).

Dedução dos coeficientes

Sendo polinómio cúbico, para o intervalo $[t_k, t_{k+1}]$ pode escrever-se

$$Q_k(t) = a_k + b_k (t - t_k) + c_k (t - t_k)^2 + d_k (t - t_k)^3$$

Condições fronteira:

$$Q(t_k) = P_k$$

$$Q(t_{k+1}) = P_{k+1}$$

$$\left(\frac{dQ_k}{dt}\right)_{t=t_k} = R_k$$

$$\left(\frac{dQ_k}{dt}\right)_{t=t_{k+1}} = R_{k+1}$$

Substituindo e resolvendo dá:

$$a_k = P_k$$

$$b_k = R_k$$

$$c_k = \frac{3(P_{k+1} - P_k)}{h_k^2} - \frac{2R_k}{h_k} - \frac{R_{k+1}}{h_k}$$

$$d_k = \frac{2(P_k - P_{k+1})}{h_k^3} + \frac{R_k}{h_k^2} + \frac{R_{k+1}}{h_k^2}$$

com $h_k = t_{k+1} - t_k$

Condições adicionais

As **4** equações deduzidas para os coeficientes referem-se a cada um dos troços (intervalos), pelo que haverá ao todo **4n** equações.

As derivadas R_i (em número de **n+1**) ainda não são conhecidas, pelo que precisamos de outras **n+1** equações.

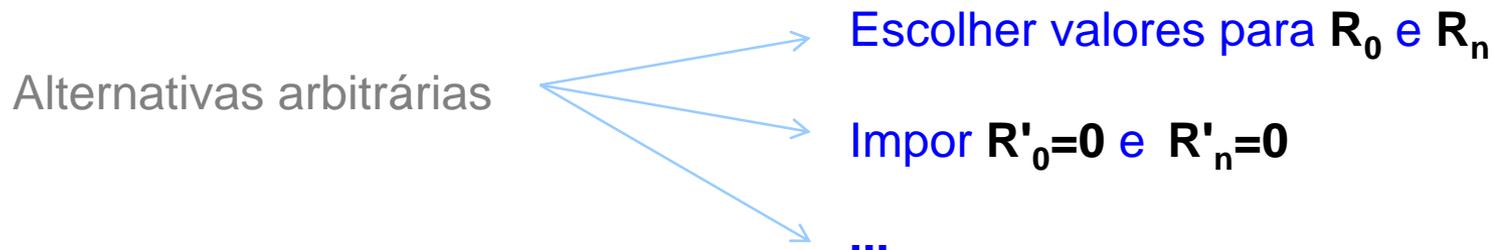
Condição para a segunda derivada:

$$\left(\frac{d^2 Q_k}{dt^2} \right)_{t=t_{k+1}} = \left(\frac{d^2 Q_{k+1}}{dt^2} \right)_{t=t_{k+1}}$$

Ao todo serão **n-1** equações, da forma:

$$2 c_k + 6 d_k (t_{k+1} - t_k) = 2 c_{k+1}$$

Para não deixarmos o sistema indeterminado, há que introduzir mais **2** equações:



Desvantagens dos Splines naturais

- ◆ Qualquer que seja a alternativa tomada para a resolução da indeterminação do sistema, o ajuste da curva resultante depende inteiramente da qualidade dos pontos dados, não se garantindo, portanto, a ausência de oscilações indesejáveis (mesmo na situação dos erros nos dados serem pequenos).
- ◆ Outro inconveniente é o do controlo global (e não local) da curva.

CONCLUSÃO:

Não é boa solução para design interactivo!

Curvas cúbicas e notação a usar

Curva no espaço 3D (também se poderia escrever em coordenadas homogêneas):

$$Q(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} a_x t^3 + b_x t^2 + c_x t + d_x \\ a_y t^3 + b_y t^2 + c_y t + d_y \\ a_z t^3 + b_z t^2 + c_z t + d_z \end{bmatrix} = \begin{bmatrix} [t^3 \ t^2 \ t \ 1] \cdot [a_x \ b_x \ c_x \ d_x]^T \\ [t^3 \ t^2 \ t \ 1] \cdot [a_y \ b_y \ c_y \ d_y]^T \\ [t^3 \ t^2 \ t \ 1] \cdot [a_z \ b_z \ c_z \ d_z]^T \end{bmatrix}$$

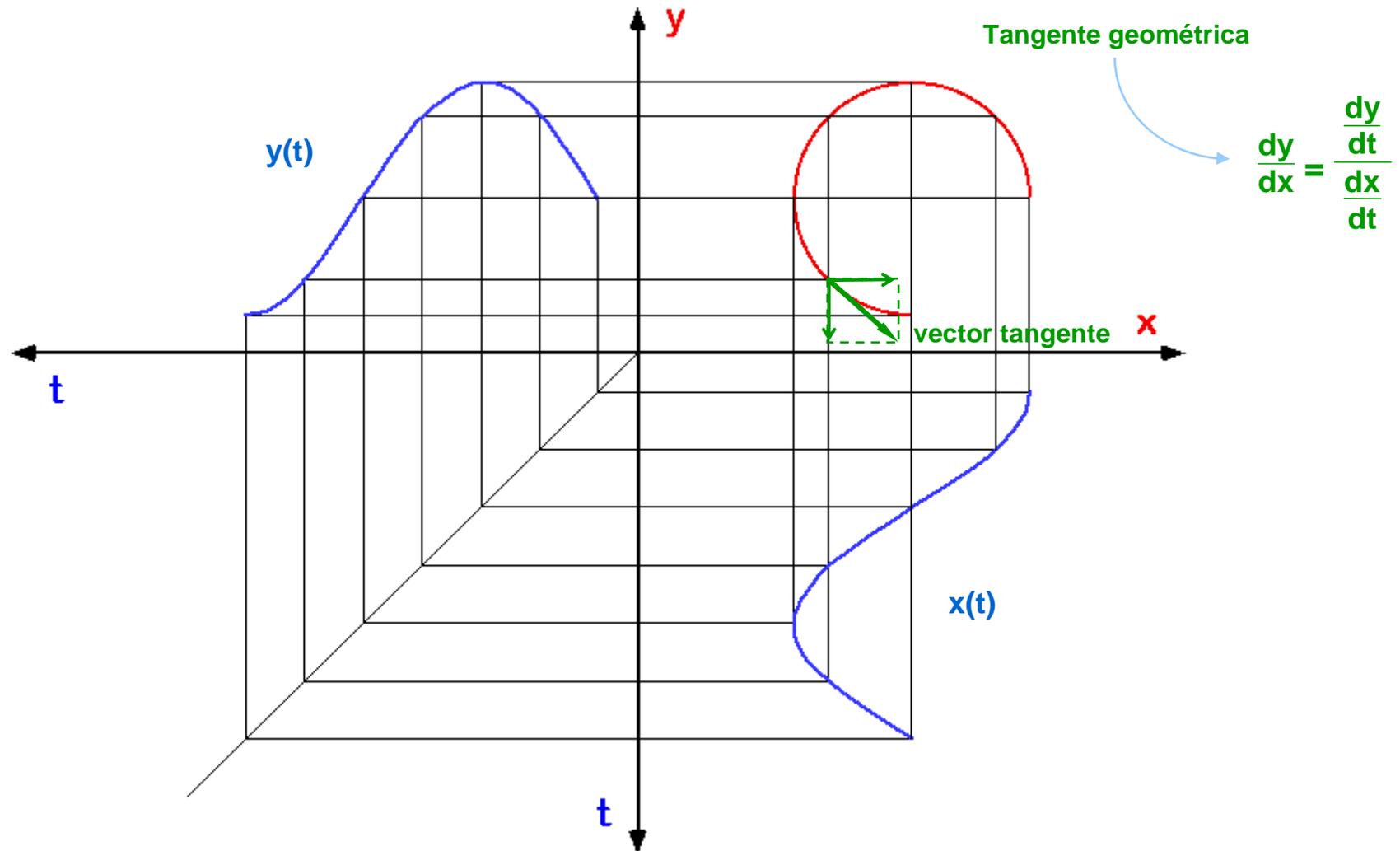
Sempre que não estiver em causa apenas uma coordenada em particular, por comodidade usar-se-á a expressão geral (vectorial)

$$Q(t) = a t^3 + b t^2 + c t + d = [t^3 \ t^2 \ t \ 1] \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = T.A$$

Conclusão:

4 coeficientes (vectoriais) arbitrários  pode-se impor 4 condições

Eliminação do parâmetro em curva 2D



Exemplo com outro tipo de curva

Forma paramétrica:

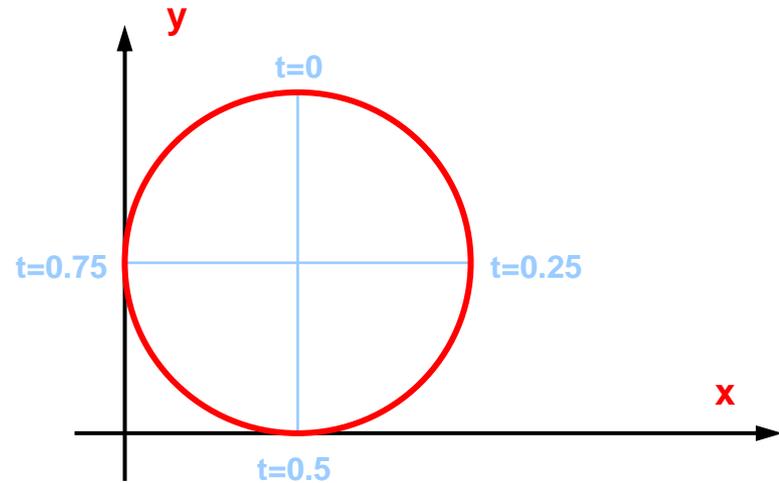
$$x = 1 + \sin(2\pi t)$$

$$y = 1 + \cos(2\pi t)$$

Componentes do vector tangente:

$$\frac{dx}{dt} = 2\pi \cos(2\pi t)$$

$$\frac{dy}{dt} = -2\pi \sin(2\pi t)$$



Módulo do vector tangente:

$$|R| = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} = 2\pi \sqrt{\cos^2(2\pi t) + \sin^2(2\pi t)} = 2\pi$$

Tangente geométrica:

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = -\operatorname{tg}(2\pi t)$$

Curvas de Hermite

Condições: 2 pontos a interpolar e vectores tangentes nesses mesmos pontos.

$$Q(0) = P_0$$

$$Q'(0) = R_0$$

$$Q(1) = P_3$$

$$Q'(1) = R_3$$

Na forma matricial:

$$\begin{bmatrix} Q(0) \\ Q(1) \\ Q'(0) \\ Q'(1) \end{bmatrix} = \begin{bmatrix} P_0 \\ P_3 \\ R_0 \\ R_3 \end{bmatrix}$$

Vector da geometria de Hermite G_H

Substituindo cada elemento $Q(t)$ por $T.A$ escrever-se-á:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot A = G_H$$

Resolvendo em ordem a A dará

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \cdot G_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot G_H = M_H \cdot G_H$$

Matriz de Hermite

pelo que uma curva de Hermite é da forma

$$Q(t) = T \cdot M_H \cdot G_H$$

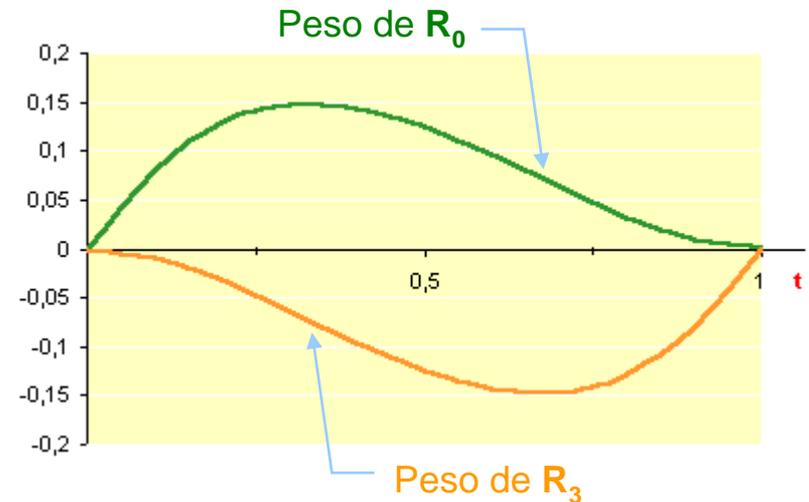
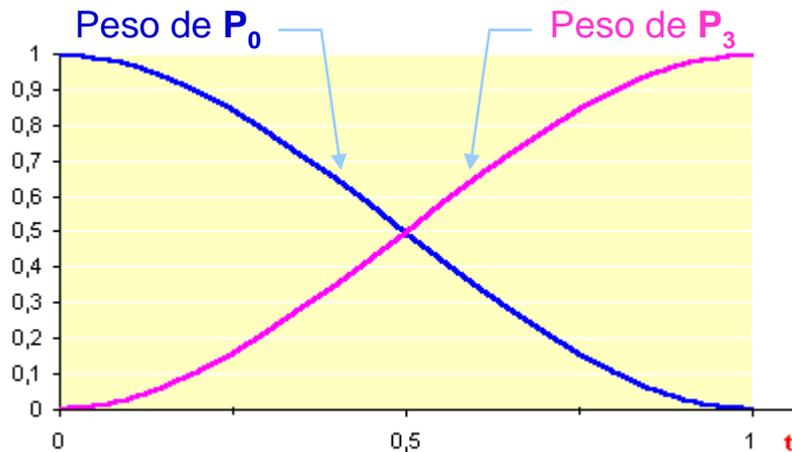
Blending functions

$$Q(t) = T \cdot M_H \cdot G_H$$

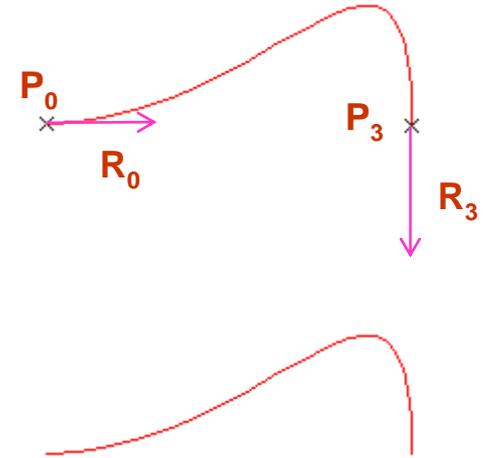
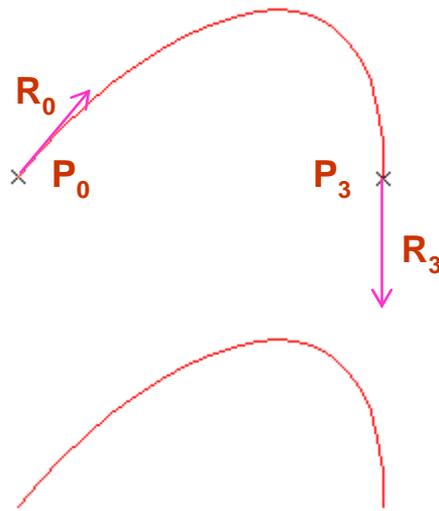
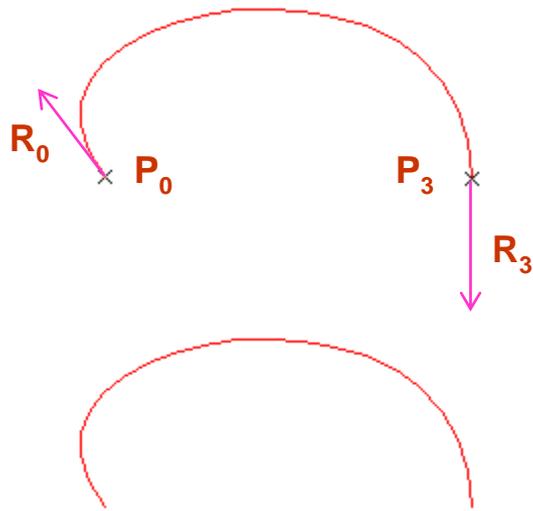
Os pesos dos elementos do vector de geometria são funções de t :

Blending Functions

$$T \cdot M_H = [2t^3 - 3t^2 + 1 \quad -2t^3 + 3t^2 \quad t^3 - 2t^2 + t \quad t^3 - t^2]$$

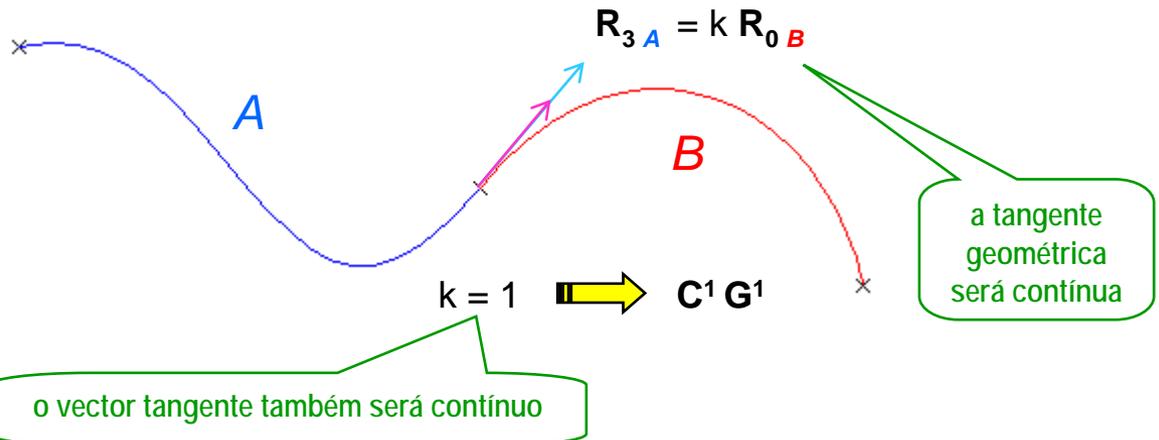
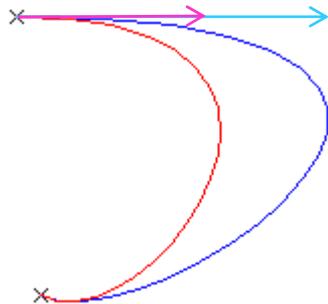


Exemplos de curvas no plano

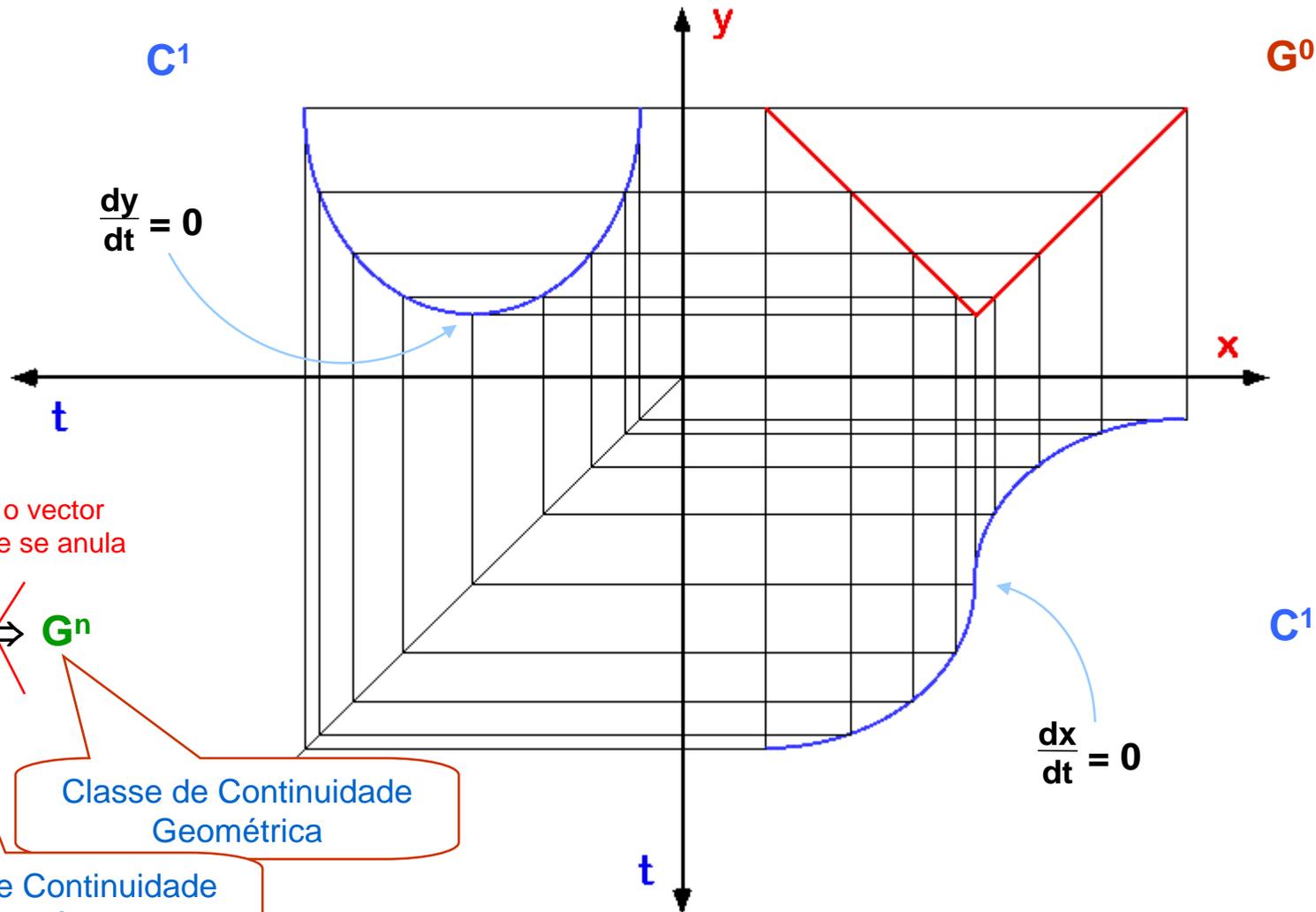


Condição $k > 0$ para $C^0 G^1$ no ponto de junção de troços:

Influência da norma do vector tangente:



Continuidades Paramétrica e Geométrica

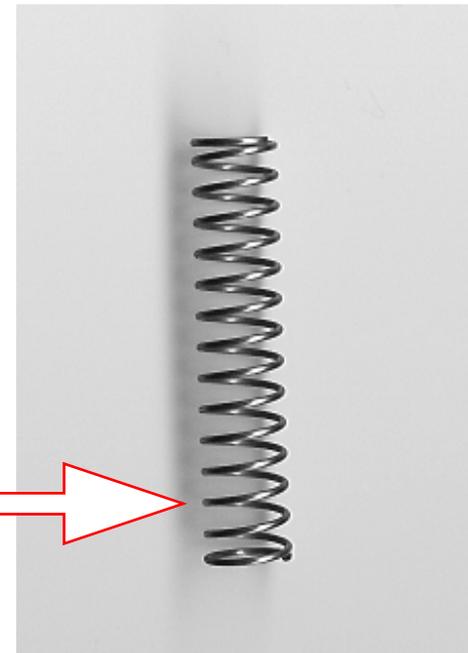


Comparação com exemplo fotográfico

Esta mola de arame não apresenta, no espaço 3D, quaisquer pontos angulosos ao longo da sua extensão...



... mas esse tipo de pontos pode aparecer numa fotografia (2D), dependendo da escolha do ponto de vista.



Curvas de Bézier

Condições: As mesmas das curvas de Hermite, introduzindo-se 2 pontos intermédios que determinam os vectores tangentes.

$$Q'(0) = R_0 = 3 (P_1 - P_0)$$

$$Q'(1) = R_3 = 3 (P_3 - P_2)$$

donde:

$$\boxed{G_H} \begin{bmatrix} P_0 \\ P_3 \\ R_0 \\ R_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \boxed{G_B}$$

Vector da geometria de Bézier

$$Q(t) = T \cdot M_H \cdot G_H$$



$$Q(t) = T \cdot M_B \cdot G_B$$

com a Matriz de Bézier:

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Numa interface para curvas de Bézier manipulam-se apenas pontos, enquanto que para curvas de Hermite há que manipular também vectores.

Curvas de Bézier

Exemplo geral da utilização de *Blending Functions* $B_{in}(t)$ de ordem n :

$$Q(t) = P_0 B_{0n}(t) + P_1 B_{1n}(t) + P_2 B_{2n}(t) + \dots + P_m B_{mn}(t)$$

com $0 \leq t \leq 1$ e $n = m + 1$

B_{in} equivale a $B_{i,n}$

Blending Functions de ordem $n=4$ para as Curvas de Bézier:

Se $Q(t) = T \cdot M_B \cdot G_B$ então

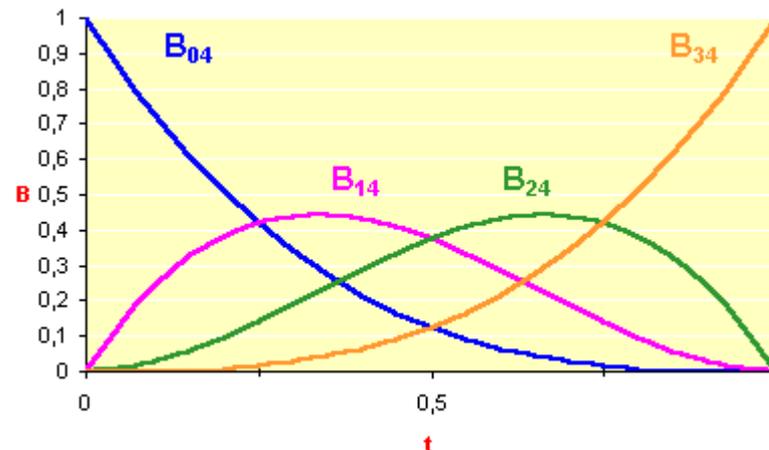
$$T \cdot M_B = [B_{04} \quad B_{14} \quad B_{24} \quad B_{34}]$$

$$B_{04}(t) = -t^3 + 3t^2 - 3t + 1$$

$$B_{14}(t) = 3t^3 - 6t^2 + 3t$$

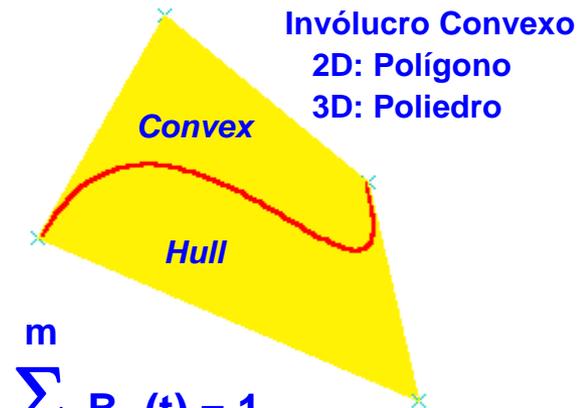
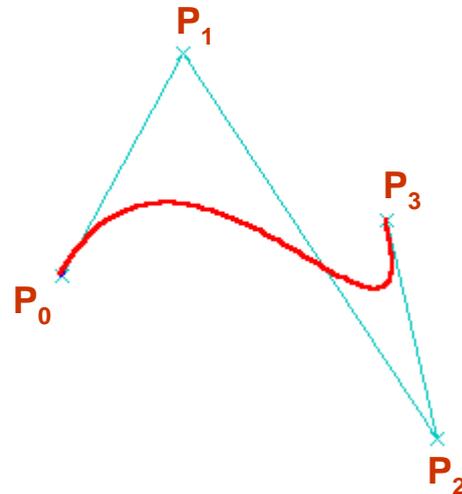
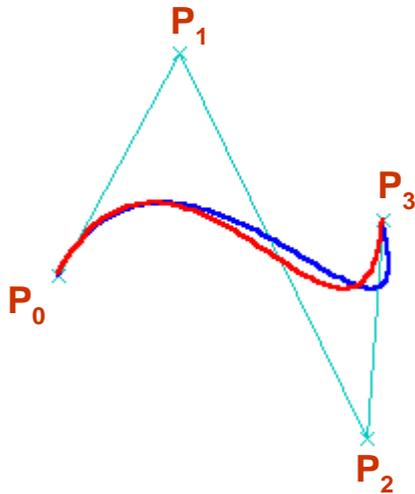
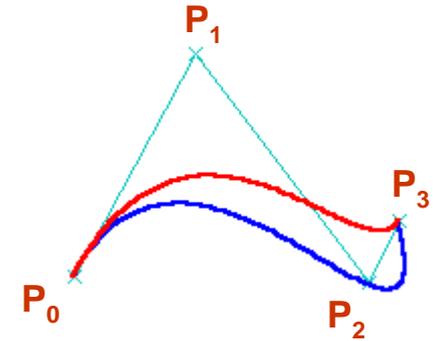
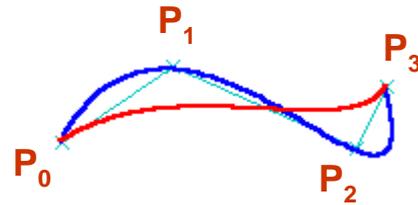
$$B_{24}(t) = -3t^3 + 3t^2$$

$$B_{34}(t) = t^3$$



Exemplo interactivo

Curva pretendida:



Invólucro Convexo
2D: Polígono
3D: Poliedro

$$\sum_{i=0}^m B_{in}(t) = 1$$

$$B_{in}(t) \geq 0$$

Aproximação Bézier-Bernstein

Generalização:

$$Q(t) = \sum_{i=0}^m P_i B_{in}(t)$$

com $0 \leq t \leq 1$ e $n = m + 1$

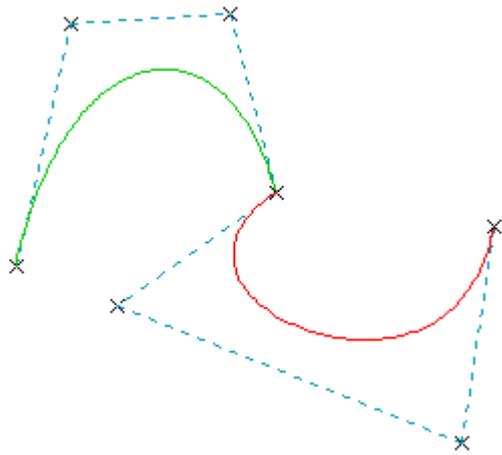
Curva de ordem n , aplicada a $m+1$ pontos
e tendo como funções de peso (*Blending Functions*) os

Polinómios de Bernstein:

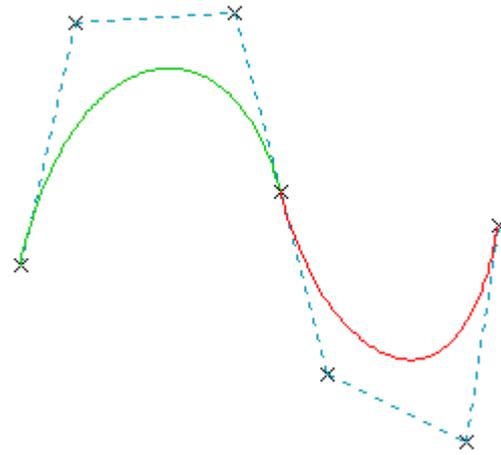
$$B_{k,n}(t) = \frac{(n-1)!}{k! (n-1-k)!} t^k (1-t)^{n-1-k}$$

As Curvas cúbicas de Bézier são o caso particular em que $n=4$ pontos.

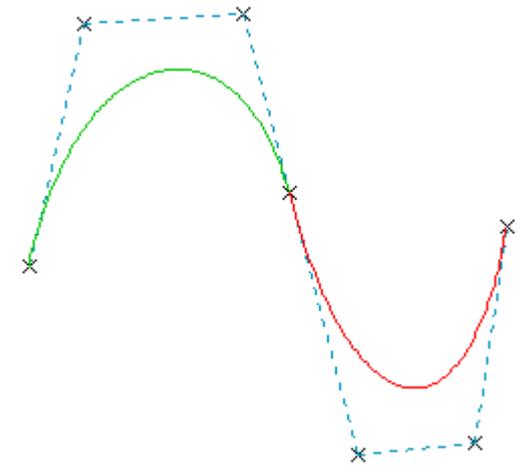
Junção de troços (segmentos) de curva



Continuidade $C^0 G^0$ na junção
(vectores tangentes com direcções
ou sentidos diferentes)



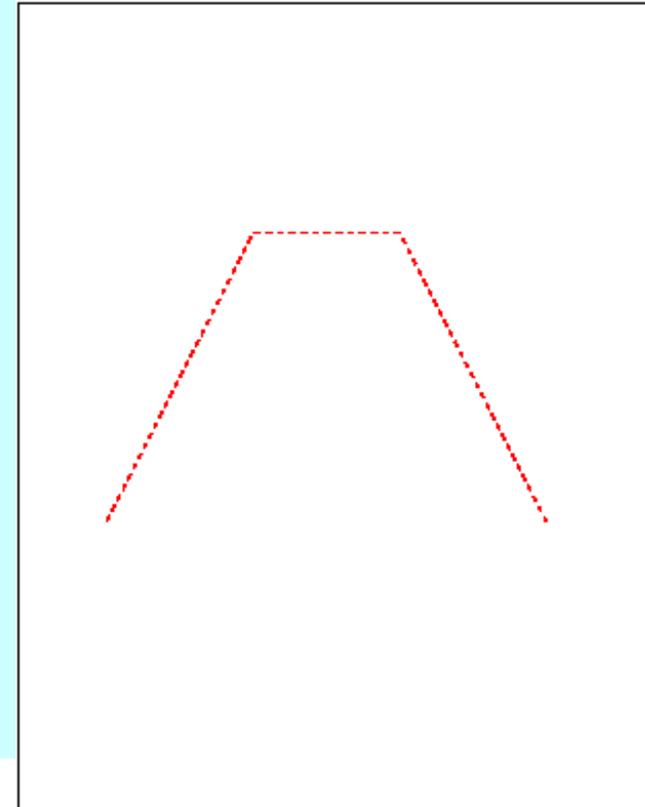
Continuidade $C^1 G^1$ na junção
(vectores tangentes iguais)



Continuidade $C^0 G^1$ na junção
(vectores tangentes apenas com a
mesma direcção e sentido)

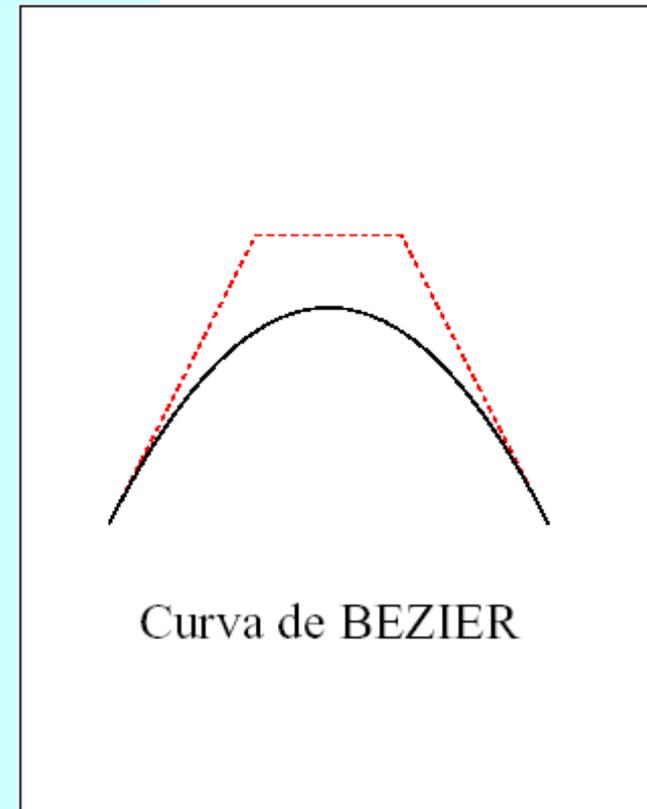
Exemplo de utilização em PostScript

```
%! PS
% Poligonal guia numa curva de Bézier (M. Próspero)
/cm {28.35 mul} def
3 cm 10 cm moveto
8 cm 20 cm lineto
13 cm 20 cm lineto
18 cm 10 cm lineto
[ 0.2 cm 0.2 cm ] 0 setdash
0.02 cm setlinewidth
1.0 0.0 0.0 setrgbcolor
stroke
showpage
```



Exemplo de utilização em PostScript

```
%! PS
% Desenho de uma curva de Bézier (M. Próspero)
/cm {28.35 mul} def
3 cm 10 cm moveto
gsave
  8 cm 20 cm lineto
  13 cm 20 cm lineto
  18 cm 10 cm lineto
  [ 0.2 cm 0.2 cm ] 0 setdash
  0.02 cm setlinewidth
  1.0 0.0 0.0 setrgbcolor
  stroke
grestore
8 cm 20 cm 13 cm 20 cm 18 cm 10 cm curveto
stroke
% Legenda:
/Times-Roman fi ndfont
50 scalefont
setfont
4 cm 6 cm moveto
(Curva de BEZIER) show
showpage
```



Algoritmo de DE CASTELJAU

A partir dos pontos dados \mathbf{P}_i (apresentados agora na forma \mathbf{P}_{0i}) definem-se pontos auxiliares:

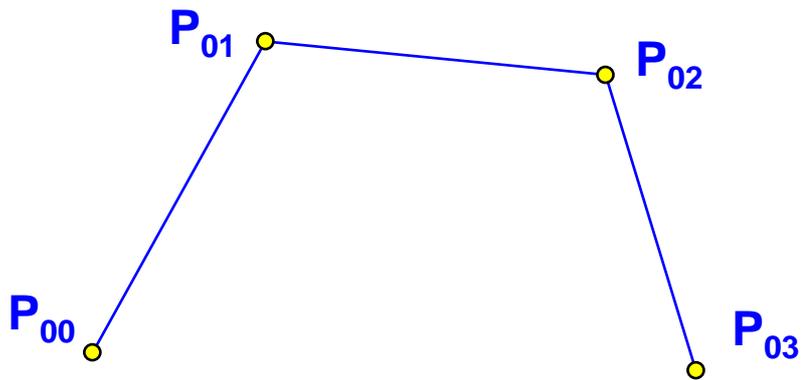
$$\mathbf{P}_{k,n}(t) = (1 - t) \mathbf{P}_{k-1,n-1}(t) + t \mathbf{P}_{k-1,n}(t)$$

Exercício: Verificar que $\mathbf{P}_{33}(t)$ corresponde à **curva cúbica de Bézier**.

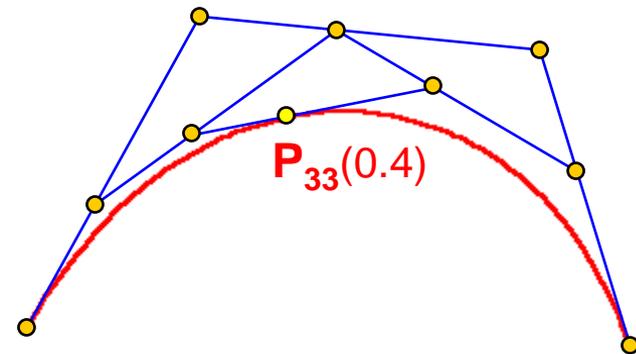
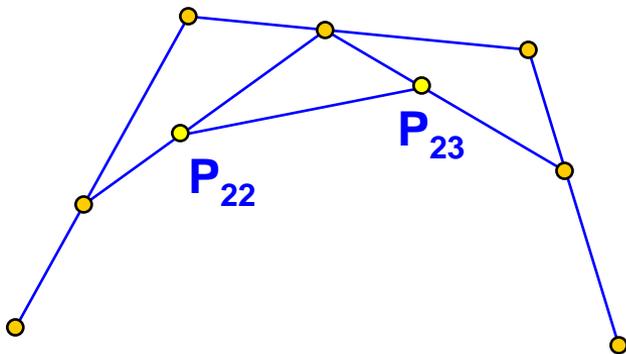
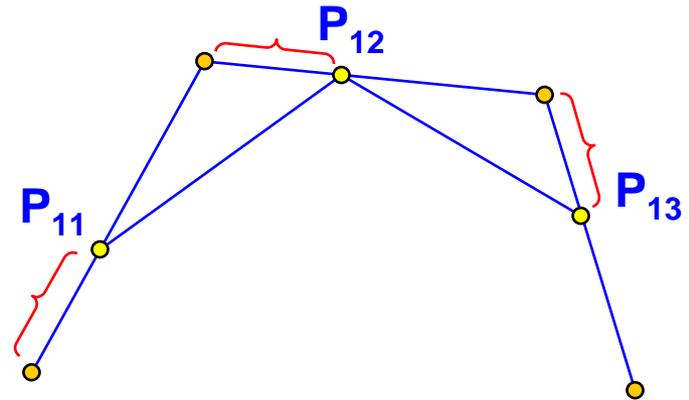
Resolução:

$$\begin{aligned} \mathbf{P}_{33}(t) &= (1 - t) \mathbf{P}_{22}(t) + t \mathbf{P}_{23}(t) \\ &= (1 - t) ((1 - t) \mathbf{P}_{11}(t) + t \mathbf{P}_{12}(t)) + \\ &\quad t ((1 - t) \mathbf{P}_{12}(t) + t \mathbf{P}_{13}(t)) \\ &= (1 - t)^2 ((1 - t) \mathbf{P}_{00} + t \mathbf{P}_{01}) + \\ &\quad 2(t - t^2) ((1 - t) \mathbf{P}_{01} + t \mathbf{P}_{02}) + t^2 ((1 - t) \mathbf{P}_{02} + t \mathbf{P}_{03}) \\ &= (1 - t)^3 \mathbf{P}_{00} + (3t - 6t^2 + 3t^3) \mathbf{P}_{01} + (3t^2 - 3t^3) \mathbf{P}_{02} + t^3 \mathbf{P}_{03} \\ &= \mathbf{Q}_{\text{Bézier}}(t) \end{aligned}$$

Aplicação geométrica



Para $t=0.4$:



Ver Applet em http://www.saltire.com/applets/advanced_geometry/spline/spline.htm

Curvas B-Spline

$$Q(t) = T \cdot M_{Bs} \cdot G_{Bs}$$

O vector da geometria G_{Bs} é igual ao de Bézier (G_B), sendo diferente a Matriz B-spline M_{Bs} :

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

Blending Functions:

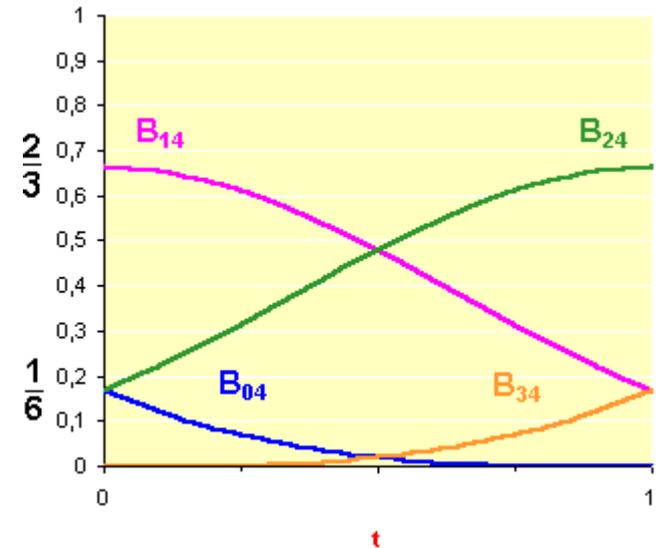
$$T \cdot M_{Bs} = [B_{04} \quad B_{14} \quad B_{24} \quad B_{34}]$$

$$B_{04}(t) = (-t^3 + 3t^2 - 3t + 1) / 6$$

$$B_{14}(t) = (3t^3 - 6t^2 + 4) / 6$$

$$B_{24}(t) = (-3t^3 + 3t^2 + 3t + 1) / 6$$

$$B_{34}(t) = t^3 / 6$$



Conclusão:

Início da curva \Rightarrow vizinhança de P_1

Final da curva \Rightarrow vizinhança de P_2

Propriedades: C^0

Sejam dados os seguintes vectores da geometria, geradores dos troços Q_3 e Q_4 :

$$G_{Bs_3} = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$G_{Bs_4} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

Sendo

$$T \cdot M_{Bs} = \frac{1}{6} \begin{bmatrix} -t^3 + 3t^2 - 3t + 1 & 3t^3 - 6t^2 + 4 & -3t^3 + 3t^2 + 3t + 1 & t^3 \end{bmatrix}$$

e considerando o intervalo $0 \leq t \leq 1$ para cada troço, por substituição obter-se-á:

$$Q_3(1) = \frac{1}{6} \begin{bmatrix} 0 & 1 & 4 & 1 \end{bmatrix} G_{Bs_3}$$

$$Q_4(0) = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} G_{Bs_4}$$

$$Q_3(1) = Q_4(0)$$

Conclusão:  Junção dos troços num mesmo ponto, implicando continuidade C^0

Propriedades: C^1

Primeira derivada:

$$\frac{d}{dt} T \cdot M_{Bs} = \frac{1}{2} [-t^2 + 2t - 1 \quad 3t^2 - 4t \quad -3t^2 + 2t + 1 \quad t^2]$$

Considerando o intervalo $0 \leq t \leq 1$ para cada troço, por substituição obter-se-á:

$$\frac{d}{dt} Q_3(1) = \frac{1}{2} [0 \quad -1 \quad 0 \quad 1] G_{Bs_3}$$

$$\frac{d}{dt} Q_4(0) = \frac{1}{2} [-1 \quad 0 \quad 1 \quad 0] G_{Bs_4}$$

$$\frac{d}{dt} Q_3(1) = \frac{d}{dt} Q_4(0)$$

Conclusão:  Continuidade C^1 no ponto de junção dos troços

Propriedades: C^2

Segunda derivada:

$$\frac{d^2}{dt^2} \mathbf{T} \cdot \mathbf{M}_{Bs} = [-t+1 \quad 3t-2 \quad -3t+1 \quad t]$$

Considerando o intervalo $0 \leq t \leq 1$ para cada troço, por substituição obter-se-á:

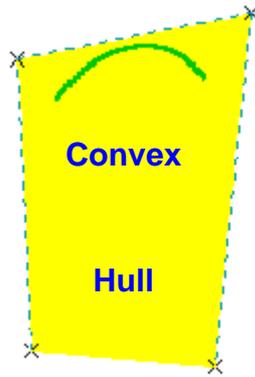
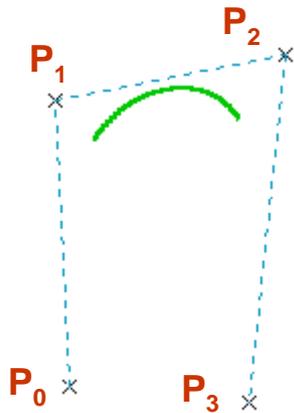
$$\frac{d^2}{dt^2} \mathbf{Q}_3(1) = [0 \quad 1 \quad -2 \quad 1] \mathbf{G}_{Bs3}$$

$$\frac{d^2}{dt^2} \mathbf{Q}_4(0) = [1 \quad -2 \quad 1 \quad 0] \mathbf{G}_{Bs4}$$

$$\frac{d^2}{dt^2} \mathbf{Q}_3(1) = \frac{d^2}{dt^2} \mathbf{Q}_4(0)$$

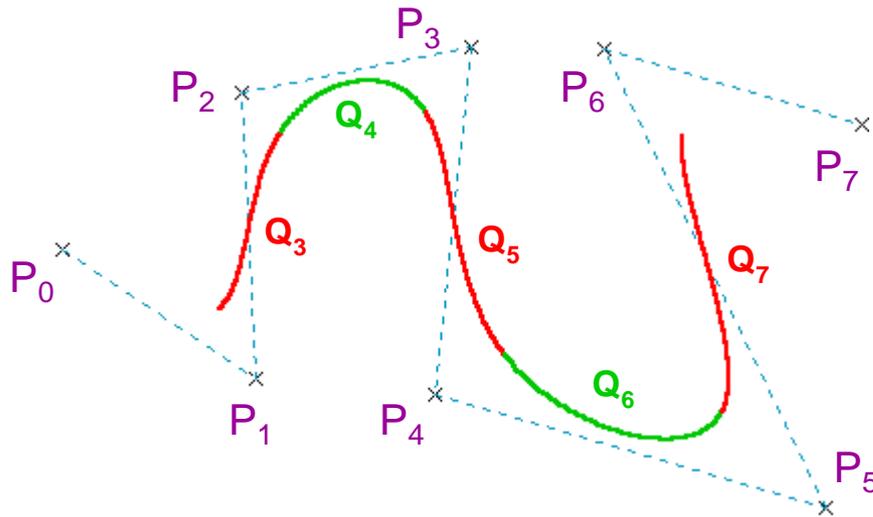
Conclusão:  Continuidade C^2 no ponto de junção dos troços

Exemplos de curvas B-spline em 2D



$$\sum_{i=0}^m B_{in}(t) = 1$$

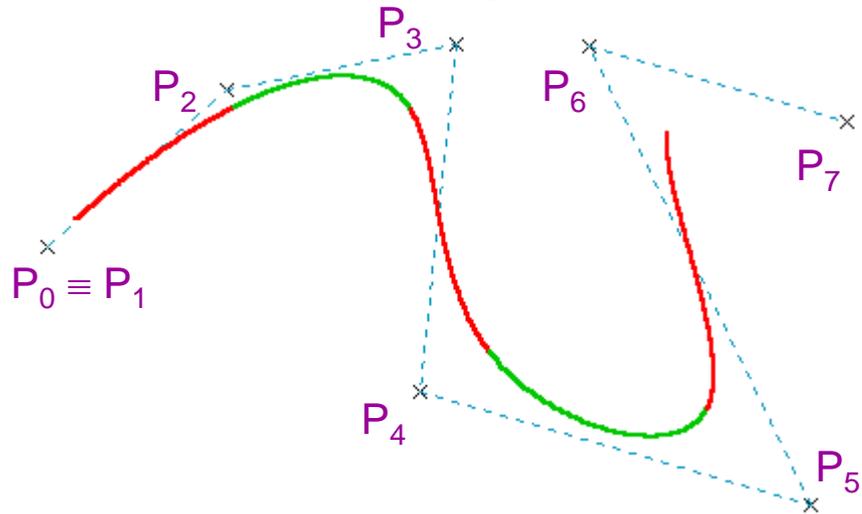
$$B_{in}(t) \geq 0$$



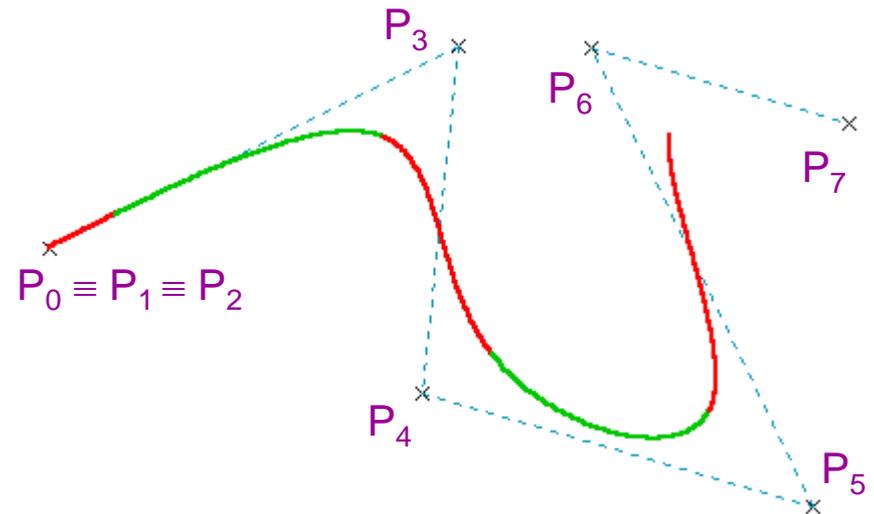
$$G_{Bsi} = \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$

Grau de multiplicidade de pontos

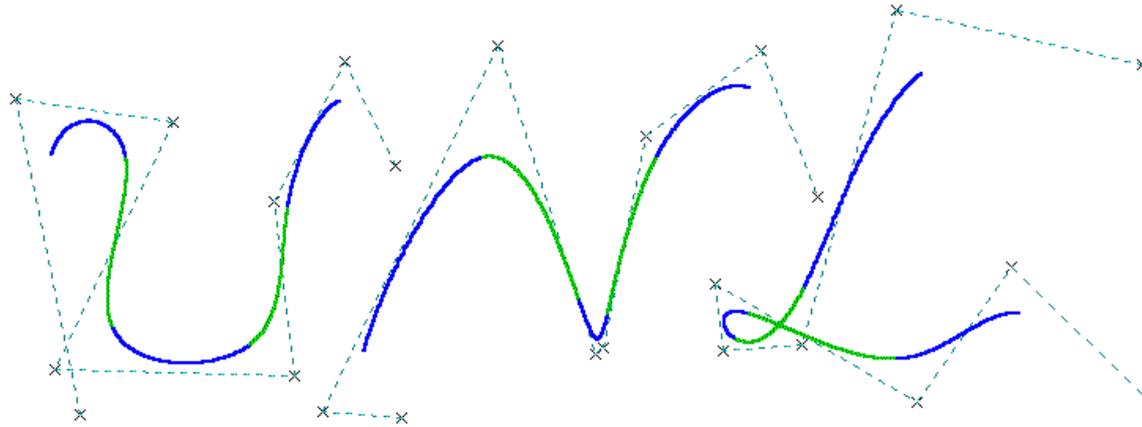
Ponto com Grau de Multiplicidade 2 :



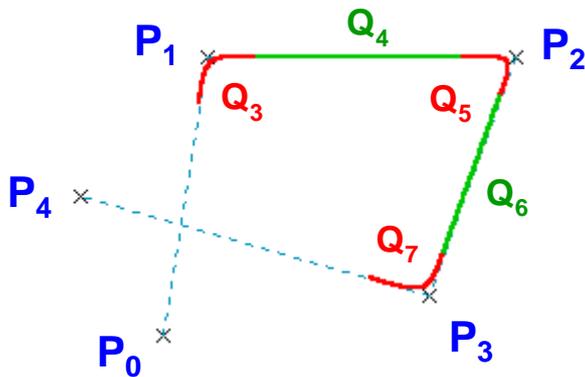
Ponto com Grau de Multiplicidade 3 :



Outros exemplos (1)



Pontos com
Grau de Multiplicidade 2 :



$$G_{Bs3} = \begin{bmatrix} P_0 \\ P_1 \\ P_1 \\ P_2 \end{bmatrix}$$

$$G_{Bs4} = \begin{bmatrix} P_1 \\ P_1 \\ P_2 \\ P_2 \end{bmatrix}$$

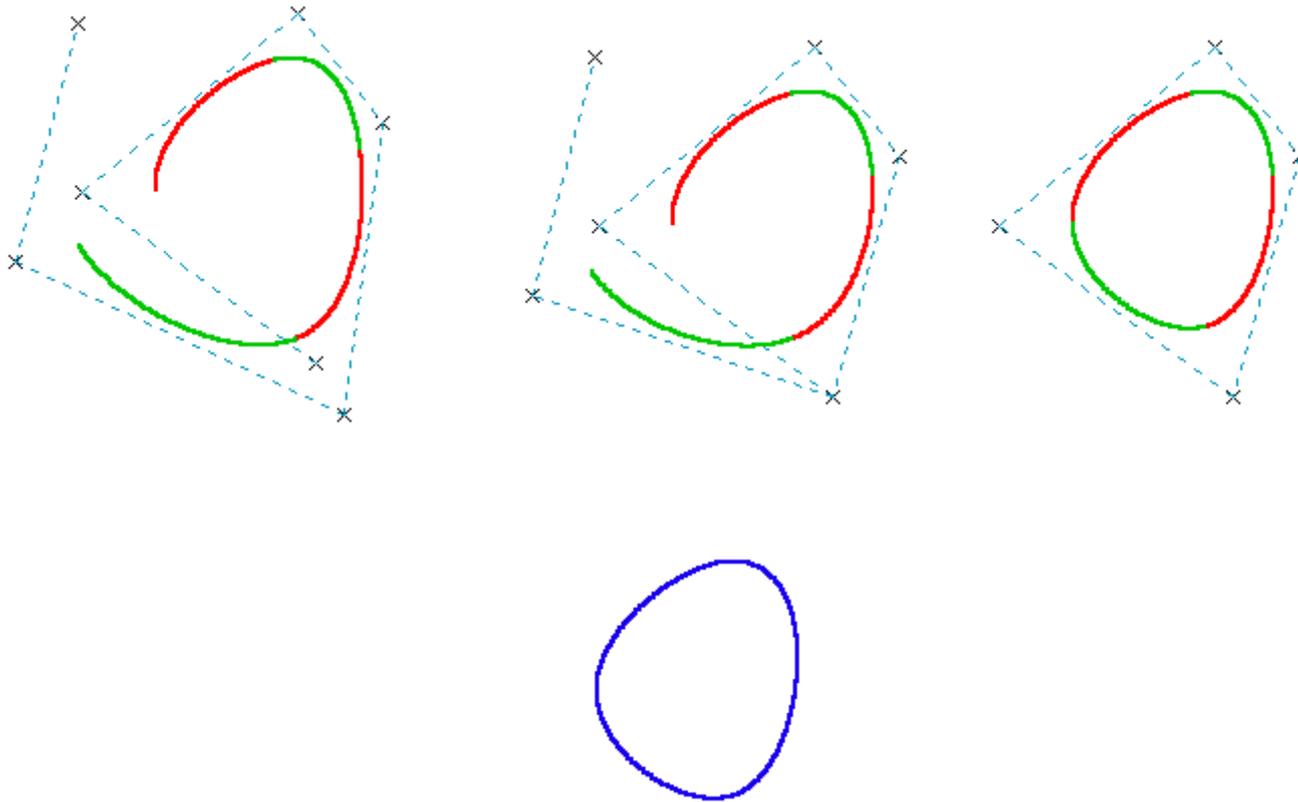
$$G_{Bs5} = \begin{bmatrix} P_1 \\ P_2 \\ P_2 \\ P_3 \end{bmatrix}$$

$$G_{Bs6} = \begin{bmatrix} P_2 \\ P_2 \\ P_3 \\ P_3 \end{bmatrix}$$

$$G_{Bs7} = \begin{bmatrix} P_2 \\ P_3 \\ P_3 \\ P_4 \end{bmatrix}$$

Outros exemplos (2)

Construção de uma curva fechada :



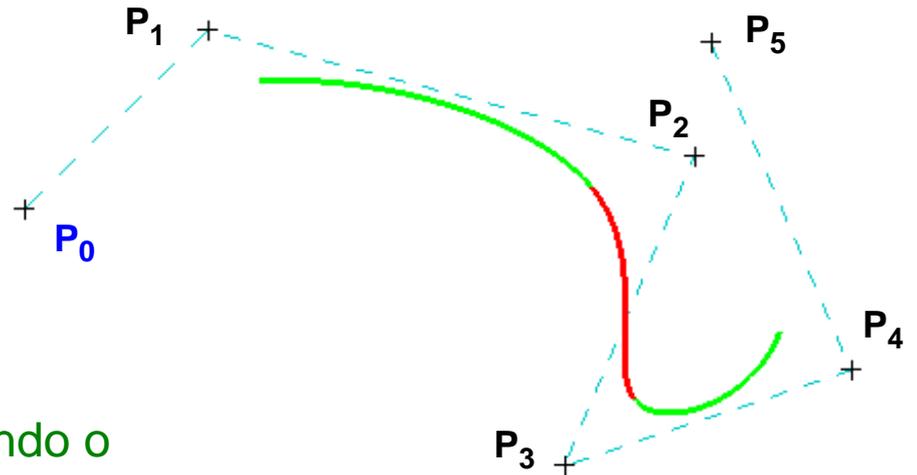
Continuidade e Interpolação

- ◆ **Resultado dos graus de multiplicidade dos pontos de controlo:**
 - Multiplicidade **1**  continuidade **$C^2 G^2$**
 - Multiplicidade **2**  continuidade **$C^2 G^1$**
 - Multiplicidade **3**  continuidade **$C^2 G^0$**
 - Multiplicidade **3**  interpola esse ponto de controlo

- ◆ **Haverá alternativa à interpolação no(s) extremo(s) de uma curva sem ser pelo aumento do grau de multiplicidade do(s) ponto(s), permitindo manter continuidade de classe $C^2 G^2$ com futuros troços adjacentes?**
 - Sim, por criação de novo(s) ponto(s) de controlo dito(s) “fantasma(s)”

Pontos de controlo “fantasmas” (1)

Admitindo que se quer interpolar P_0 :

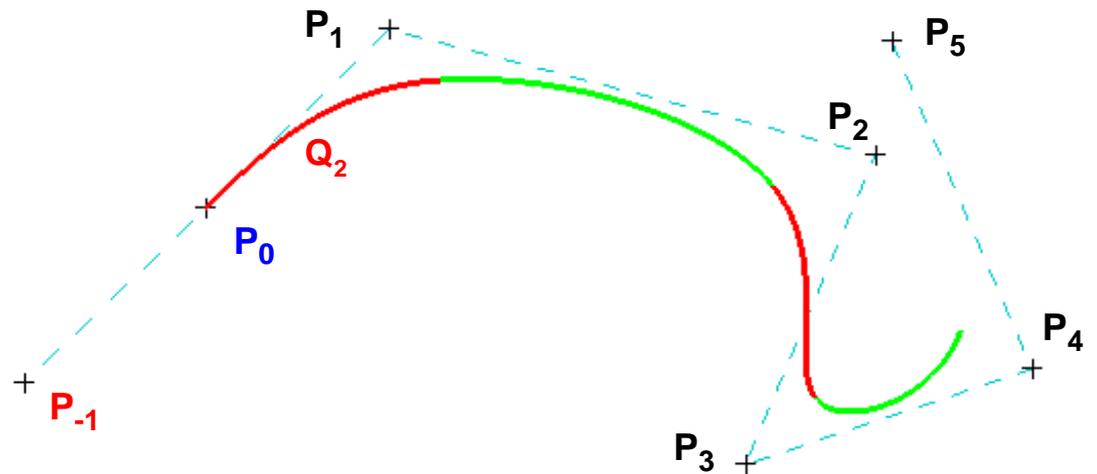


Resolve-se matematicamente encontrando o ponto “fantasma” P_{-1} que verifique a condição

$$Q_2(t=0) = T \cdot M_{Bs} \cdot \begin{bmatrix} P_{-1} \\ P_0 \\ P_1 \\ P_2 \end{bmatrix} = P_0$$

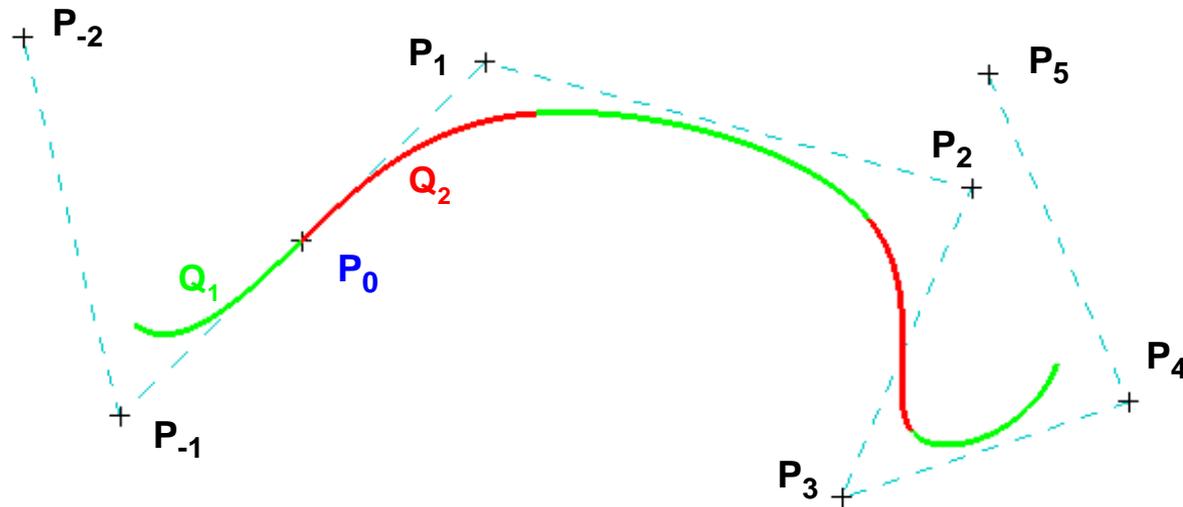
$$(P_{-1} + 4P_0 + P_1) / 6 = P_0$$

$$P_{-1} = 2P_0 - P_1$$



Pontos de controlo “fantasmas” (2)

Construção de um troço Q_1 adjacente a Q_2 , que também irá interpolar P_0 , mantendo-se continuidade $C^2 G^2$ em todas as junções de troços da curva (o que não seria possível se P_0 tivesse grau de multiplicidade superior a 1):



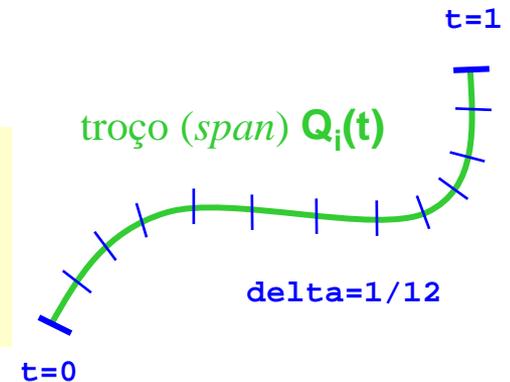
Avaliação de uma curva cúbica

Recorde-se que

$$Q_i(t) = T \cdot A = a * t^3 + b * t^2 + c * t + d$$

Algoritmo de visualização (aproximação a poligonal):

```
for (i=first_span to last_span)
  for (j=1 to 1/delta)
    Line_3D( Q_i((j-1)*delta), Q_i(j*delta) )
```



Com este algoritmo, a curva $Q_i(t)$ é avaliada $2/\delta$ vezes (e $1+1/\delta$ recorrendo à posição corrente ou a uma variável auxiliar). Mas a eficiência aumentará se se puder reduzir o número de operações aritméticas.

Um dos métodos para tal é o de **HORNER**:

$$Q_i(t) = ((a * t + b) * t + c) * t + d$$

(Extra programa: Um outro método, ainda mais rápido, é o das **Diferenças Finitas Descendentes**)

Conversão entre curvas cúbicas

Se for construído (ou se já existir num determinado sistema) um avaliador para um certo tipo de curvas cúbicas, é possível usá-lo com qualquer outra forma de curva cúbica desde que se proceda a uma conversão entre curvas.

É o que se passa com os avaliadores (eficientes) para curvas de Bézier já existentes em PostScript e em OpenGL.

PROBLEMA:

Dada a curva $Q(t) = T \cdot M_{Bs} \cdot G_{Bs}$

encontrar G_B tal que $Q(t) = T \cdot M_B \cdot G_B$

Conversão para curva de Bézier

Solução:

$$\mathbf{M}_B \cdot \mathbf{G}_B = \mathbf{M}_{Bs} \cdot \mathbf{G}_{Bs}$$

$$\mathbf{G}_B = \mathbf{M}_B^{-1} \cdot \mathbf{M}_{Bs} \cdot \mathbf{G}_{Bs}$$

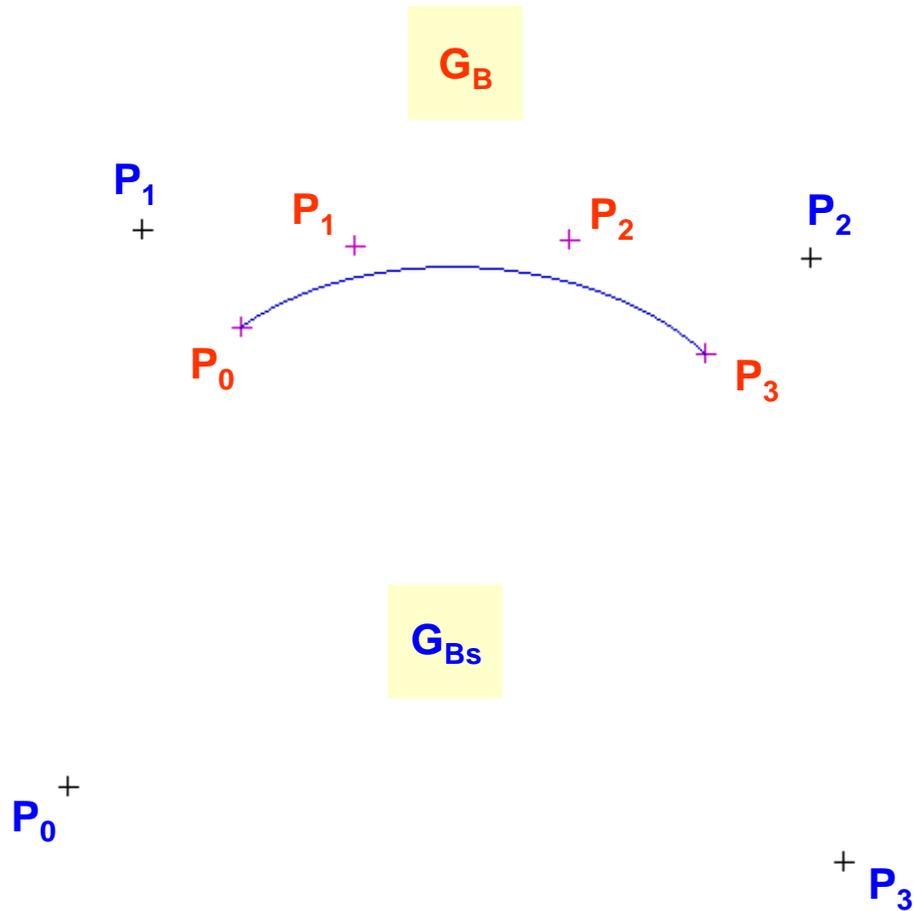
em que

$$\mathbf{M}_B^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1/3 & 1 \\ 0 & 1/3 & 2/3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Conclusão:

Qualquer curva cúbica pode ser convertida numa curva cúbica de Bézier

Exemplo de conversão B-spline → Bézier



Curvas de Catmull-Rom

ou Splines de Overhauser

$$Q(t) = T \cdot M_{CR} \cdot G_{Bs}$$

$$M_{CR} = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

O vector da geometria é igual ao de B-spline (G_{Bs})

Blending Functions:

$$T \cdot M_{CR} = [B_{04} \quad B_{14} \quad B_{24} \quad B_{34}]$$

$$B_{04}(t) = (-t^3 + 2t^2 - t) / 2$$

$$B_{14}(t) = (3t^3 - 5t^2 + 2) / 2$$

$$B_{24}(t) = (-3t^3 + 4t^2 + t) / 2$$

$$B_{34}(t) = (t^3 - t^2) / 2$$

Propriedades

Considerando o intervalo $0 \leq t \leq 1$ para cada troço:

$$Q_i(0) = \frac{1}{2} [0 \ 2 \ 0 \ 0] G_{Bs_i} = P_{i-2}$$

$$Q_i(1) = \frac{1}{2} [0 \ 0 \ 2 \ 0] G_{Bs_i} = P_{i-1}$$

$$Q_{i+1}(0) = \frac{1}{2} [0 \ 2 \ 0 \ 0] G_{Bs_{i+1}} = P_{i-1}$$

$$Q_i(1) = Q_{i+1}(0)$$

C^0 e interpolação de pontos de controlo



$$\frac{d}{dt} T \cdot M_{CR} = \frac{1}{2} [-3t^2 + 4t - 1 \quad 9t^2 - 10t \quad -9t^2 + 8t + 1 \quad 3t^2 - 2t]$$

$$\frac{d}{dt} Q_i(1) = \frac{1}{2} [0 \ -1 \ 0 \ 1] G_{Bs_i} = \frac{P_i - P_{i-2}}{2}$$

$$\frac{d}{dt} Q_{i+1}(0) = \frac{1}{2} [-1 \ 0 \ 1 \ 0] G_{Bs_{i+1}} = \frac{P_i - P_{i-2}}{2}$$

$$\frac{d}{dt} Q_i(1) = \frac{d}{dt} Q_{i+1}(0)$$

Continuidade C^1



Exemplos

Direcção dos vectores tangente

