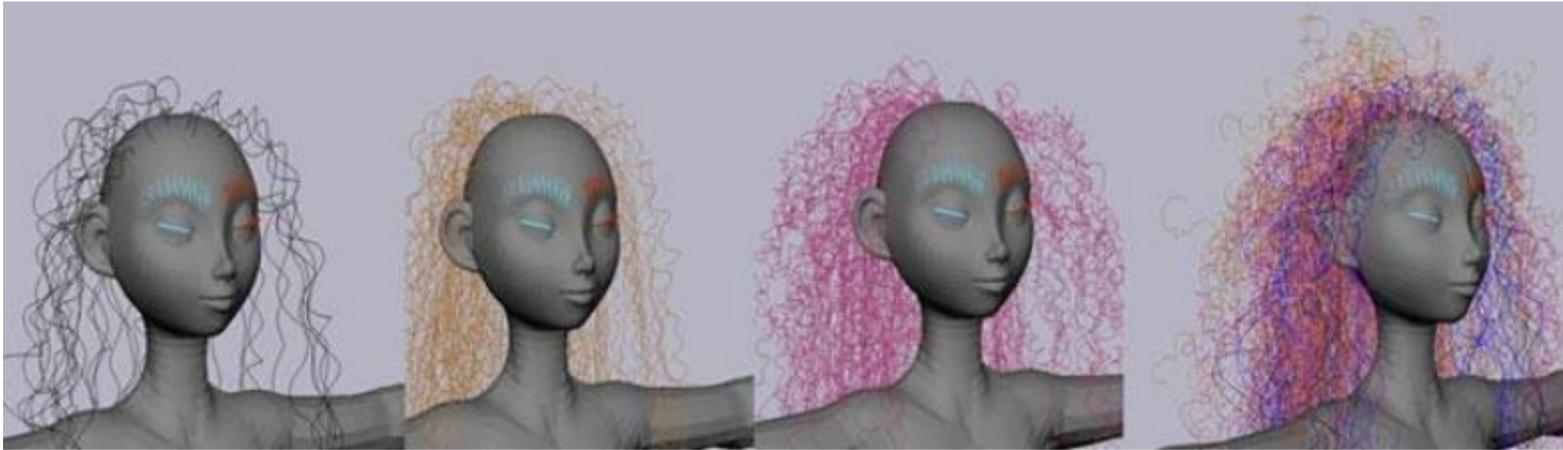


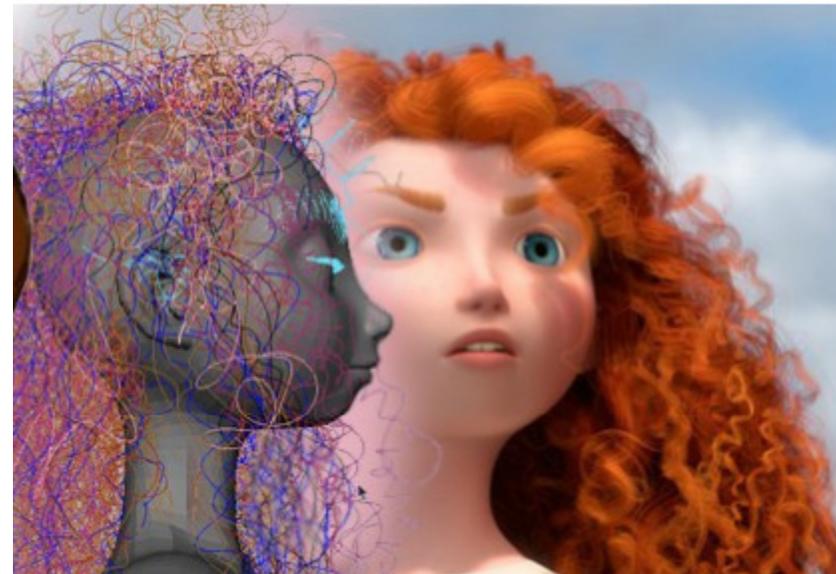
<http://computacaografica.ic.uff.br/conteudocap3.html>



aula 17

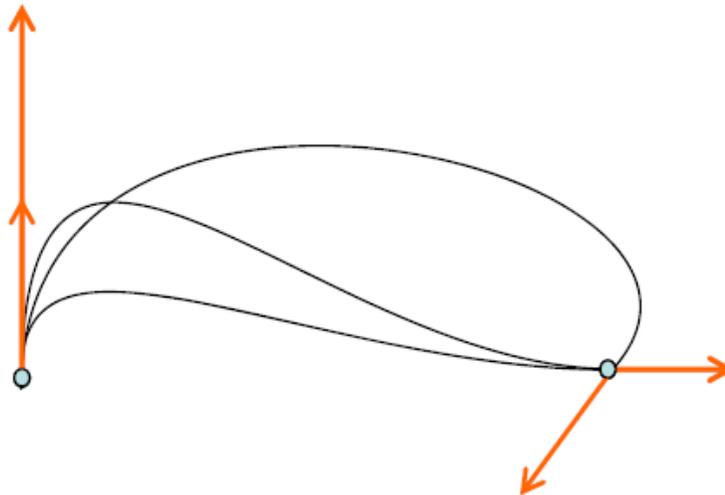
IC/UFF – 2018 - 2

**Curvas de Hermite,  
Splines e  
Superfícies em CG**



# Curvas de Hermite

- Ao invés de modelar a curva a partir de um polígono de controle (Bézier), especifica-se pontos de controle e vetores tangentes nesses pontos
- Vantagem: é fácil emendar várias curvas bastando especificar tangentes iguais nos pontos de emenda
- Exemplos (cúbicas):



## Curvas de Hermite

[https://pt.wikipedia.org/wiki/Charles\\_Hermite](https://pt.wikipedia.org/wiki/Charles_Hermite)

Criadas por Charles Hermite (1822-1901)

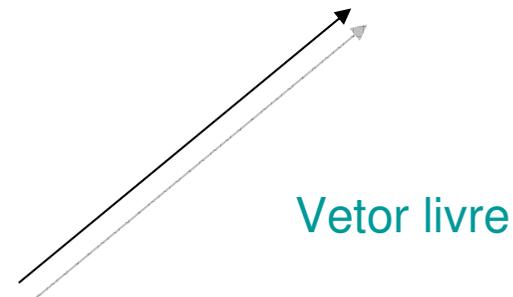
tem como idéia básica o comando por **Vetores**  
mas qual tipo de vetores?? :

**Na matemática** - um elemento com de um espaço vetorial

**Em Física** – em oposição as **grandezas escalares**, algo que se caracteriza por ter intensidade, sentido , direção e ponto de aplicação em engenharia e outras ciências

**Computação** – arranjo unidimensional - estrutura de dados utilizada no contexto da programação.

**Epidemiologia** - um agente de disseminação de doenças infecto-contagiosas



## Curva de Hermite

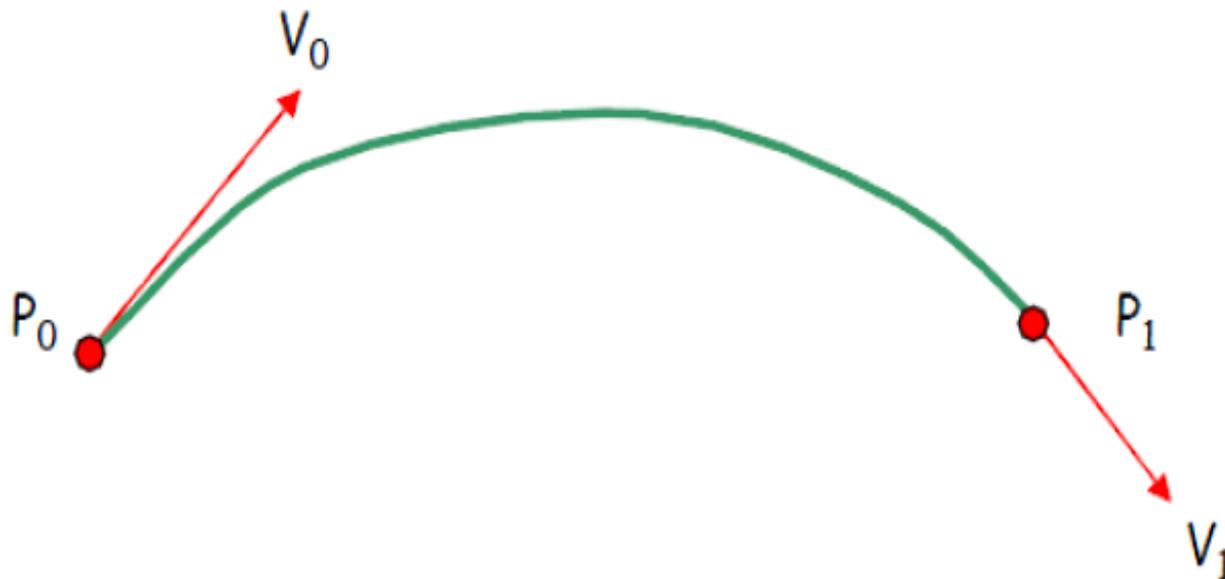
- A curva de Hermite é determinada por restrições no primeiro e último pontos
- O usuário especifica o primeiro ( $P_1$ ) e o último pontos ( $P_4$ ) bem como os vetores tangentes a  $P_1$  e  $P_4$ , chamados  $R_1$  e  $R_4$

pontos de controle =  $P_i$

# Curvas de Hermite

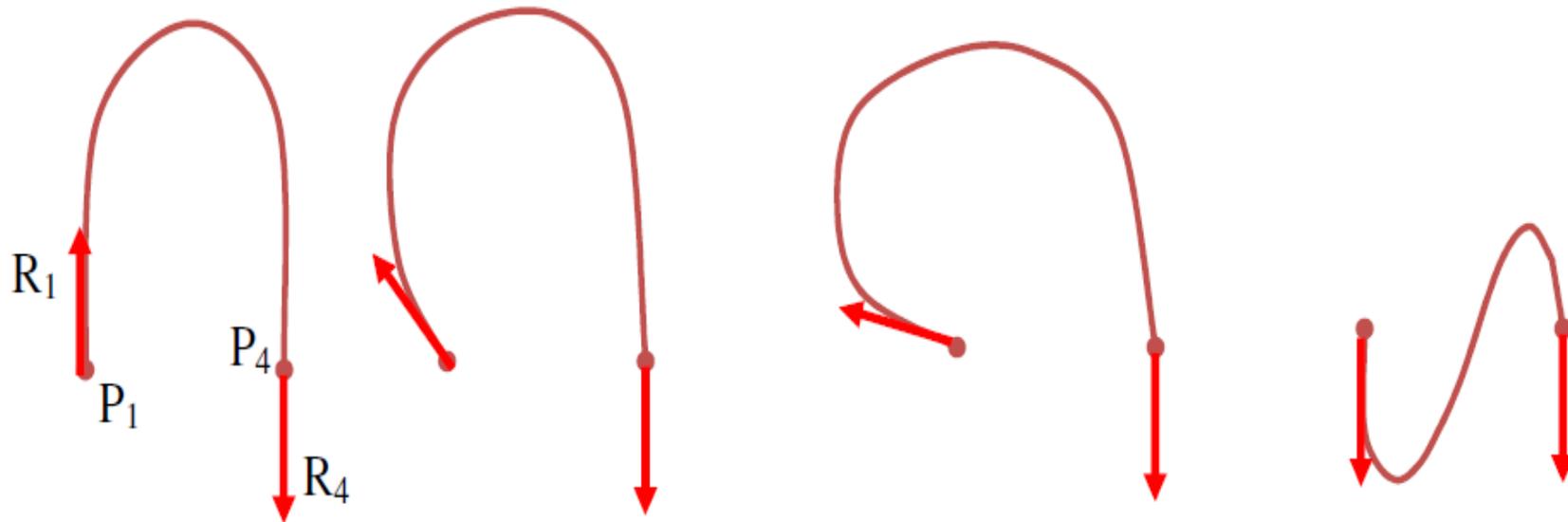
Definida a partir de restrições no ponto inicial e no ponto final.

- Os pontos propriamente ditos:  $P_0$  e  $P_1$
- Vetores tangentes nestes pontos:  $V_0$  e  $V_1$

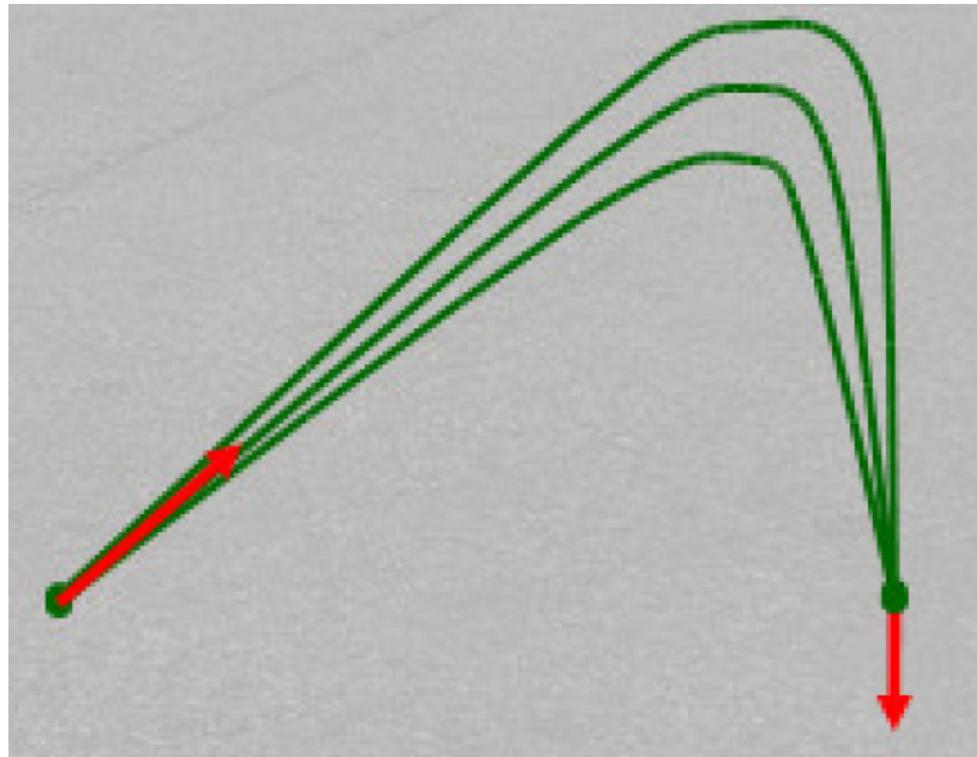


pontos de controle =  $P_i$

**Curvas de Hermite** com os mesmos pontos iniciais e finais, apenas alterando a direção da tangente



**Curvas de Hermite** com os mesmos pontos iniciais e finais, apenas alterando a intensidade da tangente



## Vantagens

- Bem fácil de implementar 😊
- Adequada para aplicações onde seja útil definir a curva em função dos vetores tangentes
- Passa nos pontos de controle (interpolação)

## Desvantagens

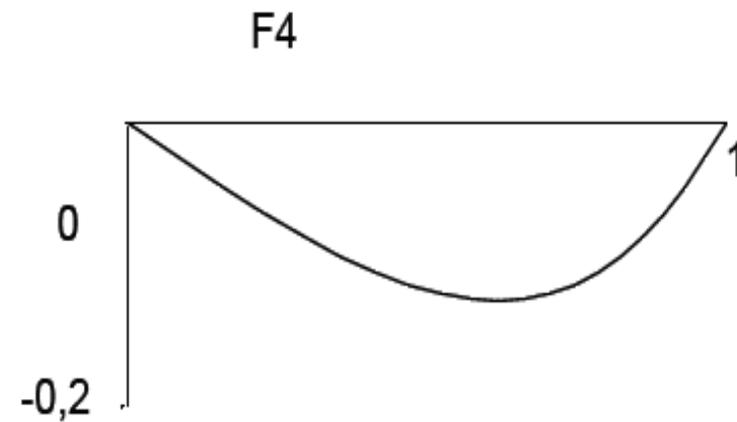
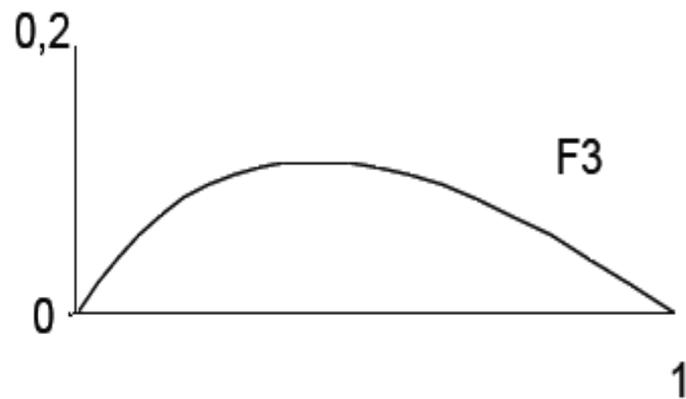
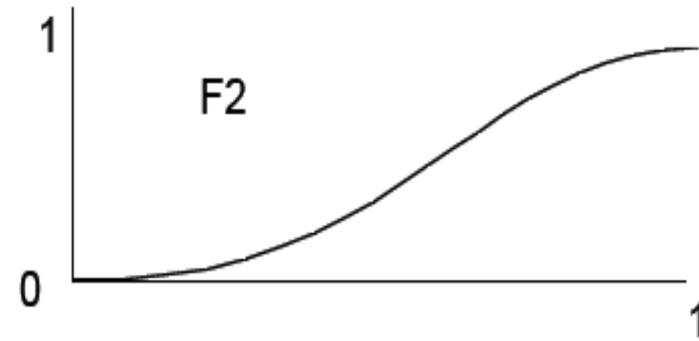
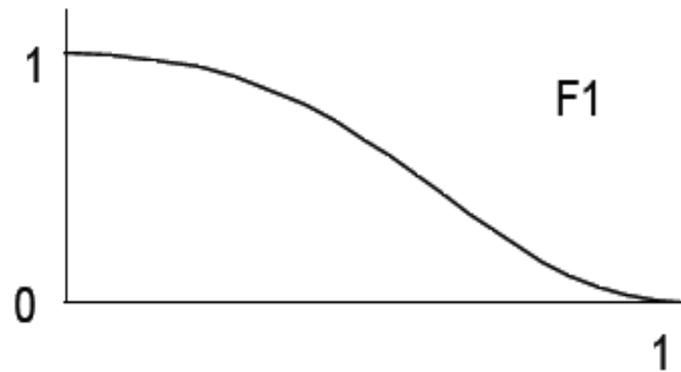
- Não garante, de forma automática, a continuidade entre os segmentos de curva
  - É necessário os vetores tangentes terem a mesma direção e sentido
- Não permite controle local
  - Alteração de um ponto de controle altera toda a curva

# Forma matricial

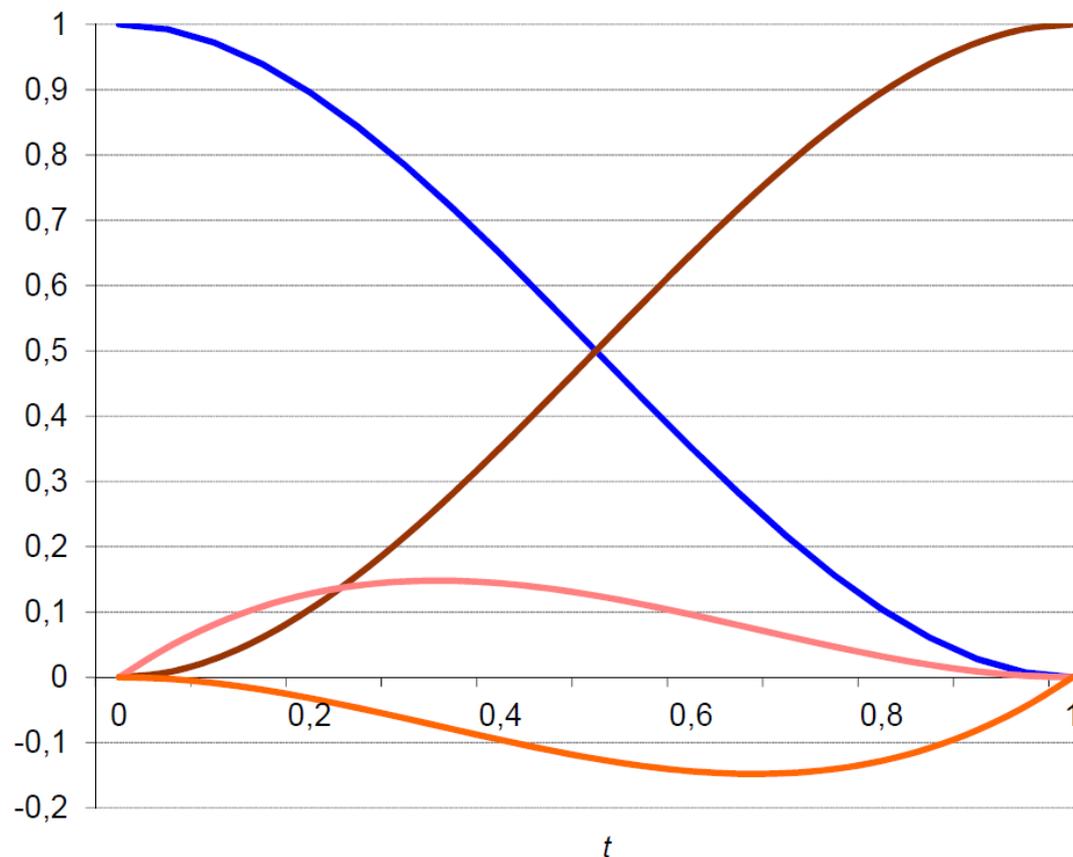
$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{bmatrix}$$

pontos de controle =  $P(0)$  e  $P(1)$

# Funções de mistura de Hermite



## Funções de mistura ou funções interpoladoras de Hermite



### Funções de Mistura:

$$F1(u) = (2u^3 - 3u^2 + 1)$$

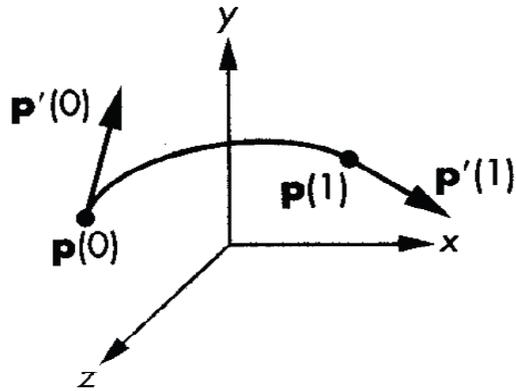
$$F2(u) = (-2u^3 + 3u^2)$$

$$F3(u) = (u^3 - 2u^2 + u)$$

$$F4(u) = (u^3 - u^2)$$

$$p(u) = F1.p(0) + F2.p(1) + F3.p'(0) + F4.p'(1)$$

pontos de controle =  $p$  e  $p'$



pontos de controle =  $p$  e  $p'$

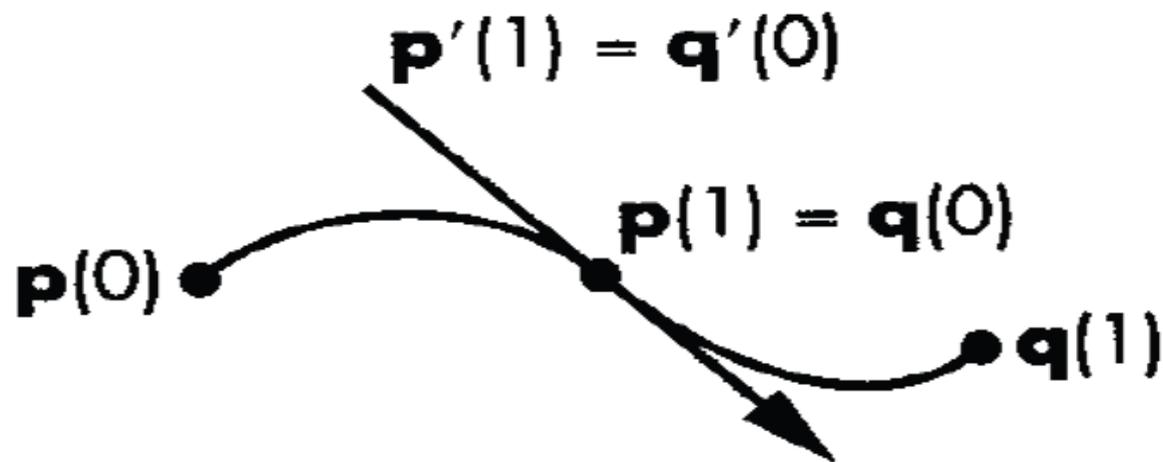
Definindo a curva de Hermite cúbica

$$p(u) = u^T c = u^T M_H p$$

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

portanto  $c = M_H p$

Como fica a curva formada pela união de 2 no ponto de união



pontos de controle =  $p$  para a curva 1 e  $q$  para a curva 2

$$Q_H(t) = TM_H G_H$$

pontos de controle =  $P_i$  e  $R_i$

$G_H$  é

$$G_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$G_{Hx}$  é a componente x de  $G_H$ :

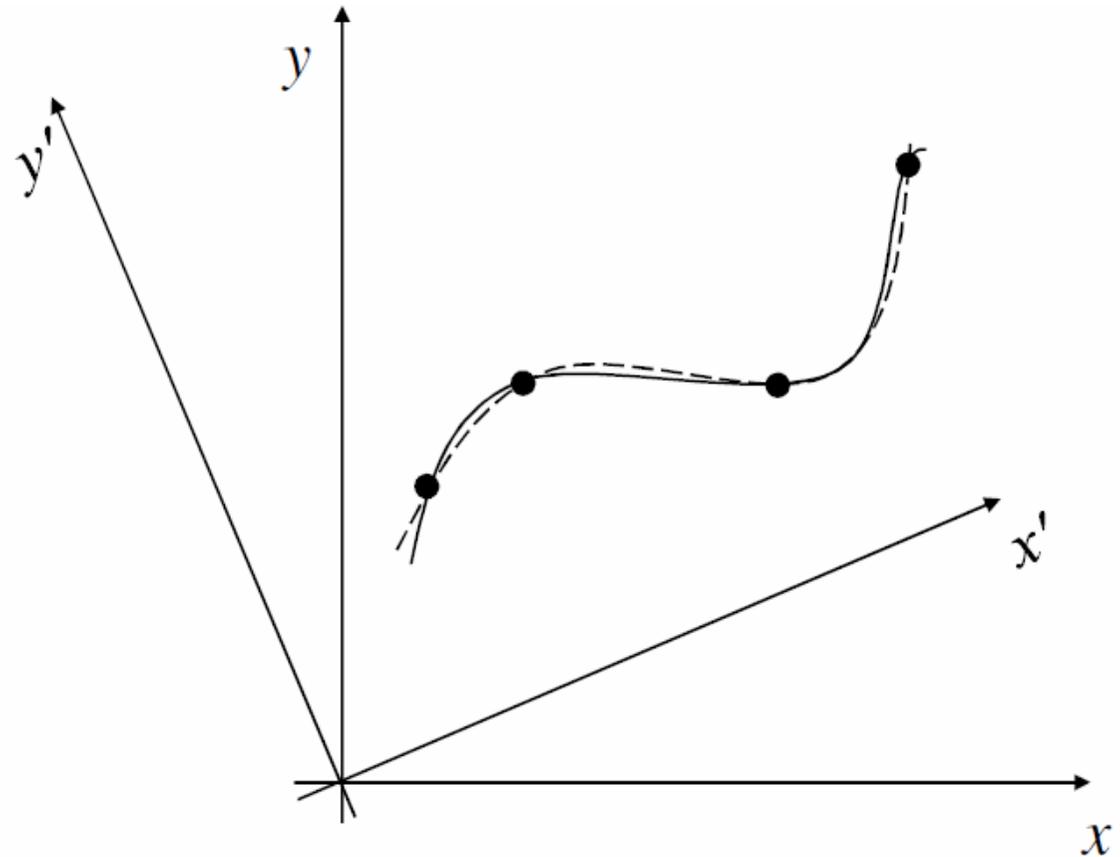
$$G_{Hx} = \begin{bmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{bmatrix}$$

$G_{Hy}$  é a componente y de  $G_H$  e

$G_{Hz}$  é a componente z de  $G_H$

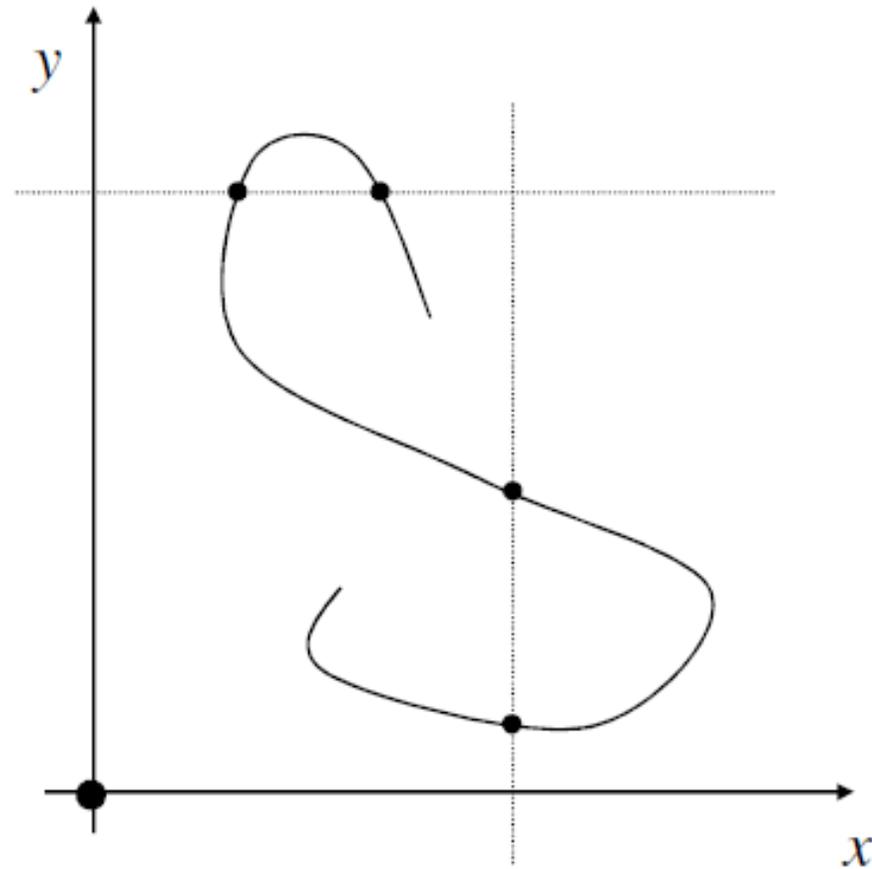
# Propriedades desejáveis de curvas para modelagem em CG

Independência dos eixos usados



# Propriedades desejáveis de curvas para modelagem em CG

Deve  
poder  
ter  
Pontos  
com  
coordenadas  
múltiplas



# Propriedades desejáveis de curvas para modelagem em CG

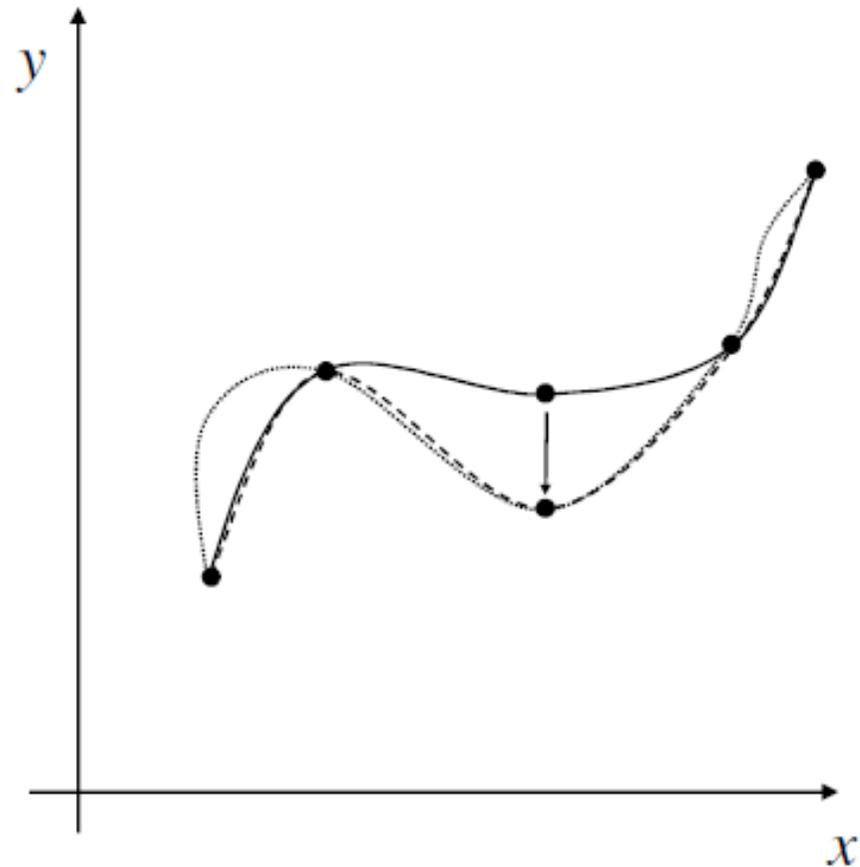
Deve ter uso

**intuitivo** e

poder ter

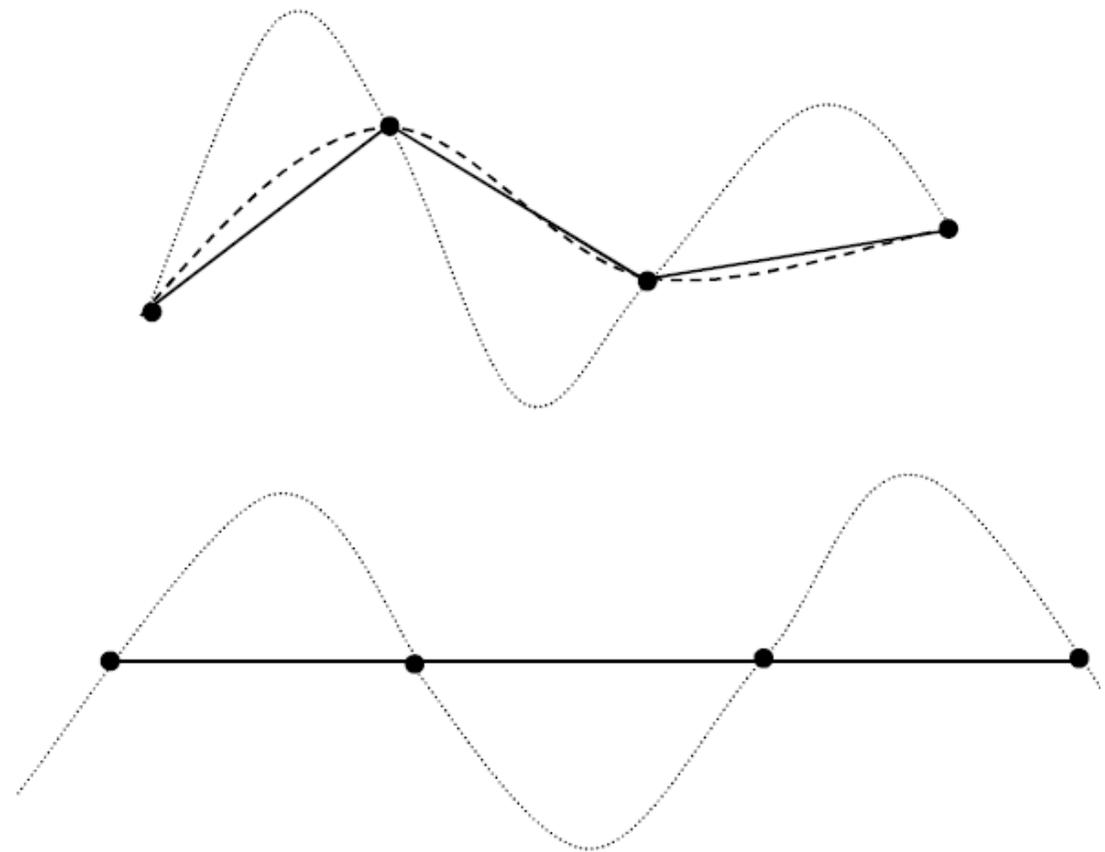
**Controle local:**

i.e. em ajuste finos:  
alterar um trecho  
não altera toda a  
curva



# Propriedades desejáveis de curvas para modelagem em CG

O número de pontos de controle local não deve estar associado ao grau da curva ou sua oscilação

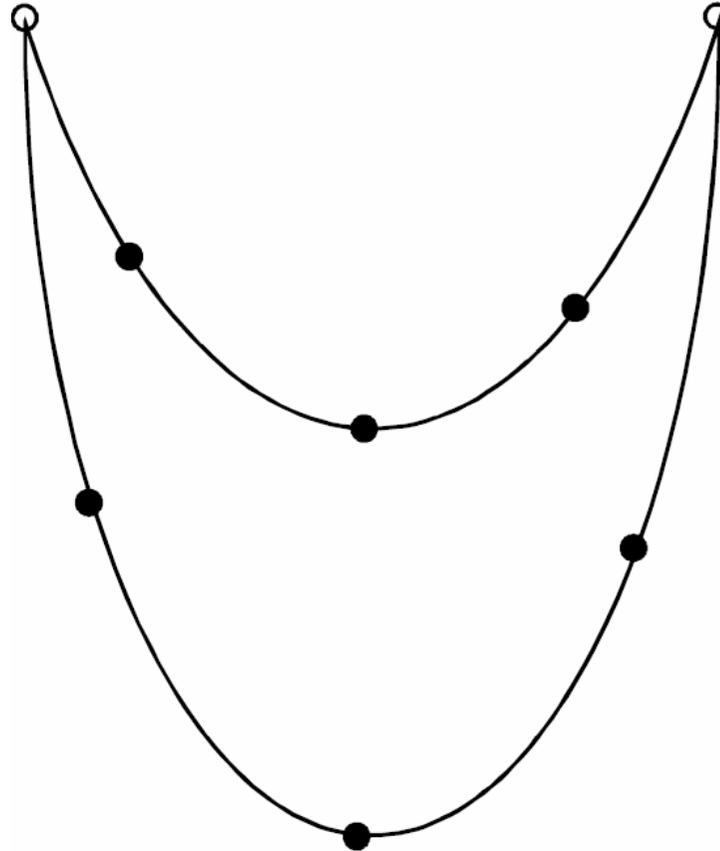


polinômio de grau elevado

formas com oscilações

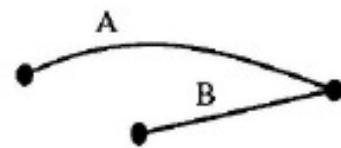
# Propriedades desejáveis de curvas para modelagem em CG

Ser possível  
representar  
diversos graus  
de  
continuidades  
que o usuário  
desejar

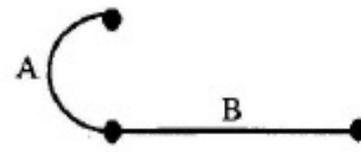


# Continuidade

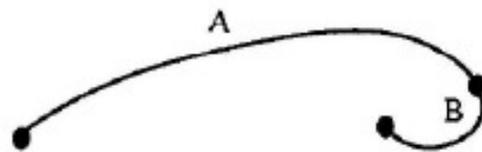
- Duas ou mais curvas nos nós (conectando pontos) para formar uma curva contínua
- Tipos de continuidade
  - Continuidade de ponto
  - Continuidade de tangente
  - Continuidade de curvatura



(a) Point continuity -  $C^0$

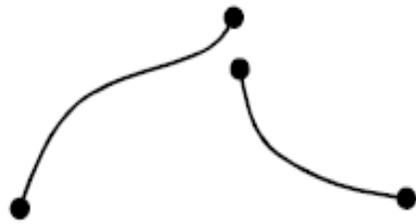


(b) Tangent continuity -  $C^1$



(c) Curvature continuity -  $C^2$

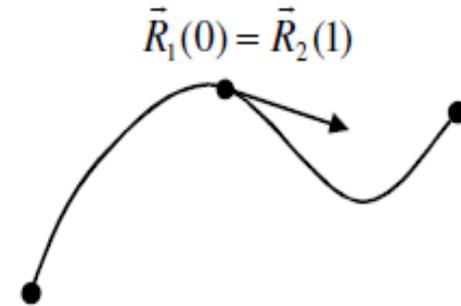
# Continuidades



Descontínua



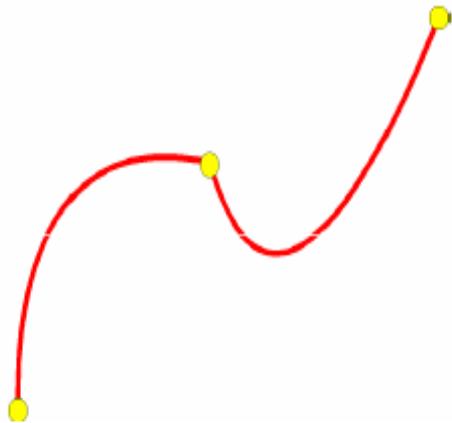
Contínua:  $C^0$  e  $G^0$



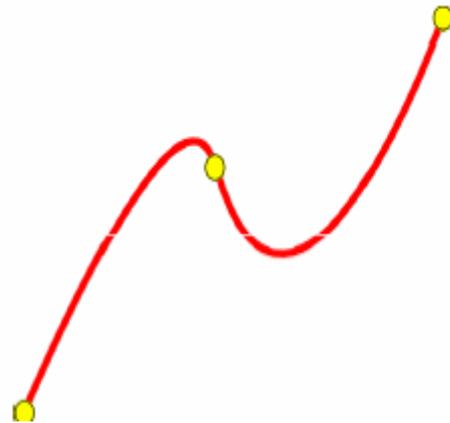
Contínua:  $C^1$  e  $G^1$

continuidades geométrica

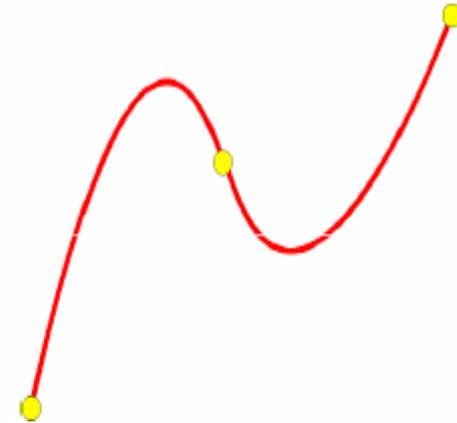
Com Hermite ficam facil de unir 2 curvas e garantir continuidades



Dois segmentos se encontram  
em um ponto



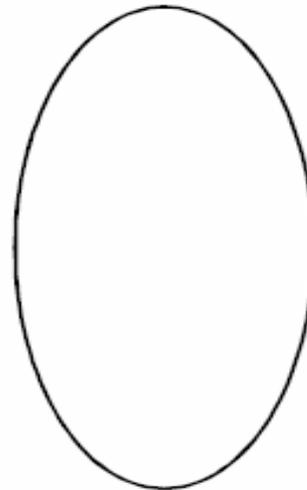
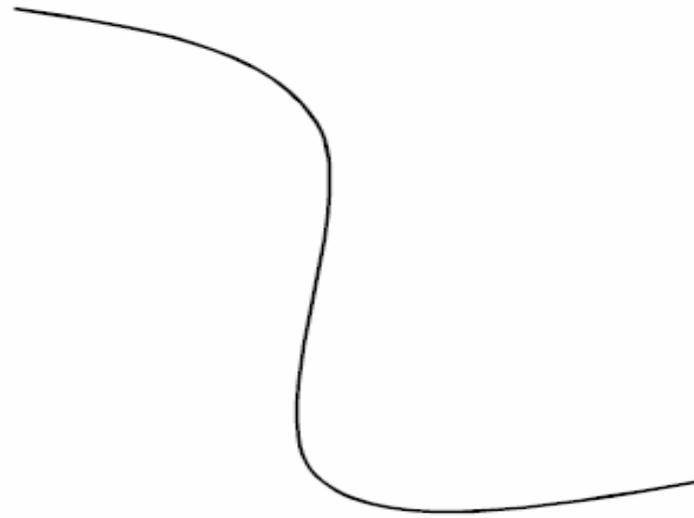
**Direção** das tangentes dos  
segmentos são  
iguais no ponto de junção



**Direção e magnitude** das  
tangentes dos segmentos  
são iguais no ponto de junção

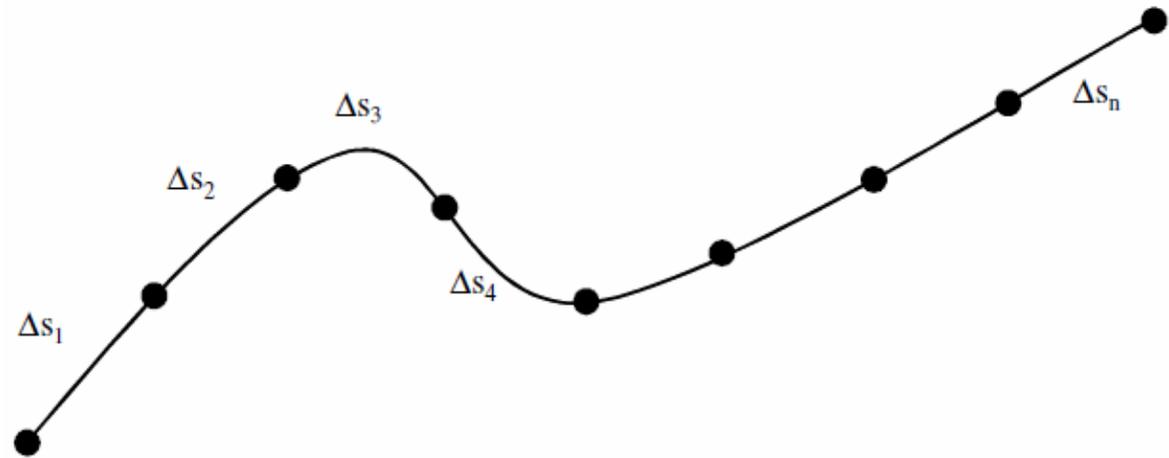
## Propriedades desejáveis de curvas para modelagem em CG

Ser possível  
representar  
curvas abertas,  
fechadas, com  
pontos de  
inflexão, etc. :  
ter a  
**versatilidade**  
que o usuário  
desejar



## Propriedades desejáveis de curvas para modelagem em CG

ter pontos  
com distâncias  
 $\approx$  constantes ao  
longo do seu  
comprimento:  
parâmetro  
uniformemente  
distribuídos.

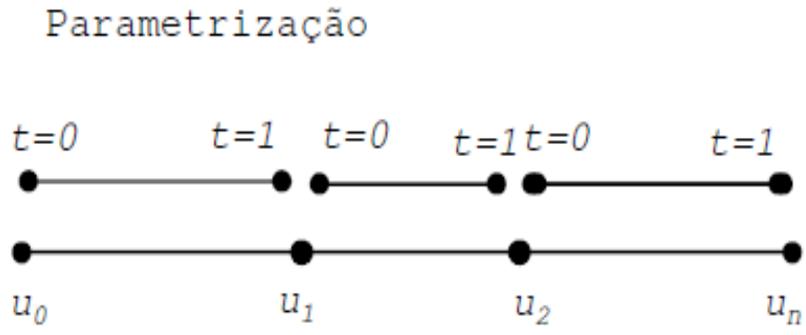
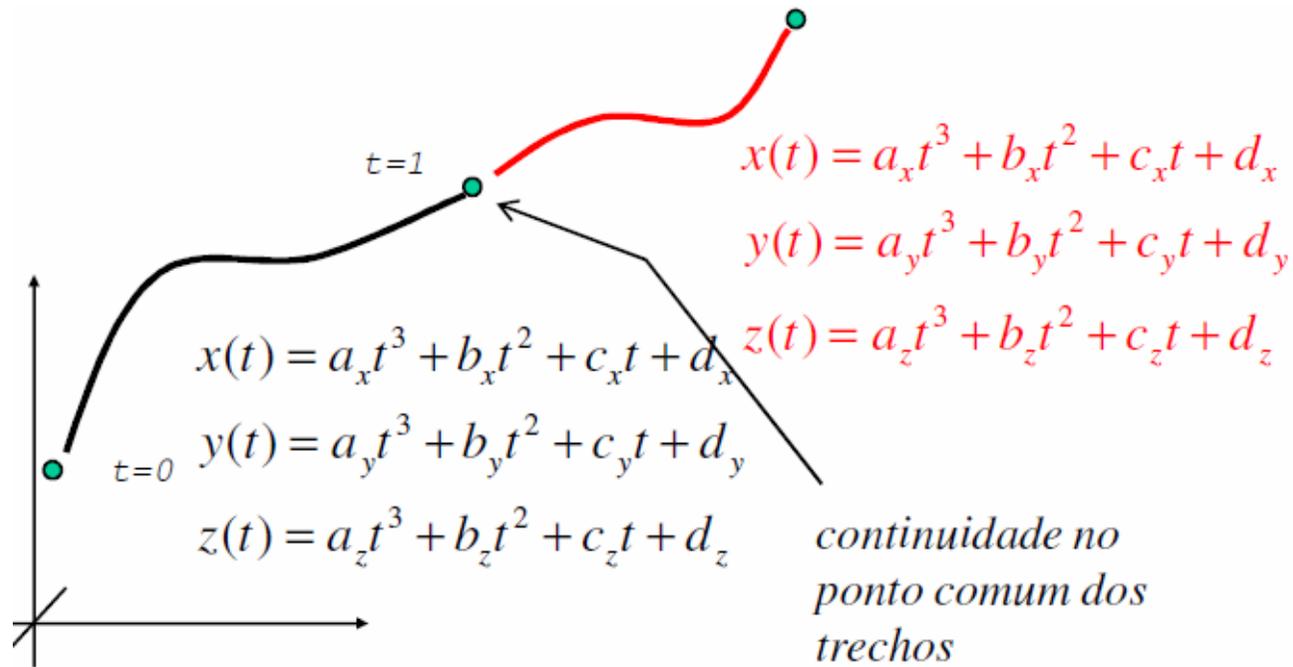


$$\Delta s_i \approx \Delta s_j$$

# Solução em CG

- Curvas de formas livres
- Representáveis por uniões
- Descritas por polinômios
- Parametrizadas
- Até grau 3
- Podendo serem re parametrizadas

# re parametrização



$t \in [0,1]$  local  
 ou  
 $u \in [u_0, u_n]$  global

# Porque geralmente se usa polinômios até terceiro grau?

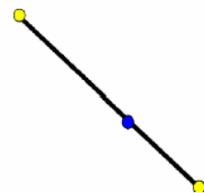
Quanto mais simples a equação da função de interpolação, mais rápida sua avaliação.

Polinômios são fáceis de avaliar

Grau menor que 3: Pouca flexibilidade.

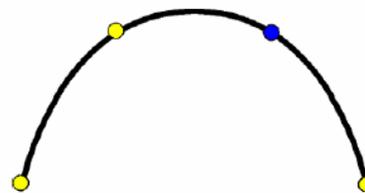
Grau maior que 3: Maior custo computacional com pouca vantagem prática.

$$f(t) = at + b$$



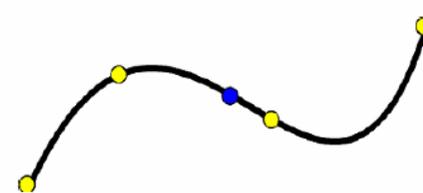
Linear

$$f(t) = at^2 + bt + c$$



Quadrático

$$f(t) = at^3 + bt^2 + ct + d$$



Cúbico

# Curvas Paramétricas Cúbicas 2D

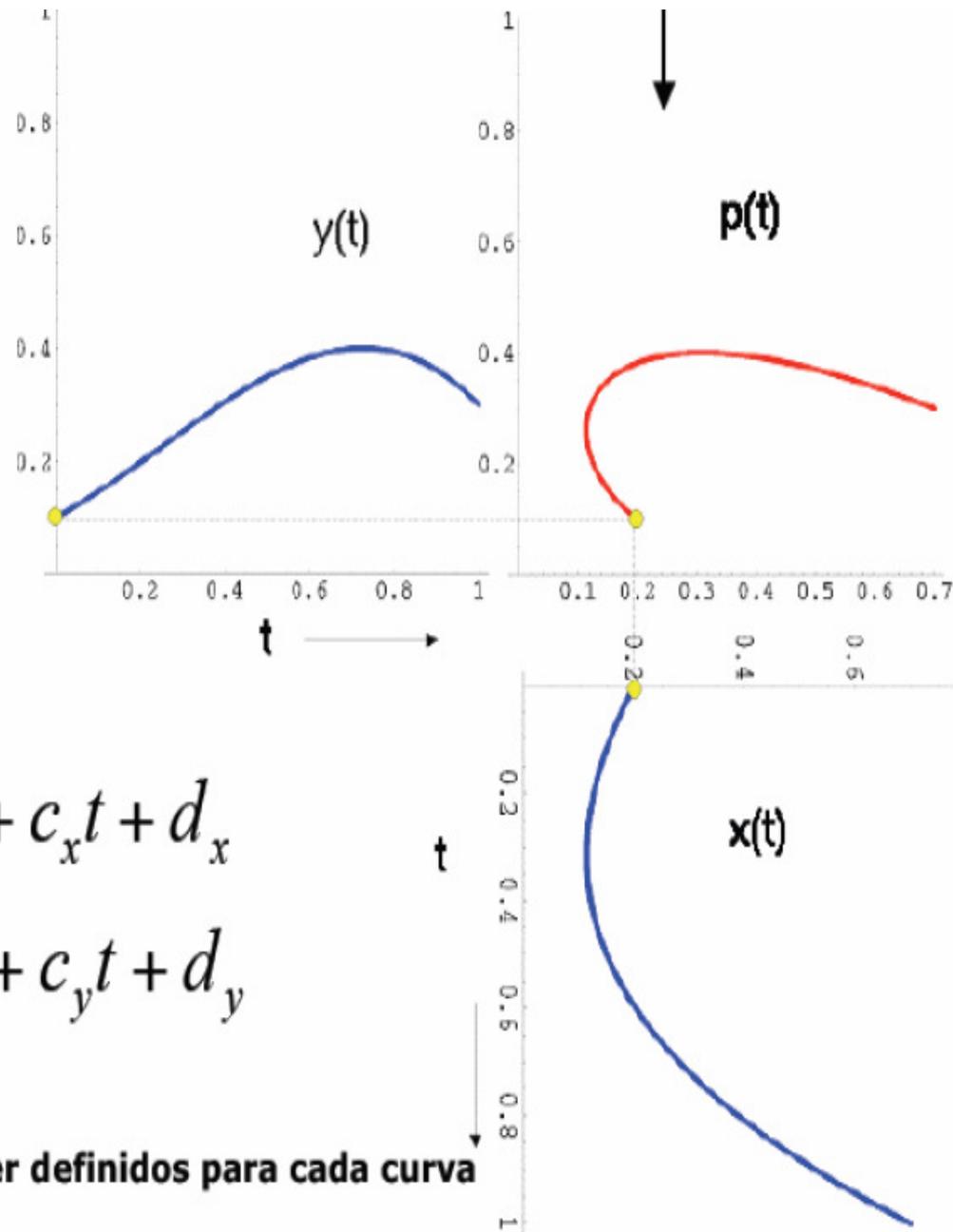
9 parâmetros  
para cada curva

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

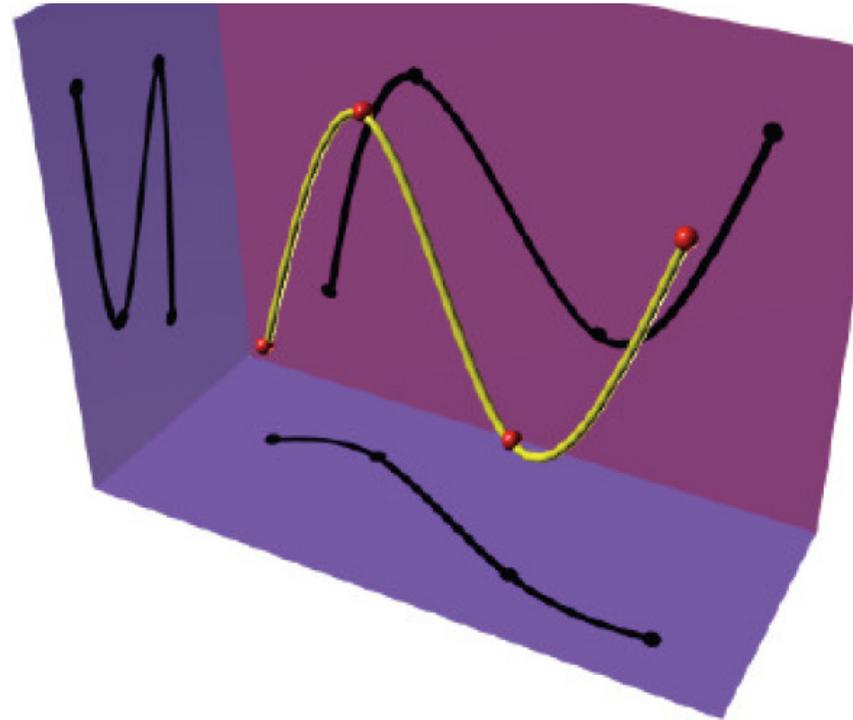


Os valores de  $a_x, b_x$ , etc devem ser definidos para cada curva



# Em 3D

- 12 parâmetros
- para cada curva



$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

# De forma genérica

$$p(u) = c_0 + c_1u + c_2u^2 + c_3u^3 = \sum_{k=0}^3 c_k u^k = \mathbf{u}^T \mathbf{c},$$

onde

$$\mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}, \quad c_k = \begin{bmatrix} c_{kx} \\ c_{ky} \\ c_{kz} \end{bmatrix}$$

Uma *spline* é uma linha flexível usada para produzir uma curva suavizada ao longo de uma série de pontos de controle.

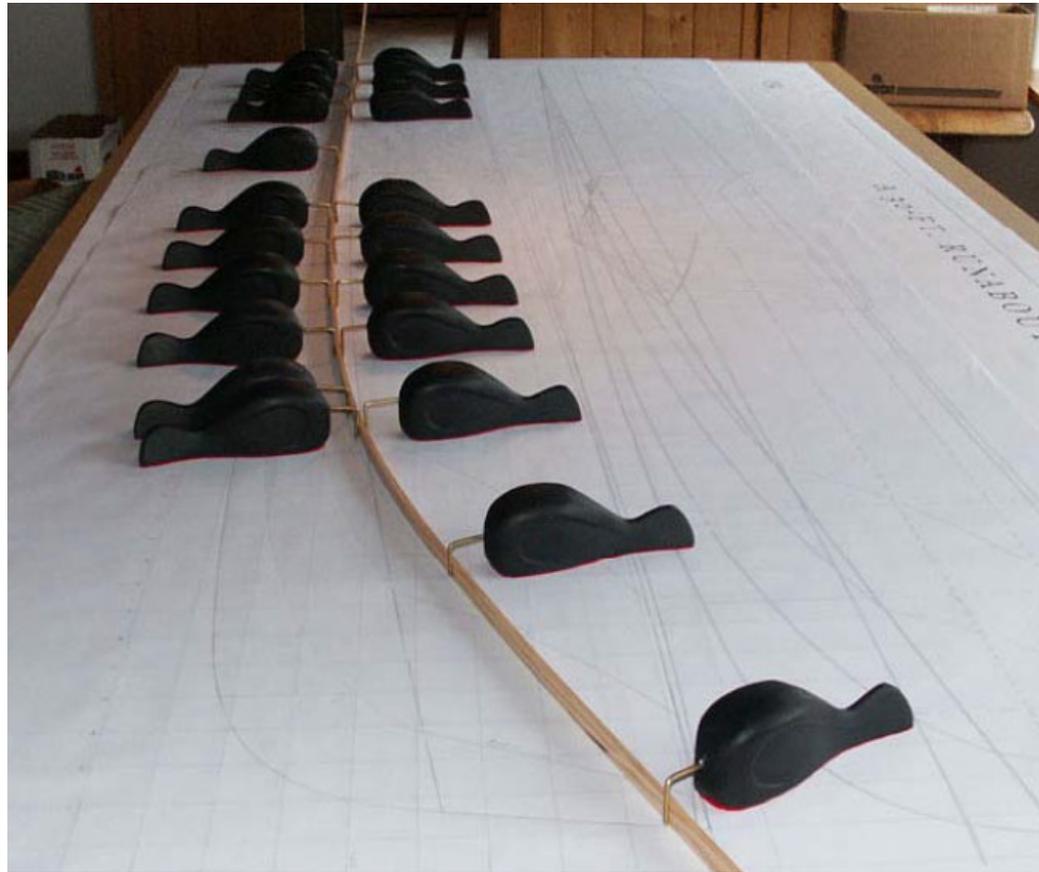
# Curvas Splines

Com maior suavidade que as anteriores (tem **curvatura contínuas**) e são conectadas formando curvas mais complexas.

**Spline** é uma curva polinomial definida por partes

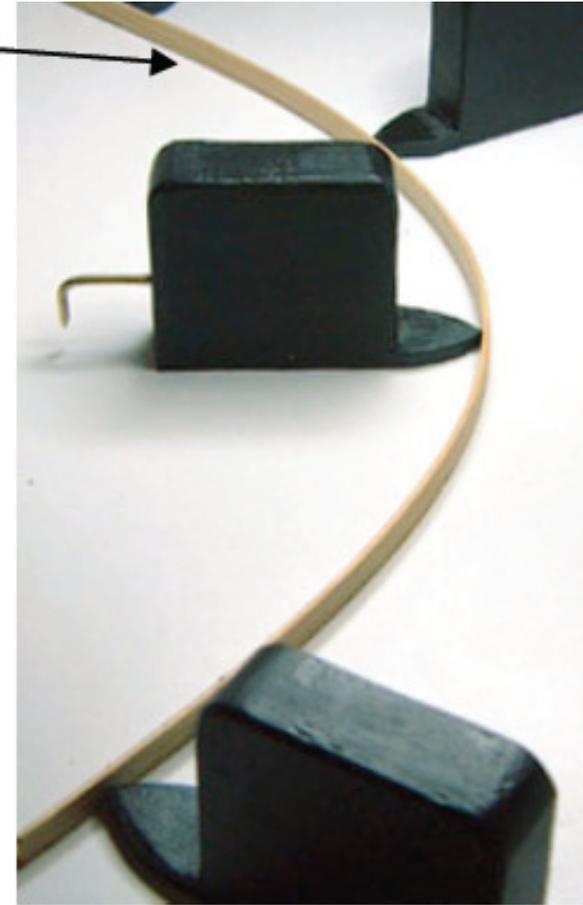
Existem vários tipos de *splines*, cuja amostragem varia de acordo com a fórmula matemática utilizada na sua construção. Elas podem ser interpoladas ou aproximadas.

# Spline física



Metal flexível com continuidade  
de curvatura:  $C^2$

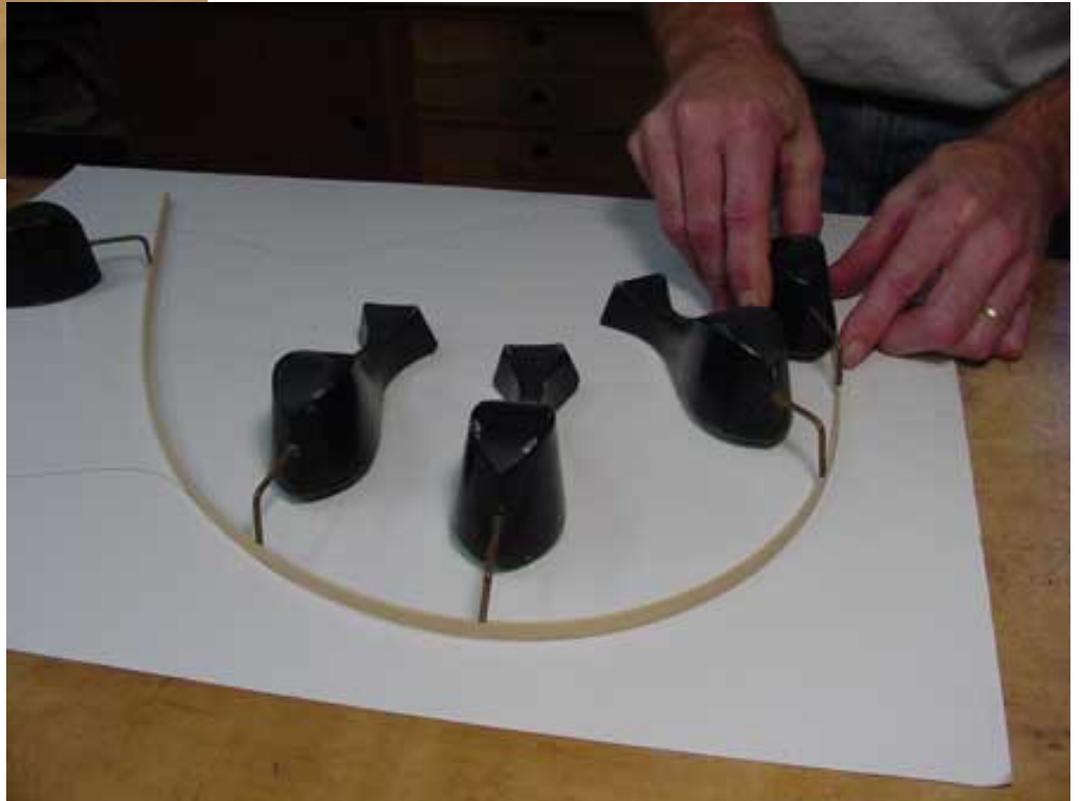
## Construindo barcos



Pesos que dão forma = “ducks”



Exemplo de como são usadas



# Uma equivalência com essa ferramenta de desenho é a

- **Spline Cubica Natural**
- Que tem continuidade  $C^2$  e passa pelos pontos de controle
- Ou seja de cara já tem **um grau a mais de continuidade** (suavidade) que as anteriores.
- Calcular as splines naturais com  $n$  pontos de controle envolve inverter uma matriz de
- **$(n+1) \times (n+1)$  pontos**

# Splines

- A base de Bézier não é própria para a modelagem de curvas longas
  - Bézier única: suporte não local
  - Trechos emendados: restrições não são naturais
- Base alternativa: B-Splines
  - Nome vem de um instrumento usado por desenhistas
  - Modelagem por polígonos de controle sem restrições adicionais
  - Suporte local
    - Alteração de um vértice afeta curva apenas na vizinhança
  - Existem muitos tipos de Splines
  - Se os nós estão equidistantemente distribuídos a spline é **uniforme**, caso contrário é **não-uniforme**.
    - Uma B-spline uniforme de grau  $d$  tem continuidade  $C^{d-1}$

# Especificando curvas

- Pontos de Controle
  - Um conjunto de pontos que influenciam a forma da curva
- Nós
  - Pontos de controle que estão sobre a curva
- Interpolando Splines
  - Curvas que passam através dos pontos de controle (nós)
- Aproximando Splines
  - Pontos de controle meramente influenciam a forma

# Spline é uma curva polinomial definida por partes

- Com maior suavidade que as anteriores (tem **curvatura contínuas**) e são conectadas formando curvas mais complexas (*knots*).

## B-spline ou basis spline

- Cardinal B-splines têm *knots* que são equidistantes uns dos outros.
- Cúbicas tem  $m+1$  pontos de controle onde,  $m \geq 3$
- Polinômios Cúbicos
  - $f(u) = a_x u^3 + b_x u^2 + c_x u + d_x$
  - $u: (0 \leq u \leq 1)$

Definida por quatro pontos de controle ( $P_1, P_2, P_3, P_4$ ).

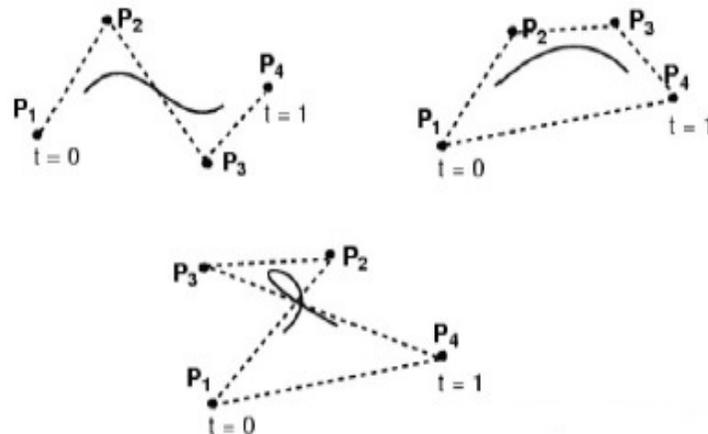
Não passa por nenhum ponto de controle.

– Curva de aproximação

Mais suave que as anteriores

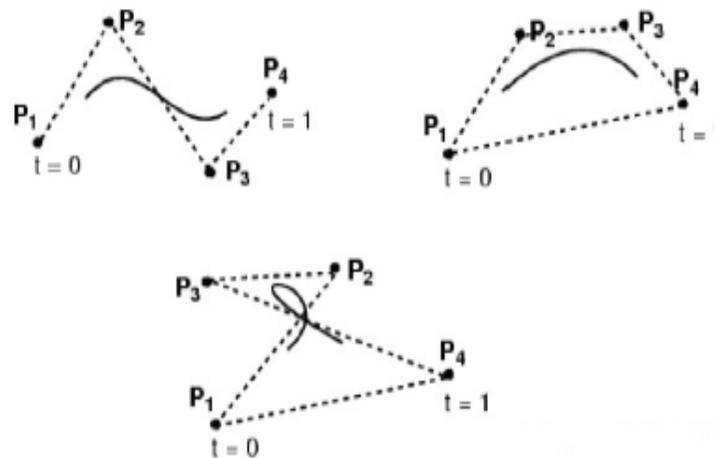
– Mais fácil garantir continuidade paramétrica

Controle local.



Cada segmento é definido por 4 pontos, cada ponto influencia 4 segmentos de curva (exceto  $P_1$  e  $P_4$ )

Knots são os pontos de junção entre os segmentos de curva



# Nós:

No espaço paramétrico global temos **nós** ou **knots** que representam os valores de  $u$  onde os segmentos  $Q_i$  têm os seus extremos. Também são designados por **nós de ligação** uma vez que são os valores de  $u$  onde os seg. de curva se unem

Por definição um  $Q_i$  é definido entre 2 nós consecutivos:  $Q_i$  define um intervalo paramétrico  $u_i \leq u \leq u_{i+1}$  (espaço de  $u$  global)

**B-Spline uniforme:** assume-se que esses nós têm valores inteiros e que o espaçamento entre nós é igual a 1 (0, 1, 2,...)

# B-Splines Uniformes

- Significa que a variável paramétrica está espaçada em intervalos uniformes ( $t=0.1, 0.2, 0.3, \text{ etc}$ )
- Cada um dos  $m-2$  segmentos é definido por 4 dos  $m+1$  pontos de controle
- Segmento  $Q_i = P_{i-3}, P_{i-2}, P_{i-1}, P_i$
- $GS = [P_{i-3} \ P_{i-2} \ P_{i-1} \ P_i] \ 3 \leq i \leq m$

Uma curva B-Spline é calculada por:

$$Q(u) = \sum_{i=0}^m B_i(u) p_i \quad \text{em que } ?? \leq u \leq ??$$

- Cada função de mistura  $B_i$  é suportada no intervalo

$$u_i - > u_{i+k}$$

- Temos  $m+1$  funções de mistura;
- Logo:

$$m + 1 + k \text{ knots } (u_0 -> u_{m+k})$$

**Número de nós:  $n^\circ$  de pts de cntrl + ordem da curva**

# B-Splines Não-Uniformes

- O intervalo entre valores da variável paramétrica não é necessariamente uniforme...
- Logo as funções de *blending* não são as mesmas para cada segmento.... Maior flexibilidade

Funções de mistura

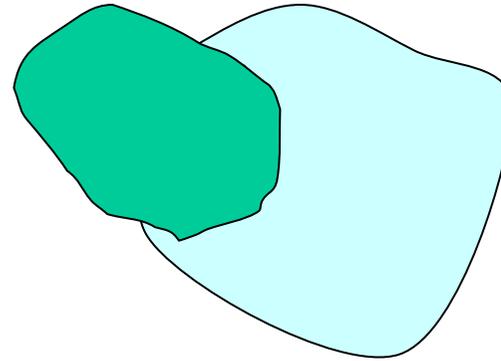


# Curvas Rational B-Spline

- Provê uma única forma matemática precisa capaz de representar as formas analíticas comuns
- Linhas, planos, curvas cônicas incluindo círculos, curvas de forma livre, superfícies quádricas e esculpidas

Invariantes após a aplicação de transformações geométricas simples e a transformação perspectiva (as não racionais “alteram” com a transformação perspectiva).

# Outras Splines



Splines de Catmull-Rom (ou Overhauser) interpolam pontos de controle, de modo similar às splines naturais

As  $\beta$ -splines controlam a forma da curva globalmente através das variáveis  $\beta_1$  (bias) e  $\beta_2$  (tensão)

Bias controla influência dos vetores tangentes enquanto que tensão puxa curva para próximo das linhas que conectam os pontos de controle

# Interpolação com splines cubicas

- Dado um conjunto de  $N+1$  coordenadas de pontos  $(P_0, P_1, P_2, \dots, P_n)$ , qual seria a função paramétrica cúbica que interpola esse pontos, ou seja precisa-se conhecer os coeficientes

$a_x, b_x, c_x, d_x, a_y, b_y, c_y, d_y$  tais que:

$$- f(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$- u: (0 \leq u \leq 1)$$

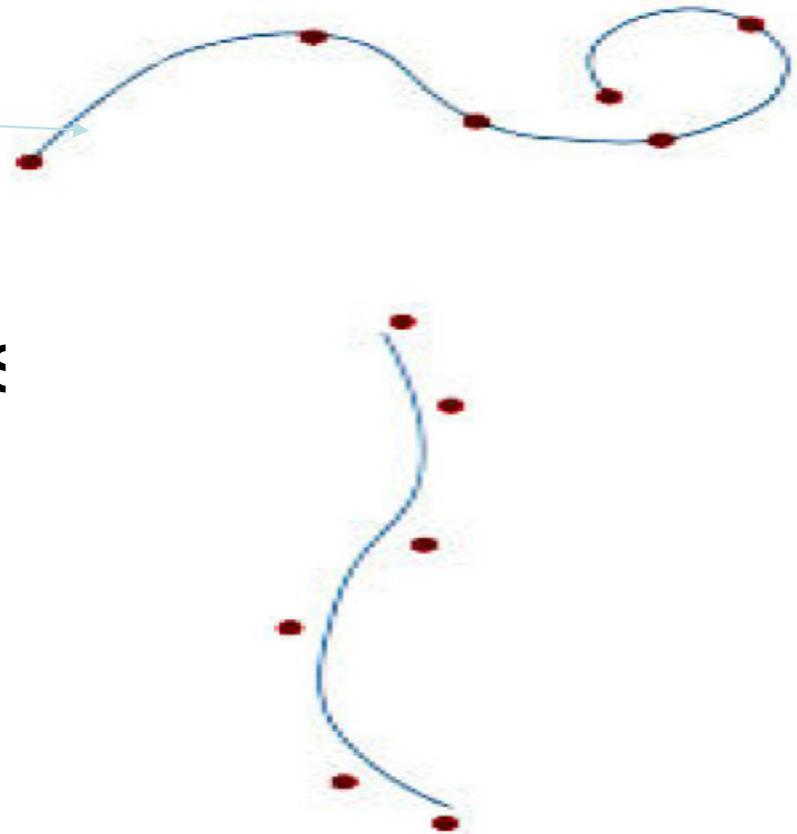
# Interpolação por Splines Cúbicas

- Usa a teoria das vigas e
- É apresentada na seção 1
- de ***Álgebra Linear***
- ***com Aplicações***

Anton e C. Rorres,

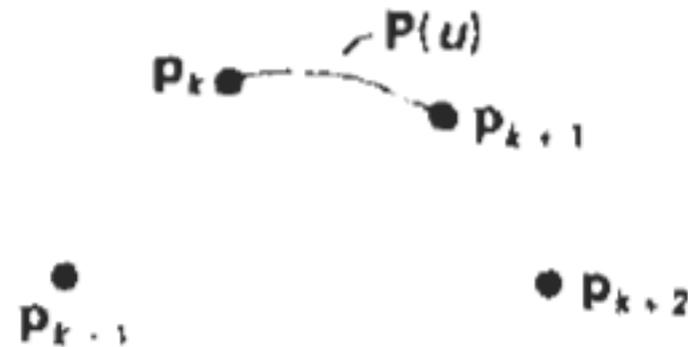
Bookman, 2001

E em Capítulo 10 (seção 10.7) de ***Computer Graphics C version*** de D. Hearne e M.P. Baker



# “Spline” se refere a um grupo de curvas em CG

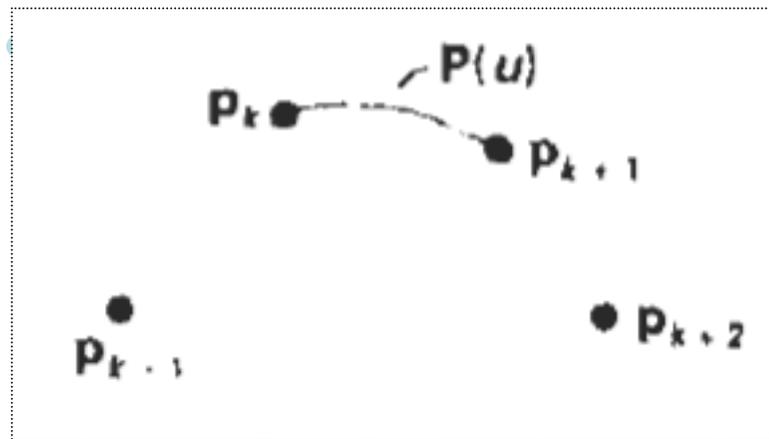
- Por exemplo a **Cardinal** é especificada por 4 pontos de controle consecutivos:



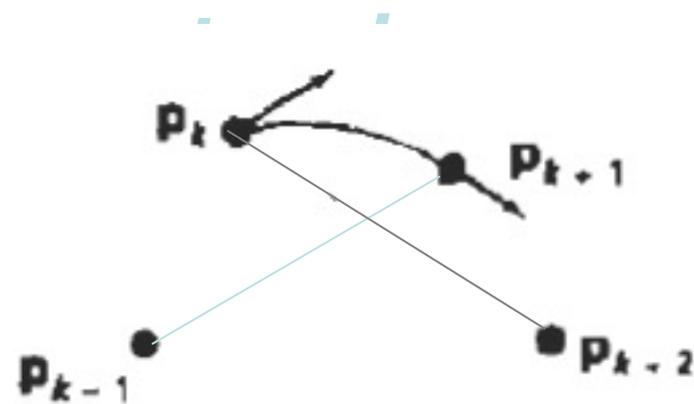
$P(u)$  é a curva e  $P_{k-1}, P_k, P_{k+1}, P_{k+2}$  são os pontos de controle

# “Cardinal Spline”

- Especificada por 4 pontos de controle consecutivos:
- Os 2 do meio definem o início e o fim da curva, e os 2 extremos ajudam a definir as inclinações da mesma nas extremidades, usando também o ponto seguinte:
- *$P(u)$  é a curva*

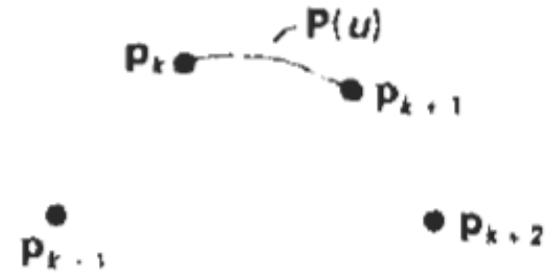


$\Rightarrow$  Os



# “Cardinal Spline”

- $P(u)$  é a curva
- $P_{k-1}, P_k, P_{k+1}, P_{k+2} \Rightarrow$  pontos de controle



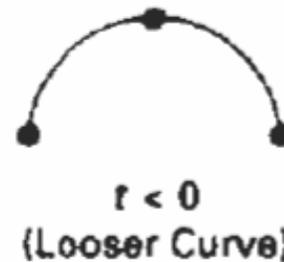
Especifica ainda um parâmetro de *tensão*  $t$  que junto com os pontos extremos ajudam a definir a influência das inclinações ao longo da curva pela expressão seguinte:

$$P(0) = P_k$$

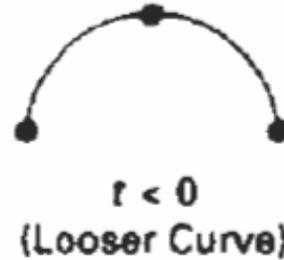
$$P(1) = P_{k+1}$$

$$P'(0) = \frac{1}{2}(1-t)(P_{k+1} - P_{k-1})$$

$$P'(1) = \frac{1}{2}(1-t)(P_{k+2} - P_k)$$



# “Cardinal Spline”



Se o parametro de *tensão*  $t = 0$  a

curva é chamada de **Catmull-Rom spline** ou **Overhauser spline**:

- Matricialmente ela fica:

$$\mathbf{P}(u) = [u^3 \ u^2 \ u \ 1] \cdot \mathbf{M}_C \cdot \begin{bmatrix} \mathbf{P}_{k-1} \\ \mathbf{P}_k \\ \mathbf{P}_{k+1} \\ \mathbf{P}_{k+2} \end{bmatrix}$$

$$s = (1 - t)/2.$$

$$\mathbf{M}_C = \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

# “Cardinal Spline”

Expandindo as expressões:

$$s = (1 - t)/2.$$

$$\mathbf{M}_C = \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

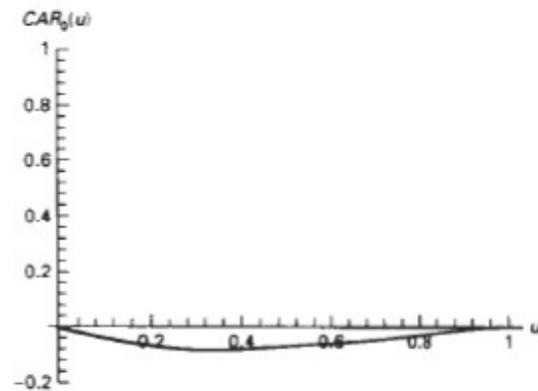
$$\mathbf{P}(u) = [u^3 \ u^2 \ u \ 1] \cdot \mathbf{M}_C \cdot \begin{bmatrix} \mathbf{P}_{k-1} \\ \mathbf{P}_k \\ \mathbf{P}_{k+1} \\ \mathbf{P}_{k+2} \end{bmatrix}$$

$$\begin{aligned} \mathbf{P}(u) &= \mathbf{p}_{k-1}(-su^3 + 2su^2 - su) + \mathbf{p}_k[(2-s)u^3 + (s-3)u^2 + 1] \\ &\quad + \mathbf{p}_{k+1}[(s-2)u^3 + (3-2s)u^2 + su] + \mathbf{p}_{k+2}(su^3 - su^2) \\ &= \mathbf{p}_{k-1}CAR_0(u) + \mathbf{p}_kCAR_1(u) + \mathbf{p}_{k+1}CAR_2(u) + \mathbf{p}_{k+2}CAR_3(u) \end{aligned}$$

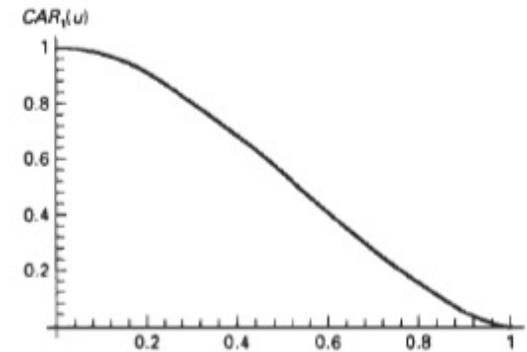
- Onde  $CAR_0$ ,  $CAR_1$ ,  $CAR_2$ ,  $CAR_3$  são as funções de mistura ou interpoladoras da *Spline Cardinal*:

# “Cardinal Spline”

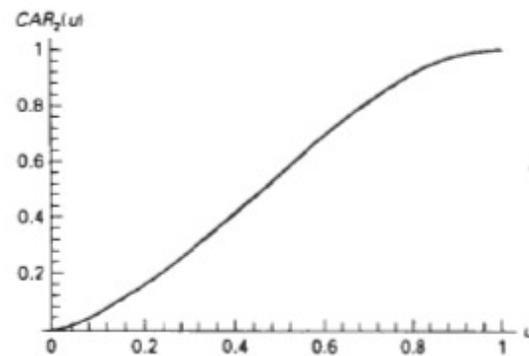
- Como ficam as  $CAR_0$ ,  $CAR_1$ ,  $CAR_2$ ,  $CAR_3$  - funções de mistura ou interpoladoras da *Spline Cardinal para  $t=0$  e  $s=1/2$* :



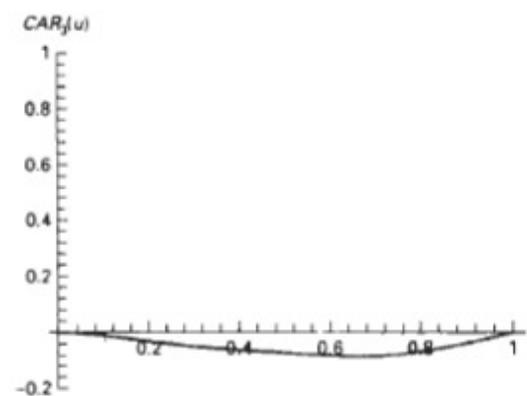
(a)



(b)



(c)



(d)

# Outras Splines

Splines de Catmull-Rom (ou Overhauser) interpolam pontos de controle, de modo similar às splines naturais

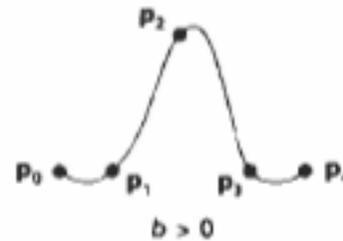
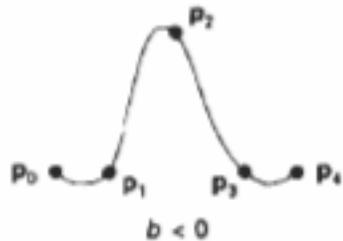
As  $\beta$ -splines controlam a forma da curva globalmente através das variáveis  $\beta_1$  (bias) e  $\beta_2$  (tensão)

Bias controla influência dos vetores tangentes enquanto que tensão puxa curva para próximo das linhas que conectam os pontos de controle

*Computer Graphics C version* de D. Hearne e M.P. Baker , p. 345-357

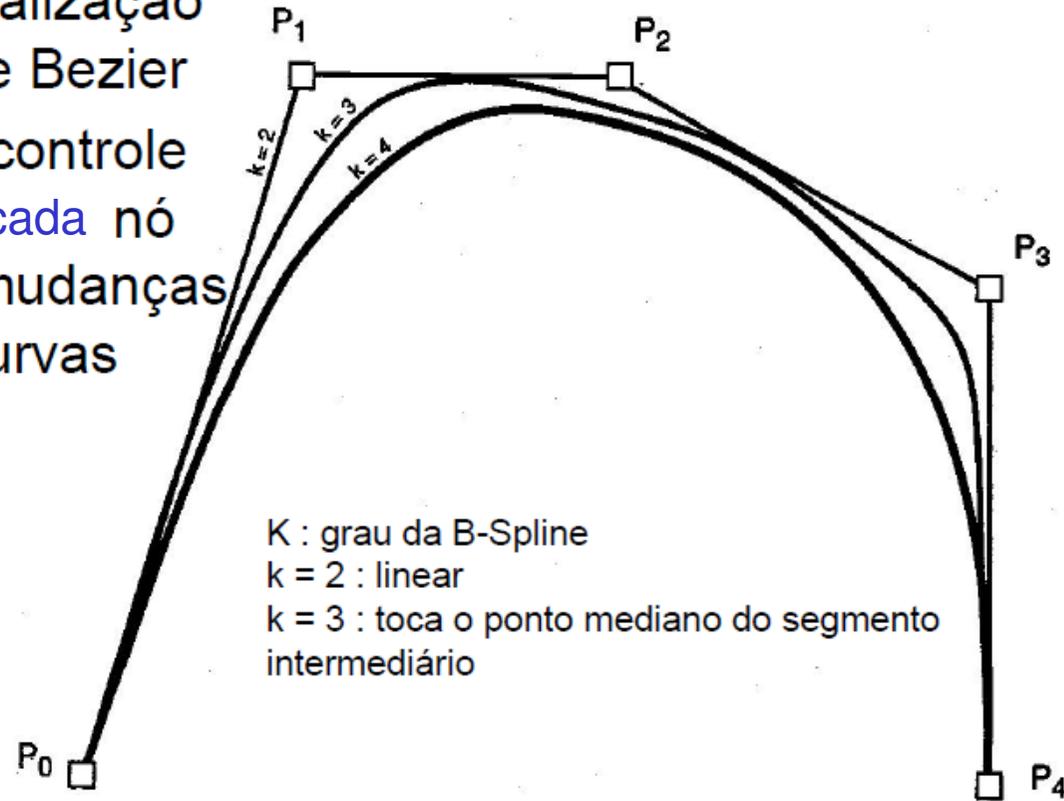
# Outras Splines

- As Korchanek-Bartel splines, são uma classe de Cardinais que além do parametro de tensão incluem mais dois:
- $b$  – bias e  $c$  – continuidade
- Dando assim ainda mais poder de flexibilidade a interpolação por splines (***Computer Graphics C version*** de D. Hearne e M.P. Baker , p. 325)



# B-Splines – é a forma de aproximar por Splines mais usada

- Uma generalização da curva de Bezier
- Pontos de controle adicionais **cada** nó permite mudanças locais às curvas



# Curvas Splines

- A base de Bézier não é própria para a modelagem de curvas longas
  - Bézier única: suporte não local
  - Trechos emendados: restrições não são naturais
- Base alternativa: B-Splines
  - Nome vem de um instrumento usado por desenhistas
  - Modelagem por polígonos de controle sem restrições adicionais
  - Suporte local
    - Alteração de um vértice afeta curva apenas na vizinhança
  - Existem muitos tipos de Splines
  - Se os nós estão equidistantemente distribuídos a spline é **uniforme**, caso contrário é **não-uniforme**.
    - Uma B-spline uniforme de grau  $d$  tem continuidade  $C^{d-1}$

# Especificando curvas

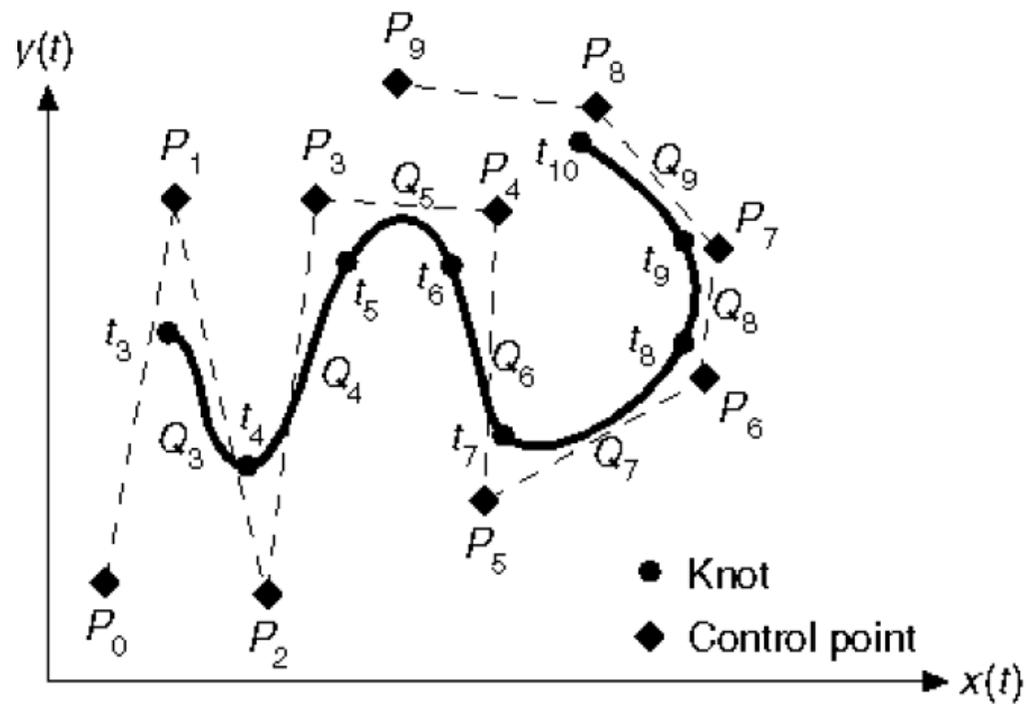
- Pontos de Controle
  - Um conjunto de pontos que influenciam a forma da curva
- Nós
  - Pontos de controle que estão sobre a curva
- Interpolando Splines
  - Curvas que passam através dos pontos de controle (nós)
- Aproximando Splines
  - Pontos de controle meramente influenciam a forma

## B-spline ou basis spline

- Características.
- o grau do polinomio interpolador é independente do número de pontos de controle ,  $m$  , *dentro de certos limites*
- Permite **controle local da forma da curva**, pelos  $p_i$  pontos de controle .

*Computer Graphics C version* de D. Hearne e M.P. Baker , p. 334-345

# Nós $\neq$ pontos de controle



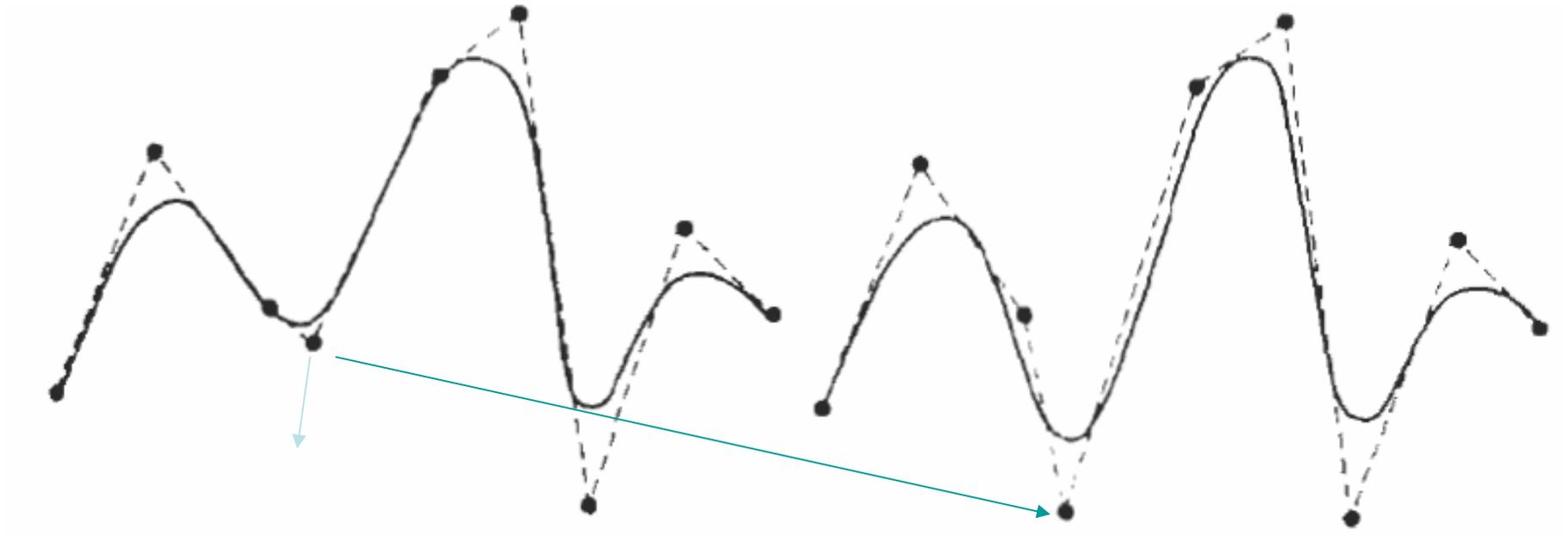
# Nós:

No espaço paramétrico global temos **nós** ou **knots** que representam os valores de  $u$  onde os segmentos  $Q_i$  têm os seus extremos. Também são designados por **nós de ligação** uma vez que são os valores de  $u$  onde os seg. de curva se unem

Por definição um  $Q_i$  é definido entre 2 nós consecutivos:  $Q_i$  define um intervalo paramétrico  $u_i \leq u \leq u_{i+1}$  (espaço de  $u$  global)

**B-Spline uniforme:** assume-se que esses nós têm valores inteiros e que o espaçamento entre nós é igual a 1 (0, 1, 2,...)

# Exemplo controle local



Uma curva B-Spline é calculada por:

$$Q(u) = \sum_{i=0}^m B_i(u) p_i \quad \text{em que } ?? \leq u \leq ??$$

- Cada função de mistura  $B_i$  é suportada no intervalo

$$u_i - > u_{i+k}$$

- Temos  $m+1$  funções de mistura;
- Logo:

$$m + 1 + k \text{ knots } (u_0 -> u_{m+k})$$

**Número de nós:  $n^o$  de pts de cntrl + ordem da curva**

Para gerar interpolações lineares, tem-se  $k=2$ , e a curva passa a ser descrita pelas funções:

$$N_{i,2}(t) = \begin{cases} \frac{(t - t_i)}{(t_{i+2} - t_i)} & \text{se } t_i \leq t \leq t_{i+1} \\ \frac{(t_{i+2} - t)}{(t_{i+2} - t_{i+1})} & \text{se } t_{i+1} \leq t \leq t_{i+2} \end{cases}$$

Dependendo do vetor de nós escolhido pode-se ter curvas uniformes e periódicas, não periódicas ou não uniformes. Se o desejado for uma B-Spline periódica definida a intervalos iguais de 1 a partir de 0(zero) teremos

$$N_{i,2}(t) = \begin{cases} t & \text{para } 0 \leq t \leq 1 \\ 2 - t & \text{se } 1 \leq t \leq 2 \end{cases}$$

Para gerar interpolações quadráticas tem-se  $k=3$  e as funções são definidas recursivamente como

$$N_{i,3}(t) = \frac{(t - t_i)}{(t_{i+2} - t_i)} N_{i,2}(t) + \frac{(t_{i+3} - t)}{(t_{i+3} - t_{i+1})} N_{i+1,2}(t)$$

Onde o valor de  $N_{i+1,2}(t)$  pode ser obtido da expressão anterior. Assim se forem usados intervalos iguais de  $t$  a partir de zero para o vetor de nós, tem-se:

$$= \begin{cases} 1/2 t^2 & \text{se } 0 \leq t \leq 1 \\ \frac{3}{4} - \left( t - \left( \frac{3}{2} \right) \right)^2 & \text{se } 1 \leq t \leq 2 \\ 1/2 (3 - t)^2 & \text{se } 2 \leq t \leq 3 \end{cases}$$

# Splines 2D:

- Spline 2D : a geração de segmentos de curvas que sejam controladas por 4 pontos dados de maneira uniforme, é equivalente a implementar a equação:

$$Q_{BS}(t) = \frac{(1-t)^3}{6} P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6} P_{i-2} +$$
$$\frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{i-1} + \frac{t^3}{6} P_i$$
$$0 \leq t \leq 1$$

- Chamada Uniforme B-splines se têm *knots* que são equidistantes uns dos outros em função do parâmetro  $u$ .
- Muito usadas em CG são as Cúbicas com  $m+1$  pontos de controle,  $P_0, P_1, P_2, \dots, P_m$  onde,  $m > 3$
- Polinômios Cúbicos
  - $f(u) = a_x u^3 + b_x u^2 + c_x u + d_x$
  - $u: (0 \leq u \leq 1)$

As funções de interpolação cúbica ( $k=4$ ) para o mesmo conjunto de nós (periódicos e uniformes) serão:

$$N_{i,4}(t) = \begin{cases} 1/6 t & 0 \leq t \leq 1 \\ 2/3 - 1/2(t-2)^3 - (t-2)^2 & 1 \leq t \leq 2 \\ 2/3 - 1/2(t-2)^3 - (t-2)^2 & 2 \leq t \leq 3 \\ 1/6 (4-t)^3 & 3 \leq t \leq 4 \end{cases}$$

Do mesmo modo pode-se recursivamente gerar qualquer tipo de B-Spline, não periódico ou não uniforme apenas escolhendo adequadamente os vetores de nós.

Definida por quatro pontos de controle ( $P_1, P_2, P_3, P_4$ ).

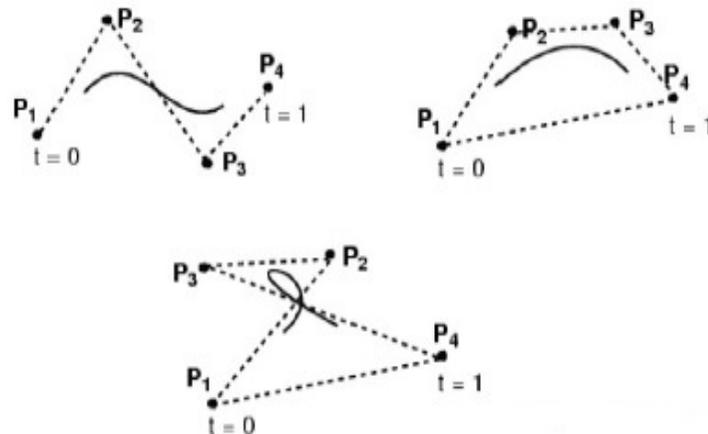
Não passa por nenhum ponto de controle.

– Curva de aproximação

Mais suave que as anteriores

– Mais fácil garantir continuidade paramétrica

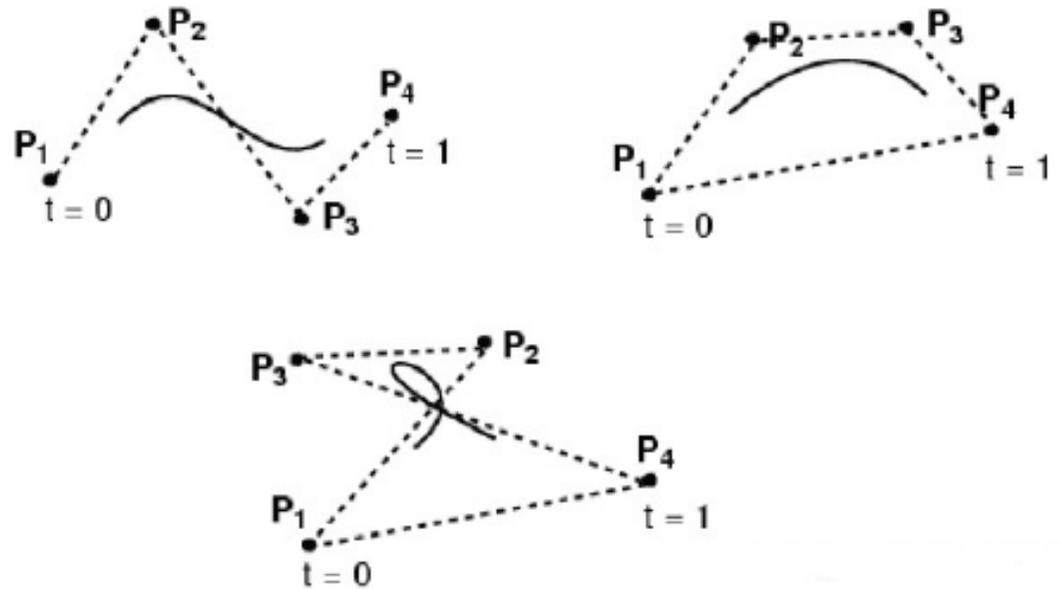
Controle local.



Porque são melhores? Porque fornecem **suporte local** e mais controle (ao contrário da Bézier que por mínima mudança nos pontos de controle, gera uma nova curva)

Genericamente:

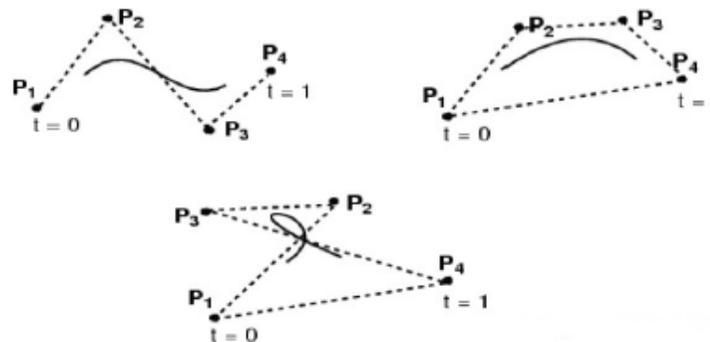
- Para  $m+1$  pontos de controle
  - $M \geq 3$   $P_0, P_1, \dots, P_n$
- Teremos curvas com  $m-2$  segmentos
  - $Q_3, Q_4, \dots, Q_m$



A curva inteira B-spline é considerada composta por segmentos de curvas spline

Cada segmento é definido por 4 pontos, cada ponto influencia 4 segmentos de curva (exceto  $P_0$  e  $P_n$ )

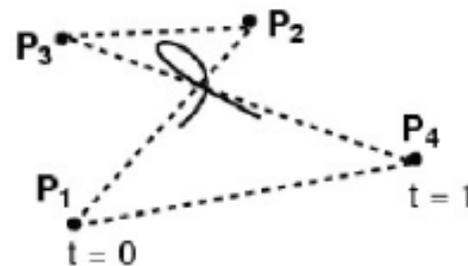
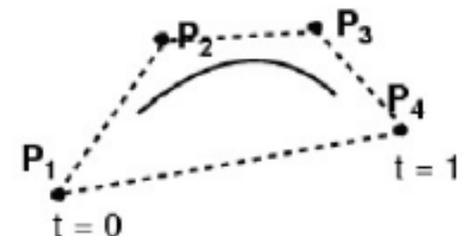
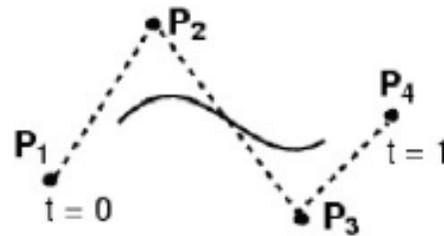
Knots são os pontos de junção entre os segmentos de curva



Porque são melhores? Porque fornecem **suporte local** e mais controle (ao contrário da Bézier que por mínima mudança nos pontos de controle, gera uma nova curva)

Genericamente:

- Para  $m+1$  pontos de controle
  - $M \geq 3$   $P_1, P_2, \dots, P_n$
- Teremos curvas com  $m-2$  segmentos
  - $Q_3, Q_4, \dots, Q_m$



# B-Splines Uniformes

- Significa que a variável paramétrica está espaçada em intervalos uniformes ( $t=0.1, 0.2, 0.3, \text{etc}$ )
- Cada um dos  $m-2$  segmentos é definido por 4 dos  $m+1$  pontos de controle
- Segmento  $Q_i = P_{i-3}, P_{i-2}, P_{i-1}, P_i$
- $GS = [P_{i-3} \ P_{i-2} \ P_{i-1} \ P_i] \ 3 \leq i \leq m$

## Spline controlada por 4 pontos

$$Q_i(u) = \sum_{j=0}^3 B_{i-j}(u) p_{i-j}$$

$$B_i = \frac{1}{6} u^3$$

$$B_{i-1} = \frac{1}{6} (-3u^3 + 3u^2 + 3u + 1)$$

$$B_{i-2} = \frac{1}{6} (3u^3 - 6u^2 + 4)$$

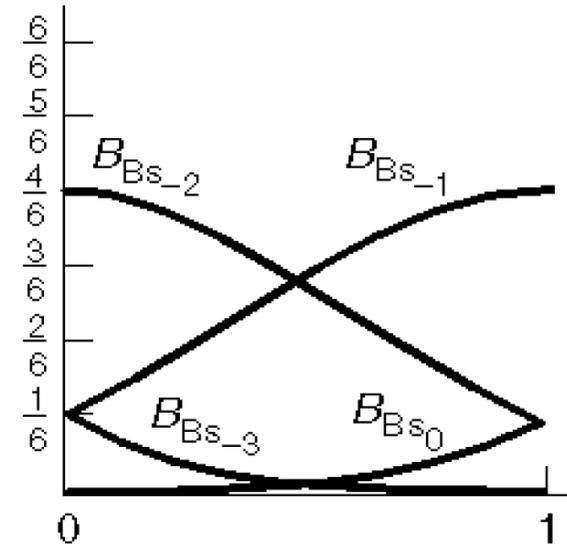
$$B_{i-3} = \frac{1}{6} (1-u)^3 \quad \text{com } 0 \leq u \leq 1$$

# Representação matricial

1/6

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

# Funções de mistura da cubica uniforme anterior

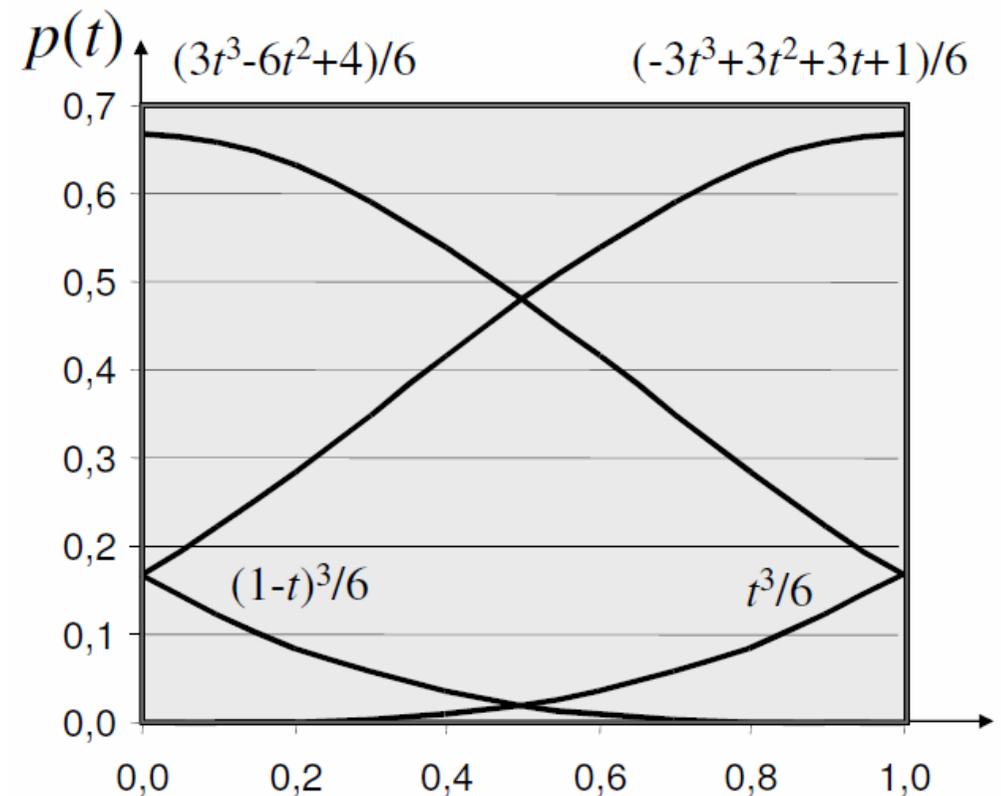


## B-Splines Uniformes

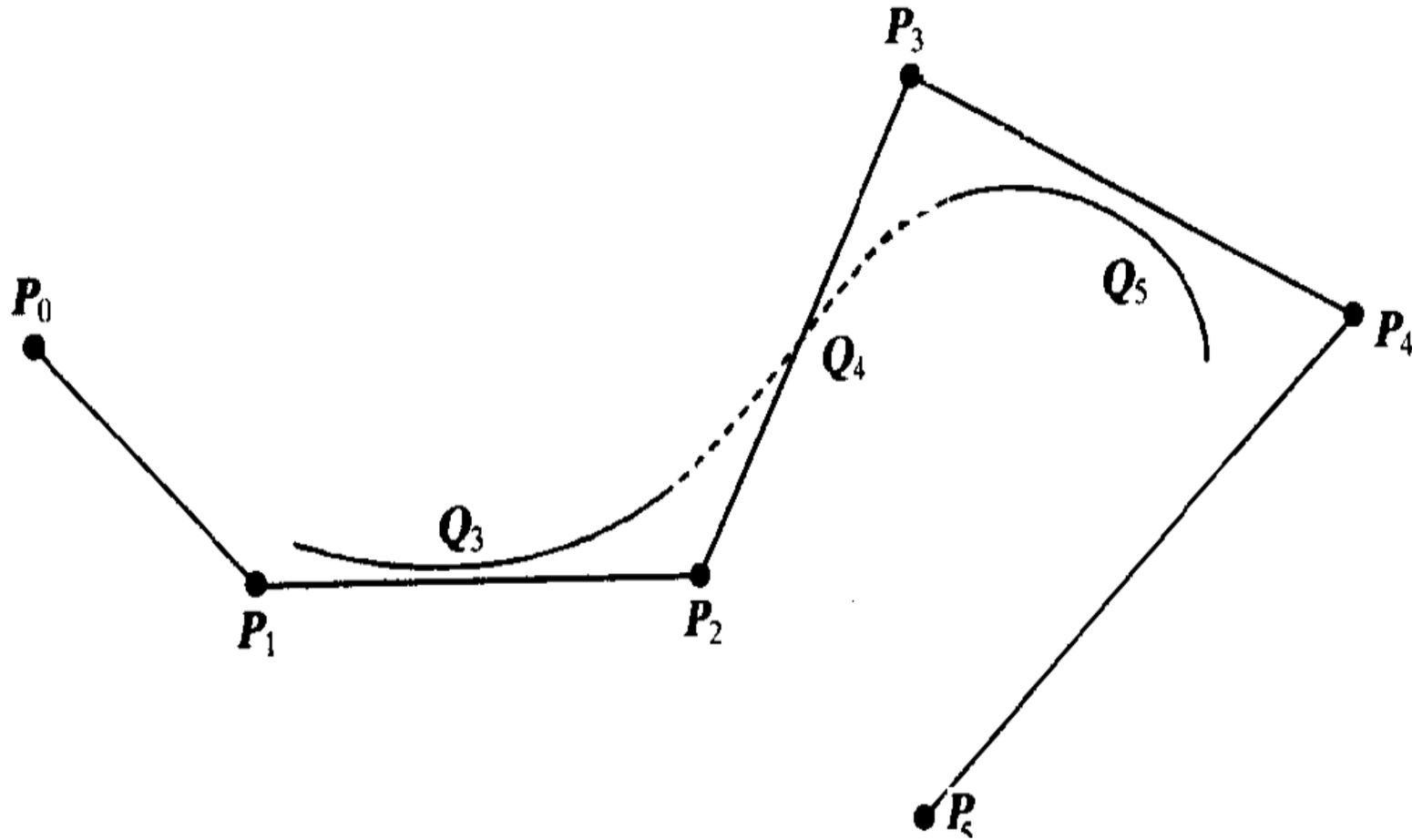
$$Q_{BS}(t) = \frac{(1-t)^3}{6} P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6} P_{i-2} +$$

$$\frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{i-1} + \frac{t^3}{6} P_i$$

$$0 \leq t \leq 1$$

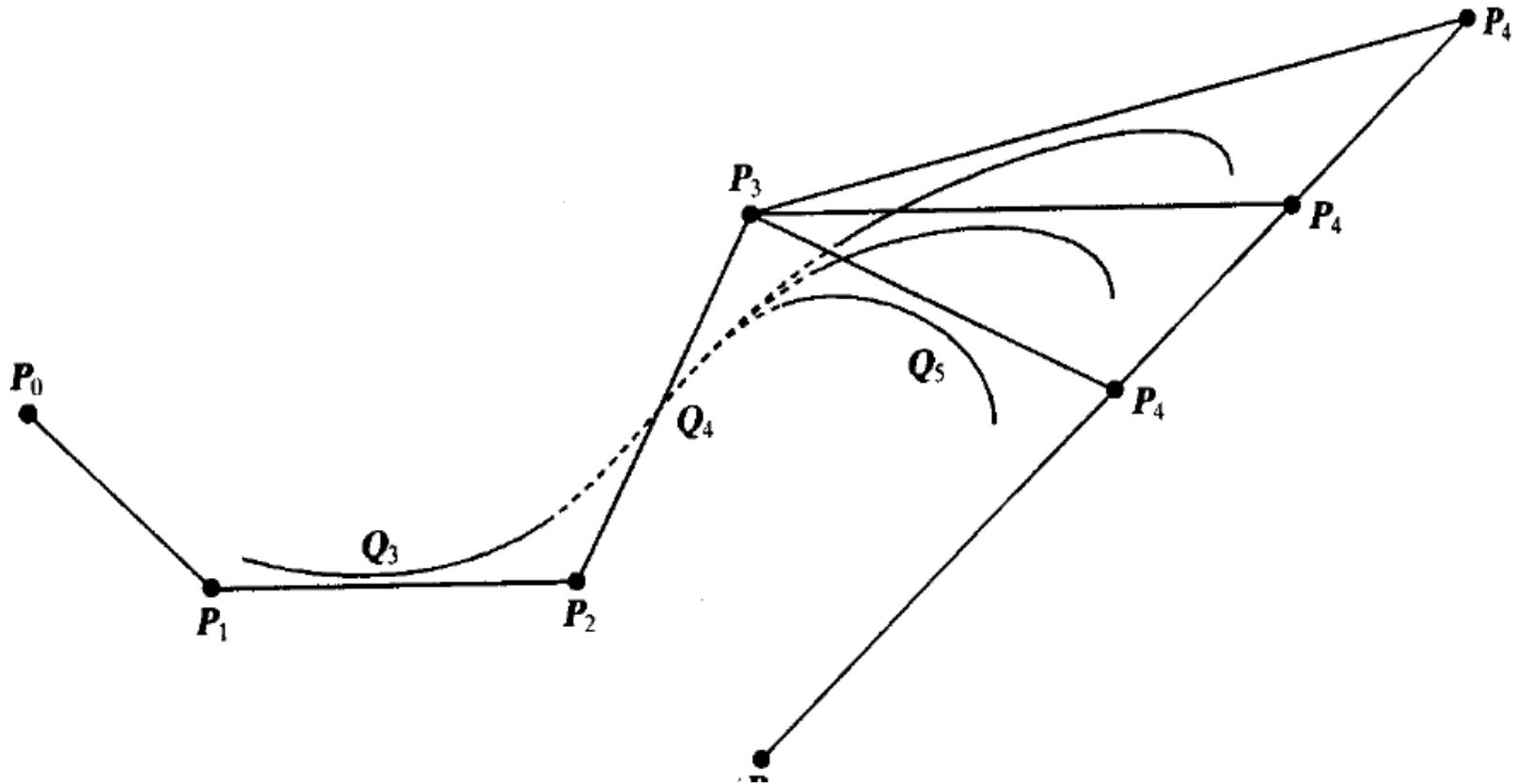


# Unido 3 curvas B-Splines



# Exemplo de controle local:

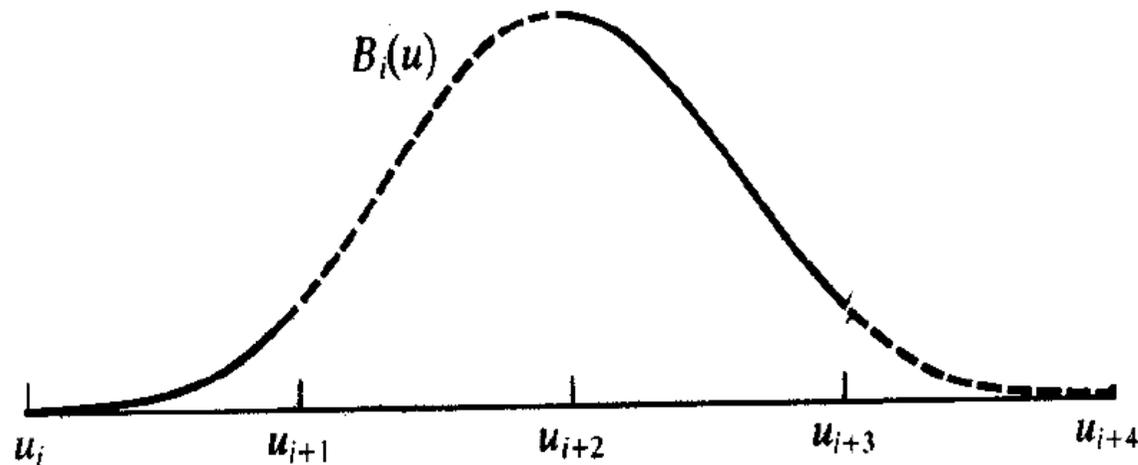
- Alterando o penúltimo ponto, não se altera o trecho inicial e só parte do trecho intermediário



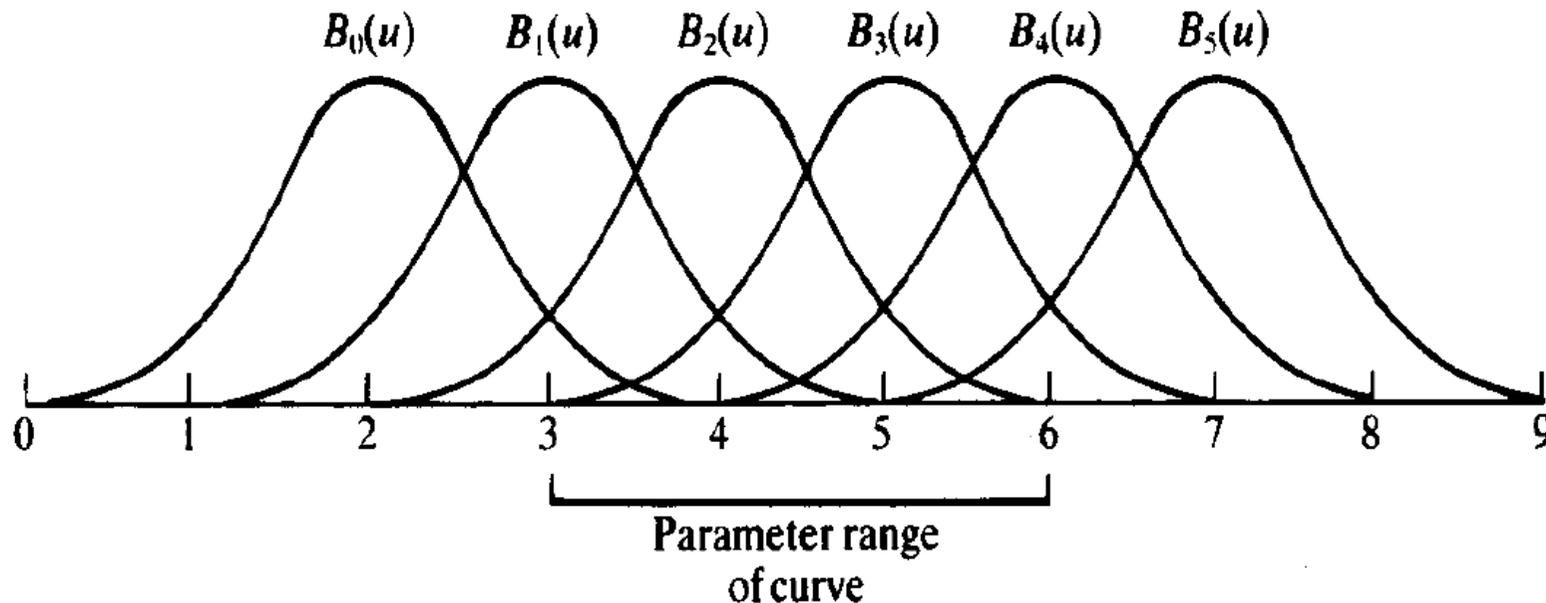
Ao ser controlada por 4 pontos, só se aproxima dos 2 centrais

Cada função de base “cobre” K intervalos

Curva B-Spline ordem 4: cada função de base é, ela própria, uma B-Spline cúbica, constituída por 4 segmentos, e simétrica



# Periódicas uniformes



Exemplo: curva cúbica com 6 pts de cntrl ( $m=5$  e  $K=4$ )

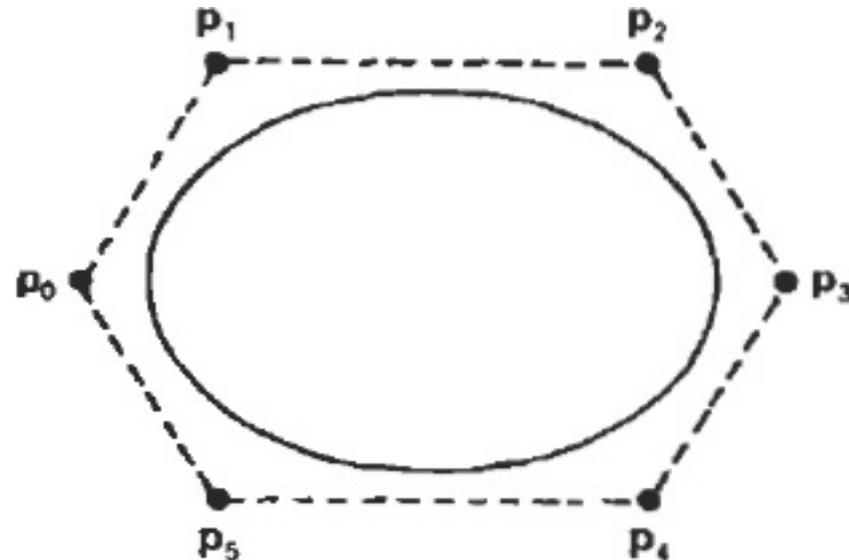
Nós uniformemente espaçados (vector de nós uniforme): cada função de base é uma cópia transladada de um nó (funções de base periódicas).

**Número total de nós: 10**

Uma curva B-Spline é calculada por:

# Para criar uma curva spline fechada:

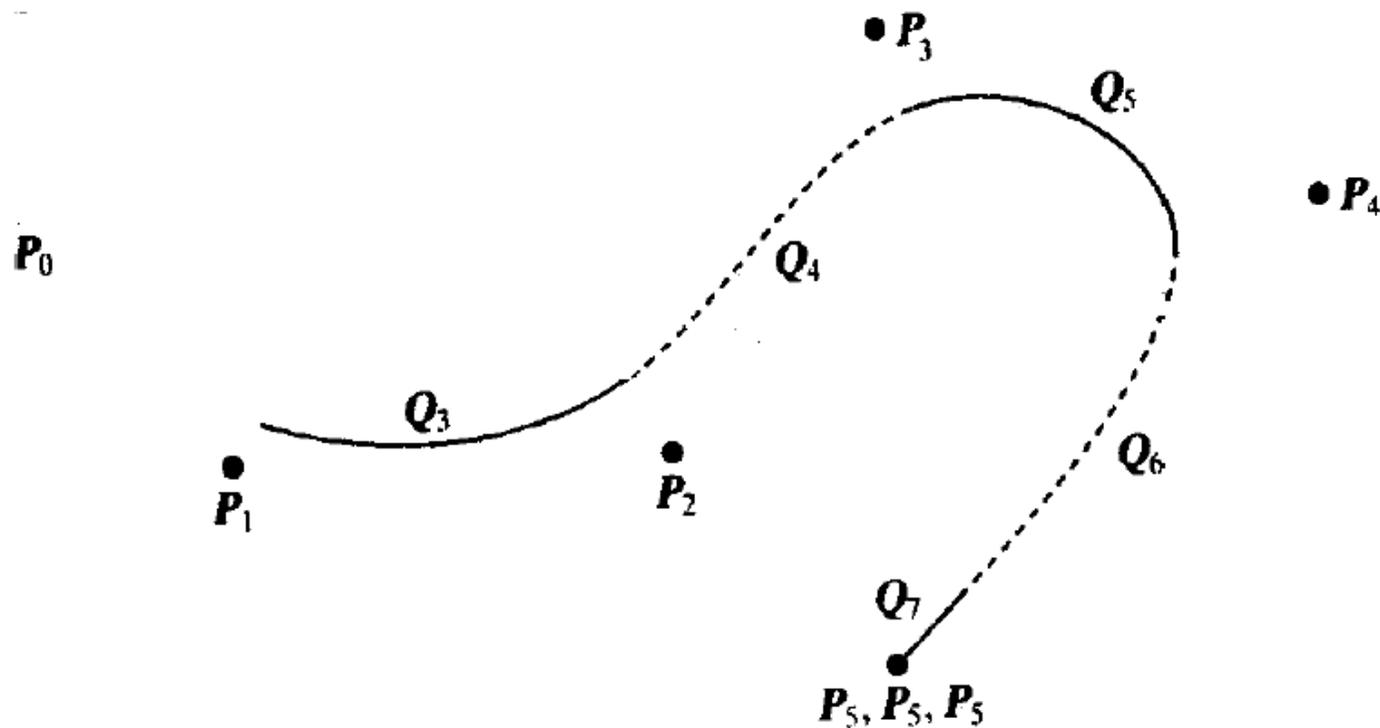
- Apenas se repete no final da seqüência dos pontos de controle da curva os 3 pontos iniciais
- $P_0, P_1, P_2, P_3 \dots \dots \dots P_m, P_0, P_1, P_2$



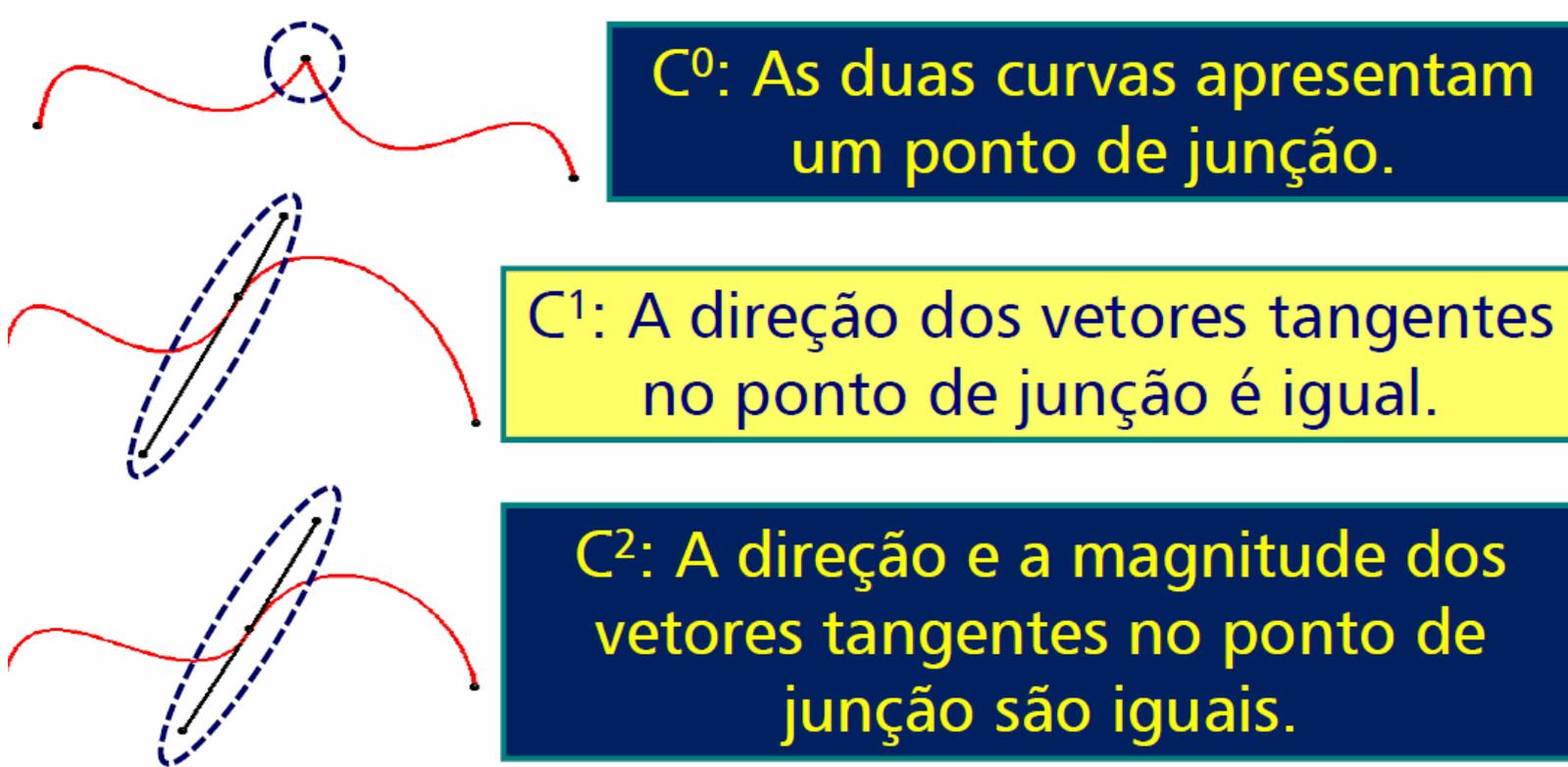
Spline com pontos controle coincidentes seguidos =>  
Ela acaba por passar pelo ponto

Três  $P_5$  coincidentes: 8 pts de controle, 6  $Q_i$ ,  $3 \leq u \leq 8$  ▼

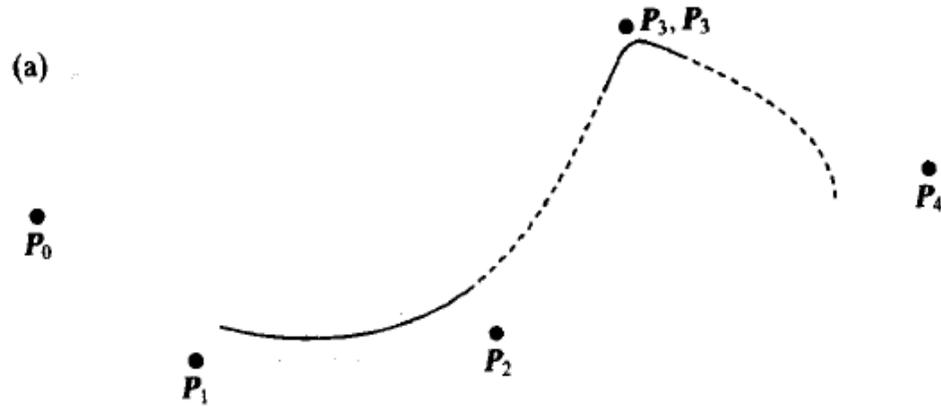
$Q_7$  ( $7 \leq u \leq 8$ ) determinado por  $P_4 P_5 P_5 P_5$ . Em  $u=8$  interpola  $P_5$



# Lembrando o significado de continuidade



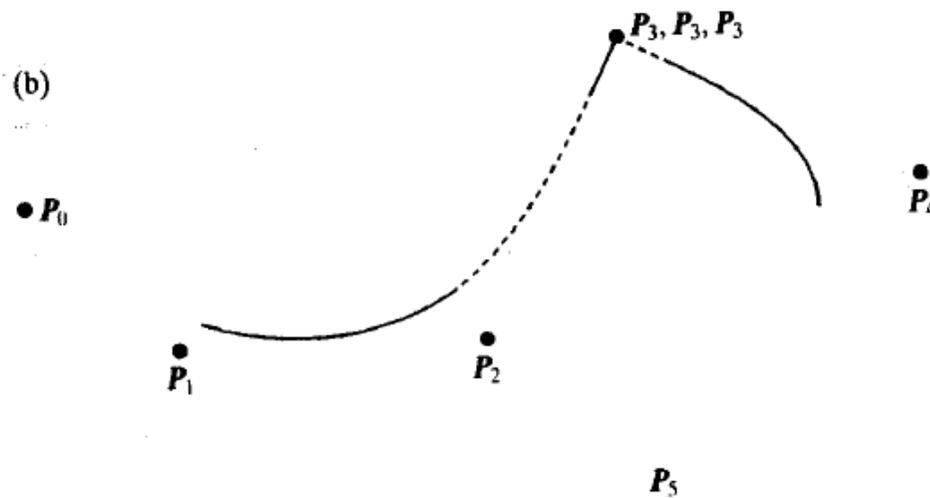
Spline com pontos controle coincidentes seguidos =>  
Ela acaba perdendo níveis de continuidade



Perda de continuidade

a) ponto duplo - G1

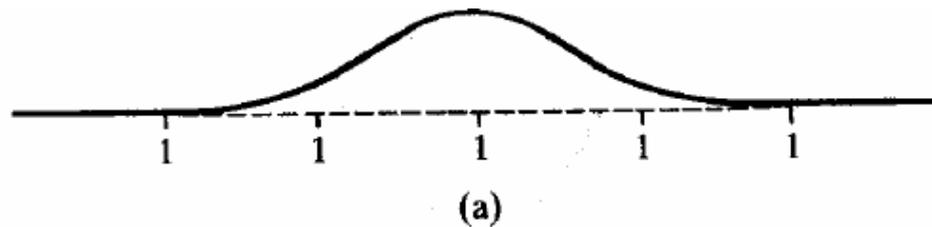
b) Ponto triplo - G0



## Spline : efeito das multiplicidades dos pontos de controle ou coincidências dos mesmos nas funções de base

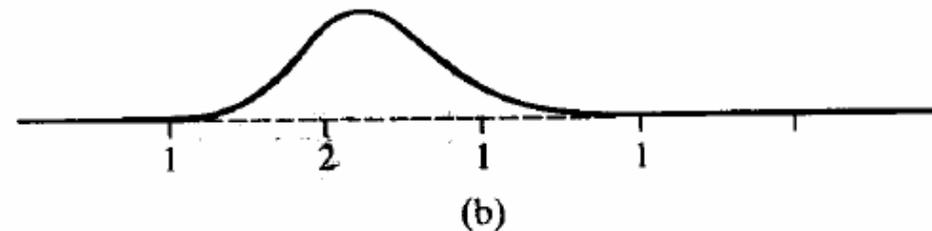
a) Multiplicidade 1:

[0, 1, 2, 3, 4]



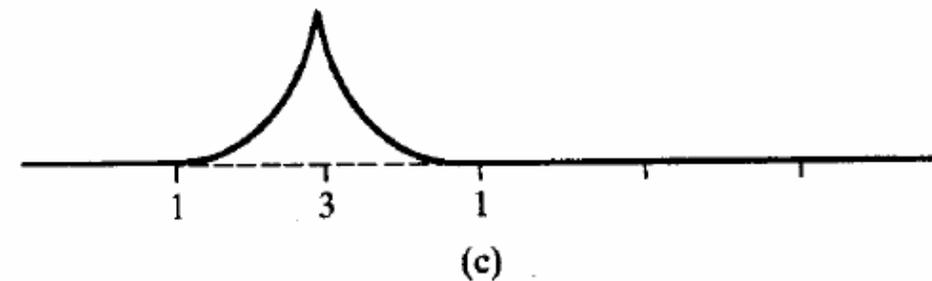
b) Multiplicidade 2:

[0, 1, 1, 2, 3]



c) Multiplicidade 3:

[0, 1, 1, 1, 2]



d) Multiplicidade 4:

[0, 1, 1, 1, 1]



# Propriedades

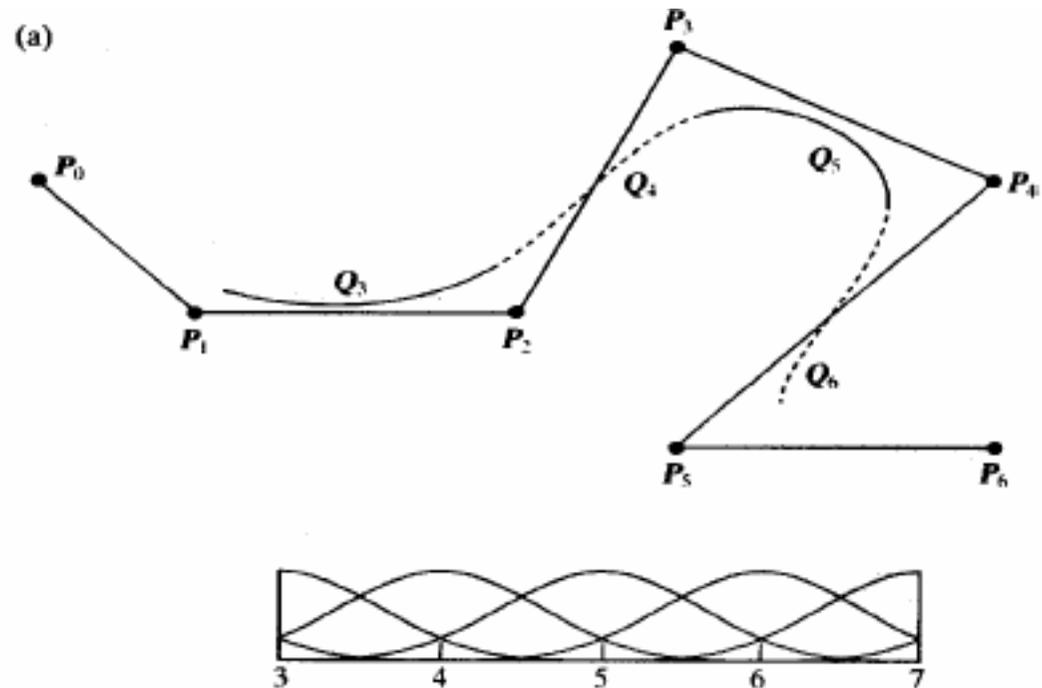
- Aumentar a multiplicidade  $m$  de um knot reduz a continuidade da paramétrica  $k-m-1$ ;
- Um knot interior de multiplicidade  $k$  transforma uma B-spline em duas B-Splines distintas cada um com o seu conjunto de pontos de controlo.

## Spline => propriedades

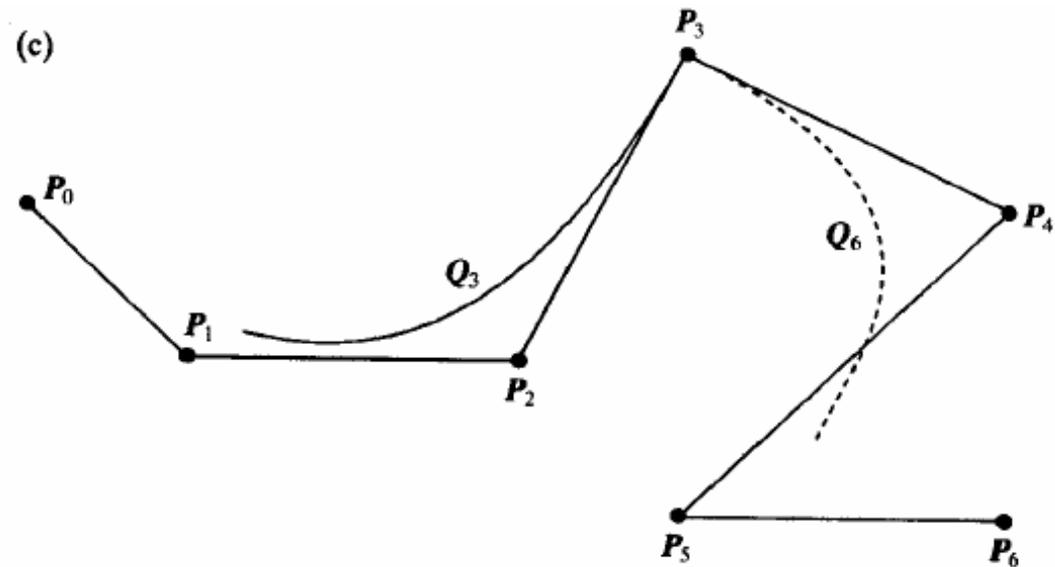
Spline com pontos controle coincidentes seguidos => perda nivel de continuidade

The effect of interior knot multiplicity on a B-spline curve.

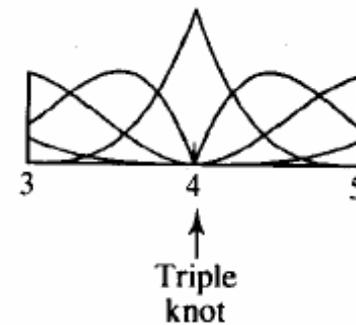
(a) A four-segment B-spline curve. The knot vector is  $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ . All B-splines are translates of each other.



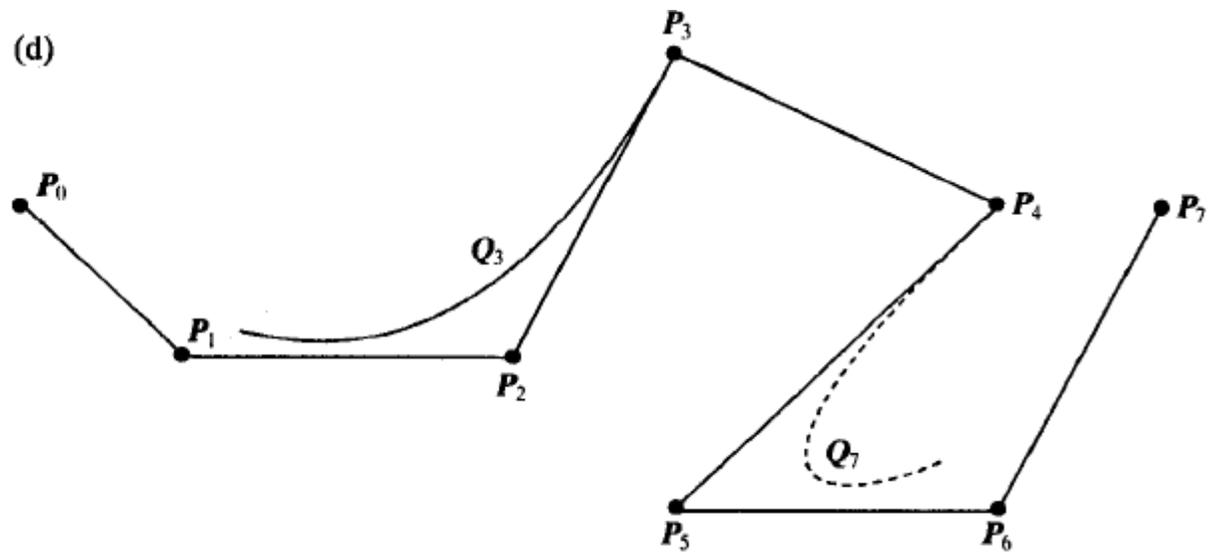
## Spline com pontos controle coincidentes seguidos



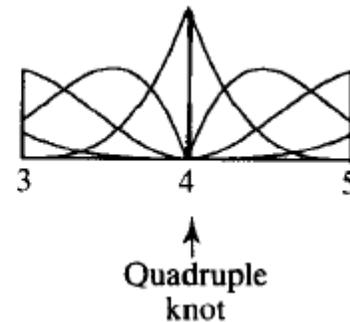
(c) Knot vector is  $[0, 1, 2, 3, 4, 4, 4, 5, 6, 7, 8]$ .  $Q_4$  and  $Q_5$  shrink to zero. Continuity between  $Q_3$  and  $Q_6$  is positional.



## Spline com pontos controle coincidentes seguidos



(d) Knot vector is  $[0, 1, 2, 3, 4, 4, 4, 4, 5, 6, 7, 8]$ . The curve reduces to a single segment  $Q_3$ . Another control point has been added to show that the curve now 'breaks' between  $P_3$  and  $P_4$ .



# B-Splines Não-Uniformes

- O intervalo entre valores da variável paramétrica não é necessariamente uniforme...
- Logo as funções de *blending* não são as mesmas para cada segmento.... Maior flexibilidade

Funções de mistura



# Curvas racionais

A curva Rational cúbica é dada pelas seguintes razões:

$$x(u) = \frac{X(u)}{W(u)}, y(u) = \frac{Y(u)}{W(u)}, z(u) = \frac{Z(u)}{W(u)}$$

Onde  $X(u)$ ,  $Y(u)$ ,  $Z(u)$  e  $W(u)$  são curvas cúbicas polinomiais cujos pts. de ctrl são definidos em coordenadas homogêneas.

Curva no espaço homogêneo:

$$Q(u) = [X(u) \ Y(u) \ Z(u) \ W(u)]$$

Tanto as curvas de Bézier quanto as B-Splines possuem ambas as formas: inteira e racional.

|          | Forma Inteira                 | Forma Racional  |
|----------|-------------------------------|---|
| Bézier   | $\sum_{i=0}^n B_i J_{n,i}(t)$ | $\frac{\sum_{i=0}^n w_i B_i J_{n,i}(t)}{\sum_{i=0}^n w_i J_{n,i}(t)}$ |
| B-Spline | $\sum_{i=0}^n B_i J_{i,k}(t)$ | $\frac{\sum_{i=0}^n w_i B_i N_{i,k}(t)}{\sum_{i=0}^n w_i N_{i,k}(t)}$ |

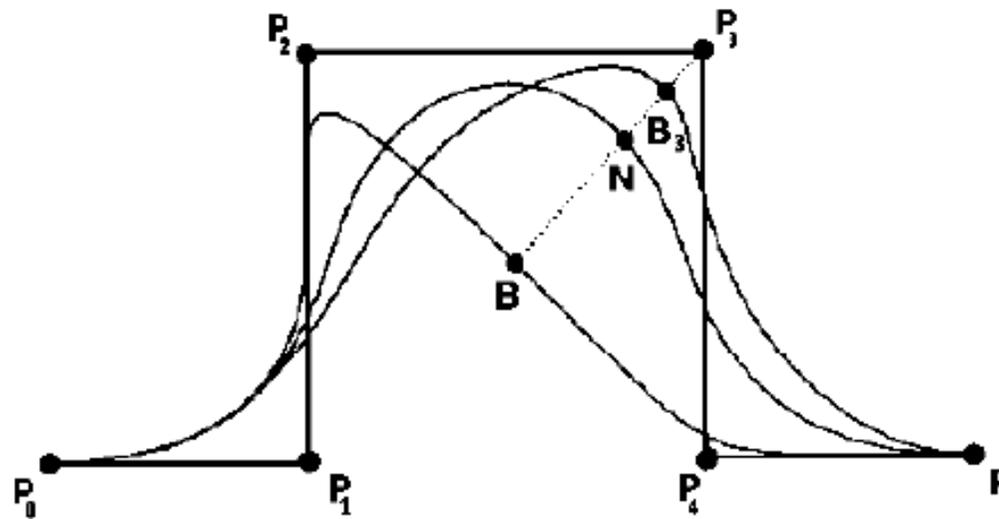
# NURBS

- ***Non-uniform rational B-splines***
- B-spline não-uniforme racional
- *Rational* significa que os segmentos de curva são expressos por razões entre polinômios cúbicos

*Computer Graphics C version* de D. Hearne e M.P. Baker , p. 357-349

# NURBS

- Non-Uniform Rational B-Spline
- O peso dos pontos de controle é a diferença



# Curvas Rational B-Spline

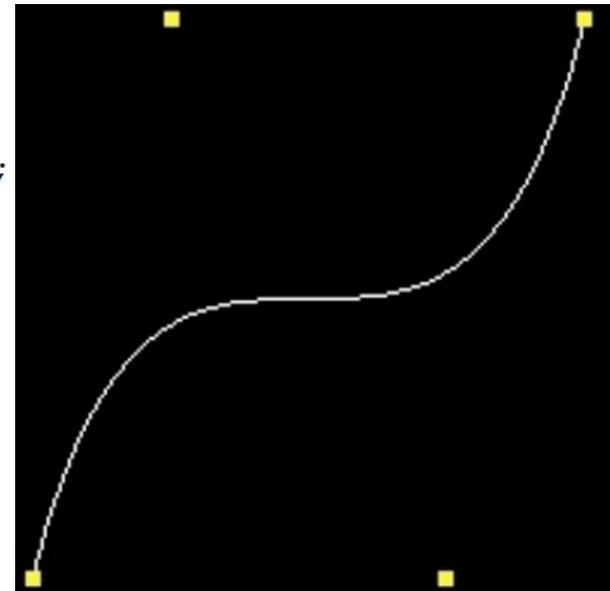
- Provê uma única forma matemática precisa capaz de representar as formas analíticas comuns
- Linhas, planos, curvas cônicas incluindo círculos, curvas de forma livre, superfícies quádricas e esculpidas

São invariantes a rotação, translação, scaling e transformações perspectivas

## Usuário fornece os pontos $X[i], Y[i]$ e:

```
i = 0;
while(i+3 < TotMarks) { //TotMarks = número total de pontos na curva
    RangeX = fabs (X[i+2] - X[i+1]);
    RangeY = fabs (Y[i+2] - Y[i+1]);
    if(RangeX > RangeY) Step = 1.0/RangeX;
    else Step = 1.0/RangeY;

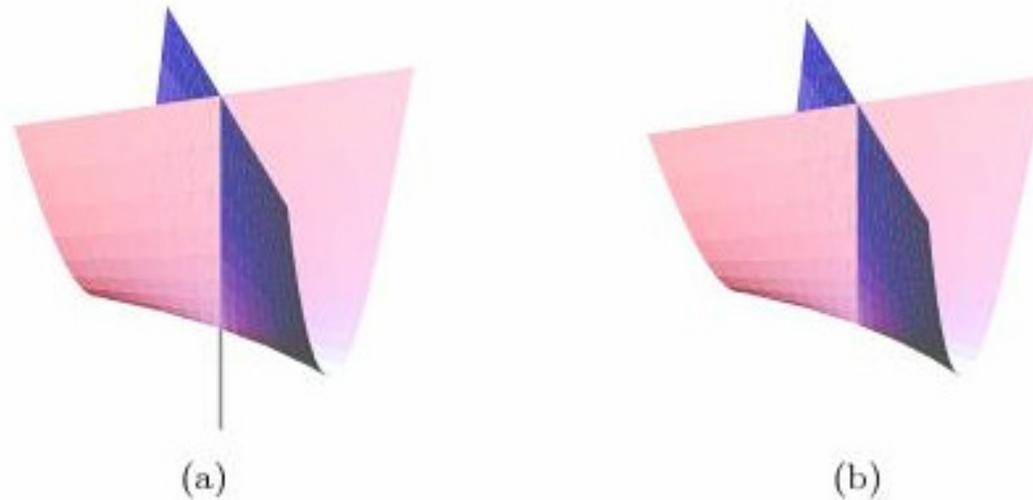
    for(t = 0; t <= 1; t += Step) {
        x = (((-1*pow(t,3) +3*pow(t,2) -3*t +1)*X[i] +
            ( 3*pow(t,3) -6*pow(t,2) +0*t +4)*X[i+1] +
            (-3*pow(t,3) +3*pow(t,2) +3*t +1)*X[i+2] +
            ( 1*pow(t,3) +0*pow(t,2) +0*t +0)*X[i+3])/6);
        y = (((-1*pow(t,3) +3*pow(t,2) -3*t +1)*Y[i] +
            ( 3*pow(t,3) -6*pow(t,2) +0*t +4)*Y[i+1] +
            (-3*pow(t,3) +3*pow(t,2) +3*t +1)*Y[i+2] +
            ( 1*pow(t,3) +0*pow(t,2) +0*t +0)*Y[i+3])/6);
        if(t == 0) MoveTo (hdc, x, y);
        else LineTo (hdc, x, y);
    }
    i++;
}
```



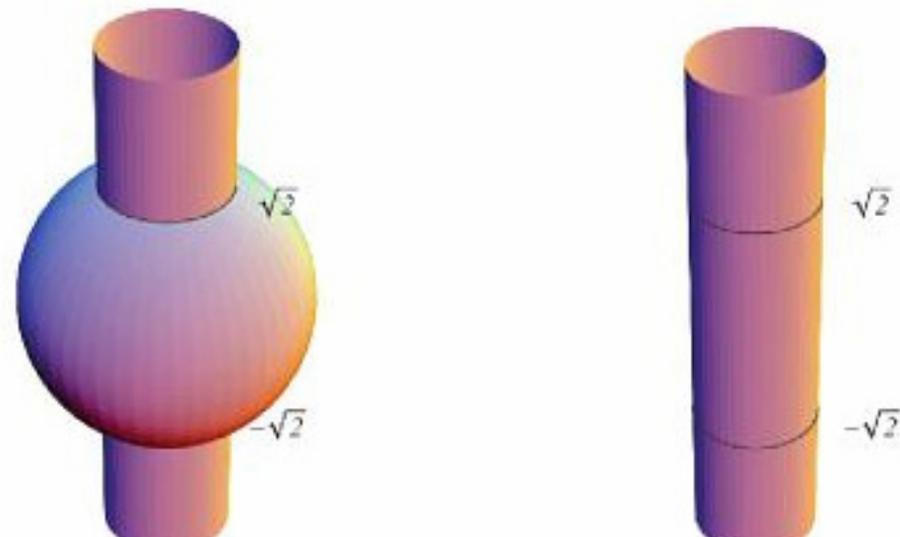
## Bibliografia

- Abel Gomes, Irina Voiculescu, Joaquim Jorge, Brian Wyvill, Callum Galbraith  
**Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms**, Springer, 2009
- **“Computer Graphics: Principles and Practice”**, Foley, van Dam, Feiner and Hughes; Capítulo 11
- **“3D Computer Graphics”**, A. Watt, Capítulo 6

# Superfícies



**Fig. 1.4.** (a) Whitney umbrella with-handle  $x^2 - zy^2 = 0$ ; (b) Whitney umbrella without-handle  $\{x^2 - zy^2 = 0\} - \{z < 0\}$ .



# Bidimensionais :

## **Superfícies Paramétricas**

$$x = f(u,v)$$

$$y = g(u,v)$$

$$z = h(u,v)$$

As próprias equações podem ser usadas para gerar as superfícies.

Por exemplo:

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

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

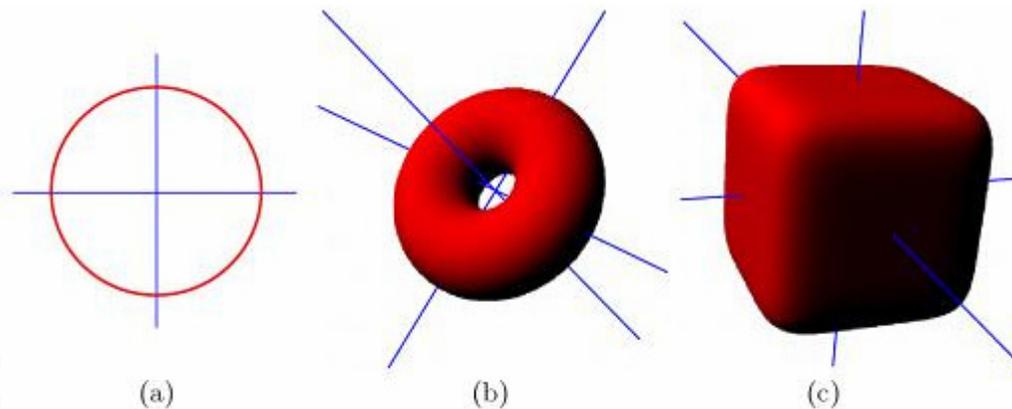
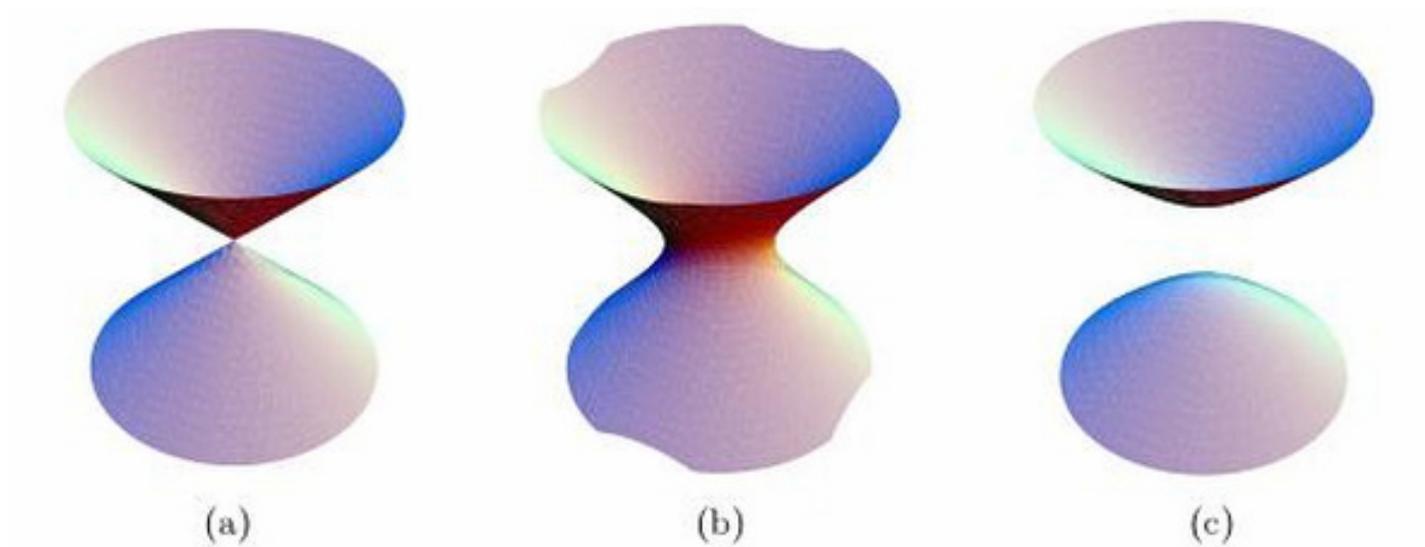


Fig. 0.1. (a) A unit circle; (b) a torus; (c) a blended cube.

$$f(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} = \begin{pmatrix} \cos u (R + r \cos v) \\ \sin u (R + r \cos v) \\ r \sin v \end{pmatrix} \quad \begin{cases} x = \frac{2t}{1+t^2} \\ y = \frac{1-t^2}{1+t^2} \end{cases} \quad \text{where } t \in [0, 1]$$

As equações podem ser usadas para gerar as superfícies.



**Fig. 1.3.** (a) Cone  $x^2 + y^2 - z^2 = 0$ ; (b) hyperboloid of one sheet  $x^2 + y^2 - z^2 = a^2$ ; (c) hyperboloid of two sheets  $x^2 + y^2 - z^2 = -a^2$ .

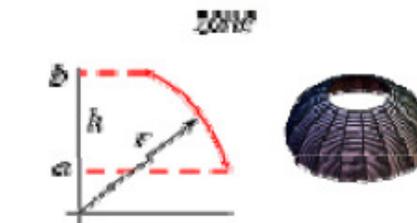
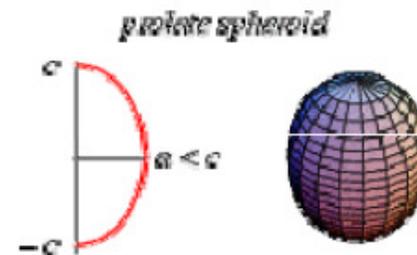
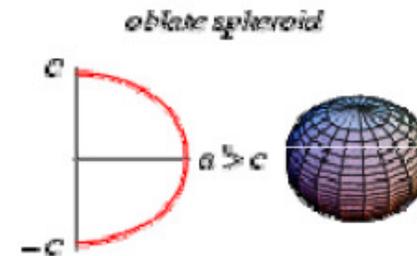
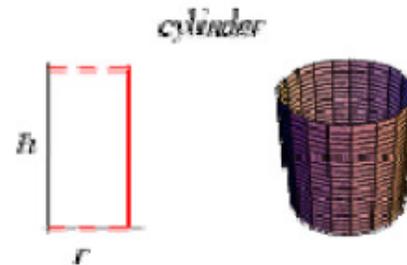
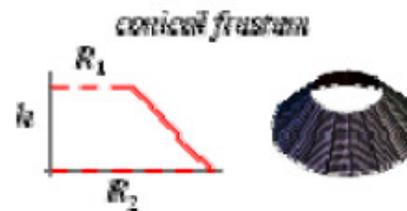
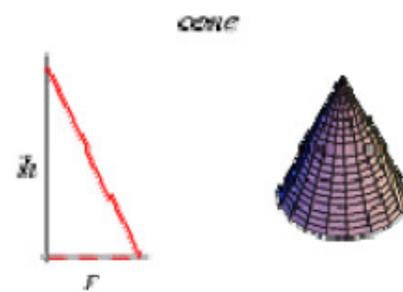
# Outras formas de geração:

## Superfície por caminho

- Superfície criada atravessando uma curva ao longo de um caminho no espaço

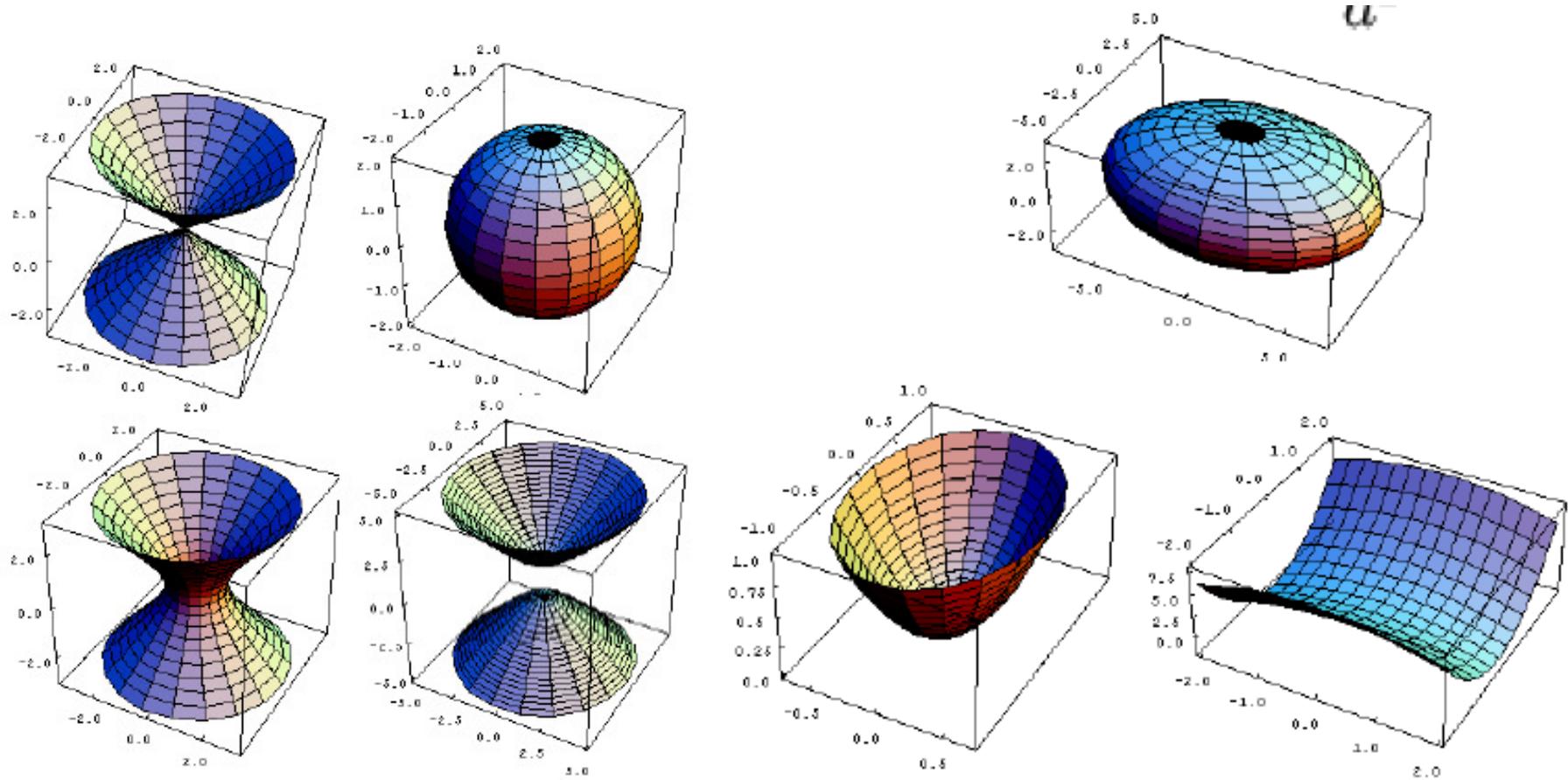
# Revolução

- Superfície criada pela rotação de uma curva sobre um plano em torno de uma linha reta (o eixo de rotação) que está sobre o mesmo plano.



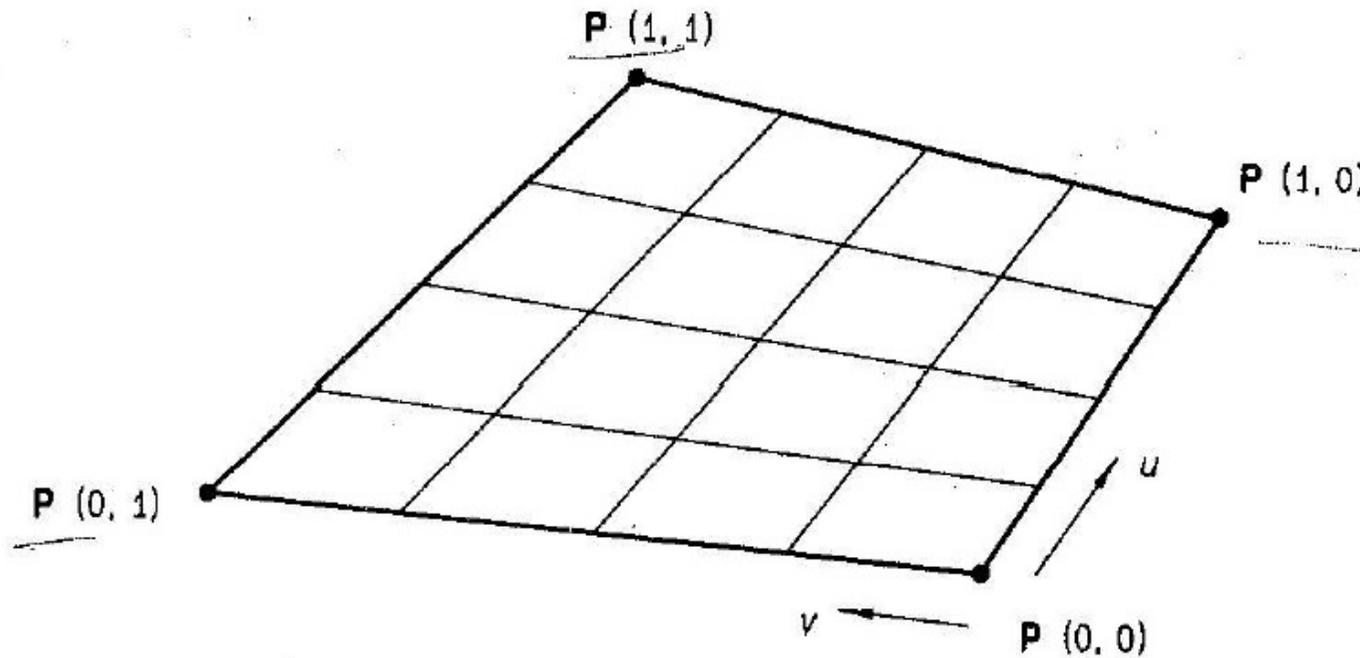
# Quádricas

$$\pm \frac{x^2}{a^2} \pm \frac{y^2}{b^2} \pm \frac{z^2}{c^2} = 1$$



# Geradas por interpolação

- Interpolação linear entre quatro pontos que não estão no mesmo plano



# A forma mais simples

- é considerar o espaço dos parâmetros representados por uma área unitária limitada pelos pontos (0,0) (0,1) (1,0) e (1,1).
- Essa área pode ser vista como o produto cartesiano dos dois eixos normalizados ortogonais. De modo que qualquer ponto do interior desta região é definido univocamente.
- Se for desejado gerar uma superfície a partir de quatro pontos: A, B, C e D, esses pontos devem ser associados aos limites do espaço dos parâmetros (0,0), (1,0), (1,1) e (0,1), então o interior é gerado empregando-se **duas interpolações lineares** sucessivas.
- Na primeira interpolação serão geradas as retas AD e CB, que correspondem aos limites com parâmetro  $u=0$  e  $u=1$ .
- Qualquer ponto  $E(v)$  sobre a reta AD será definido como:

$$E(v) = (1-v) A + vD,$$

de modo que se  $v=0$ ,  $E(v)=E(0)$  é o próprio ponto A e se  $v=1$ ,  $E(v)=E(1)$  corresponde ao ponto D.

Do mesmo modo, os pontos  $F(v)$  sobre a reta BC serão obtidos pela interpolação linear das coordenadas de B e C:

$$F(v) = (1-v) B + vC.$$

Com os pontos E e F gera-se o interior da superfície a partir de outra interpolação linear, usando agora o parâmetro  $u$ :

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

# interpolação bi-linear:

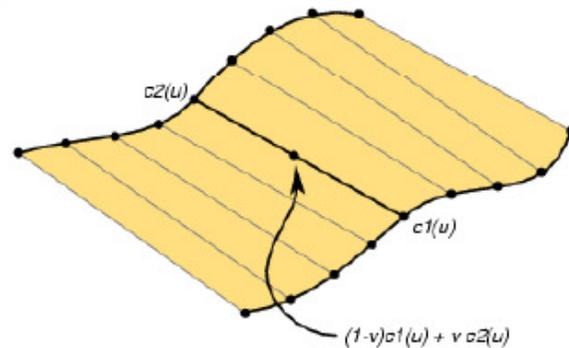
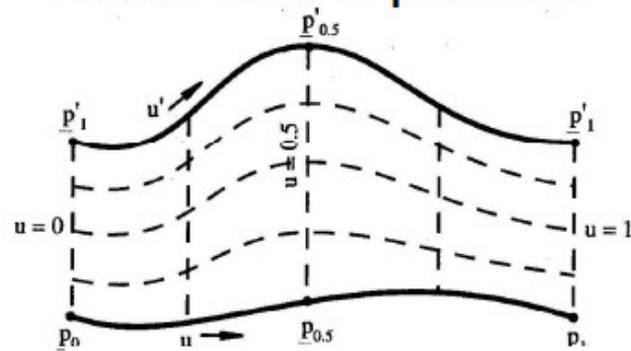
- As expressões anteriores podem ser reunidas resultando em duas interpolações lineares ou uma interpolação bi-linear:

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

- É fácil verificar que a superfície gerada só será um plano se os quatro pontos forem co-planares (estiverem sobre um mesmo plano). Neste caso as fronteiras serão limitadas por segmentos de reta.
- Se as fronteiras (ou limites) forem definidos por curvas e não por retas a forma mais simples de gerar o interior é usar expressão dessas curvas para gerar os pontos de  $E(v)$  e de  $F(v)$ .
- Essa é a idéia de geração de superfícies denominada “**lofting**”, **usada desde a antiguidade na construção de caravelas, naus, embarcações e navios.**
- No **lofting** as curvas dos limites opostos, nas direções  $u$  ou  $v$  dos parâmetros são usadas para a geração da superfície.

# Lofting

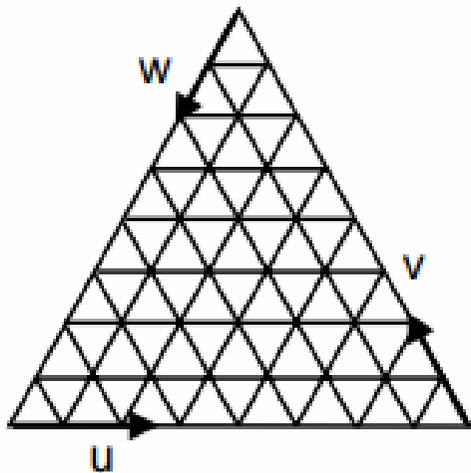
- Contruída juntando 2 curvas por linhas retas entre pontos



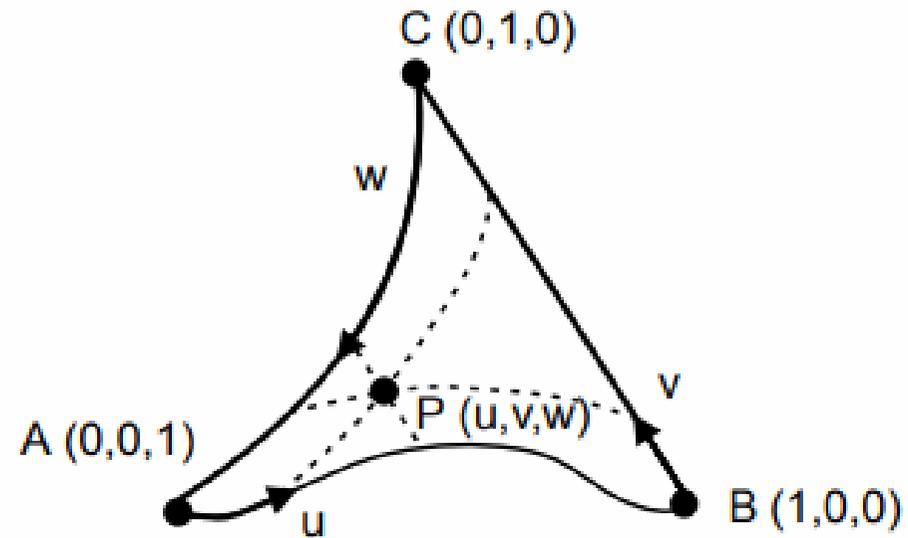
# interpolações trilineares

- no caso da definição da superfície por três curvas de fronteira.
- Nesta um ponto do interior é definido por três parâmetros  $u$ ,  $v$  e  $w$ .
- Mas como são 2D sempre podem ser descritas com apenas dois parâmetros.
- Assim, obviamente há a restrição adicional de que  $w + v + u = 1$  em qualquer ponto.

# interpolações trilineares



espaço dos parâmetros



superfície gerada pelas curvas de fronteira

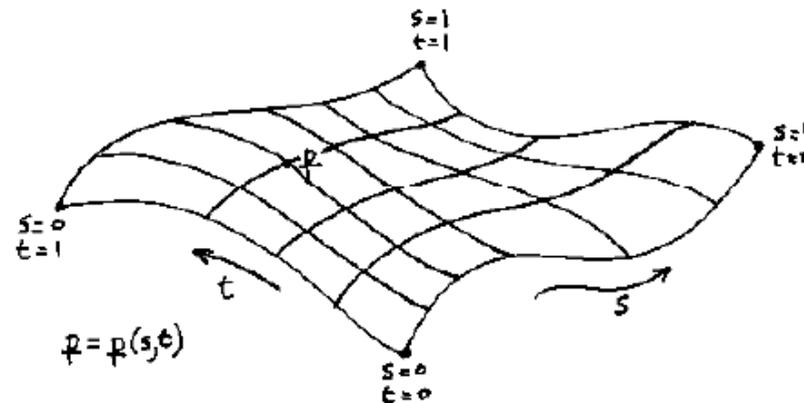
# Superfícies de Formas Livres

- A idéia básica de construir uma superfície pelas interpolações dos contornos pode ser generalizada para quando uso das curvas de CG (Hermite, Bézier, Splines ou Racionais), gerando superfícies que podem ser descritas por curvas criadas a partir de pontos de controle das suas formas.

# Patch – ou construções por retalho

## Superfície Linear de Coons

- Interpolação entre quatro curvas de fronteira
- Similar a superfície regrada em duas direções



# Superfícies Paramétricas Bicúbicas

- As superfícies paramétricas bi-cúbicas são definidas como combinação de curvas cúbicas e 16 pontos de controle  $P_{ij}$ , como:

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

onde, como no caso das curvas de Bézier,  $B_{i,j}$  define os pontos de controle da superfície e  $J_{i,n}(s)$ ,  $J_{j,m}(t)$  são as funções de Bernstein nas direções  $s$  e  $t$  respectivamente.

As funções não precisam ter o mesmo grau nas duas direções, podendo ser cúbica na direção  $s$  e quadrática na direção  $t$ .

A superfície toma a forma definida pelos pontos de controle  $B_{i,j}$ .

Os pontos dos quatro cantos da superfície gerada coincidem com os 4 pontos de controle nestas posições.

# Na forma matricial com cúbicas

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

- onde  $S = [s^3 \ s^2 \ s \ 1]$ ,  $T = [t^3 \ t^2 \ t \ 1]$ , (o sobrescrito T indica que é usado a transposta das matrizes e vetores),  $M_b$  é a matriz de Bezier e  $G_b$  uma matriz com os 16 pontos de controle de definição da superfície.

# superfície de Bézier

definida por dois polinômios de grau 3, bicúbica de Bézier, pode ser escrita como:

$$P(s,t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} M_b G_b M_b^T \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Sendo

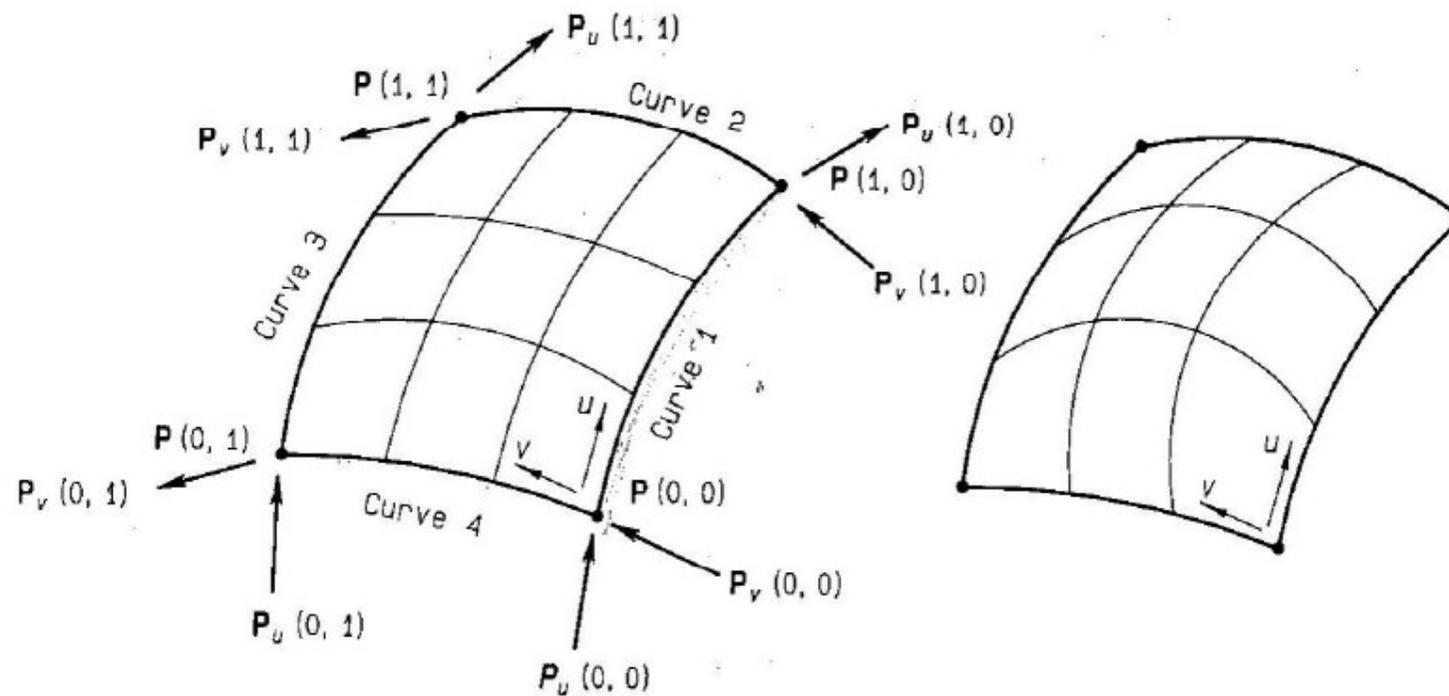
$$M_b = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

e os pontos de controle representados pela matriz

$$G_b = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$$

# Patches

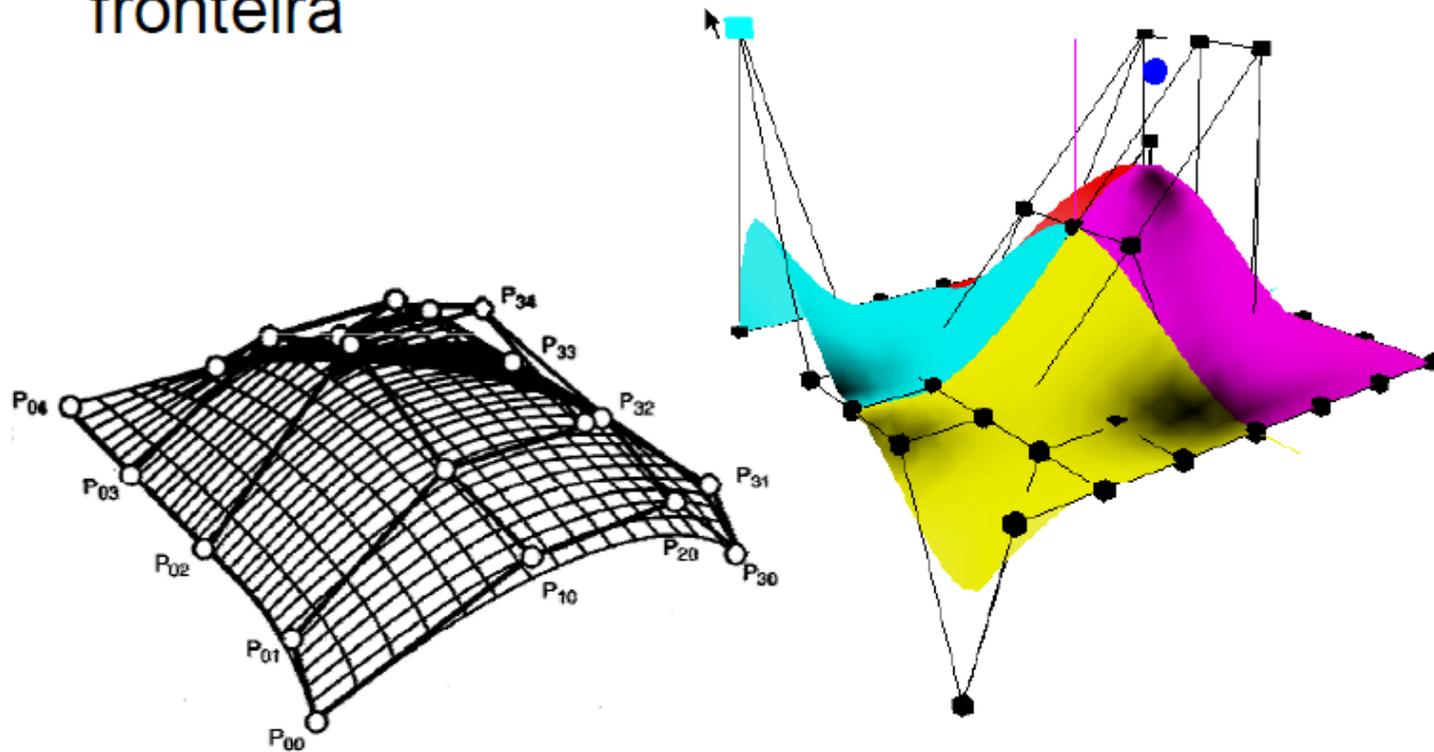
- Curvas cúbicas paramétricas como quatro curvas de fronteira



## Remendos cúbicos

# Superfície de Bezier

- Curvas Bezier como quatro curvas de fronteira



# Superfícies de Bézier

- Um ponto qualquer da superfície de Bézier pode ser obtido pela definição dos parâmetros  $s$  e  $t$  na expressão

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

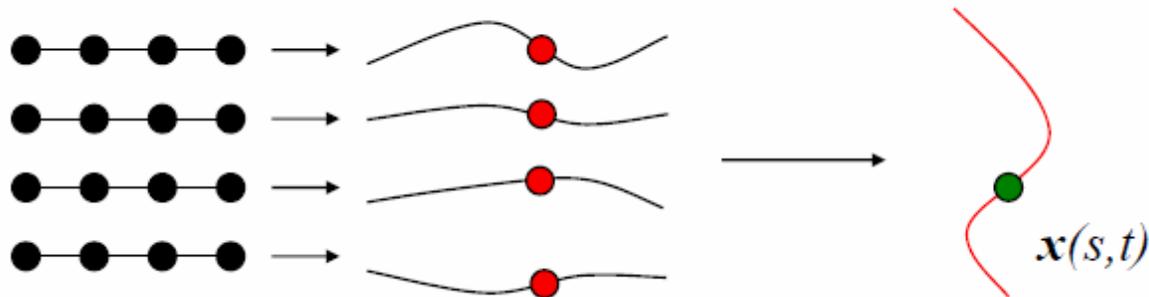
---

onde, como no caso das curvas de Bézier,  $B_{i,j}$  define os pontos de controle da superfície e  $J_{i,n}(s)$ ,  $J_{j,m}(t)$  são as funções de Bernstein nas direções  $s$  e  $t$  respectivamente.

As funções não precisam ter o mesmo grau nas duas direções, podendo ser cúbica na direção  $s$  e quadrática na direção  $t$ , por exemplo.

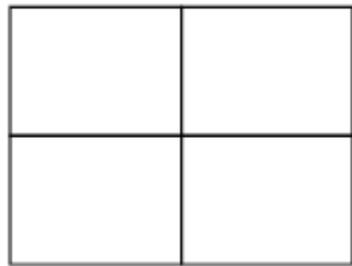
# Retalhos de Bézier

- Curvas na fronteira são curvas de Bézier
- *Qualquer* curva para  $s$  ou  $t$  constante é uma curva Bézier
- Podemos pensar assim:
  - Cada linha da grade com 4 pontos de controle define uma curva de Bézier para o parâmetro  $s$
  - Ao avaliar cada curva para um mesmo  $s$  obtemos 4 pontos de controle “virtuais”
  - Pontos de controle “virtuais” definem uma curva Bézier em  $t$
  - Avaliando esta curva em um dado  $t$  resulta no ponto  $\mathbf{x}(s,t)$

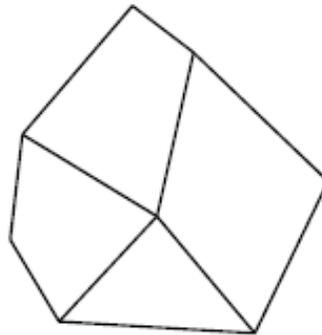


# Malhas de Retalhos Bézier

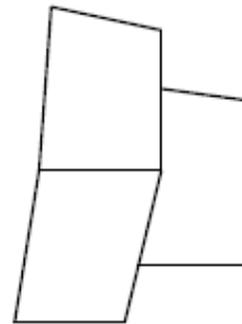
- São malhas compostas de diversos retalhos unidos ao longo de suas fronteiras
  - As arestas das grades de controle precisam se justapor perfeitamente
  - As grades precisam ser retangulares



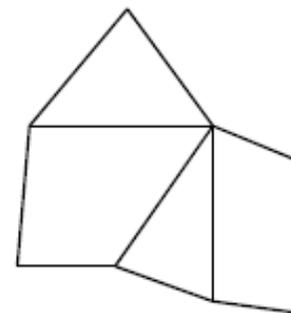
OK



OK



Não



Não

# Superfícies de B-Spline

- As superfícies B-Spline, como as anteriores, podem ser representadas pela expressão:

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

onde  $N_{i,k}(s)$  e  $N_{j,l}(t)$  são as mesmas funções de B-Spline definidas para as curvas B-Spline e  $B_{i,j}$  os pontos de controle.

Os arrays de nós nas duas direções de parametrização podem ser classificados como periódicos uniformes, não periódicos ou não uniformes, como as curvas. As superfícies B-Splines periódicas uniformes são geradas usando vetores de nós uniformes. Essas superfícies tem o mesmo tipo de controle local das curvas B-splines. Variando apenas um ponto de controle apenas uma parte da curva é afetada.

# Superfícies de Hermite

- Dois parâmetros são necessários, ambos variando entre 0 e 1. Se esses parâmetros forem chamados de  $s$  e  $t$ , a superfície bicúbica pode ser escrita como:

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

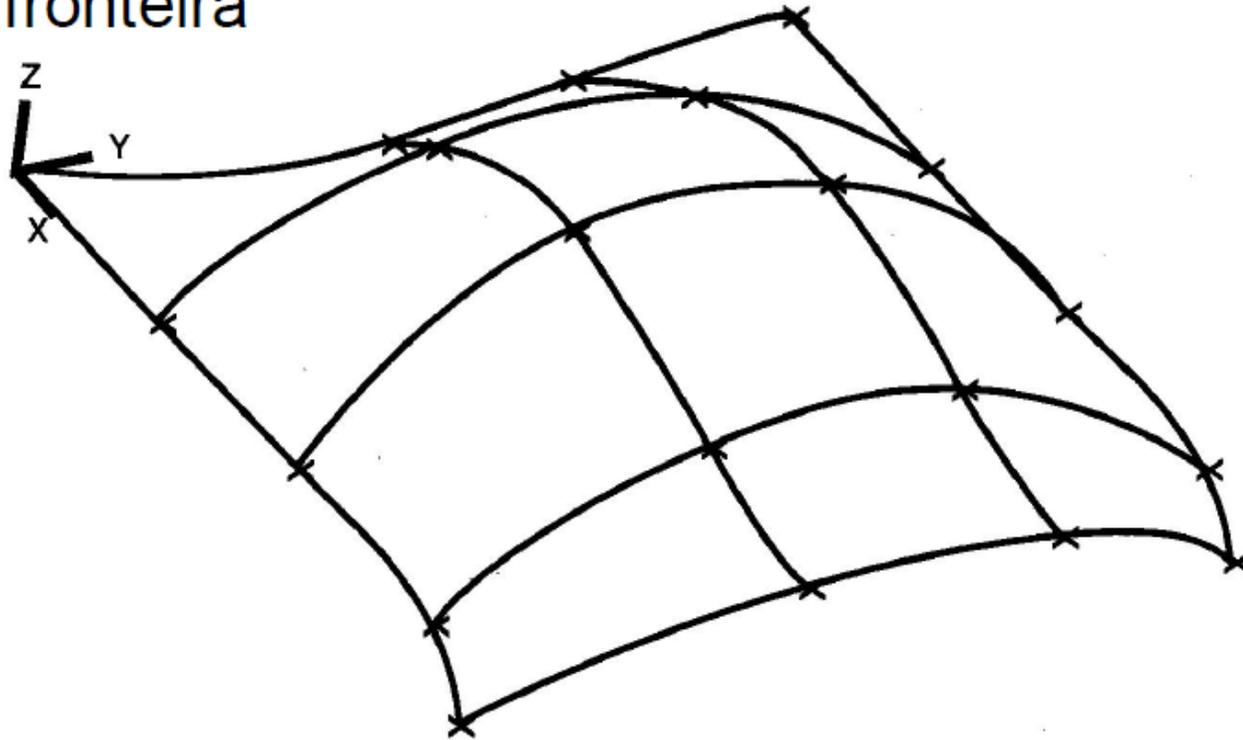
onde  $S = [s^3 \ s^2 \ s \ 1]$ ,  $T = [t^3 \ t^2 \ t \ 1]$ , e

Essa matriz deve conter os 4 pontos limites dos cantos da superfície, suas derivadas em relação aos parâmetros  $s$  e  $t$  (vetores tangentes) nos pontos dos cantos e suas derivadas cruzadas nestes pontos (vetores de torção ou “twist vectors”).

$$G_h = \begin{bmatrix} P(0,0) & P(0,1) & \frac{dP}{dt}(0,0) & \frac{dP}{dt}(0,1) \\ P(1,0) & P(1,1) & \frac{dP}{dt}(1,0) & \frac{dP}{dt}(1,1) \\ \frac{dP}{ds}(0,0) & \frac{dP}{ds}(0,1) & \frac{d^2P}{dsdt}(0,0) & \frac{d^2P}{dsdt}(0,1) \\ \frac{dP}{ds}(1,0) & \frac{dP}{ds}(1,1) & \frac{d^2P}{dsdt}(1,0) & \frac{d^2P}{dsdt}(1,1) \end{bmatrix}$$

# Superfície B-Splines

- Curvas B-Spline como quatro curvas de fronteira



formulação matricial  $P(s, t) = S M_b G_b M_b^T T^T$

- onde os parâmetros são representados pelos vetores  $S = [s^3 \ s^2 \ s \ 1]$  e  $T = [t^3 \ t^2 \ t \ 1]$ ,  $G_b$  representa a matriz formada pelos dezesseis pontos de controle e as matrizes  $M_b$  são as mesmas já usadas para curvas para curvas

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

# Superfícies Racionais

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

# NURBS

- O termo NURBS é a abreviatura de Non-Uniform Rational B-Splines Surfaces, ou seja, é uma B-Spline racional (originária da razão de 2 polinômios).

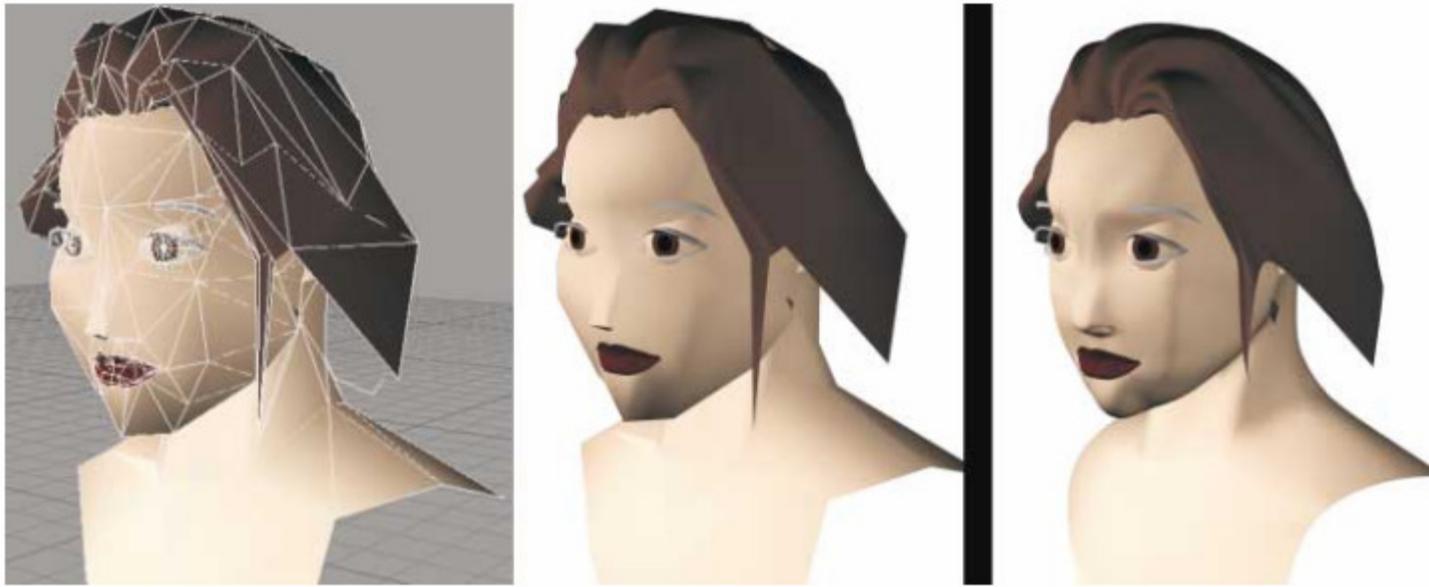


FIGURA 3.30. NURMS aplicado em um modelo de face de poucos polígonos.

---

## Bibliografia

- Abel Gomes, Irina Voiculescu, Joaquim Jorge, Brian Wyvill, Callum Galbraith  
**Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms**, Springer, 2009
- **“Computer Graphics: Principles and Practice”**, Foley, van Dam, Feiner and Hughes; Capítulo 11
- **“3D Computer Graphics”**, A. Watt, Capítulo 6