

A, B, C, D, E, F, G

A, B, D, E, F, G

A, B, C, D, E, G

A, B, C, E, F, G

Capitulo 3

A, B, C, D

Curvas e Superfícies

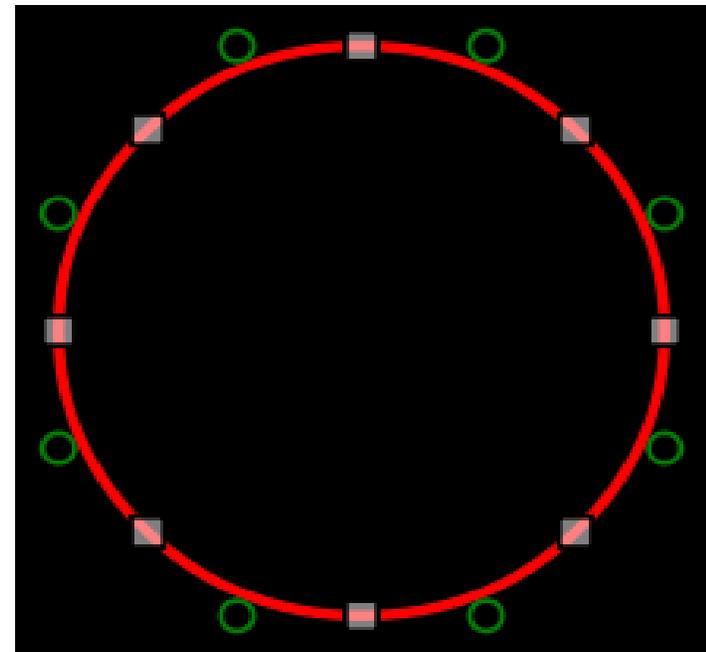
2014 – IC / UFF

Onde se usa: Qualquer representação de curvas
Modelação da Fronteira (B-Rep)-> Representação pela
descrição paramétrica da sua superfície;

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

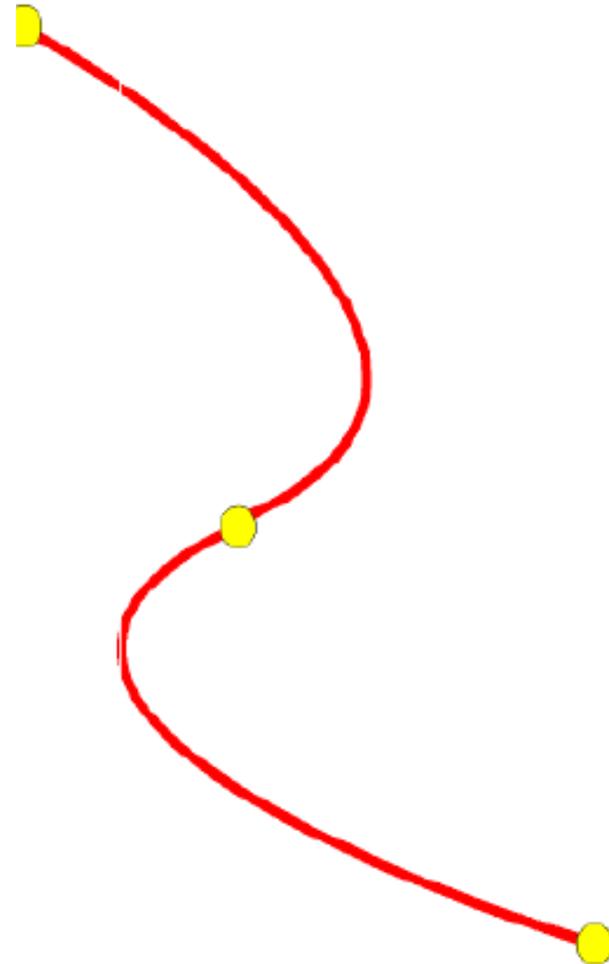
Um segmento de curva quadrática de Bézier é definido por 2 pontos extremos e 1 de controle.

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



Elementos 1D

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



Curvas



- Formas de representação:
 - Procedural (exemplo *curvas fractais*)
 - Conjunto de pontos (digitalizadores: x_i, y_i)
 - Analítica:
 - Explícita : $y = f(x)$
 - Implícita : $x+y=0$
 - Paramétrica : $x = f(t), y = f(t)$

Representação analítica

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

Também podem ser

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

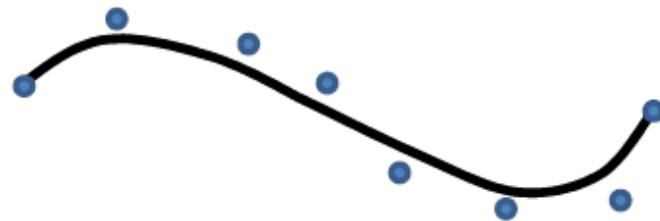
Representação analítica de curva definida por ponto

- Interpolação
 - Analiticamente definindo uma curva a partir de um conjunto de pontos conhecido
- Ajustada
 - Uma curva que passa por todos os pontos conhecidos
- Satisfatória
 - Uma curva que passa perto de pontos conhecidos

Interpolação X Aproximação



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



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

Gerar uma curva **suave** que passe por **pontos** específicos

A

A blue circular dot representing point A.

B

A blue circular dot representing point B.

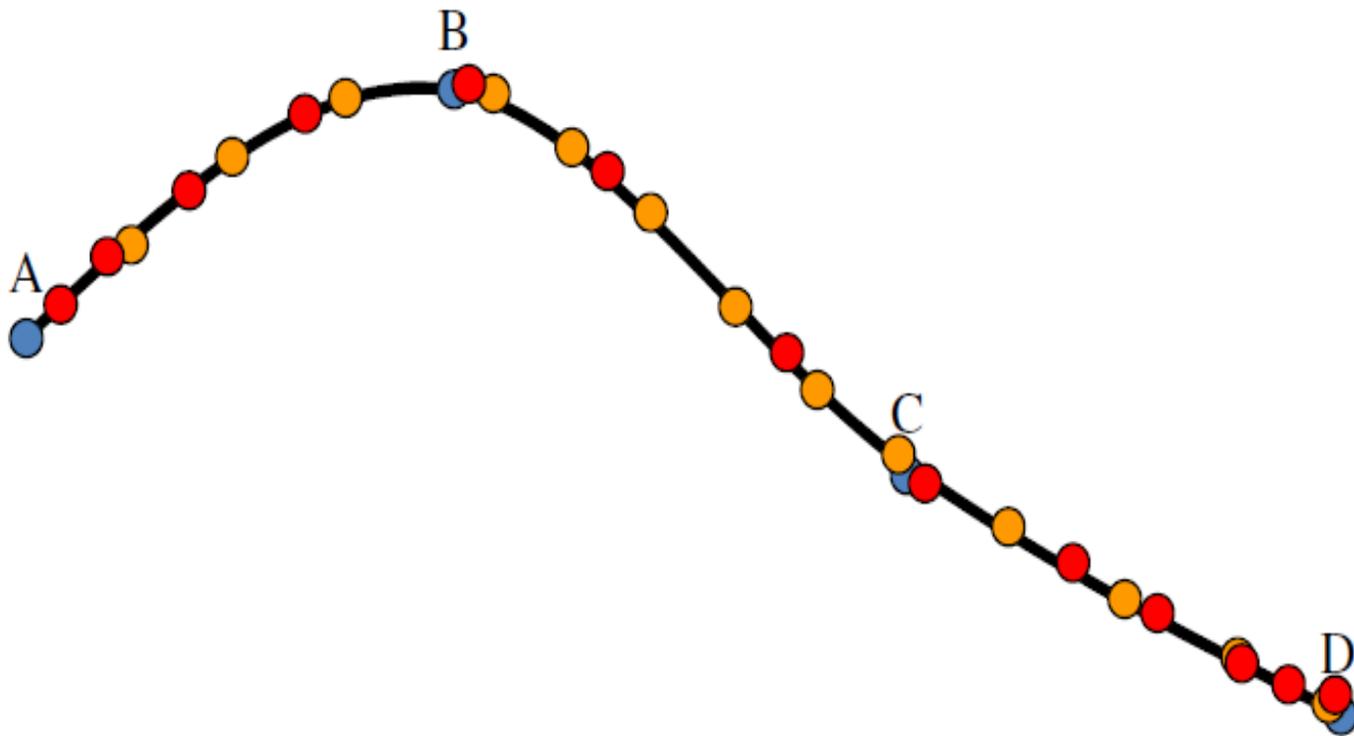
C

A blue circular dot representing point C.

D

A blue circular dot representing point D.

gerar uma curva no espaço, distribuindo **pontos** de maneira **suave**

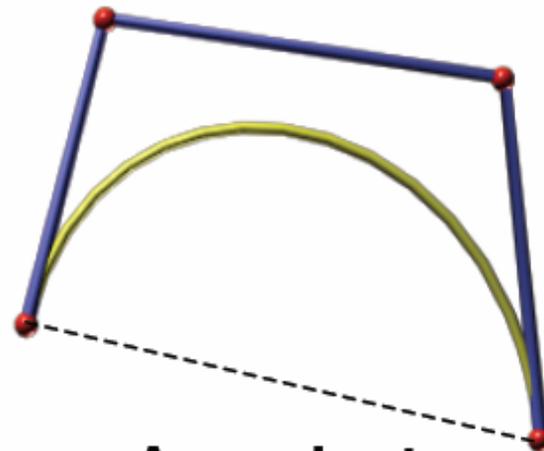


Dado um número n de pontos para traçar uma curva:

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



Interpolate



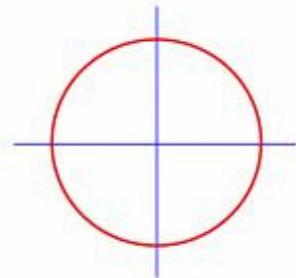
Approximate

Não paramétrica vs paramétrica

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

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

Exemplo
circunferência
representações
paramétricas

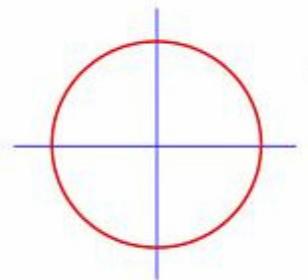


$$f : [0, 2\pi] \rightarrow \mathbb{R}^2, f(\theta) = (x(\theta), y(\theta))$$

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

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

Exemplo circunferência representações não paramétricas



Explícita $y = f(x)$

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

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

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

Implícita $f(x, y) = 0$

E essas?

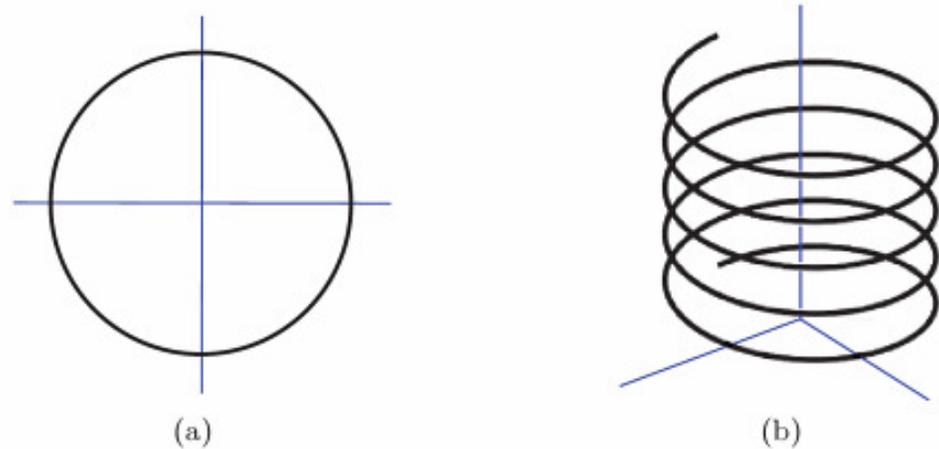


Fig. 1.1. (a) Image and (b) graph of $f(t) = (\cos t, \sin t)$.

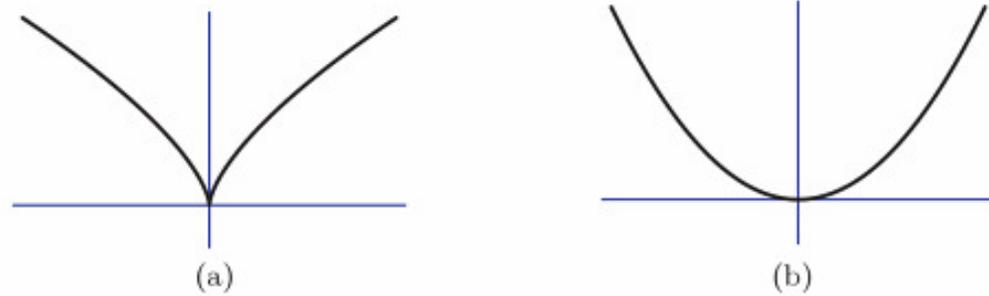


Fig. 1.2. (a) Cuspidal cubic $x^3 = y^2$ and (b) parabola $y = x^2$ as *images* of different metrizations.

Explícita $y = f(x)$

Implícita $f(x, y) = 0$

Representação implícita

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

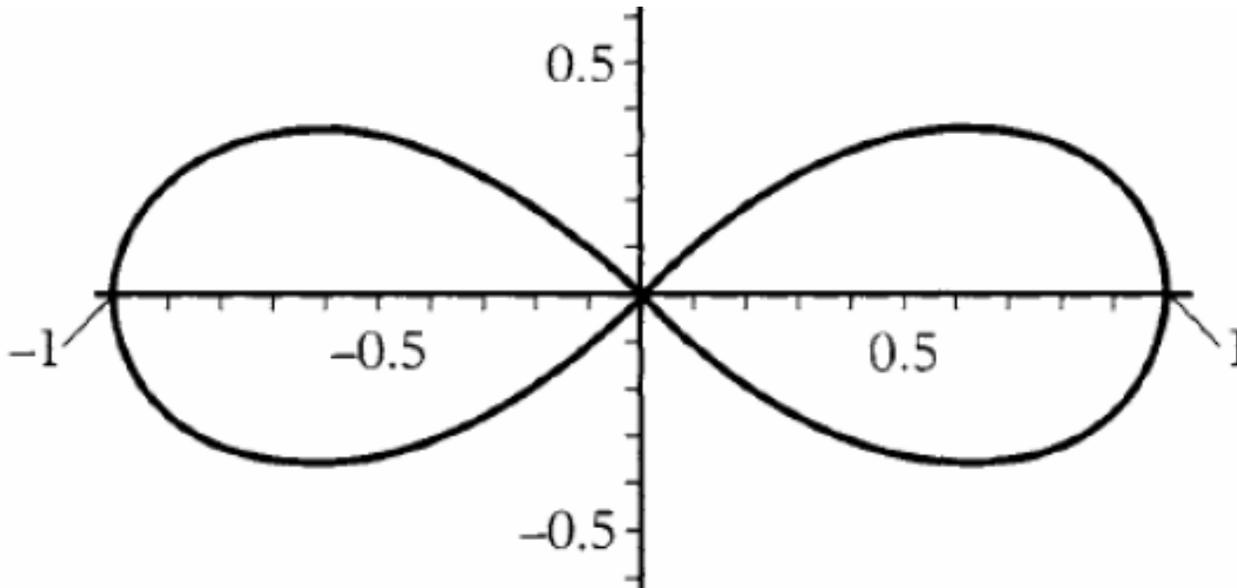
Outros exemplos:

- Lemniscata de Bernoulli => símbolo infinito

Quarto grau!

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

$$\text{Implícita } f(x, y) = 0$$



Curvas não paramétricas

- Linha
- Círculo
- Parábola
- Elipse
- Hipérbole

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

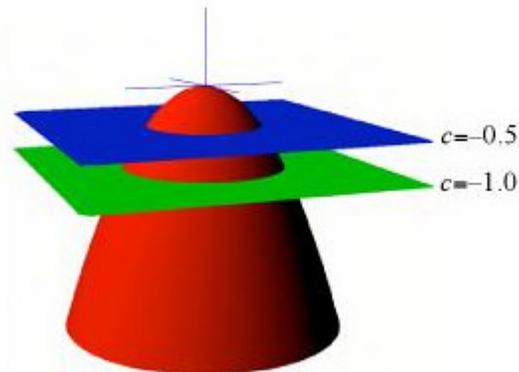
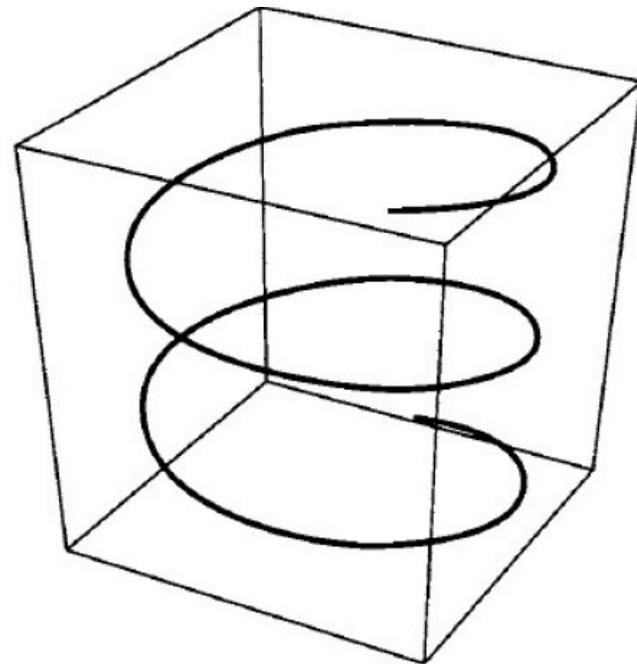


Fig. 1.7. The paraboloid $-z = x^2 + y^2$ in \mathbb{R}^3 .

Curvas paramétricas

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

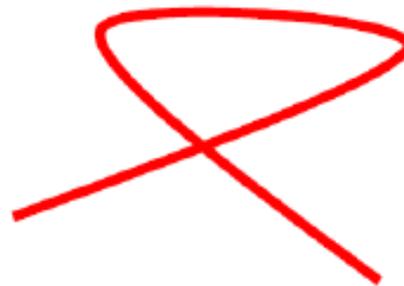
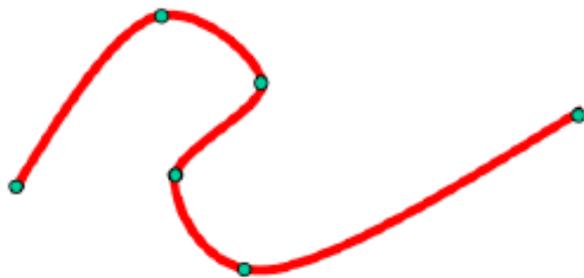
$$x = \cos(t), y = \sin(t), z = t/5$$



Peculiaridades das curvas em CG

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

- É difícil definir a equação não paramétrica de uma curva que passe por um conjunto de pontos pré-definidos.
- Não permite a representação de curvas com laços



Peculiaridades das curvas em CG

Para CG, representações paramétricas costumam ser as mais convenientes

Assim, genericamente, uma curva 3D é

$$- Q(t)=[x(t) \ y(t) \ z(t)]$$

$x(t)$, $y(t)$, $z(t)$ são chamadas de funções-base
(*base functions*)

$$P(u) = (X(u), Y(u), Z(u))$$

$$0 \leq u \leq 1$$

$$P(u, v) = (X(u, v), Y(u, v), Z(u, v))$$

$$0 \leq u \leq 1$$

$$0 \leq v \leq 1$$

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

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

$$dy/dx = \text{infinito} \Rightarrow dx/du = 0$$

- Elementos geométricos definidos parametricamente são inerentemente limitados ($0 \leq u \leq 1$).
- As expressões paramétricas são facilmente traduzidas na forma de vectores e matrizes.
- Utilização de um só modelo matemático para representar qualquer curva ou superfície.

Reta na forma paramétrica

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

$$- P_x = P_{x0} + at$$

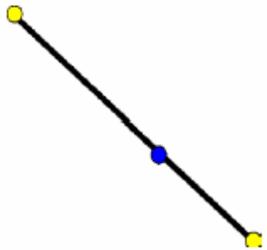
$$- P_y = P_{y0} + at$$

$$- P_z = P_{z0} + at$$



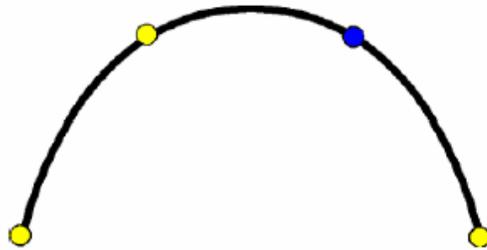
Parametrizando polinômios

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



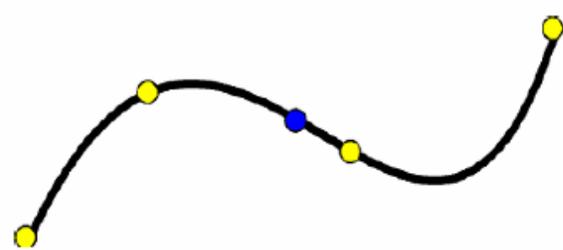
Linear

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



Quadrático

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



Cúbico

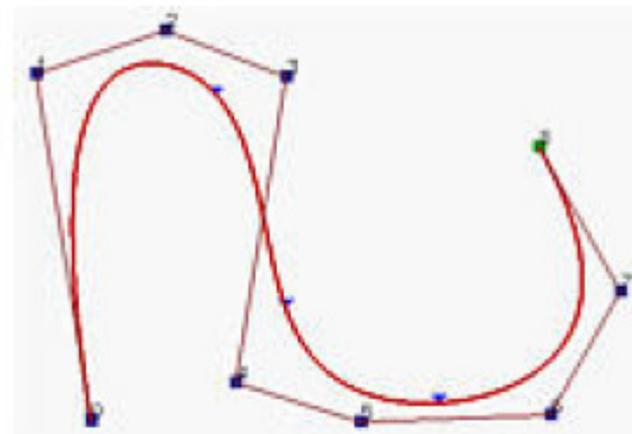
Peculiaridades das curvas em CG

A curva é definida através de um conjunto de **pontos de controle** que influenciam a forma da curva.

Os nós são pontos de controle que pertencem à curva.

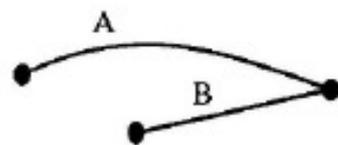
A curva pode ser interpolada, passando nesse caso por todos os pontos de controle, ou pode ser aproximada, passando apenas em alguns pontos de controle ou mesmo nenhum.

Os pontos de controle definem a fronteira de um polígono designado por *convex hull*.

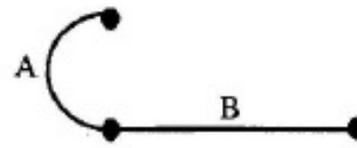


Continuidade

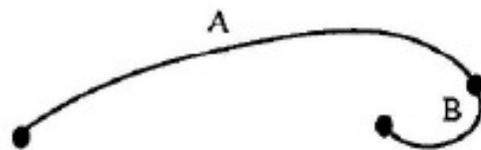
- Duas ou mais curvas nos nós (conectando pontos) para formar uma curva contínua
- Tipos de continuidade
 - Continuidade de ponto
 - Continuidade de tangente
 - Continuidade de curvatura



(a) Point continuity - C^0



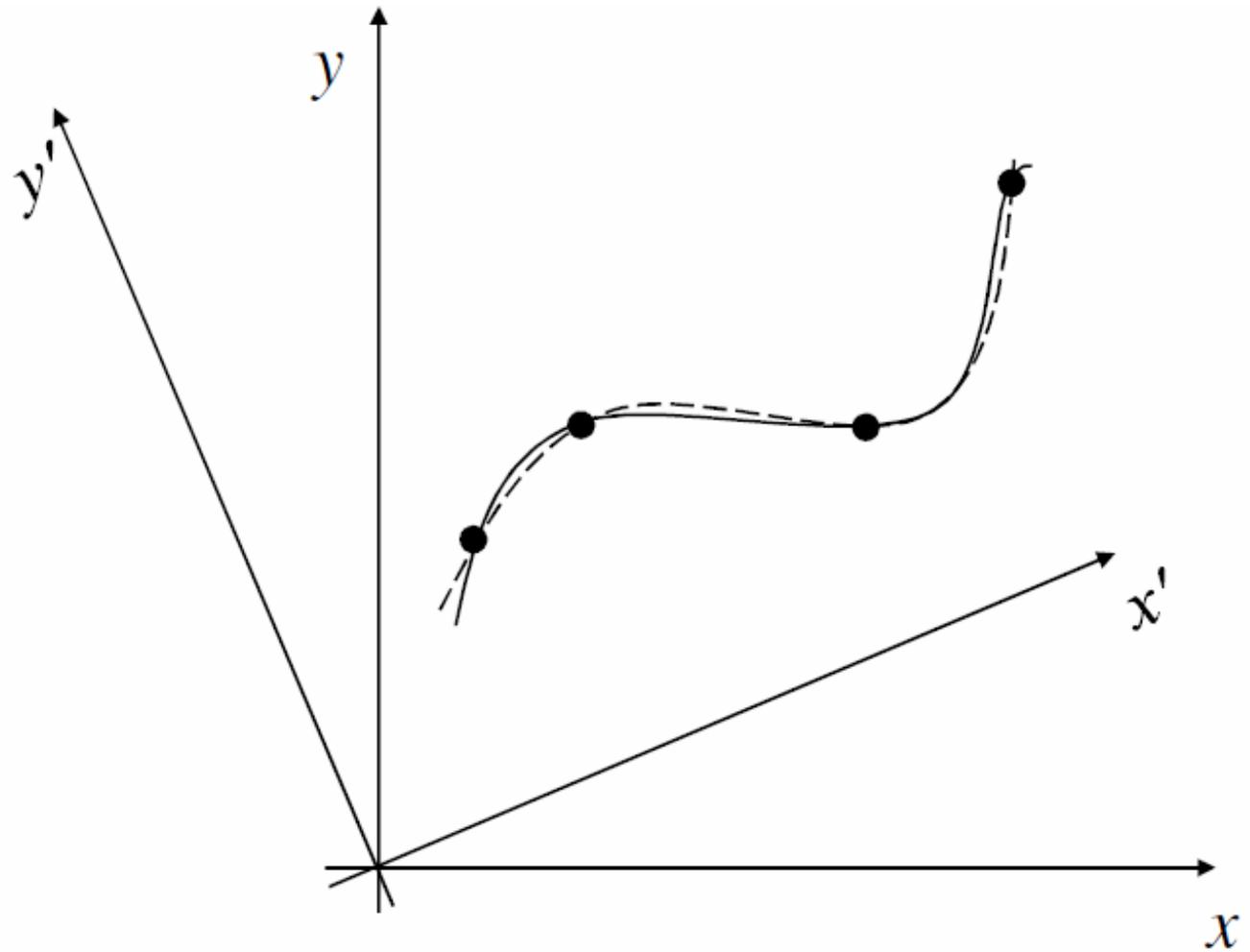
(b) Tangent continuity - C^1



(c) Curvature continuity - C^2

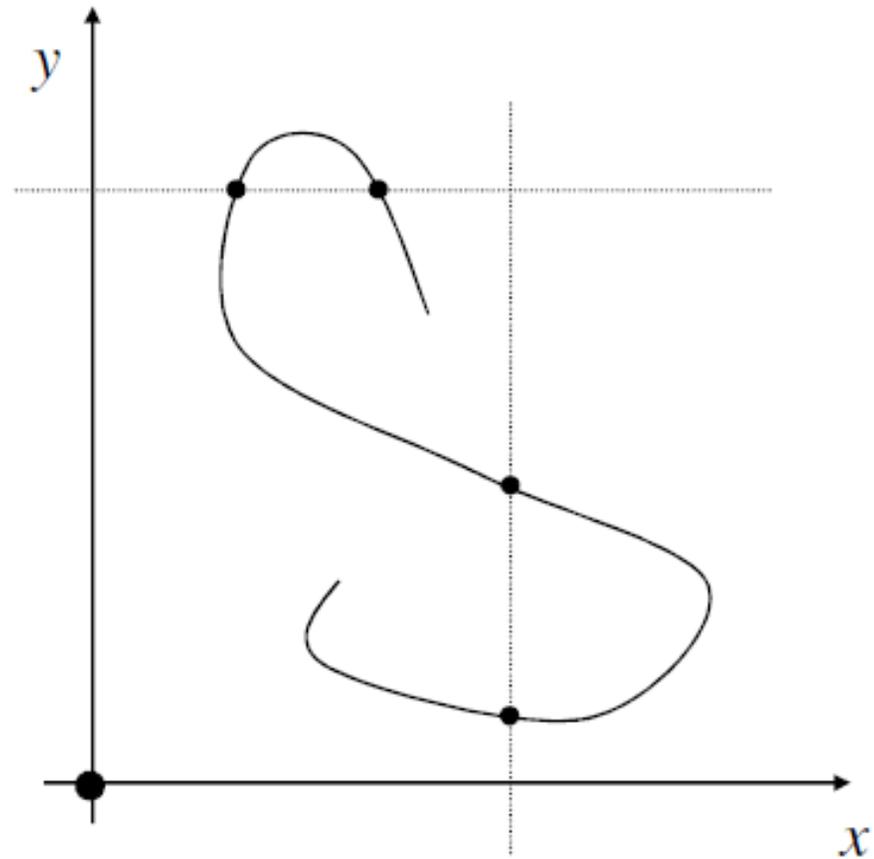
Propriedades desejáveis de curvas para modelagem em CG

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

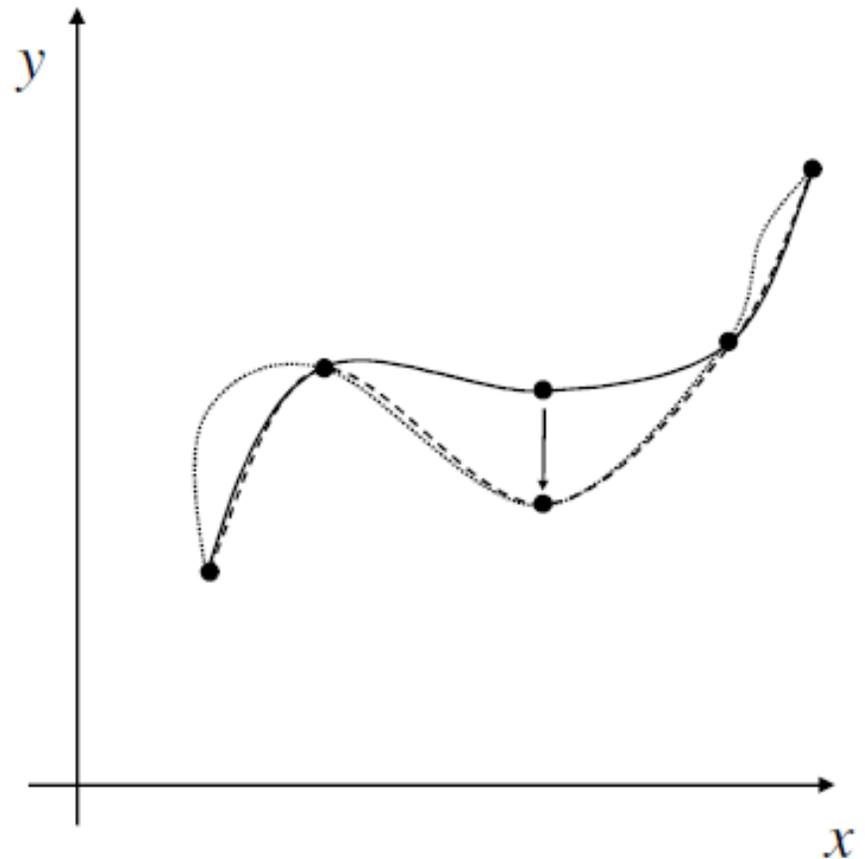
Controle local:

i.e. em ajuste finos:

alterar um trecho

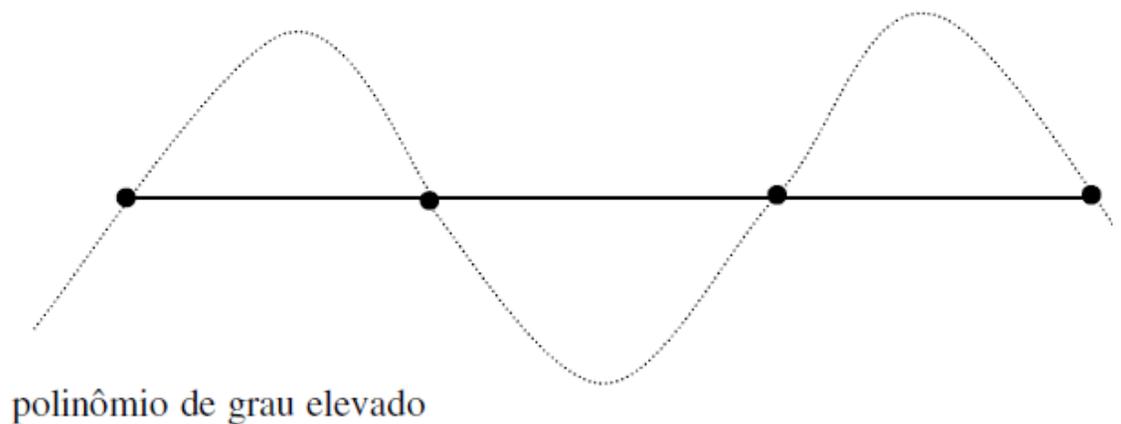
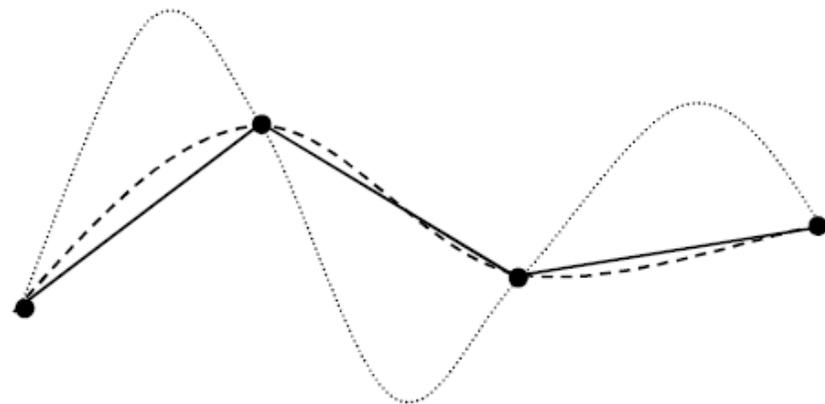
não altera toda a

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

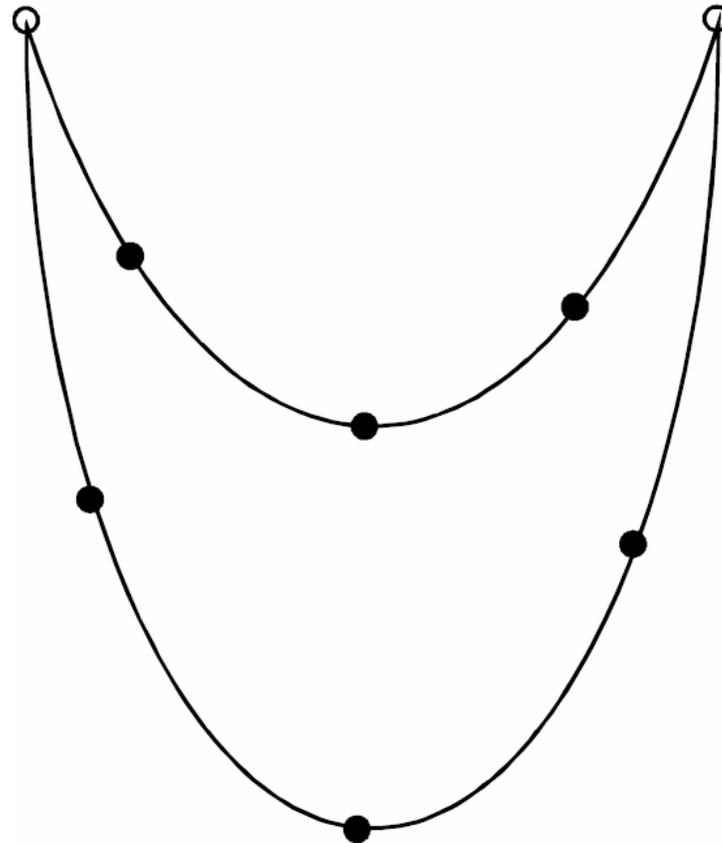


polinômio de grau elevado

formas com oscilações.

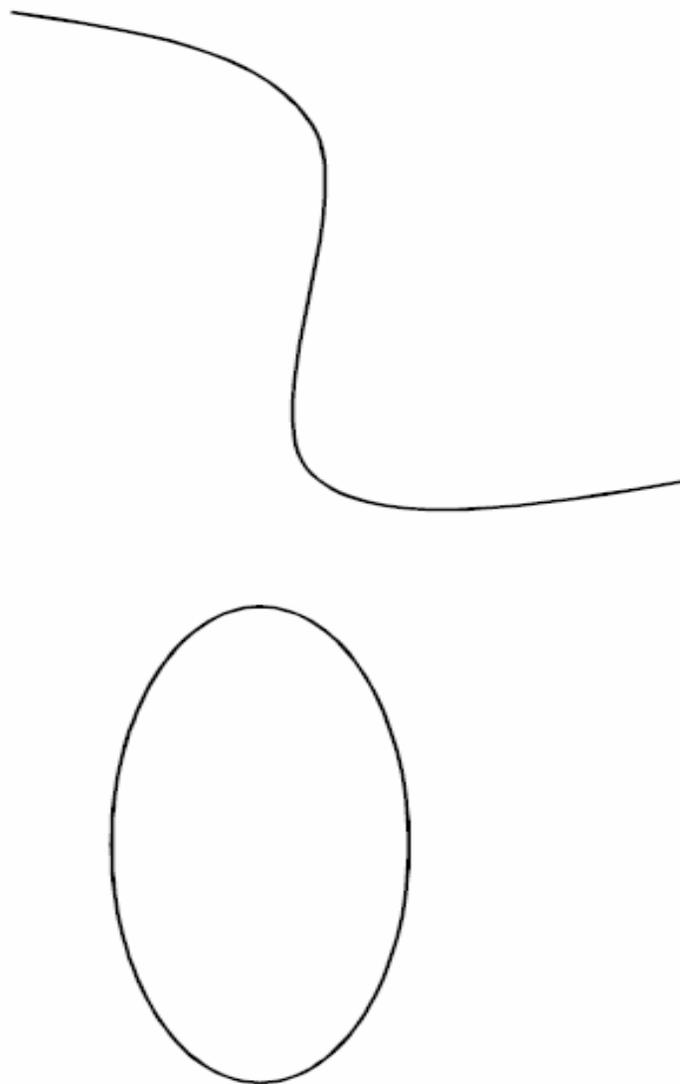
Propriedades desejáveis de curvas para modelagem em CG

Ser possível
representar
diversos graus
de
continuidades
que o usuário
desejar



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

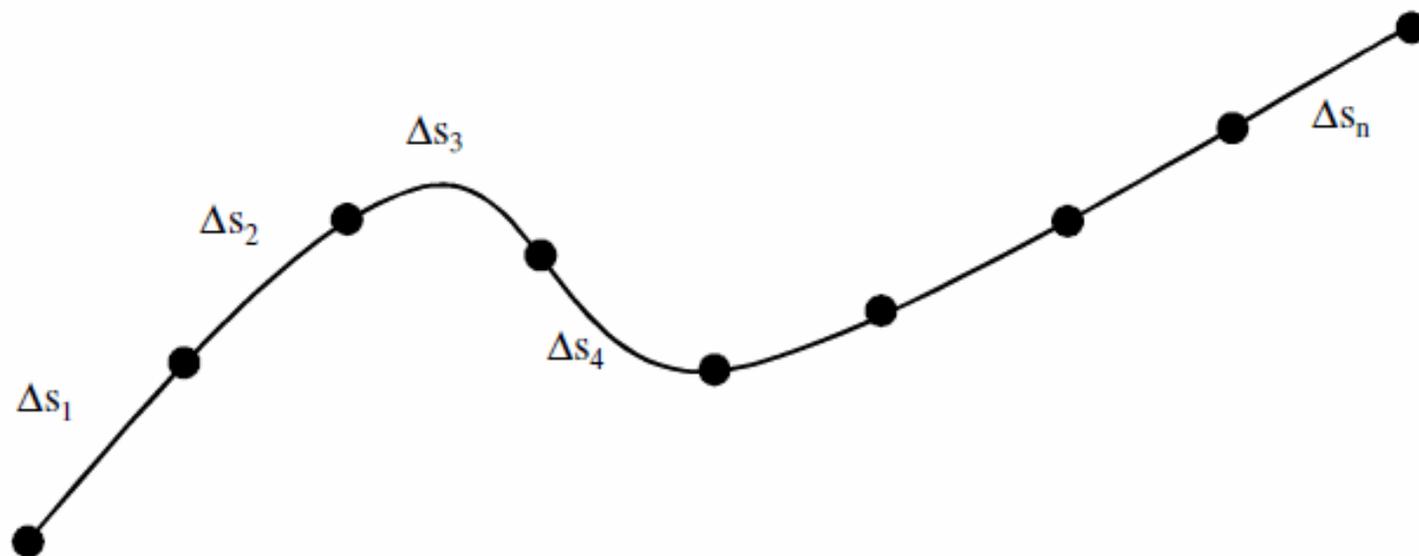
ter pontos

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
- Representadas por uniões
- Descritas por polinômios
- Parametrizadas
- Até grau 3
- Com continuidade paramétrica

Porque polinômios até terceiro grau?

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

Polinômios são fáceis de avaliar

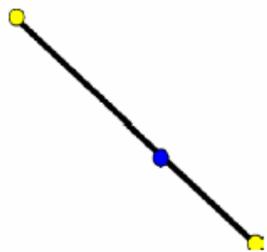
Grau menor que 3: Pouca flexibilidade.

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

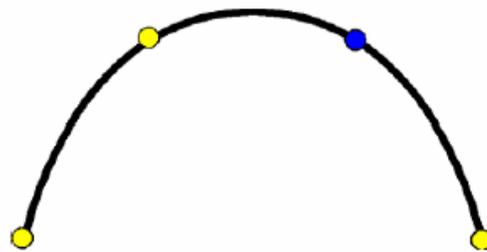
$$f(t) = at + b$$

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

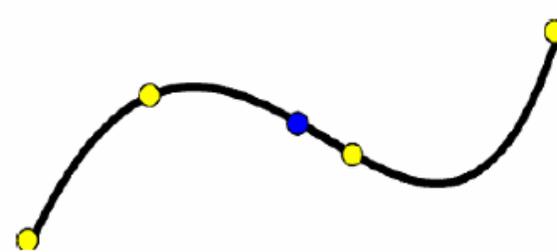
$$f(t) = at^3 + bt^2 + ct + d$$



Linear



Quadrático



Cúbico

Curvas Paramétricas Cúbicas 2D

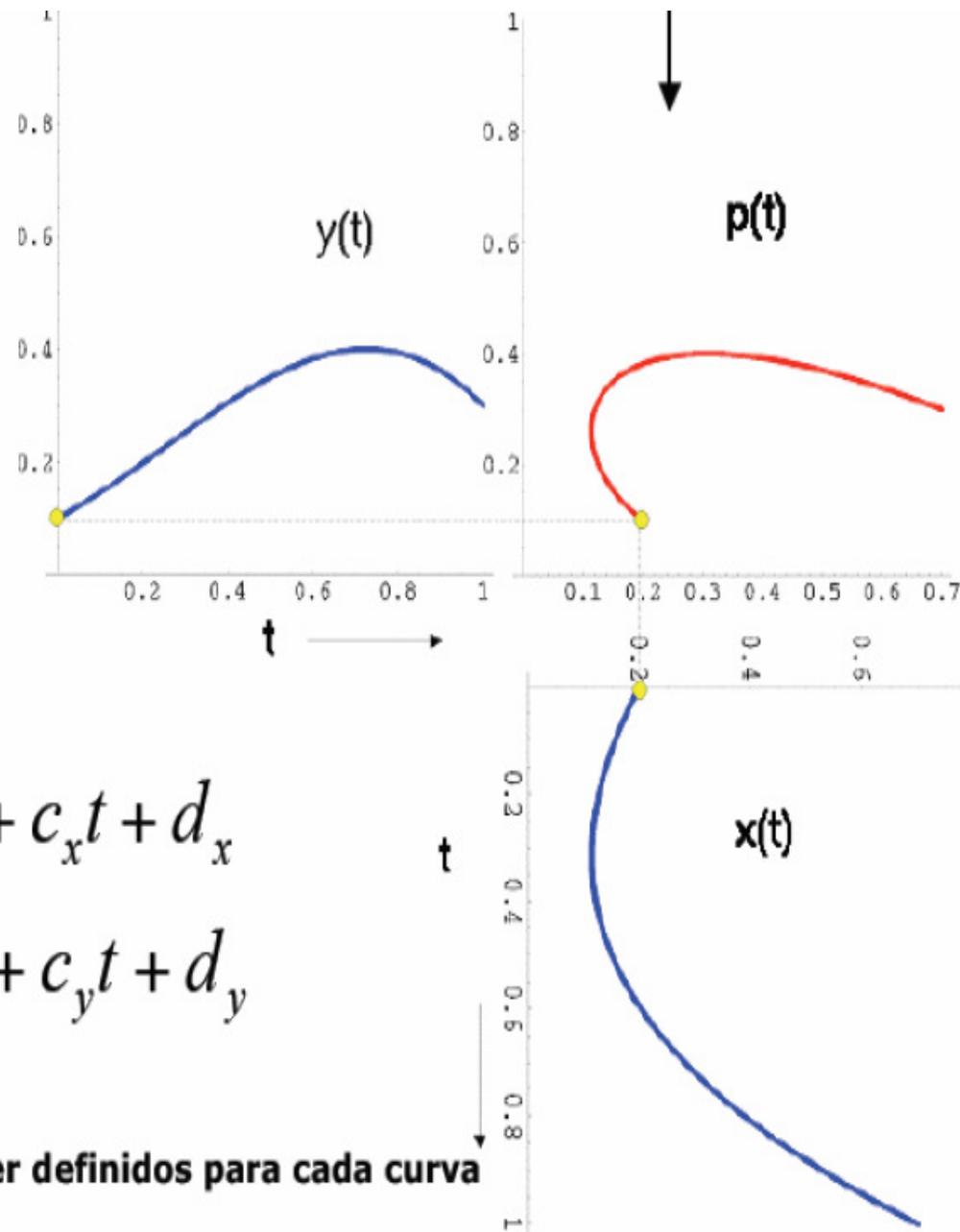
9 parâmetros
para cada curva

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

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



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



Em 3D

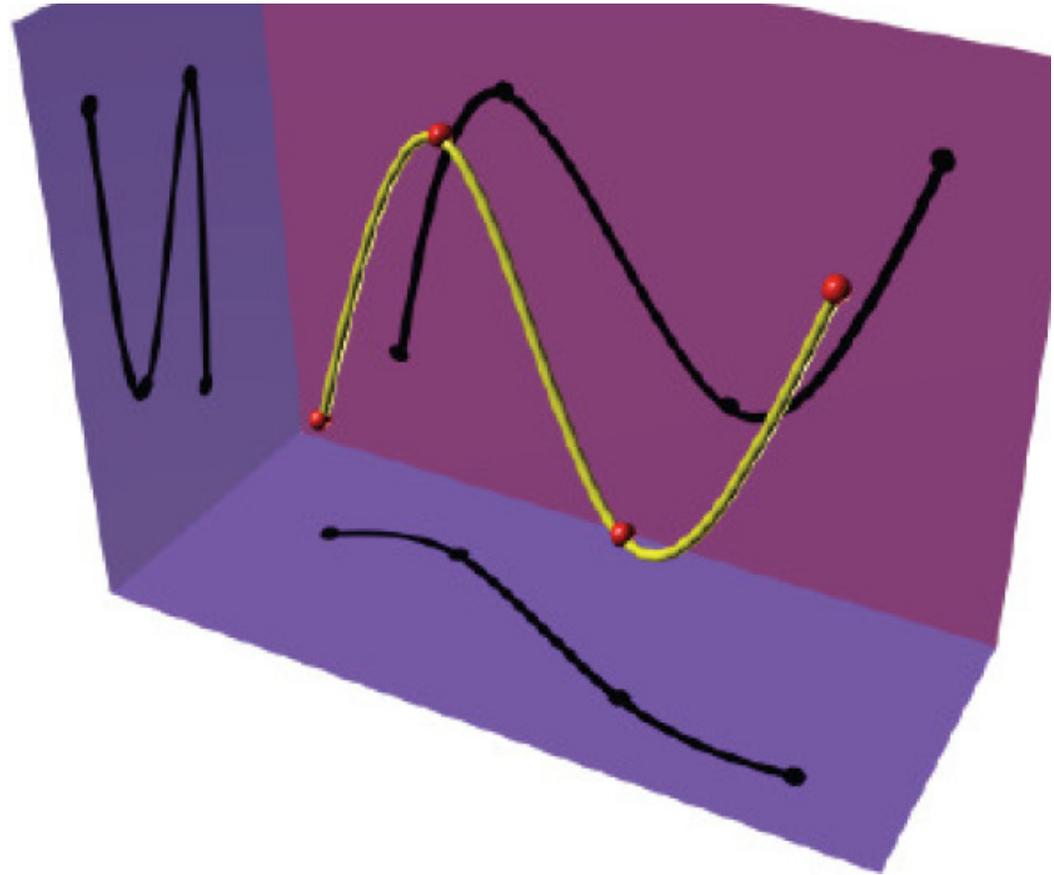
Curva polinomial paramétrica:

$$P(u) = c_0 + c_1u + c_2u^2 + \dots + c_ku^k$$

$$\begin{cases} x(u) = c_{0x} + c_{1x}u + c_{2x}u^2 + \dots + c_{kx}u^k \\ y(u) = c_{0y} + c_{1y}u + c_{2y}u^2 + \dots + c_{ky}u^k \\ z(u) = c_{0z} + c_{1z}u + c_{2z}u^2 + \dots + c_{kz}u^k \end{cases}$$

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

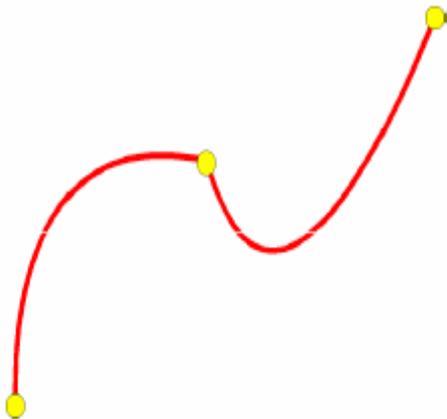
$$p(u) = c_0 + c_1u + c_2u^2 + c_3u^3 = \sum_{k=0}^3 c_k u^k = \mathbf{u}^T \mathbf{c},$$

onde

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

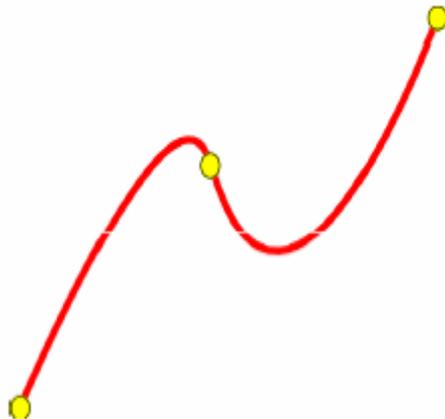
continuidade paramétrica e geométrica

Foley at al p. 480 - 483



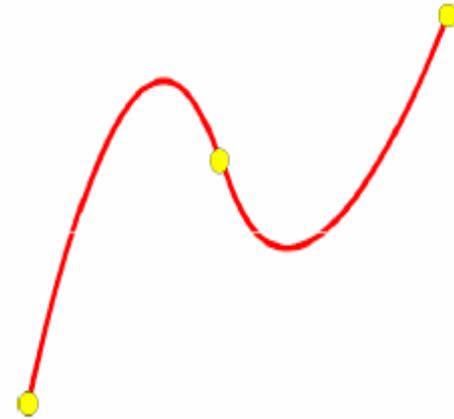
Continuidade Geométrica G0

Dois segmentos se encontram em um ponto



Continuidade Geométrica G1

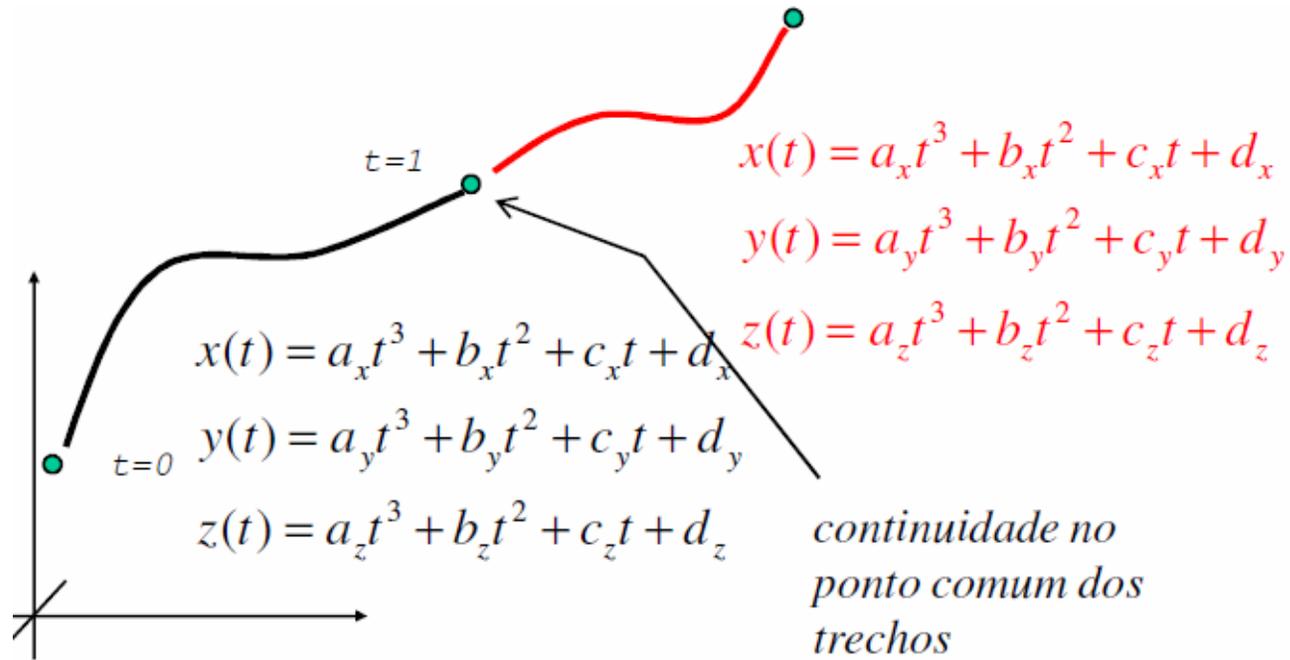
Direção das tangentes dos segmentos são iguais no ponto de junção



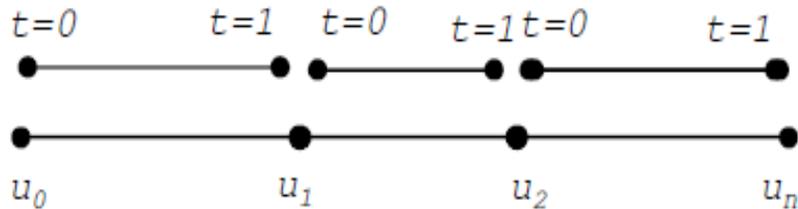
Continuidade Paramétrica C1

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

Com continuidade paramétrica



Parametrização



$t \in [0,1]$ local

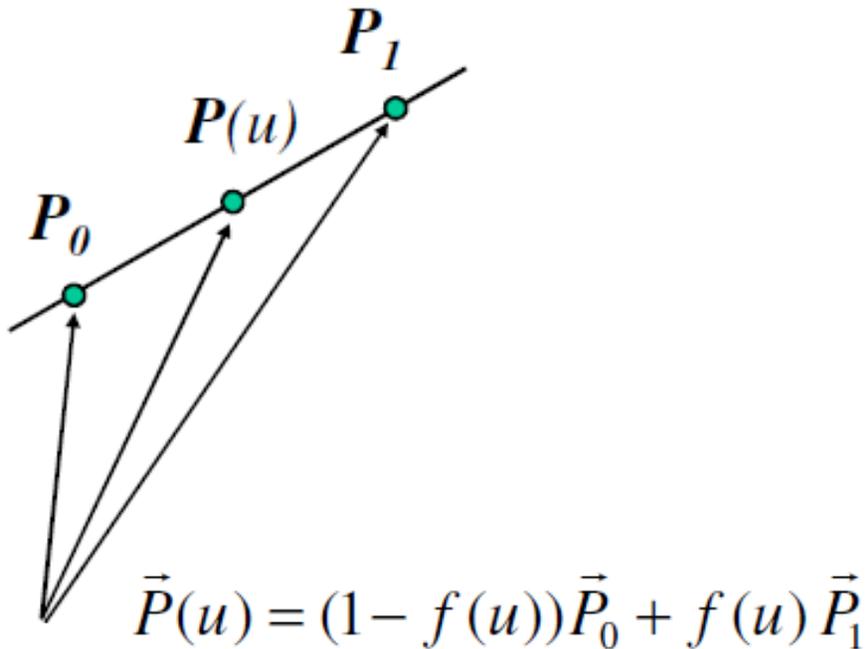
ou

$u \in [u_0, u_n]$ global

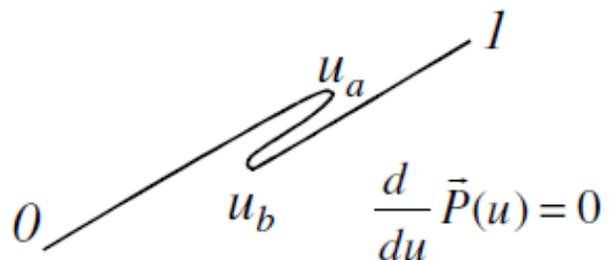
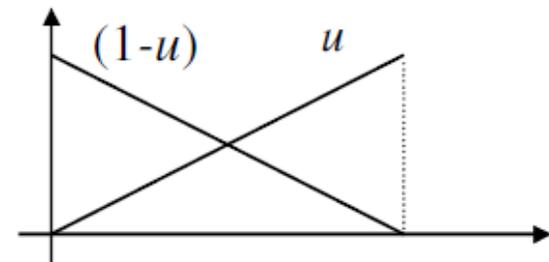
Requisitos para os parâmetros:

Parâmetro genérico: u

Parâmetro de comprimento: $s = s(u)$

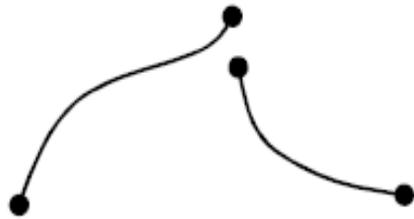


$$\vec{P}(u) = (1 - u)\vec{P}_0 + u\vec{P}_1$$



Com continuidade paramétrica
 Se $u_2 > u_1 \Rightarrow s(u_2) > s(u_1)$

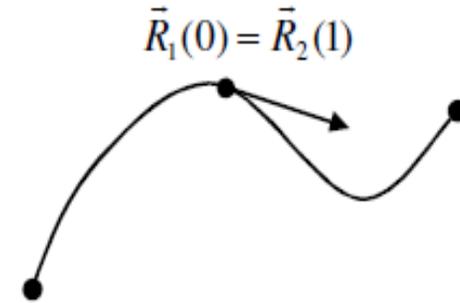
Continuidade geométrica x paramétrica



Descontínua

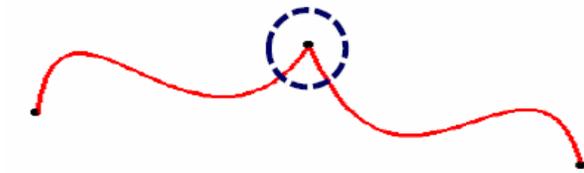


Contínua: C^0 e G^0

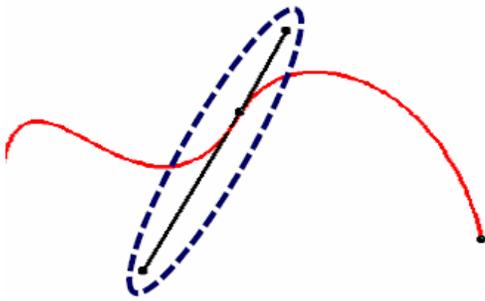


Contínua: C^1 e G^1

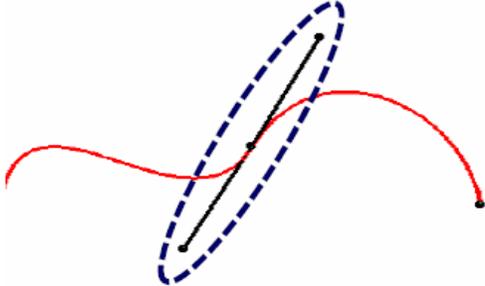
parametricas



C^0 : As duas curvas apresentam um ponto de junção.



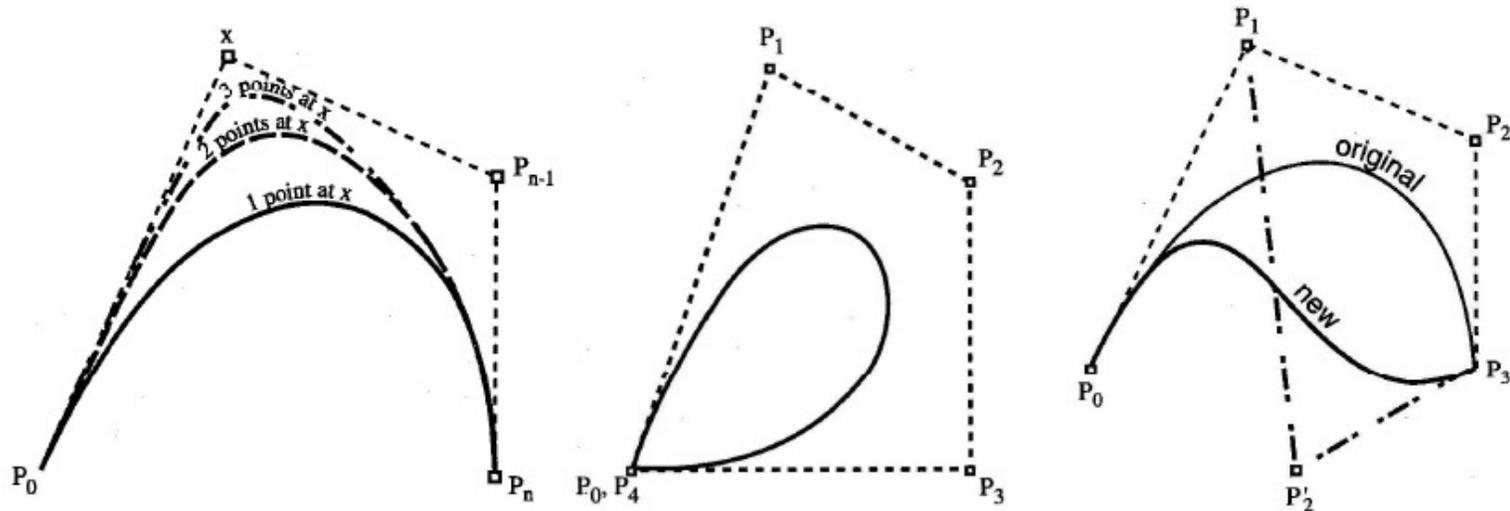
C^1 : A direção dos vetores tangentes no ponto de junção é igual.



C^2 : A direção e a magnitude dos vetores tangentes no ponto de junção são iguais.

Curva de Bezier

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



Curva polinomial desenvolvida em 1962 por Pierre Bézier.

Utilizada no projeto de automóveis (Renault).

Baseada no algoritmo de De Casteljaou em 1957.

Curva de aproximação.

Controle global.

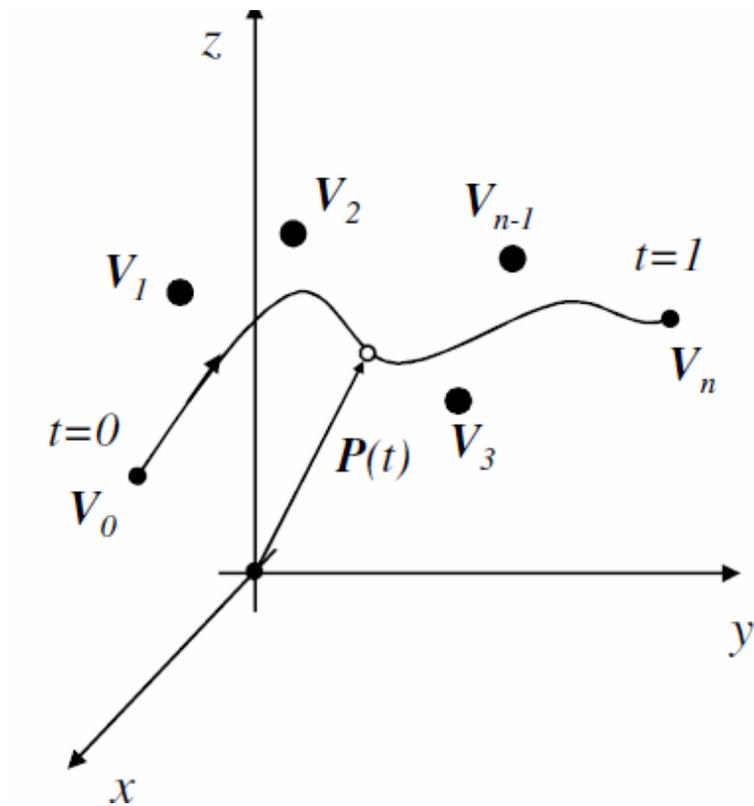


P. de Casteljaou, 1959 (Citroën)

P. de Bézier, 1962 (Renault) - UNISURF

Forest 1970: Polinômios de Bernstein

Forma geral:



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

onde:

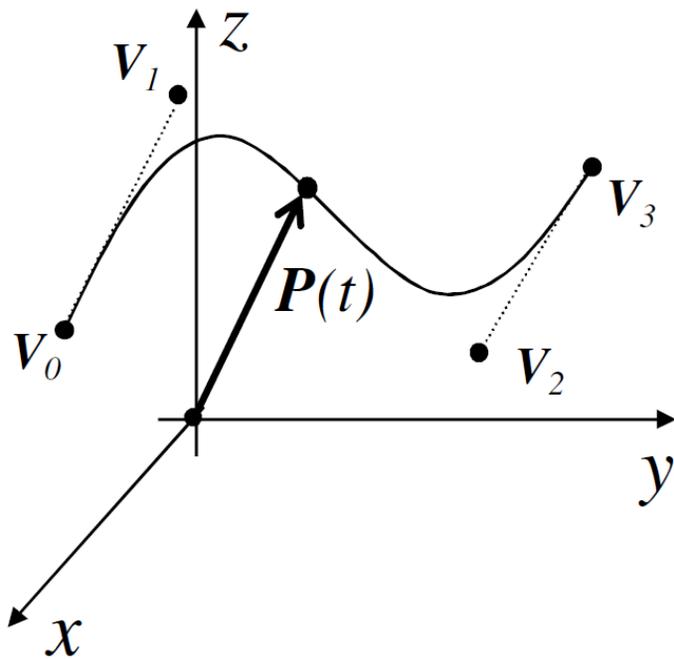
$$B_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

pol. Bernstein

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

coef. binomial

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



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

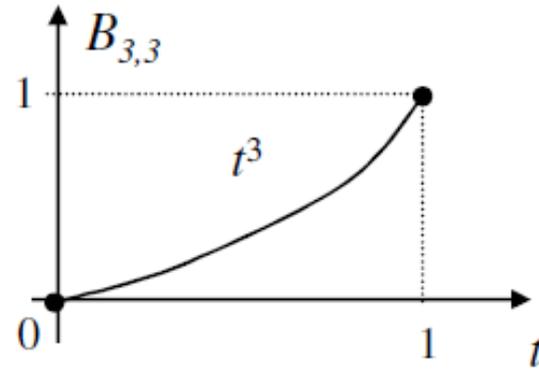
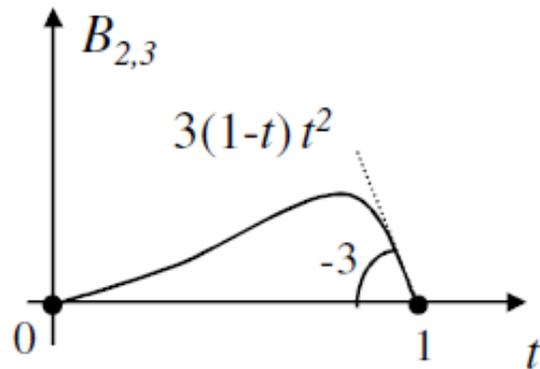
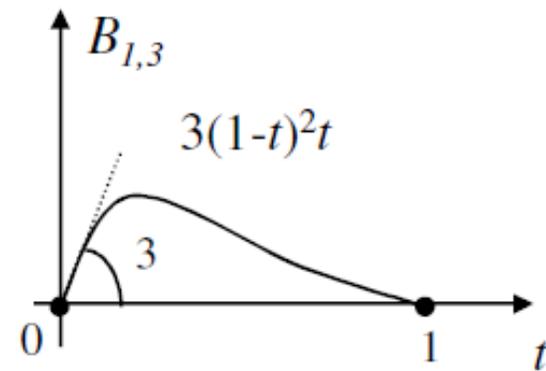
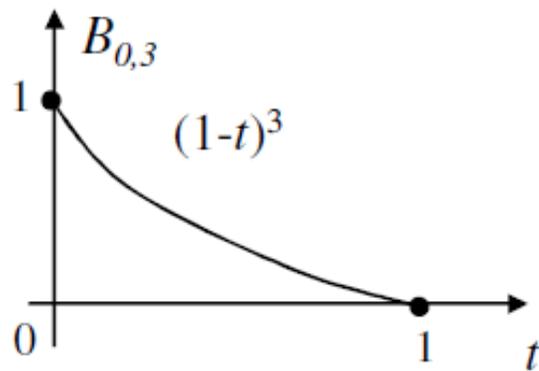
$$B_{1,3}(t) = \binom{3}{1} (1-t)^{3-1} t^1 = 3(1-t)^2 t$$

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

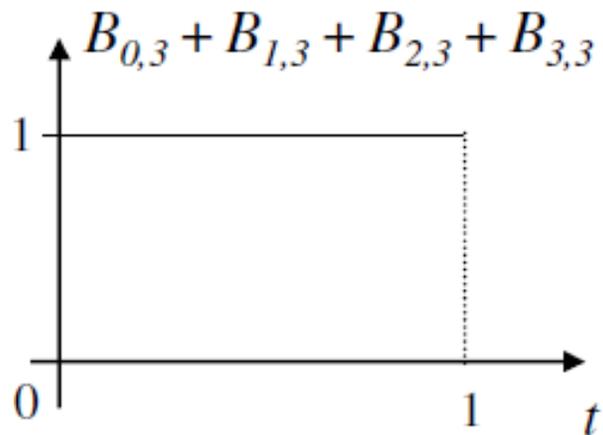
$$B_{3,3}(t) = \binom{3}{3} (1-t)^{3-3} t^3 = t^3$$

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

Polinômios cúbicos de **Bernstein**

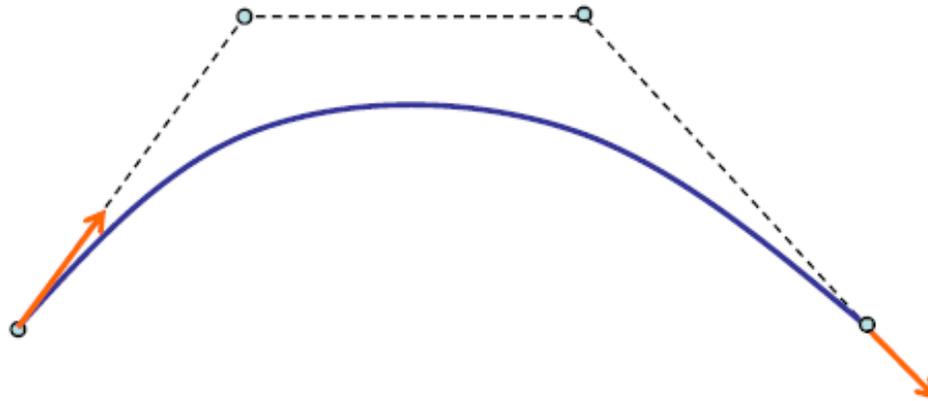


A soma dos
Polinômios Cúbicos de Bernstein
resulta:



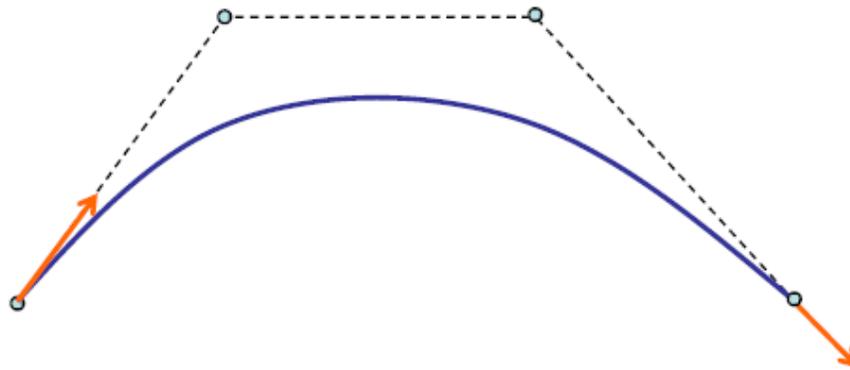
Propriedades de Curva de Bézier

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



Cont.

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



Demonstrando essas propriedades para uma Bezier cúbica:

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

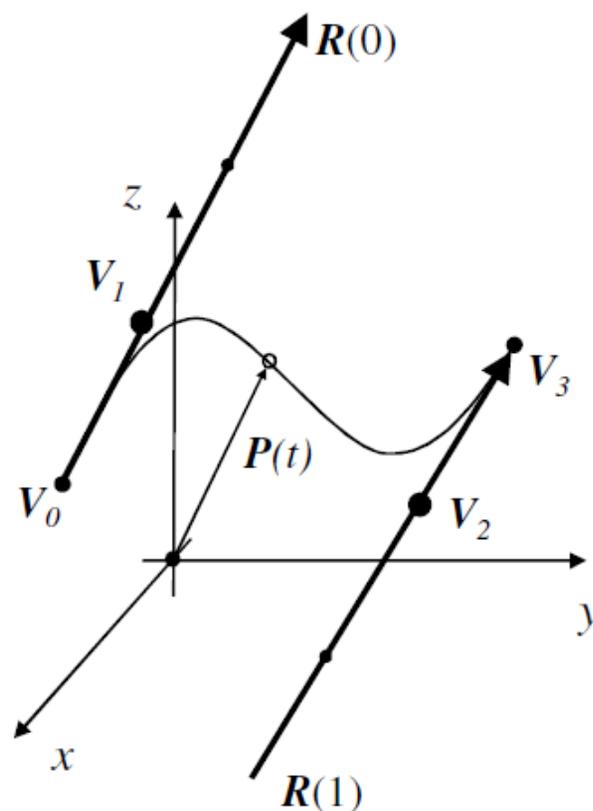
$$\frac{d}{dt} \vec{P}(t) = -3(1-t)^2 \vec{V}_0 + [-6(1-t)t + 3(1-t)^2] \vec{V}_1 + [-3t^2 + 6(1-t)t] \vec{V}_2 + t^3 \vec{V}_3$$

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

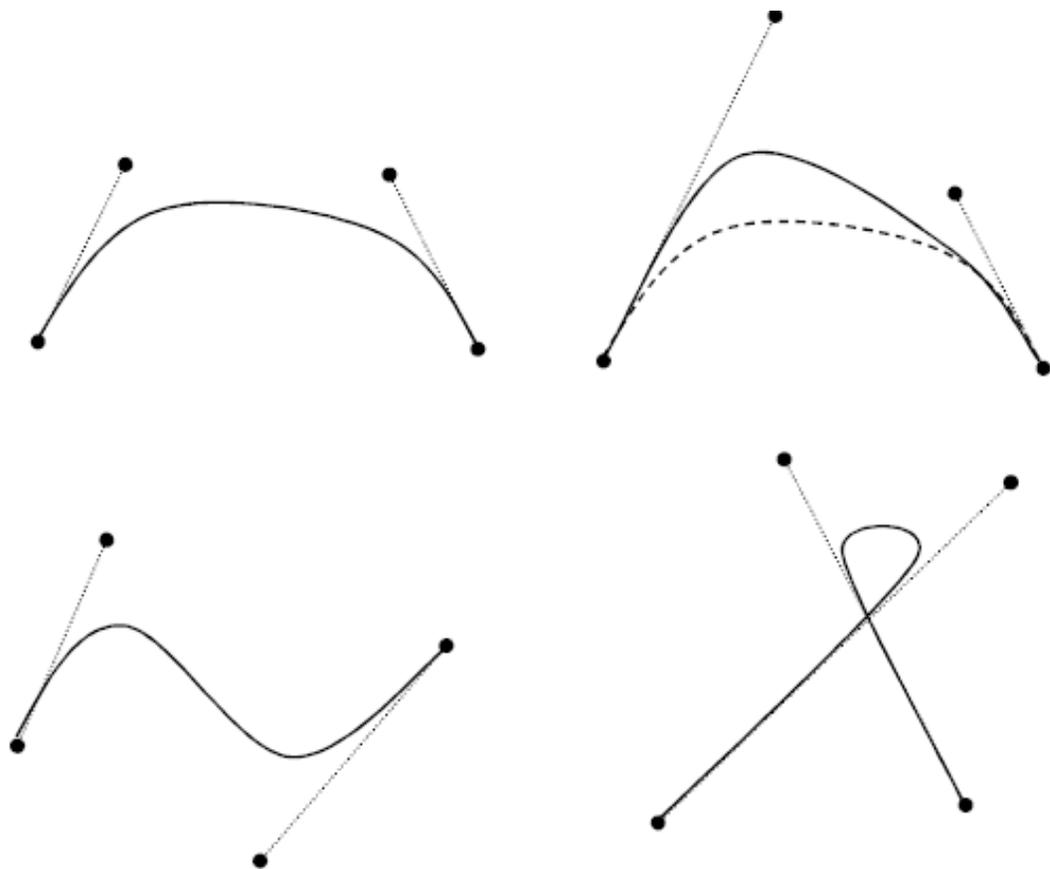
$$\vec{P}(1) = \vec{V}_3$$

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

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



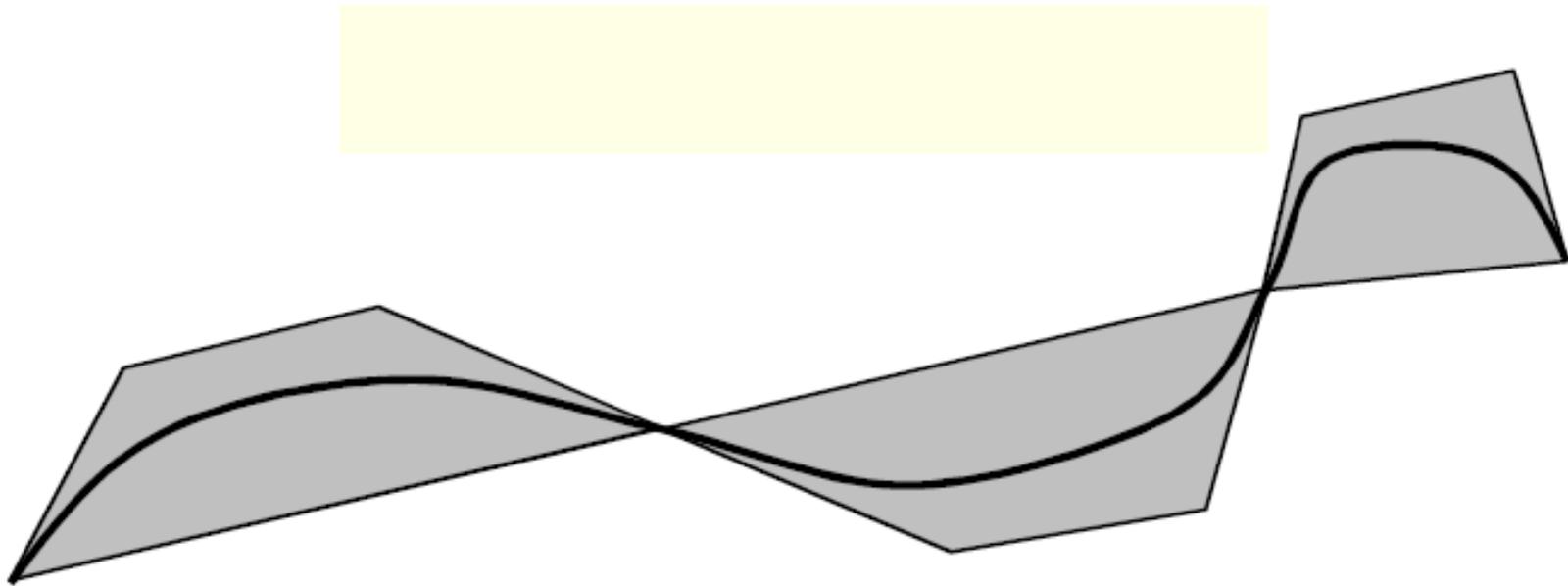
A ordem e posição dos pontos controla a curva!



Fecho convexo

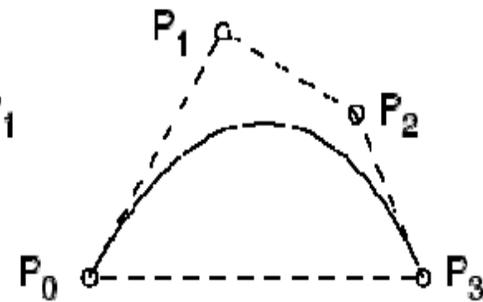
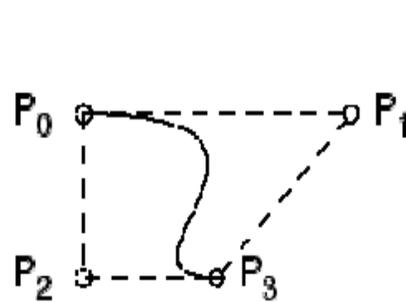
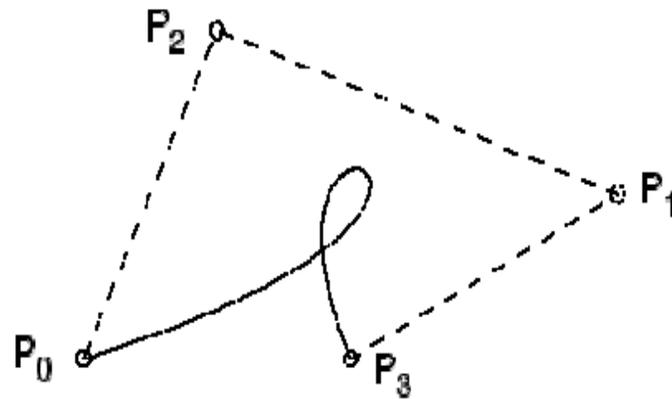
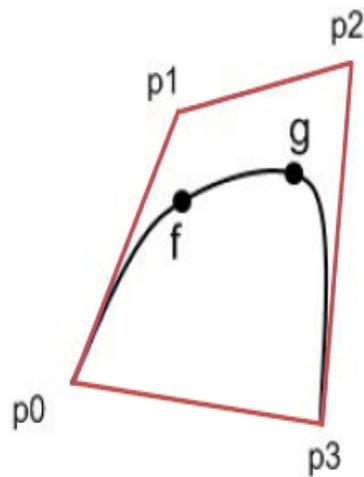
$$\vec{P}(t) = \sum_{i=0}^n \alpha_i \vec{V}_i \quad \text{com} \quad \sum_{i=0}^n \alpha_i = 1$$

- Convex hull

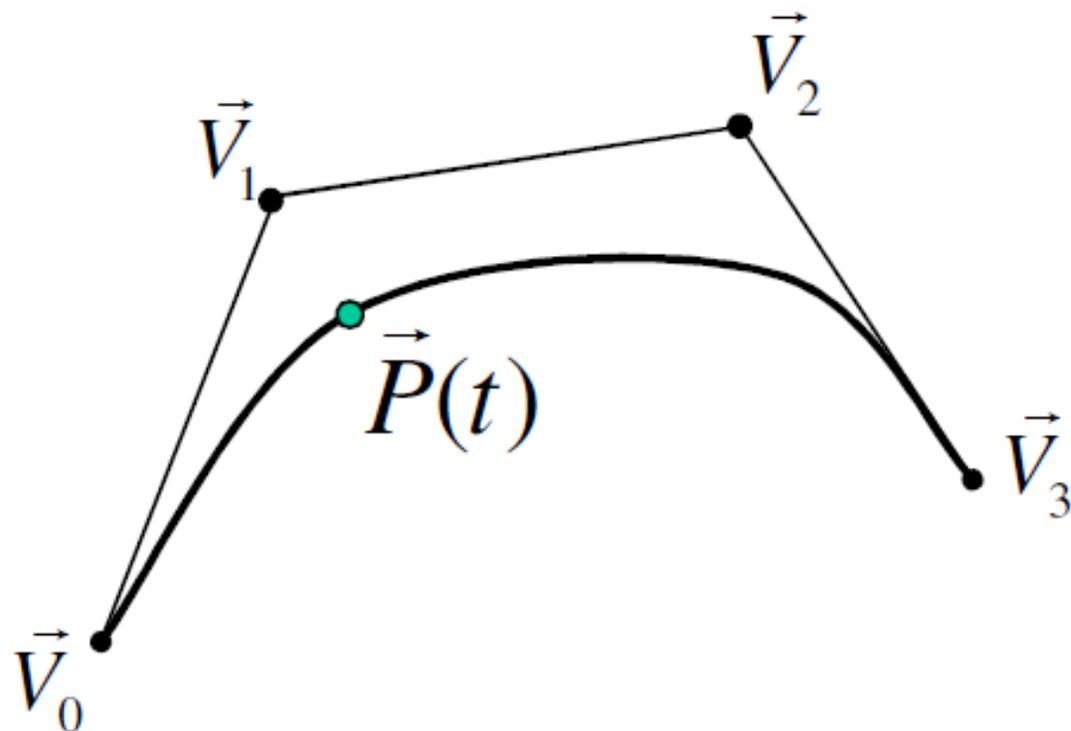


Propriedade: *Convex Hull*

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



Representação
matricial :



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

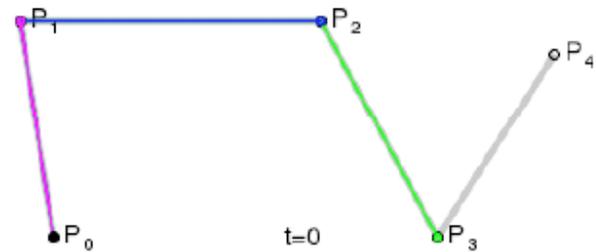
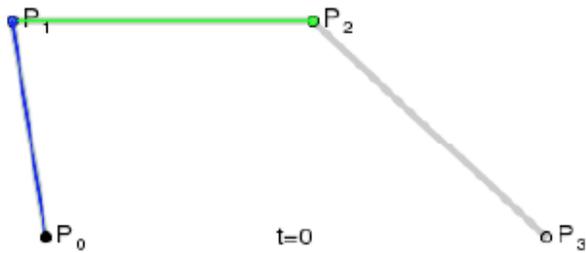
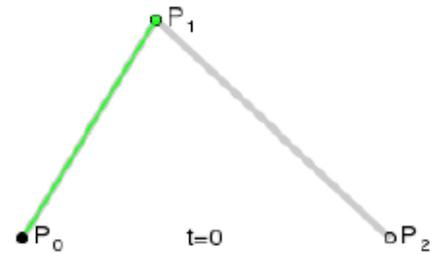
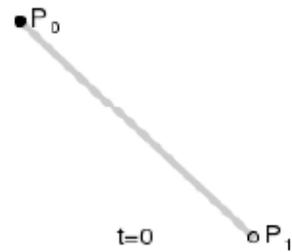
$$\vec{P}(t) = \left\langle t^3 \quad t^2 \quad t \quad 1 \right\rangle \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & -3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{0x} & V_{0y} & V_{0z} \\ V_{1x} & V_{1y} & V_{1z} \\ V_{2x} & V_{2y} & V_{2z} \\ V_{3x} & V_{3y} & V_{3z} \end{bmatrix}$$

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

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

$Q(t) = TMG$

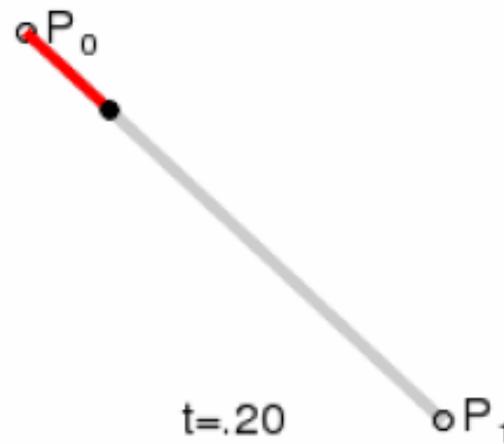
Outras formas de Bezier



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

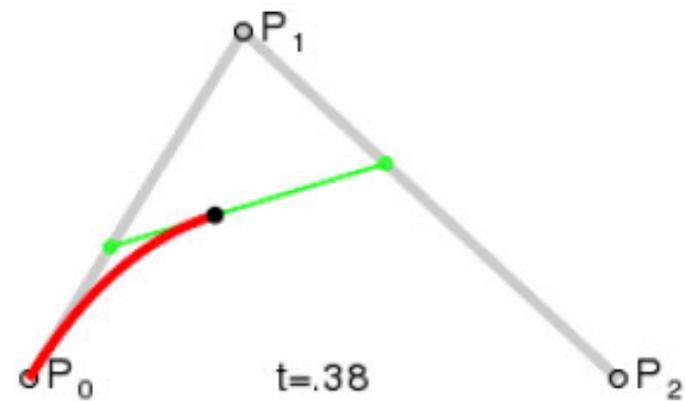
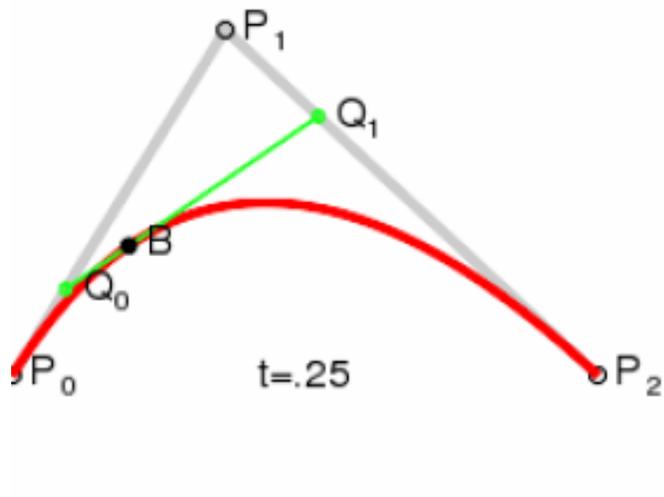
Outras formas de Bezier

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

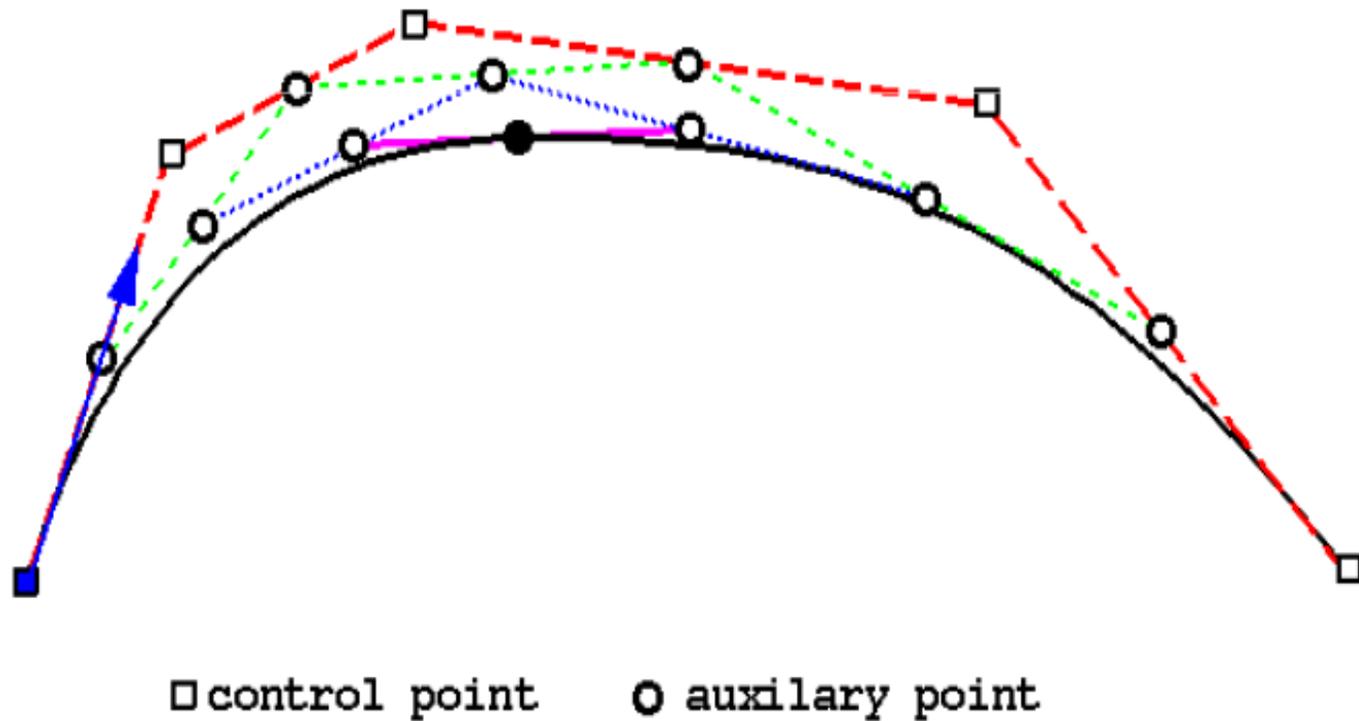


Outras formas de Bezier

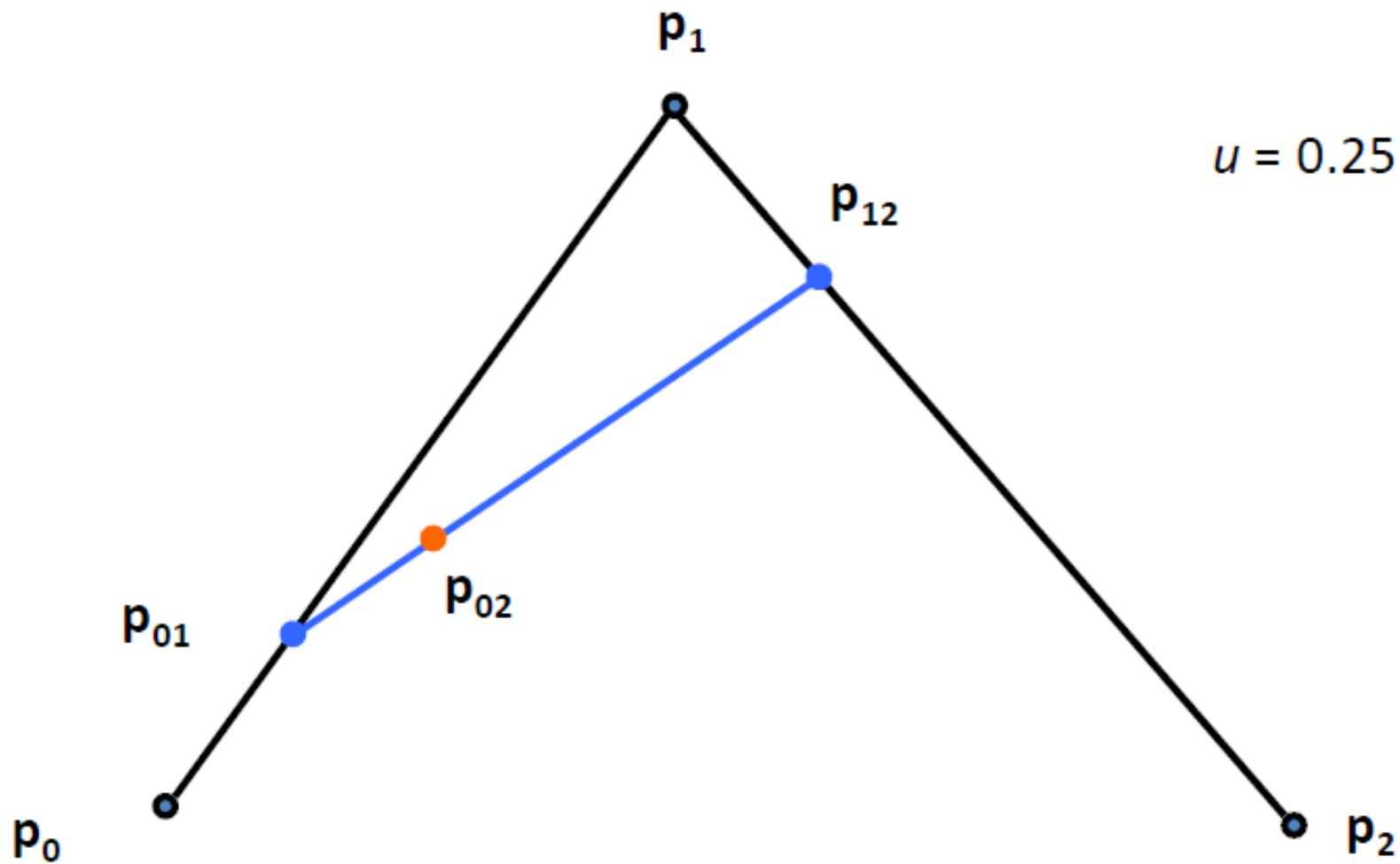
$$P(t) = (1 - t)^2 \mathbf{P}_0 + 2 t (1 - t) \mathbf{P}_1 + t^2 \mathbf{P}_2$$

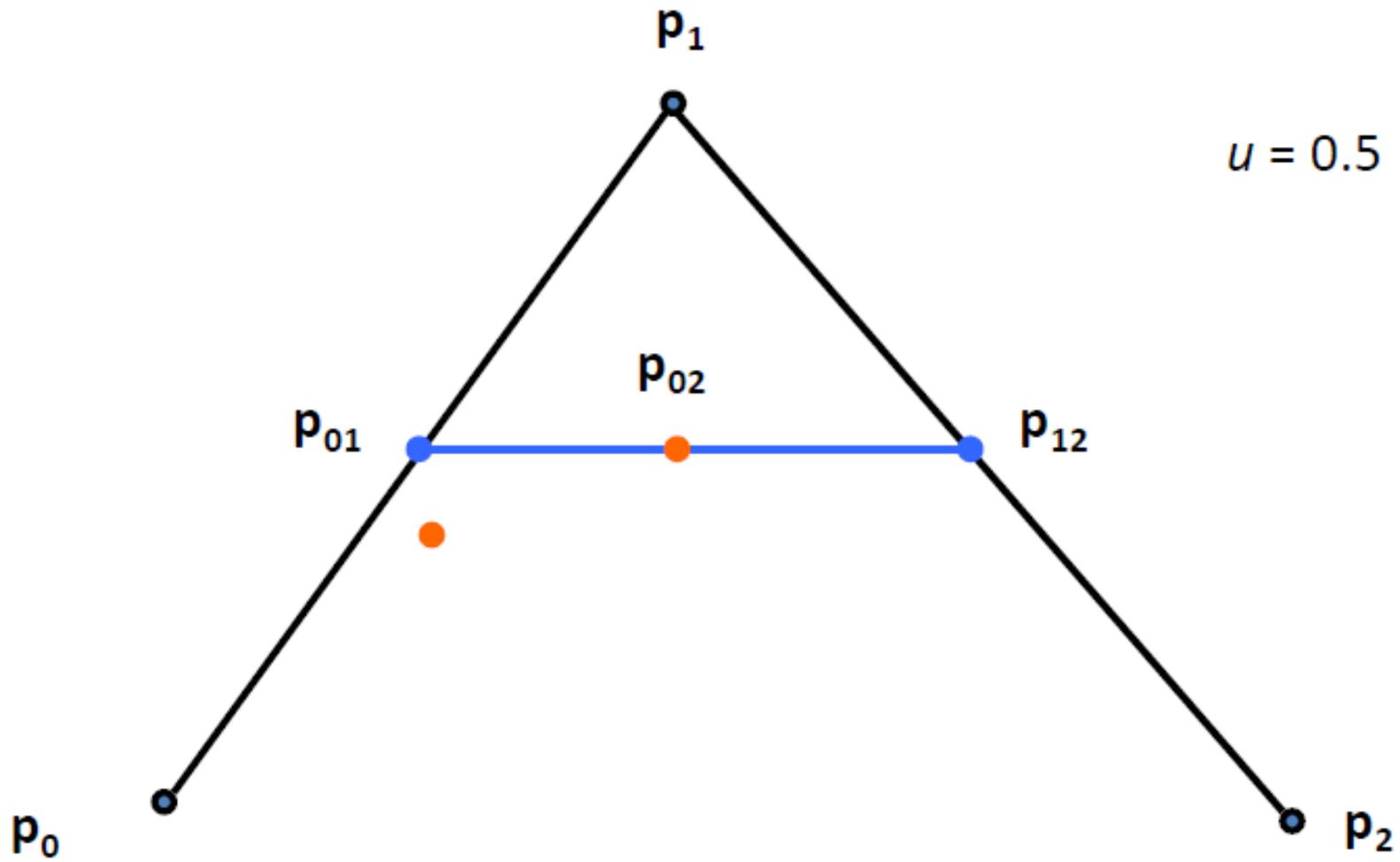


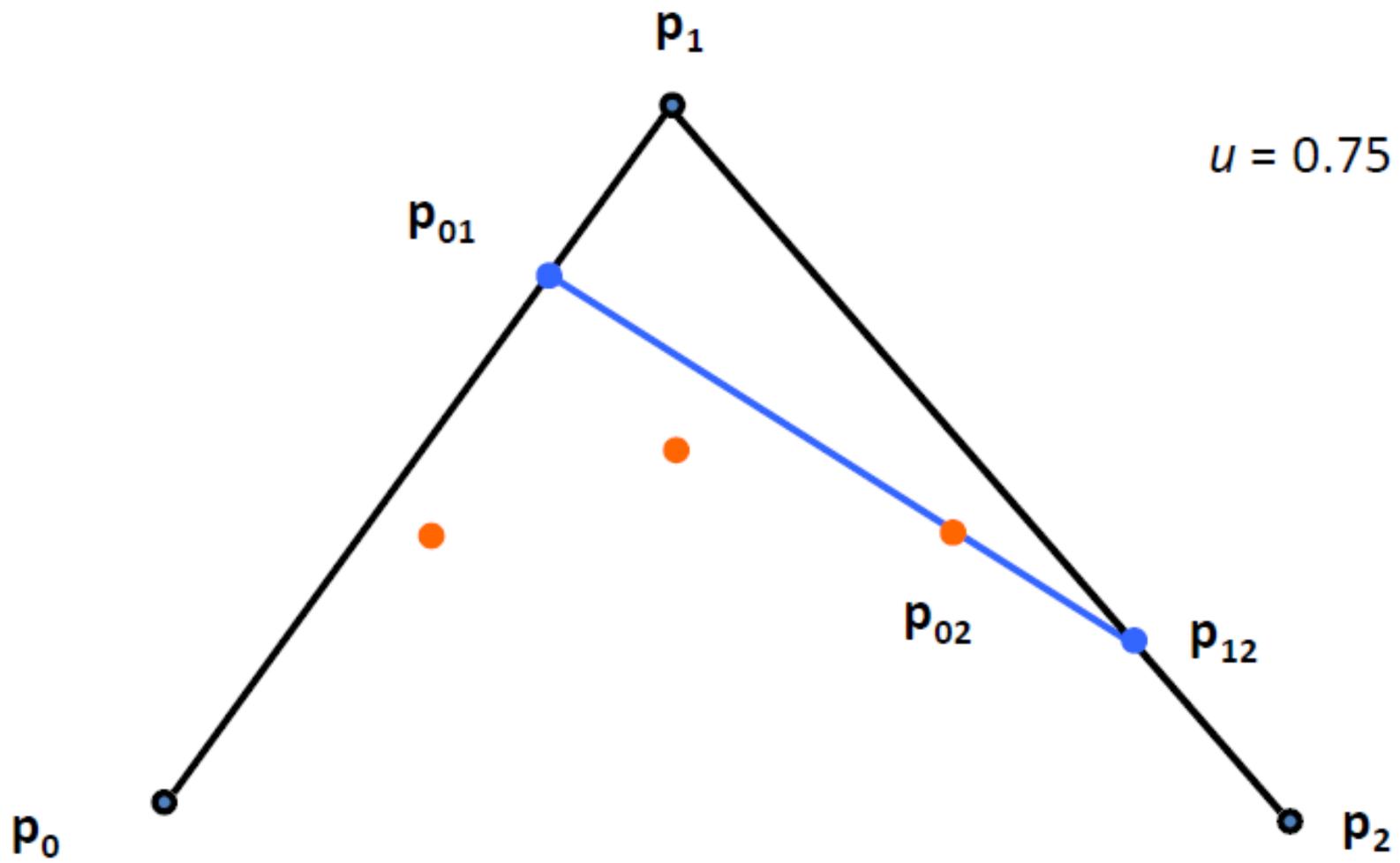
De Casteljau: algoritmo geométrico para construção de curvas Bézier.

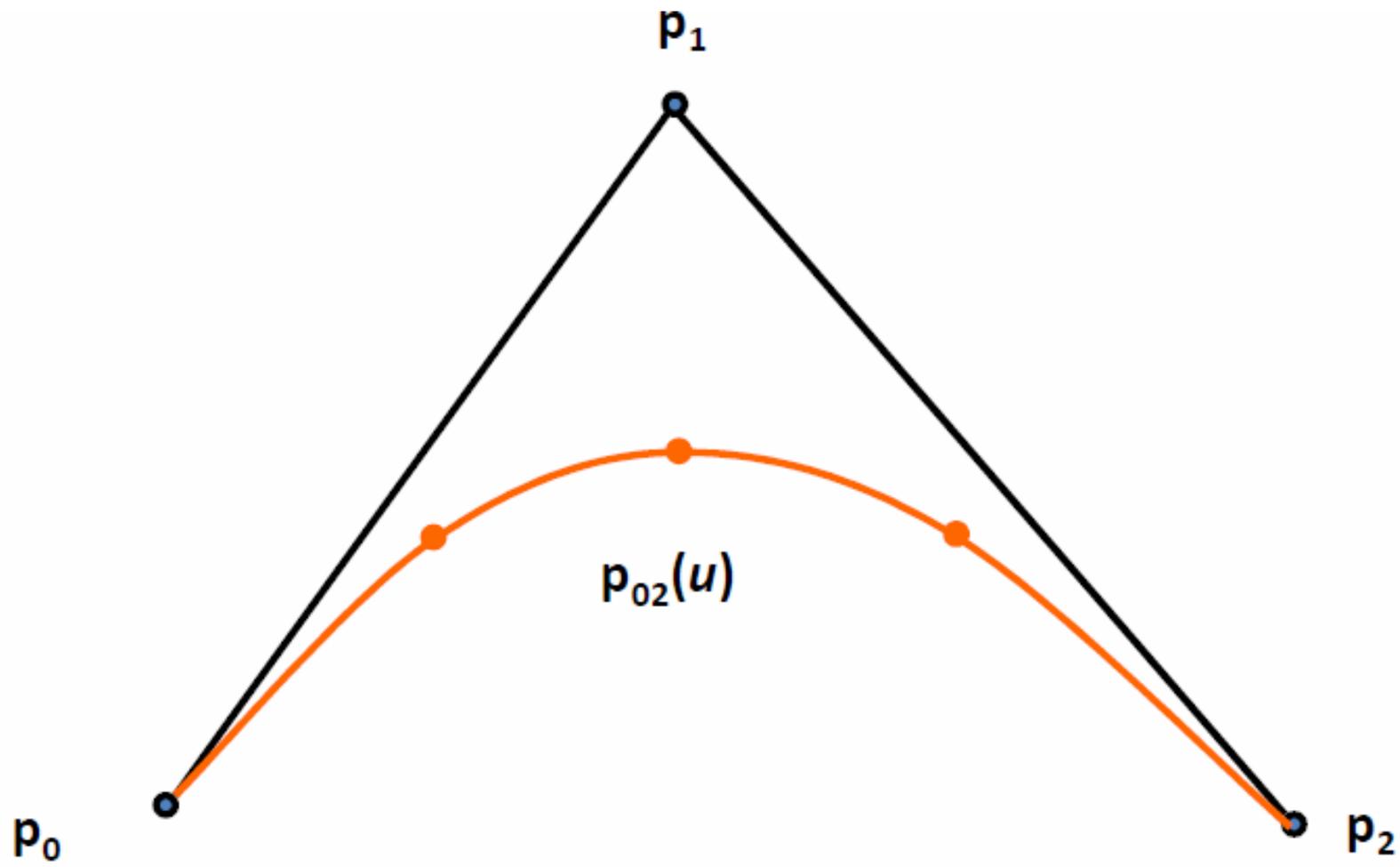


Algoritmo geométrico



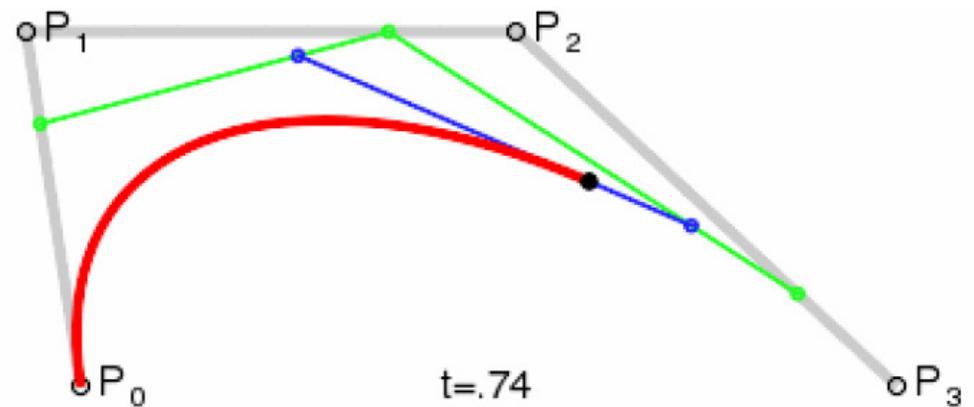
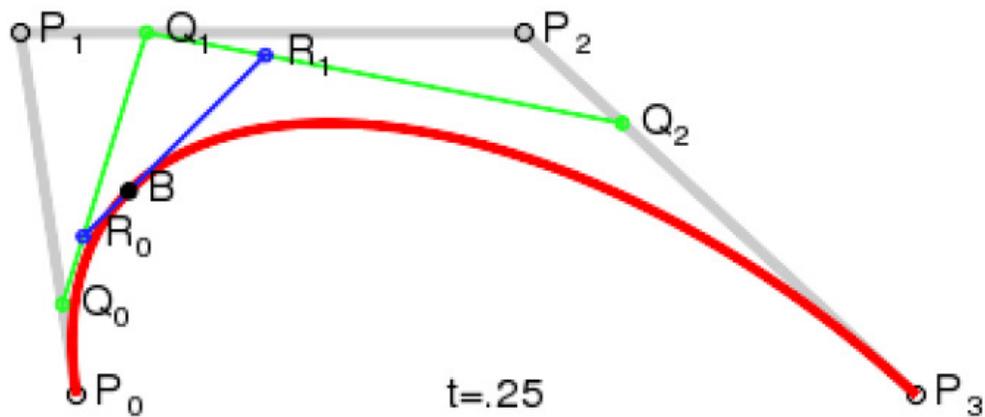






Outras formas de Bezier

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

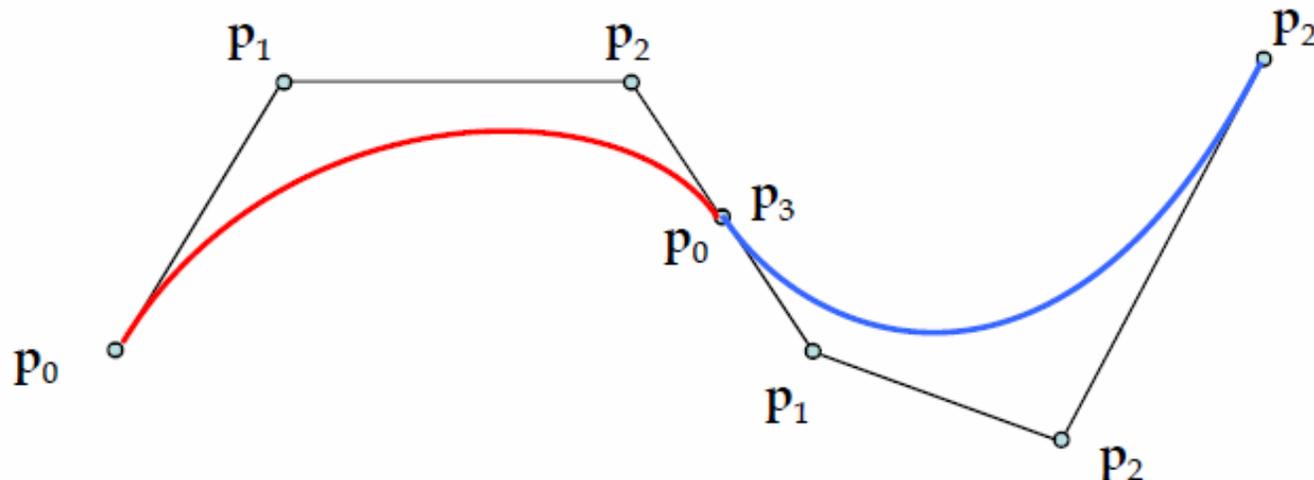


Cont.

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

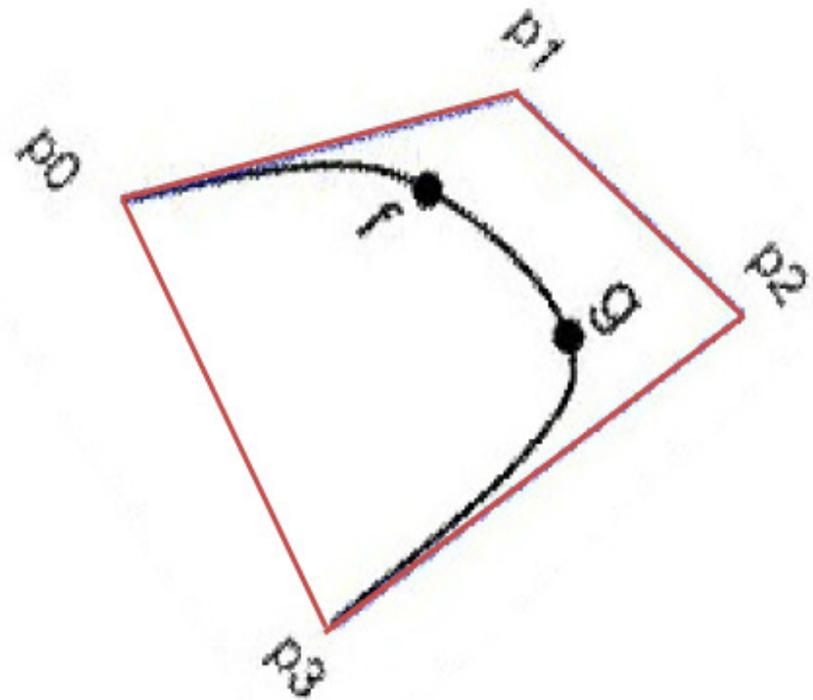
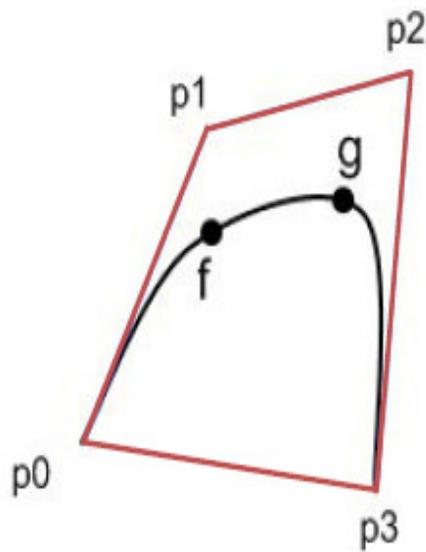
Emendando Curvas Bézier

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



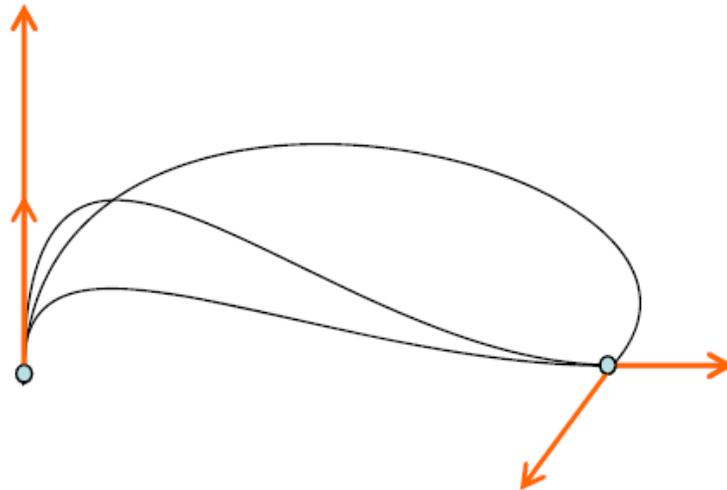
Transformações

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



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):



- 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

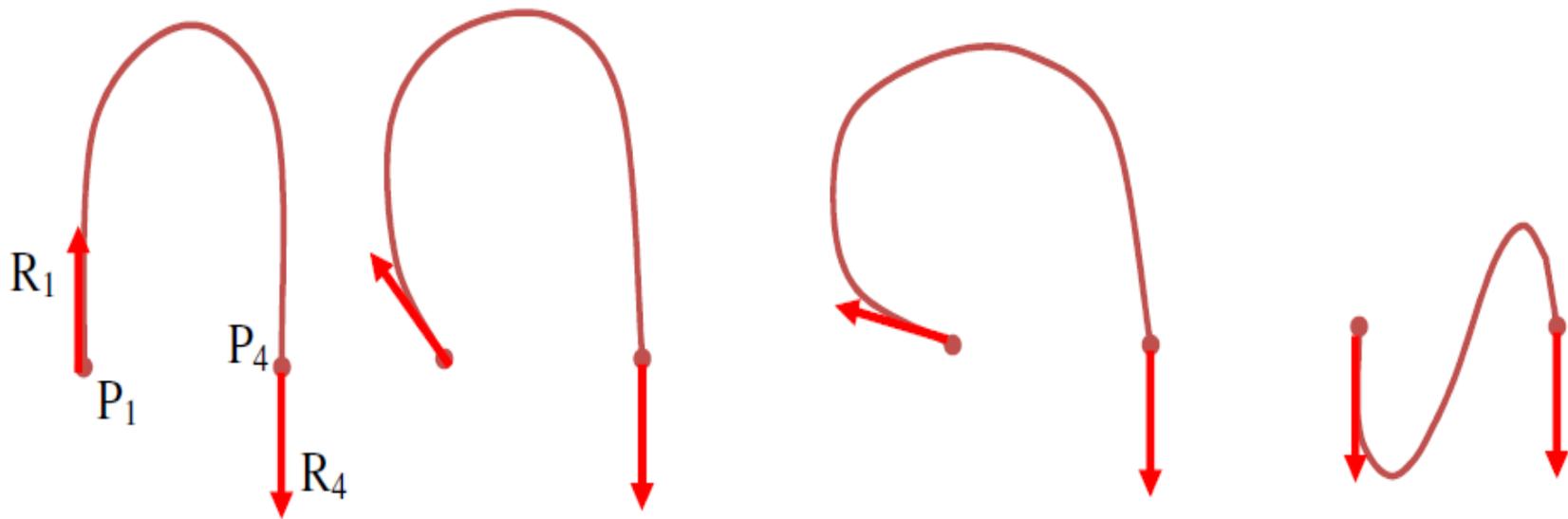
Curvas de Hermite

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

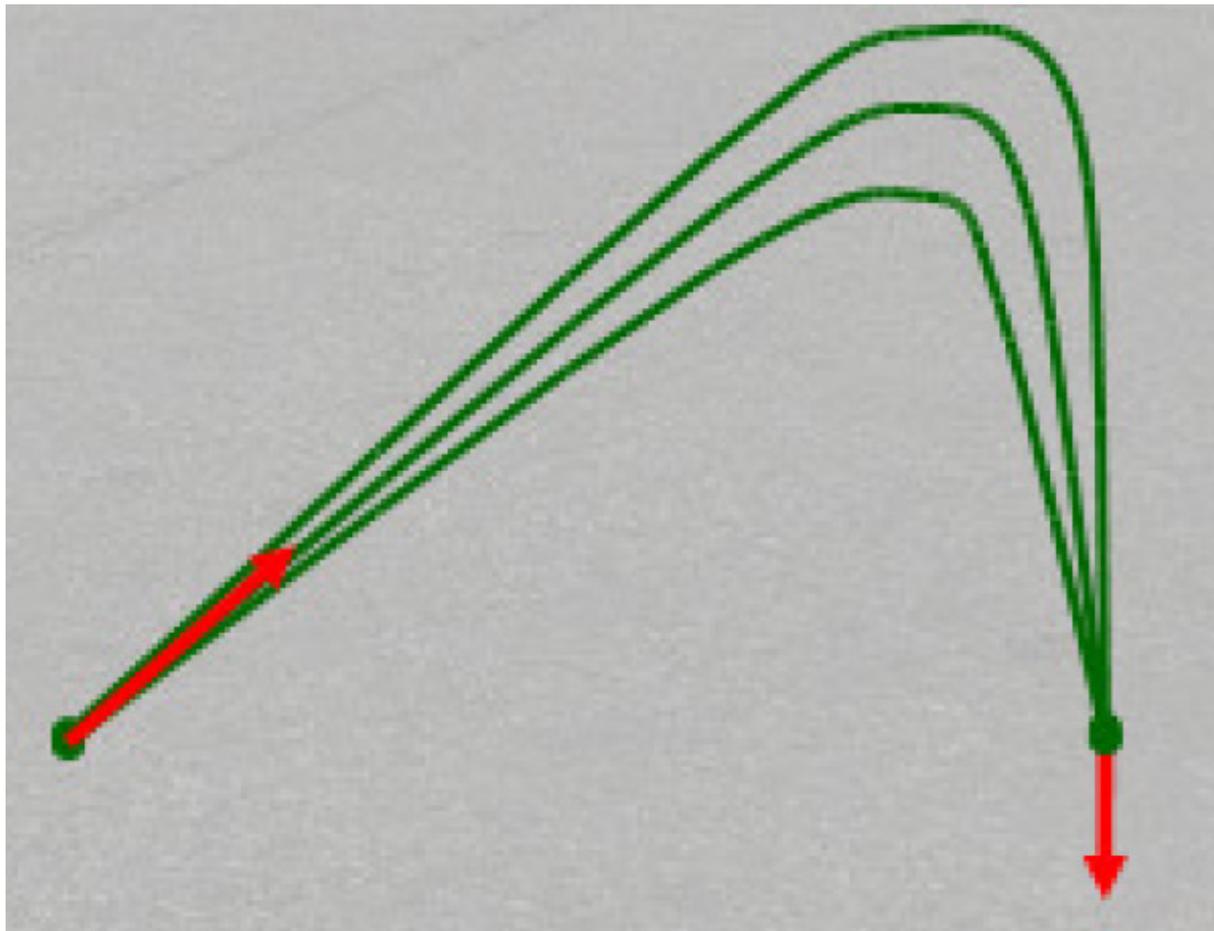
- Os pontos propriamente ditos: P_0 e P_1
- Vetores tangentes nestes pontos: V_0 e V_1



Mesmos pontos iniciais e finais,
apenas alterando a direção da
tangente



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)

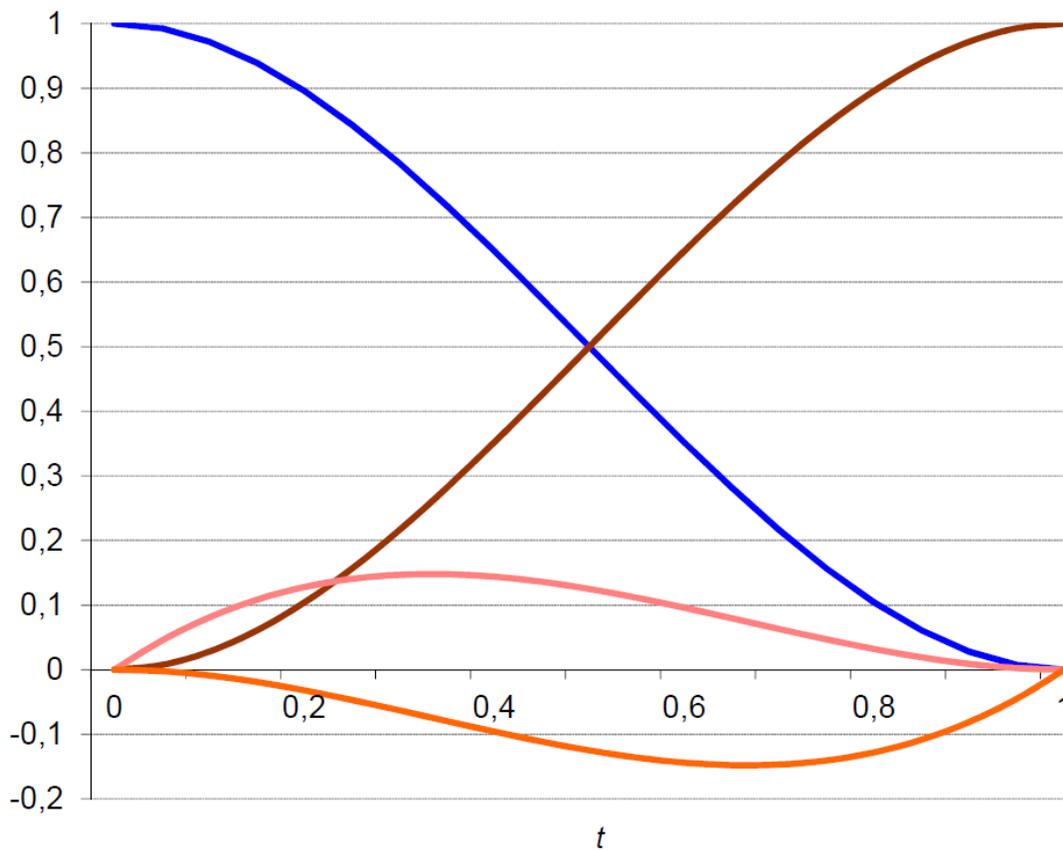
Desvantagens

- 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

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

Funções de mistura



Coeficientes Geométricos:
 $p(0)$, $p(1)$, $p'(0)$, $p'(1)$:

Funções de Mistura:

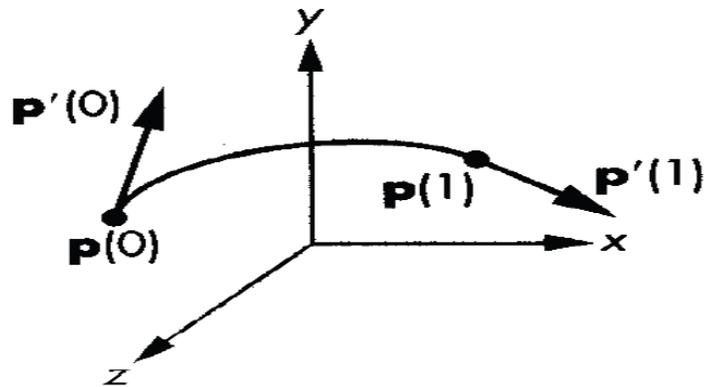
$$F1(u) = (2u^3 - 3u^2 + 1)$$

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

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

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

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

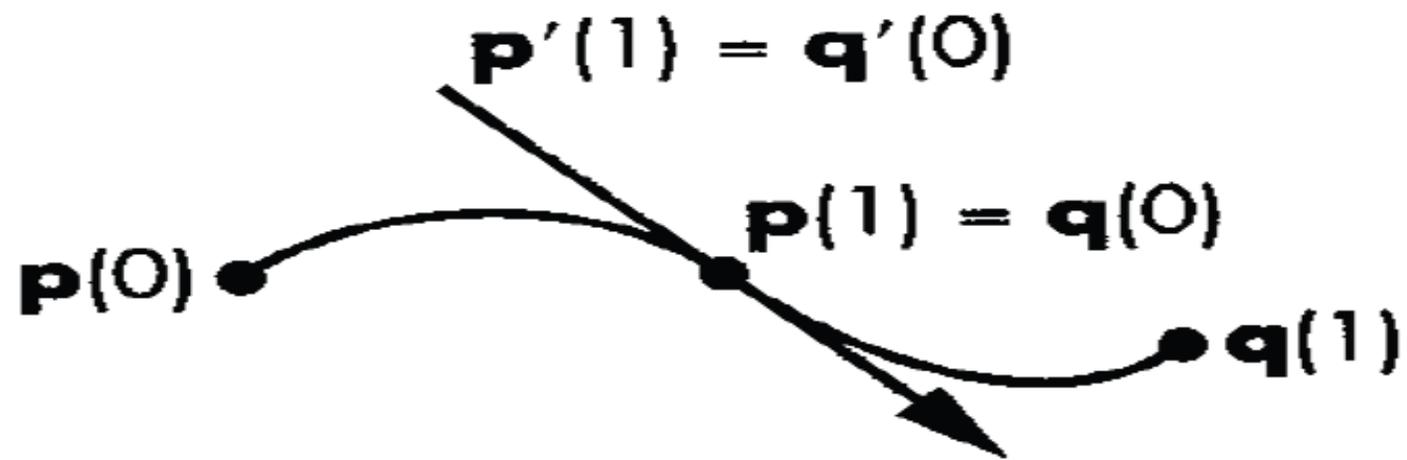


Definition of the
Hermite cubic.

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

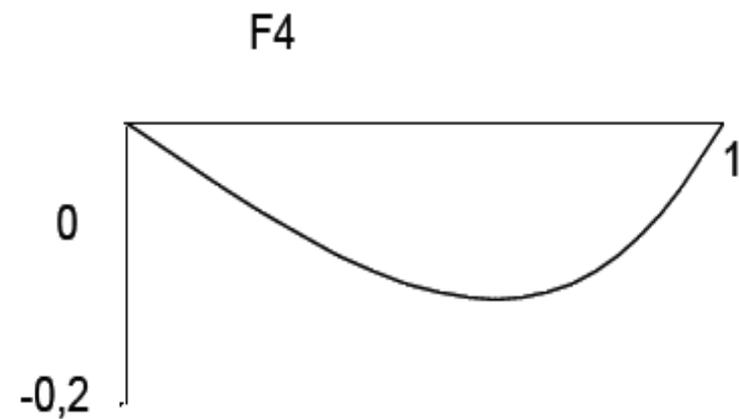
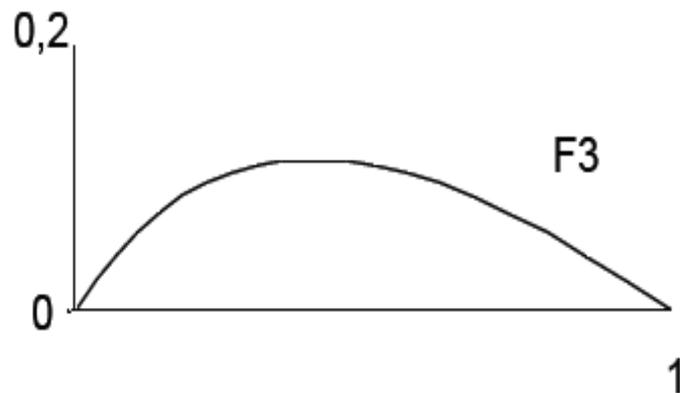
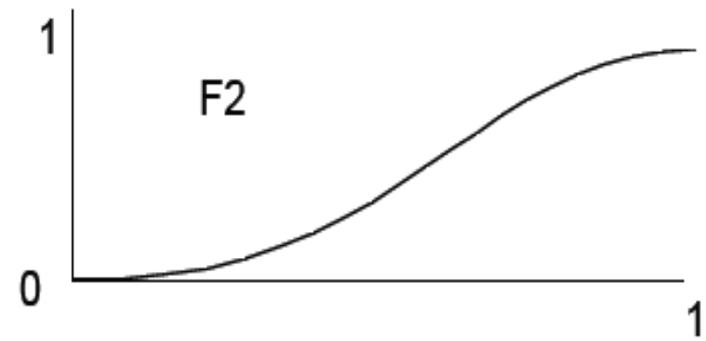
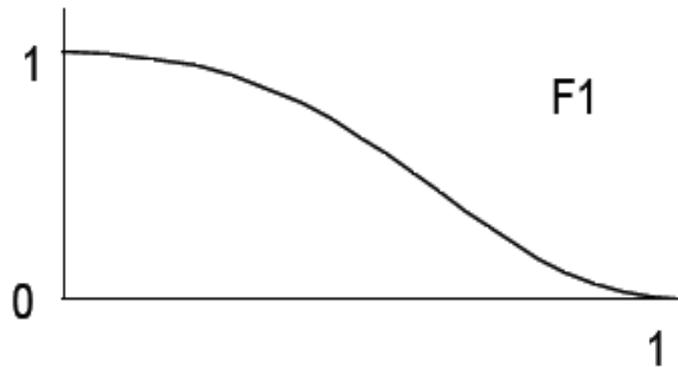
$$M_H = A^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

portanto $c = M p$



Hermite form at
join point.

Funções de mistura de Hermite



$$Q_H(t) = TM_H G_H$$

G_H é

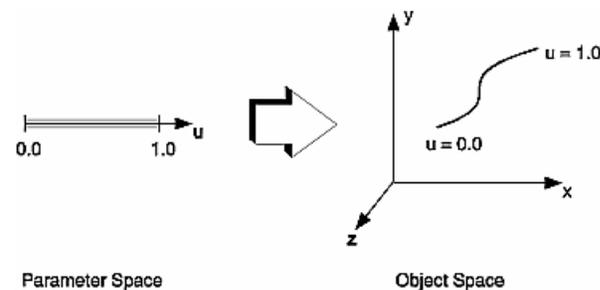
$$G_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

G_{Hx} é a componente x de G_H :

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

Curvas Splines

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

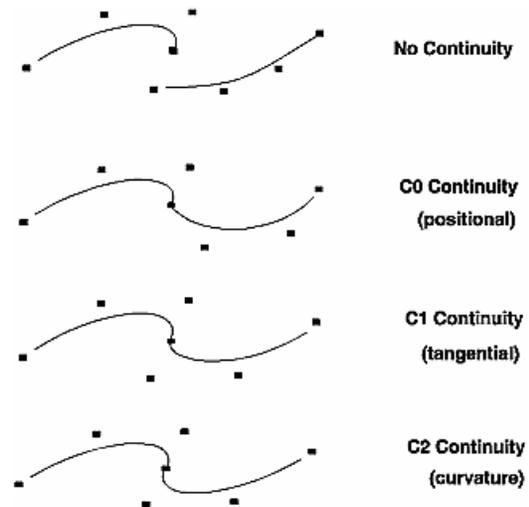
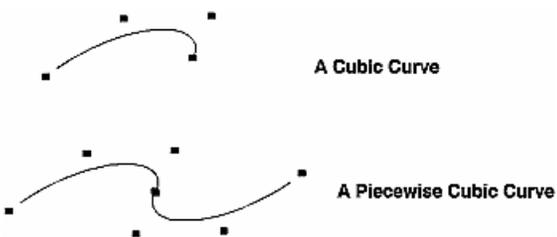


Splines

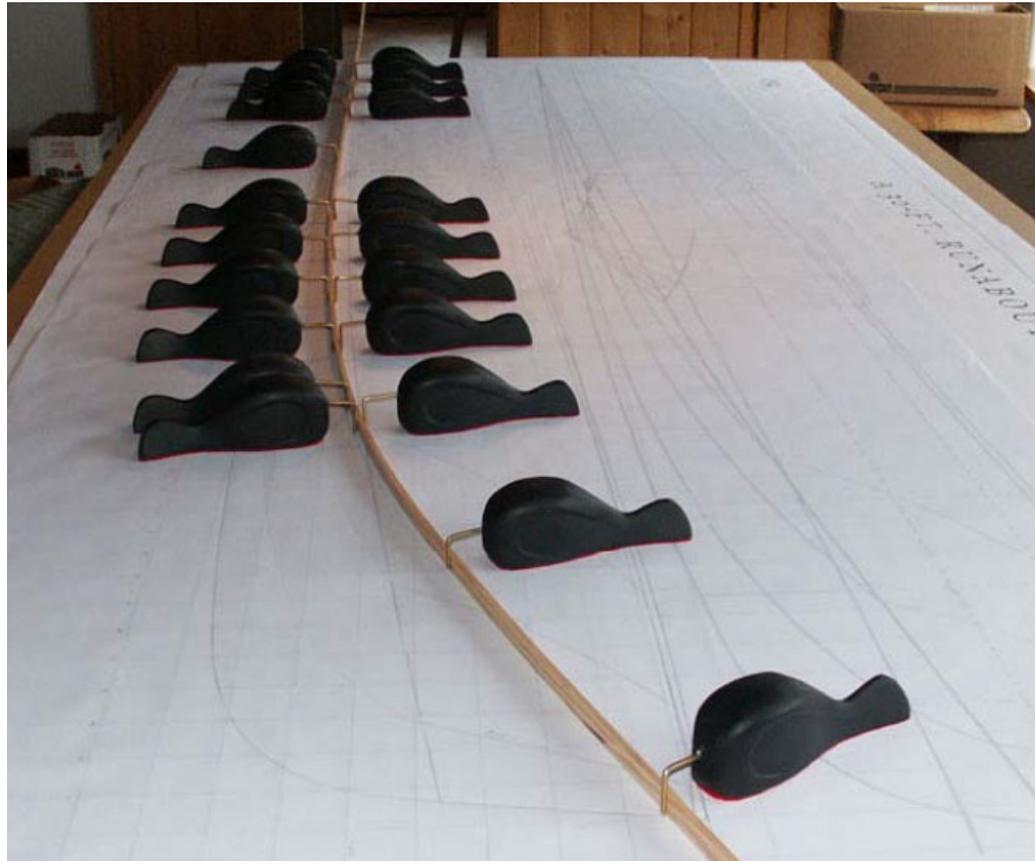
- A base de Bézier não é própria para a modelagem de curvas longas
 - Bézier única: suporte não local
 - Trechos emendados: restrições não são naturais
- Base alternativa: B-Splines
 - Nome vem de um instrumento usado por desenhistas
 - Modelagem por polígonos de controle sem restrições adicionais
 - 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}

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*).



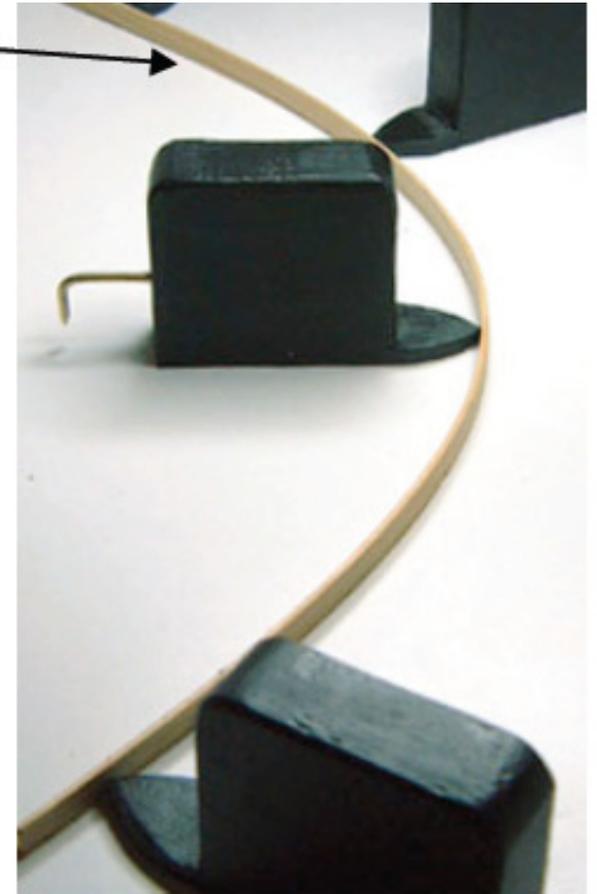
Spline física



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

Metal flexível com continuidade
de curvatura: C^2

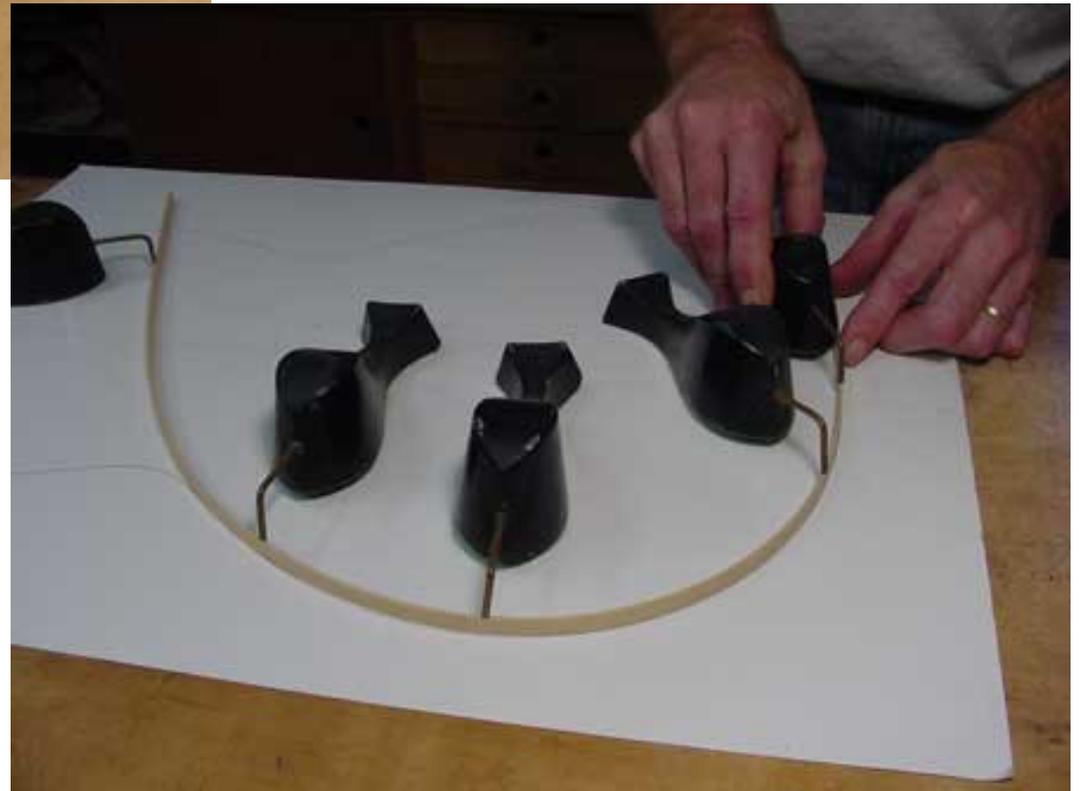
Construindo barcos



Pesos que dão forma = “ducks”



Exemplo de como são usadas



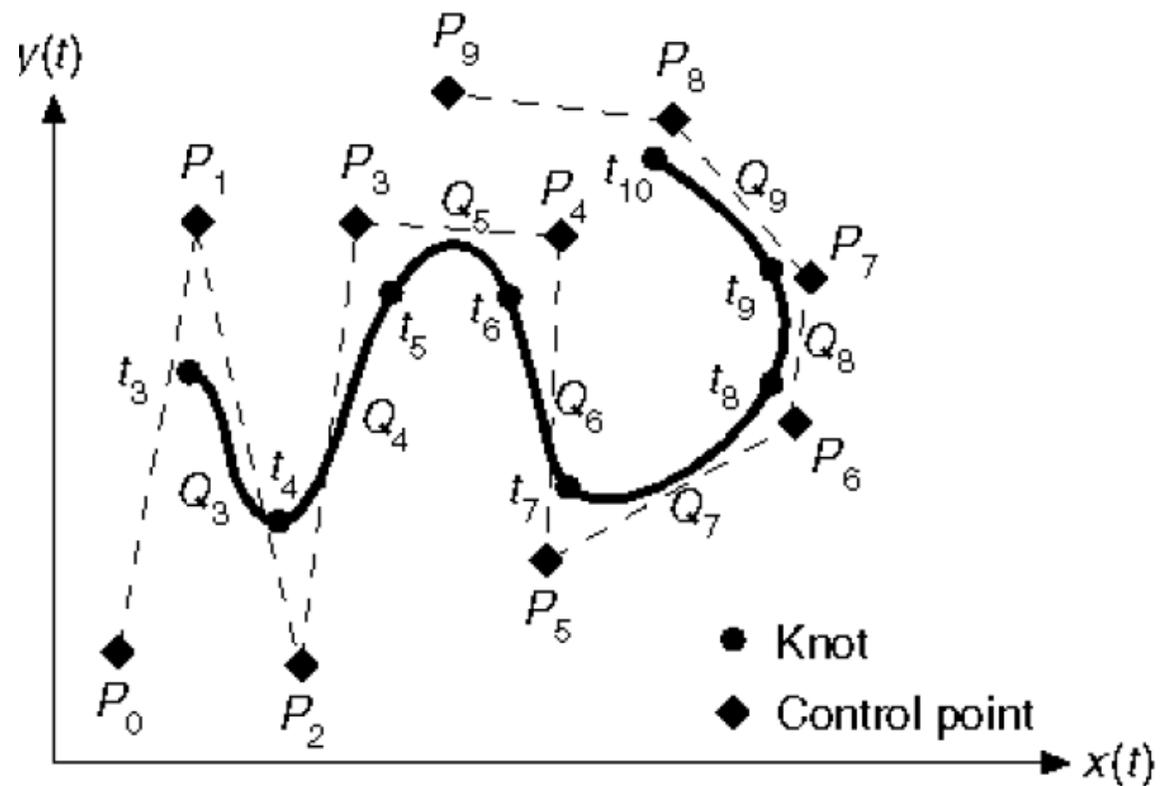
Especificando curvas

- 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

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

Nos + pontos de controle



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

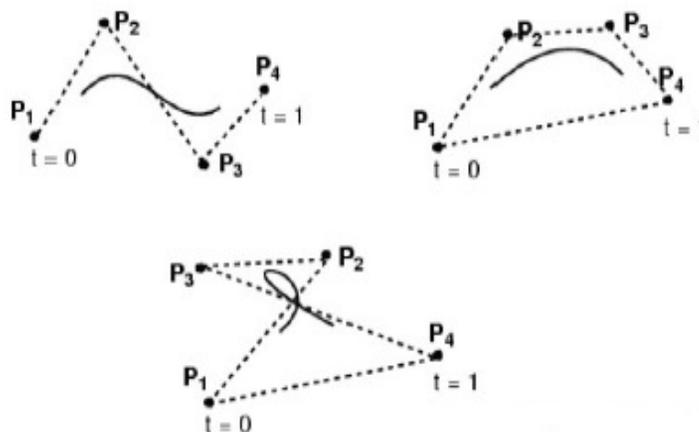
Não passa por nenhum ponto de controle.

– Curva de aproximação

Mais suave que as anteriores

– Mais fácil garantir continuidade paramétrica

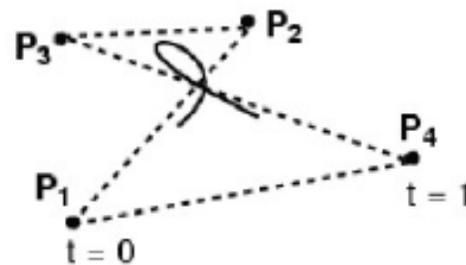
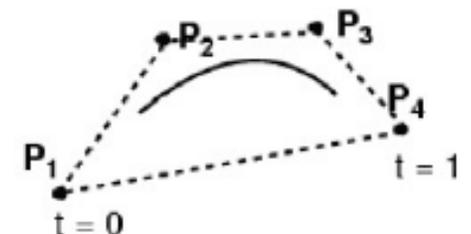
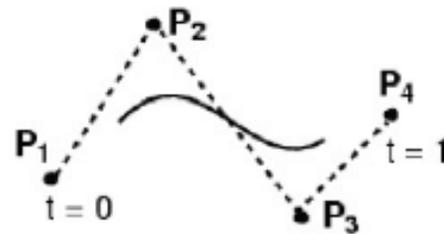
Controle local.



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:

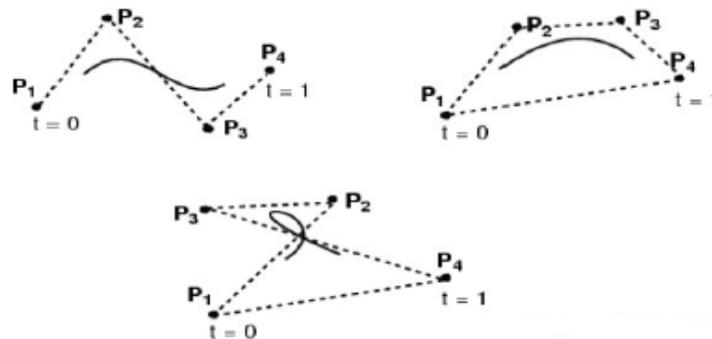
- Para $m+1$ pontos de controle
 - $M \geq 3$ P_0, P_1, \dots, P_n
- Teremos curvas com $m-2$ segmentos
 - Q_3, Q_4, \dots, Q_m

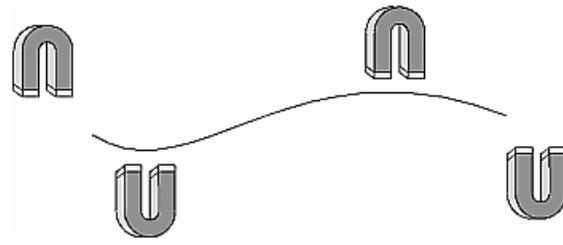


A curva inteira B-spline é considerada composta por segmentos de curvas spline

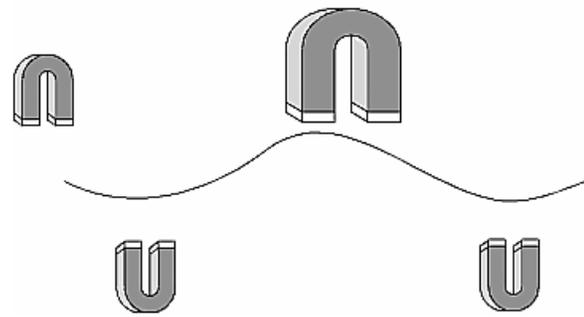
Cada segmento é definido por 4 pontos, cada ponto influencia 4 segmentos de curva (exceto P_0 e P_n)

Knots são os pontos de junção entre os segmentos de curva





w = 10.0



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)

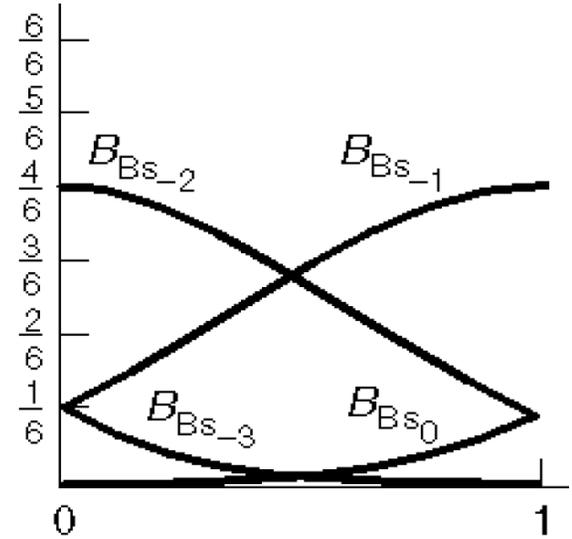
B-Spline uniforme: assume-se que esses nós têm valores inteiros e que o espaçamento entre nós é igual a 1 (0, 1, 2,...)

Representação matricial

1/6

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

Funções de mistura

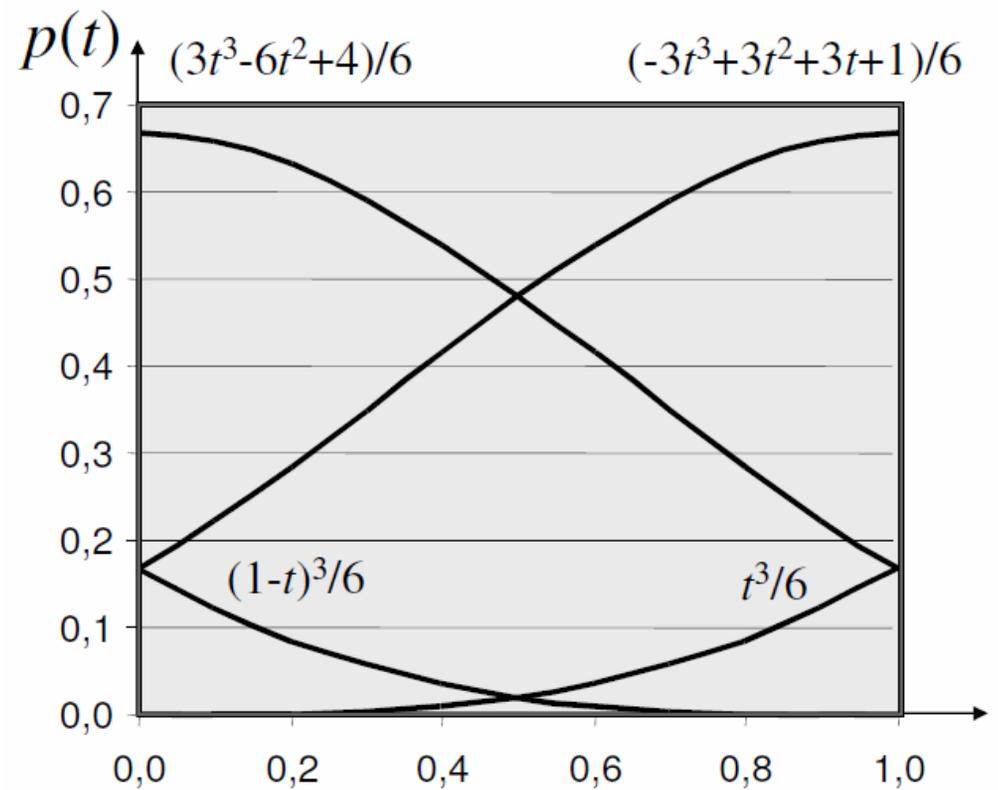


B-Splines Uniformes

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

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

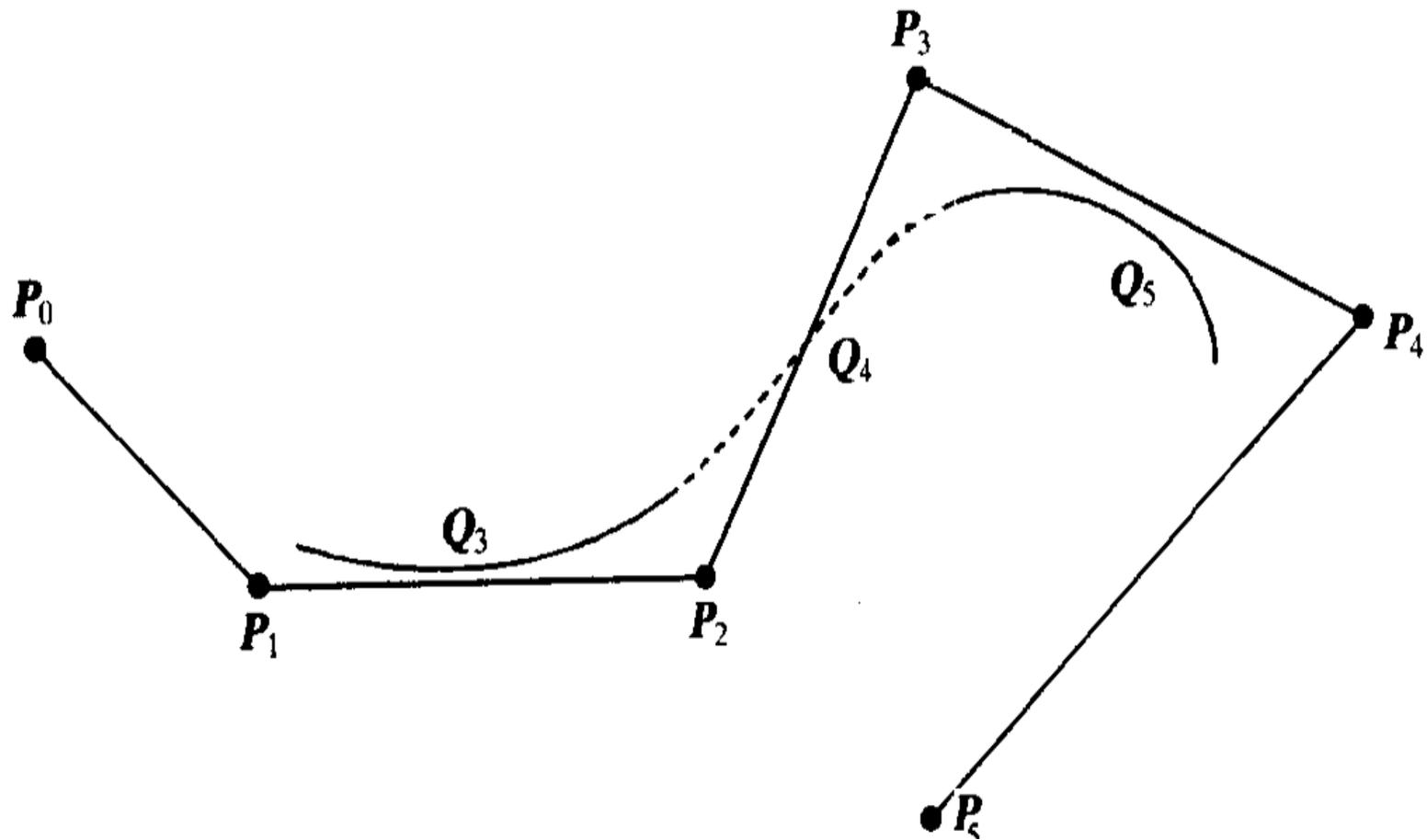
$$0 \leq t \leq 1$$



B-Splines Uniformes

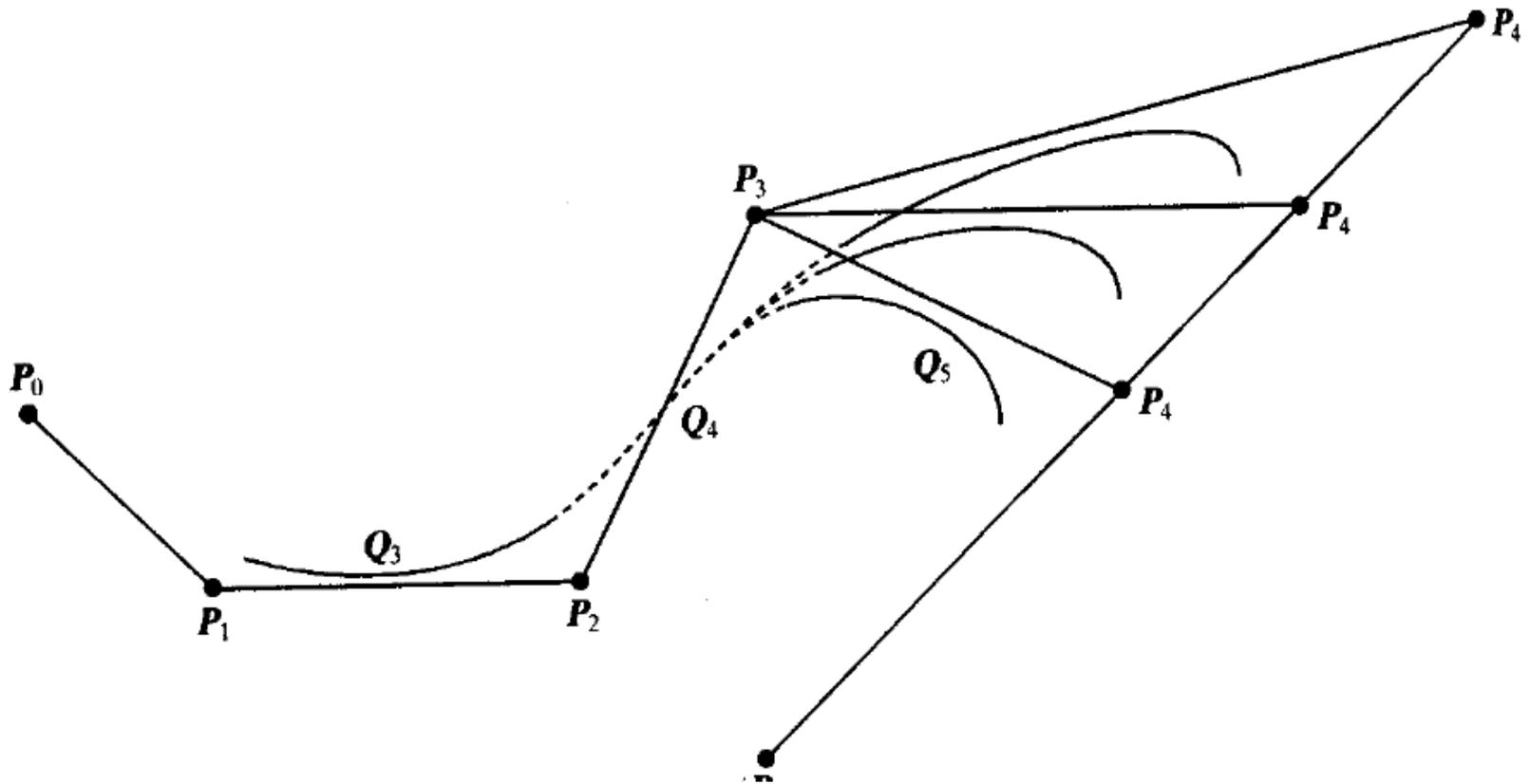
- Significa que a variável paramétrica está espaçada em intervalos uniformes ($t=0.1, 0.2, 0.3, \text{etc}$)
- Cada um dos $m-2$ segmentos é definido por 4 dos $m+1$ pontos de controle
- Segmento $Q_i = P_{i-3}, P_{i-2}, P_{i-1}, P_i$
- $GS = [P_{i-3} \ P_{i-2} \ P_{i-1} \ P_i] \ 3 \leq i \leq m$

Unido 3 curvas B-Splines



Exemplo de controle local:

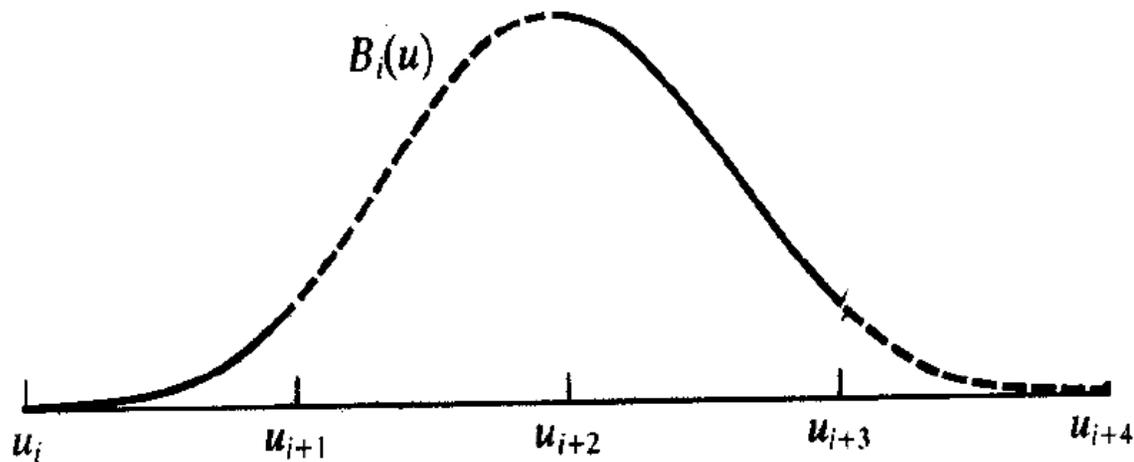
Alterando o penúltimo ponto, não se altera o trecho inicial e só parte do trecho intermediário



Ao ser controlada por 4 pontos, só se

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



Para criar uma curva spline fechada:

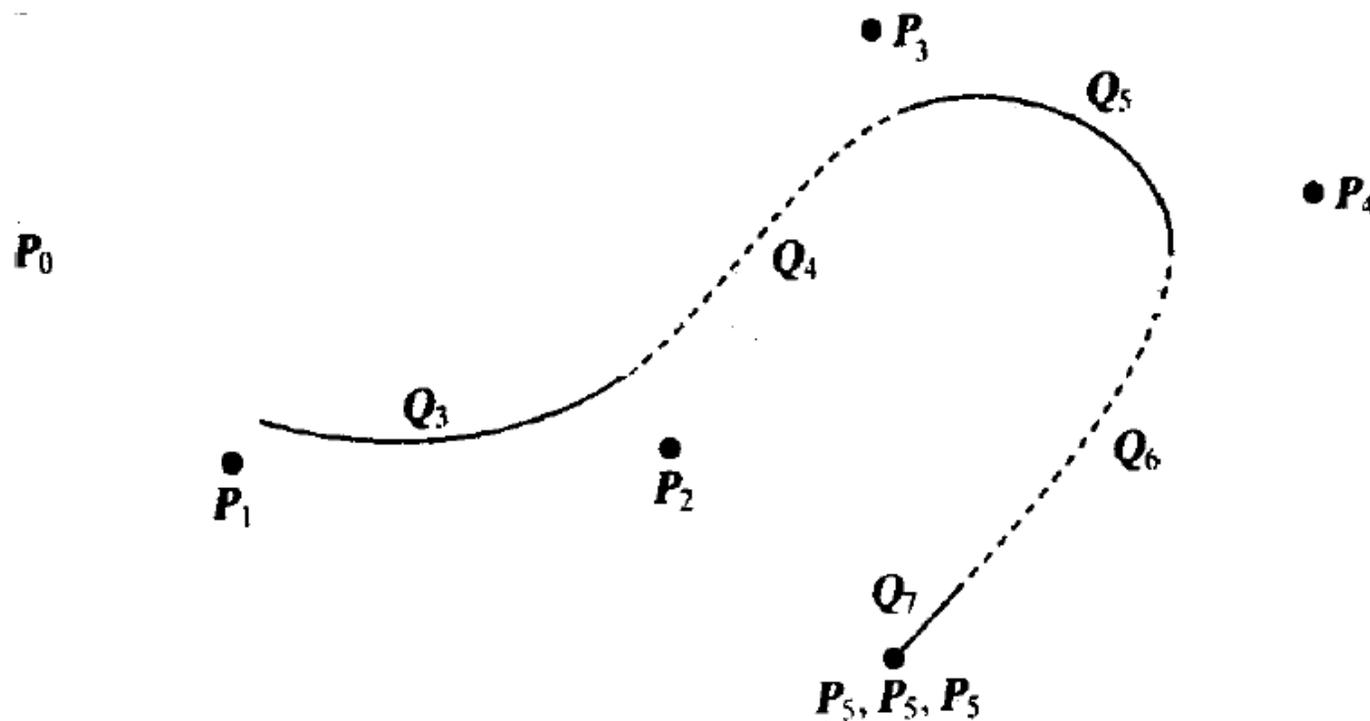
Apenas se repete no final da seqüência dos pontos de controle da curva os 3 pontos iniciais

$P_0, P_1, P_2, P_3 \dots \dots P_m, P_0, P_1, P_2$

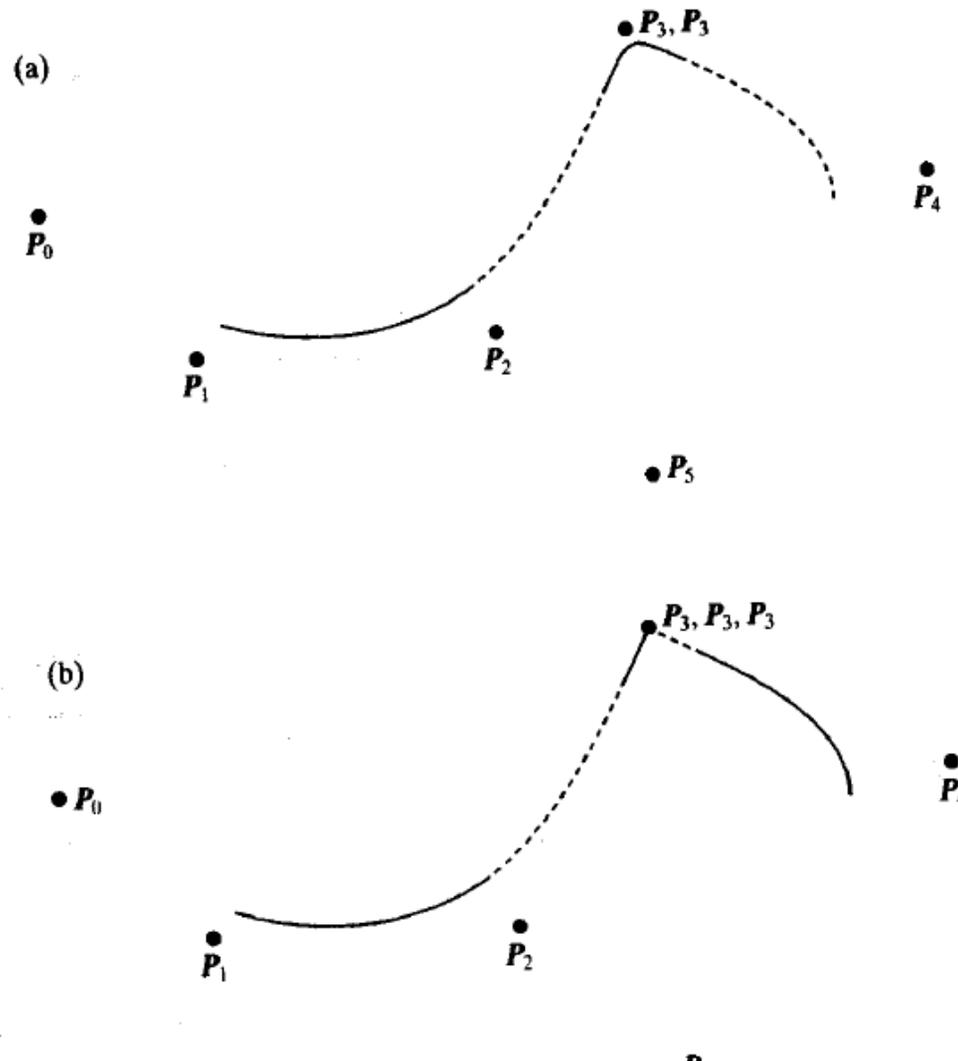
Spline com pontos controle coincidentes seguidos =>
Ela acaba por passar pelo ponto

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



Spline com pontos controle coincidentes seguidos =>
Ela acaba perde nivel de continuidade



Perda de continuidade

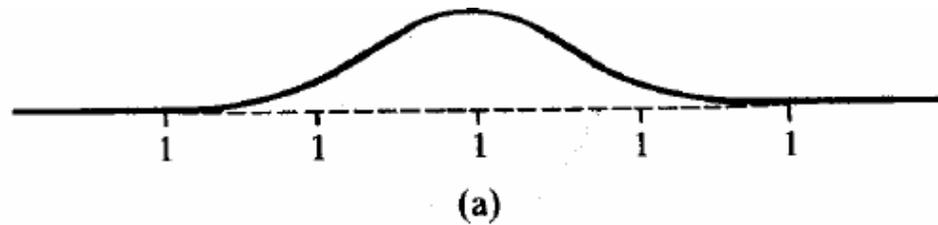
a) ponto duplo -G1

b) Ponto triplo - G0

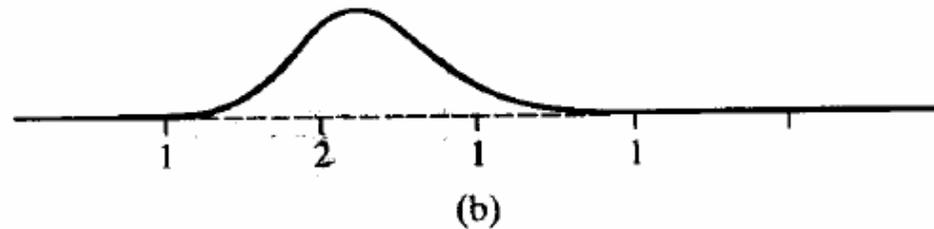


Spline : efeito das multiplicidades dos pontos de controle ou coincidências dos mesmos nas funções de base

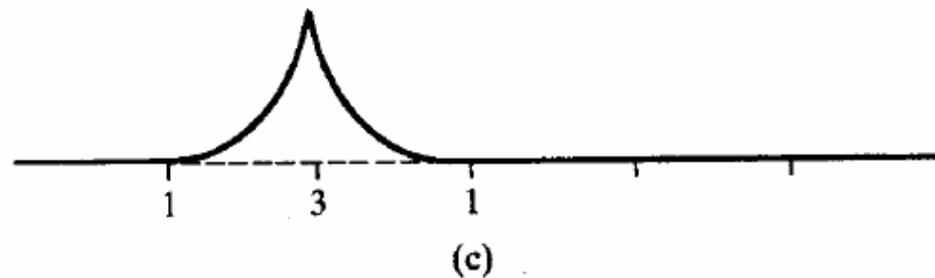
a) Multiplicidade 1:
[0, 1, 2, 3, 4]



b) Multiplicidade 2:
[0, 1, 1, 2, 3]



c) Multiplicidade 3:
[0, 1, 1, 1, 2]

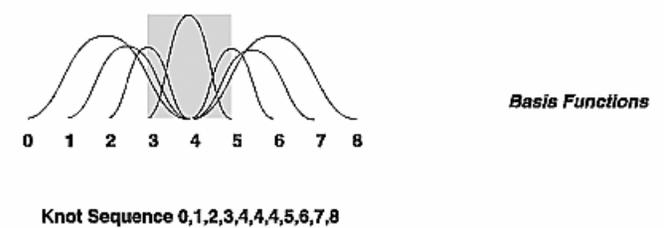
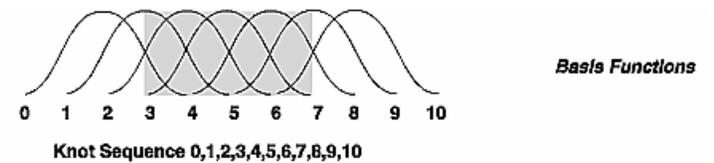
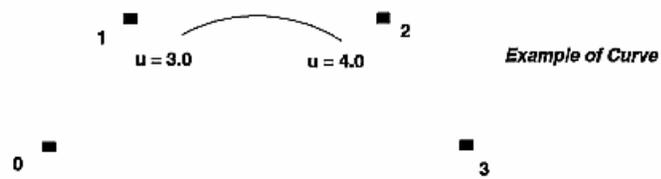
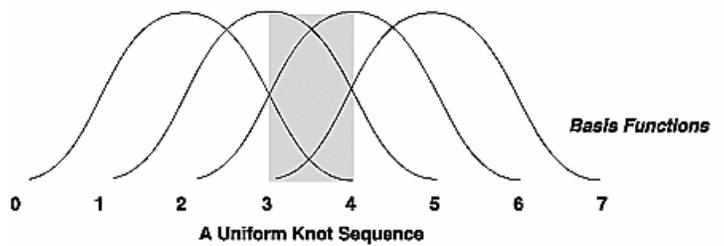
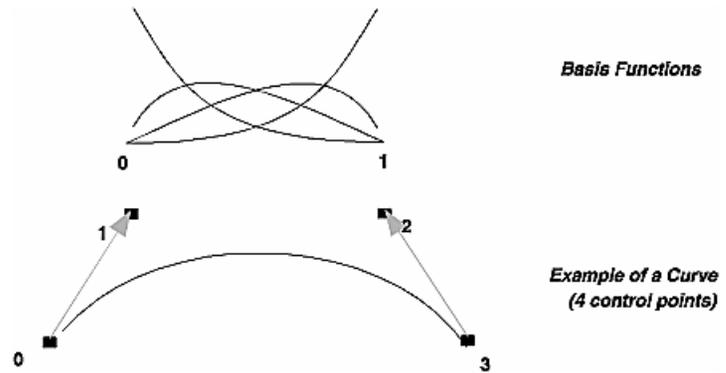
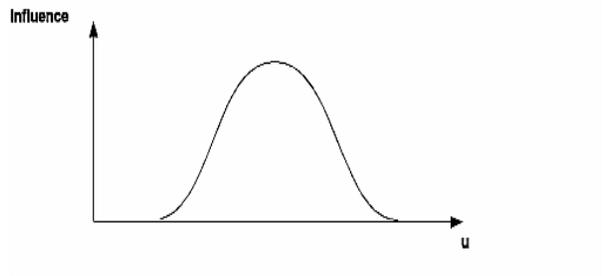


d) Multiplicidade 4:
[0, 1, 1, 1, 1]



Propriedades

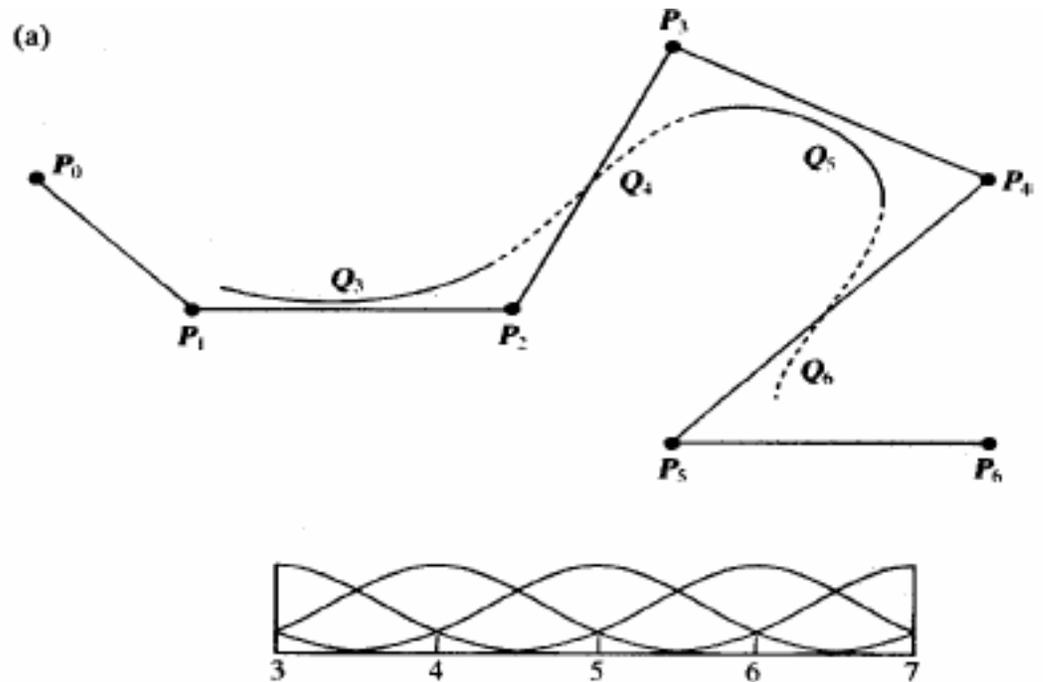
- Aumentar a multiplicidade m de um knot reduz a continuidade da paramétrica $k-m-1$;
- Um knot interior de multiplicidade k transforma uma B-spline em duas B-Splines distintas cada um com o seu conjunto de pontos de controlo.



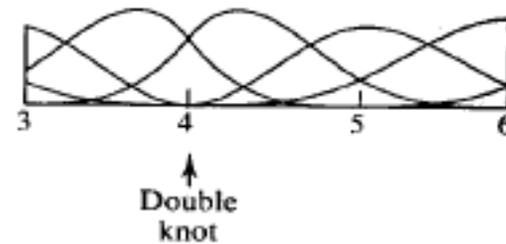
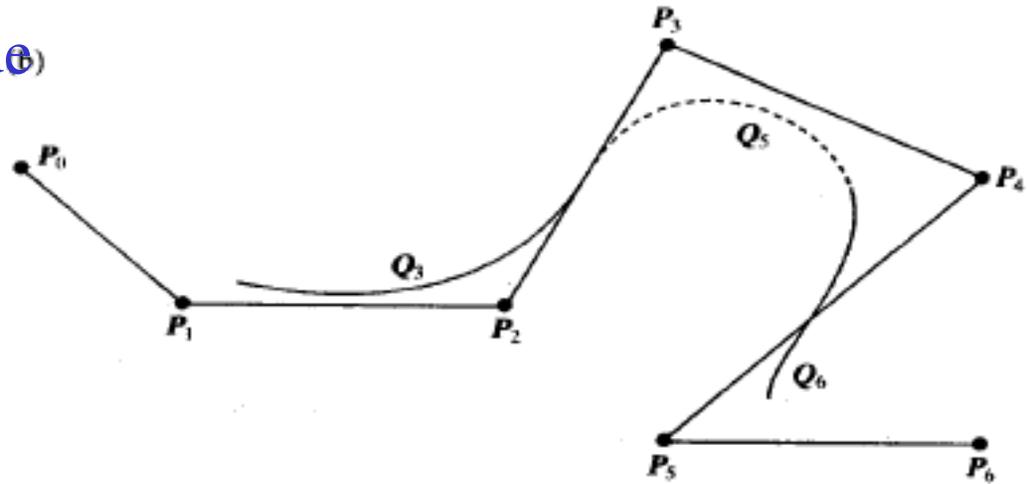
Spline => propriedades

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.

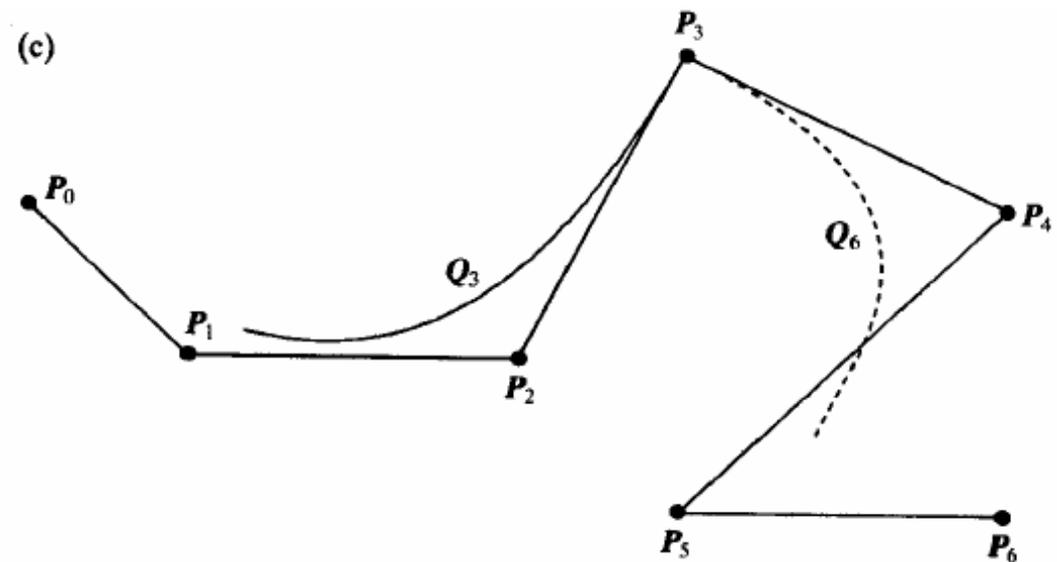


Spline com pontos controle coincidentes seguidos => perda nivel de continuidade

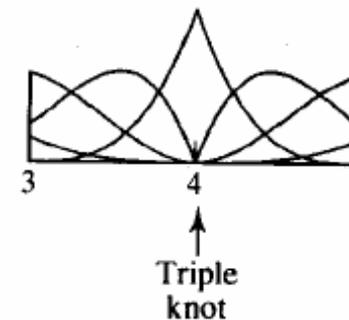


(b) Knot vector is
 $[0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 9]$.
 Q_4 shrinks to zero.

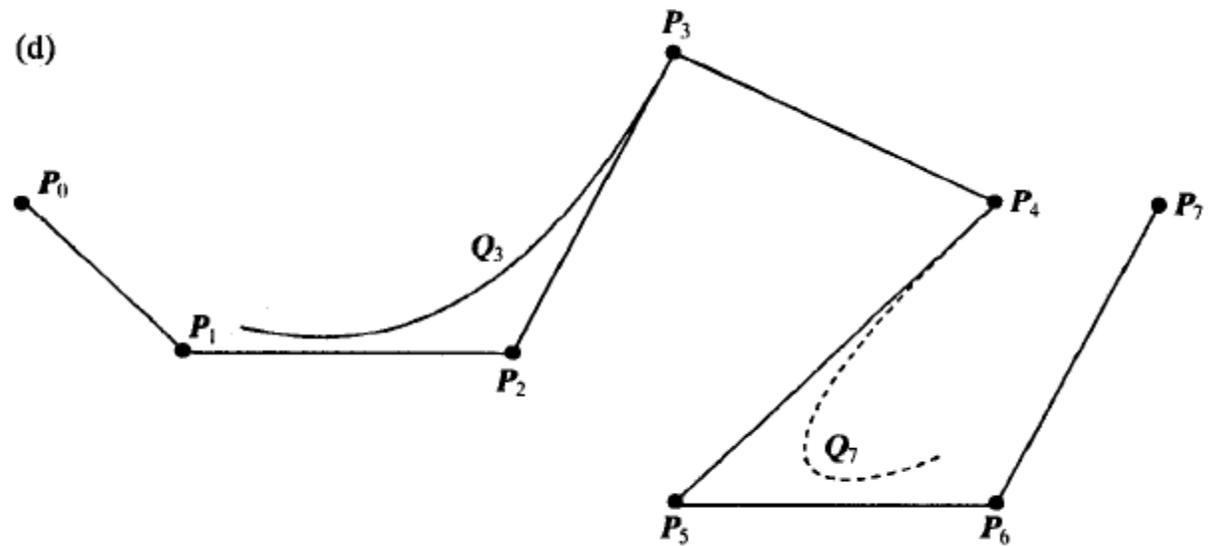
Spline com pontos controle coincidentes seguidos



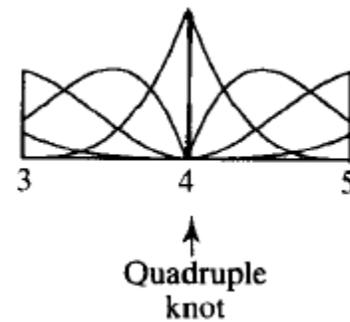
(c) Knot vector is $[0, 1, 2, 3, 4, 4, 4, 5, 6, 7, 8]$. Q_4 and Q_5 shrink to zero. Continuity between Q_3 and Q_6 is positional.

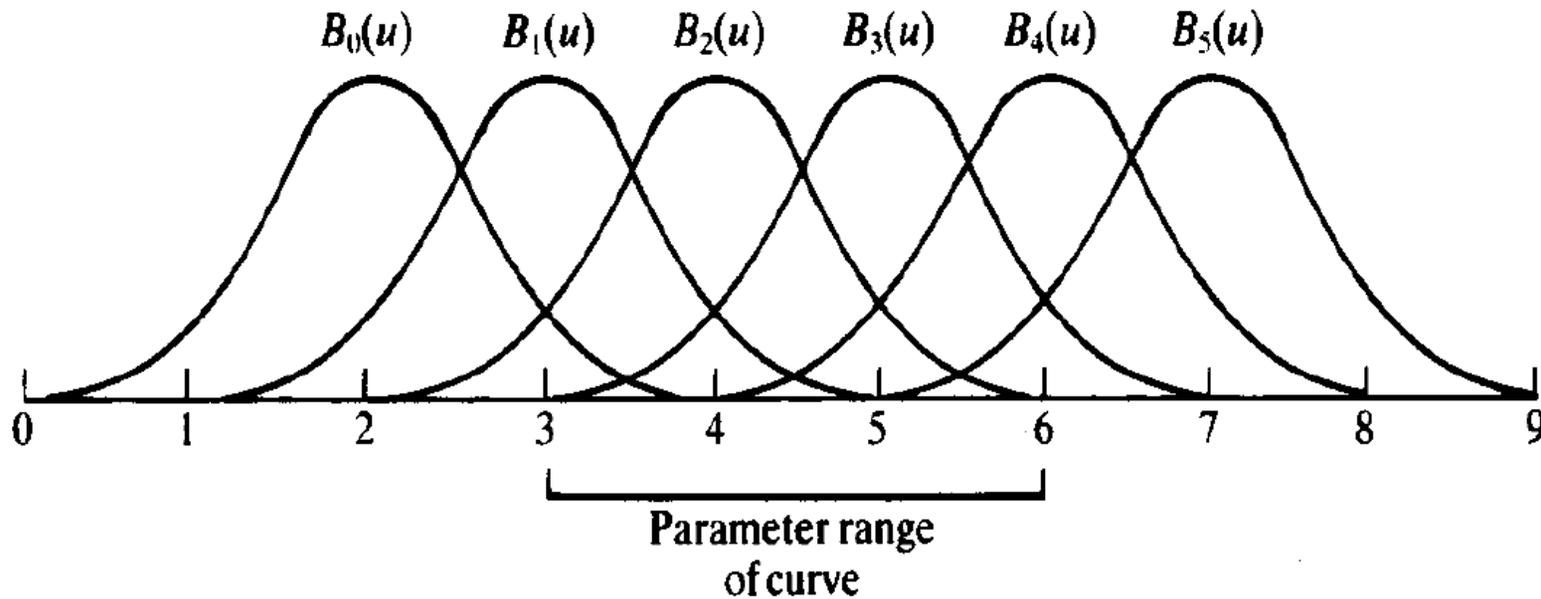


Spline com pontos controle coincidentes seguidos



(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 .





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:

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:

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

Número de nós: n° de pts de cntrl + ordem da curva

Spline controlada por 4 pontos

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

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

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

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

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

B-Splines Não-Uniformes

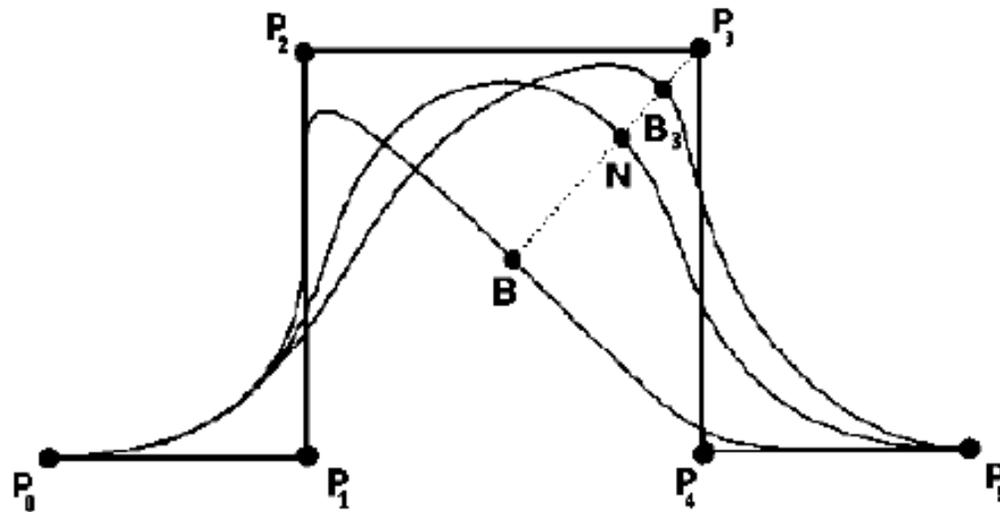
- O intervalo entre valores da variável paramétrica não é necessariamente uniforme...
- Logo as funções de *blending* não são as mesmas para cada segmento.... Maior flexibilidade

Funções de mistura



NURBS

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



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
- 

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

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:

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

Trabalho 12:

Implente em qualquer linguagem a geração de segmentos de curvas que sejam controladas por 4 pontos dados (uma spline 2D) . Isso é implemente a equação:

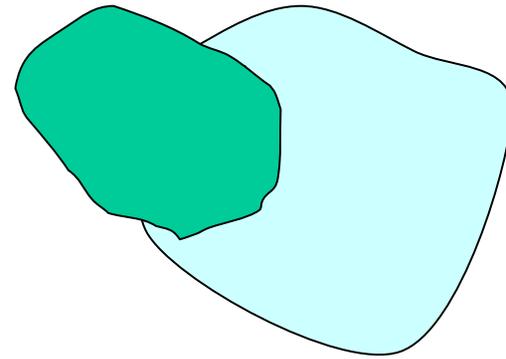
$$Q_{BS}(t) = \frac{(1-t)^3}{6} P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6} P_{i-2} +$$
$$\frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{i-1} + \frac{t^3}{6} P_i$$
$$0 \leq t \leq 1$$

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++;
}
```

Outras Splines



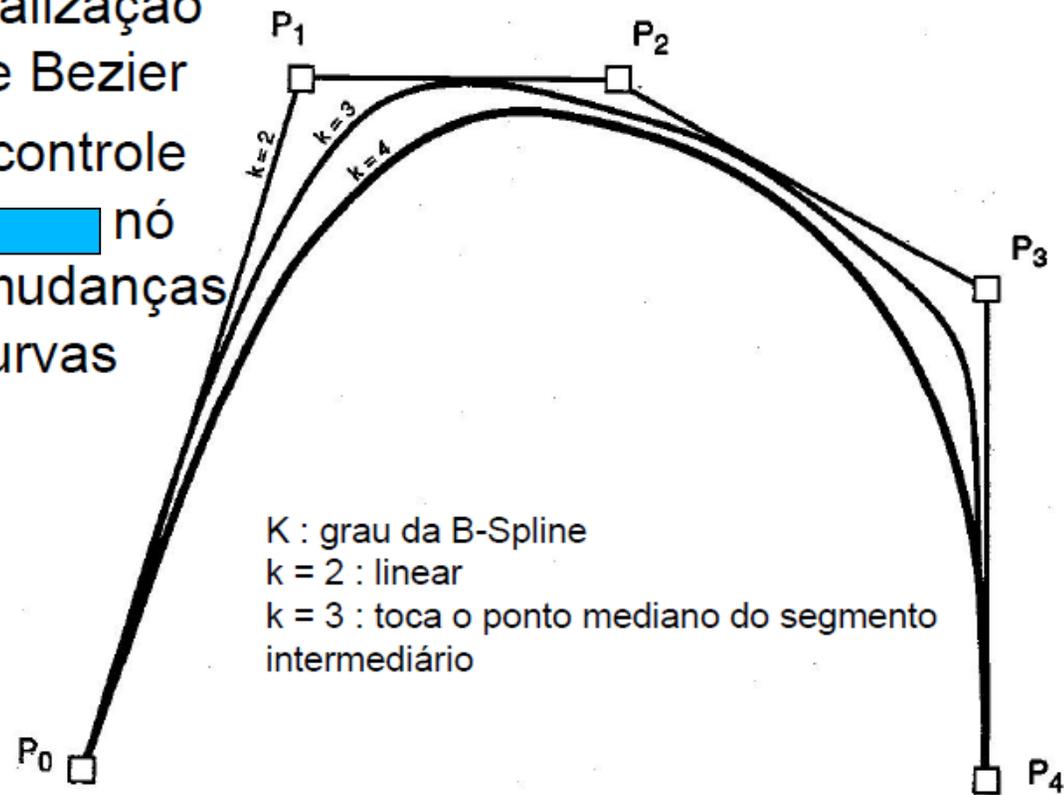
Splines de Catmull-Rom (ou Overhauser) interpolam pontos de controle, de modo similar às splines naturais

As β -splines controlam a forma da curva globalmente através das variáveis β_1 (bias) e β_2 (tensão)

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

B-Splines

- Uma generalização da curva de Bezier
- Pontos de controle adicionais nó permitem mudanças locais às curvas



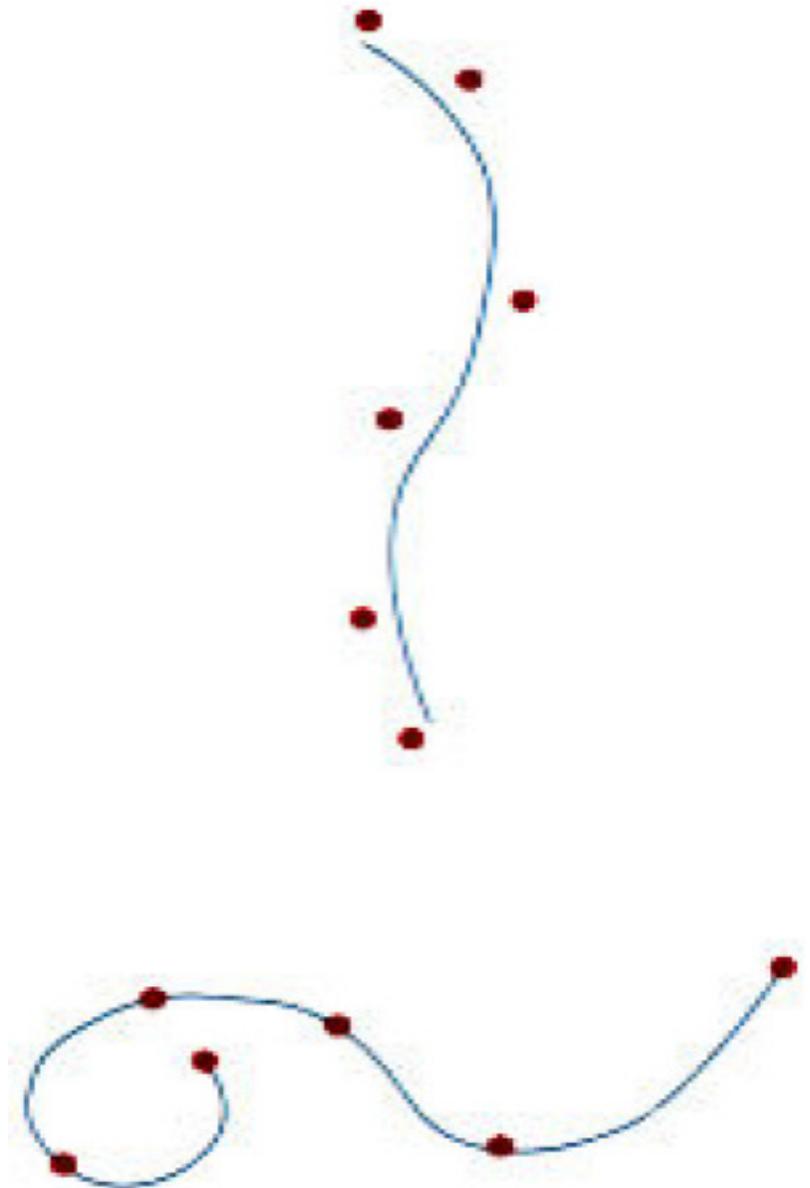
Interpolação por Splines Cubicas

Veja seção 11.5

De

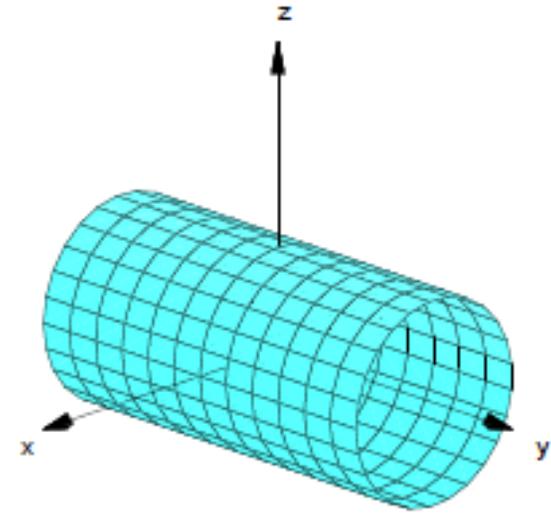
*Algebra Linear
com Aplicações*

A. Anton e C. Rorres,
Bookman, 2001



Superfícies

Formas de geração:

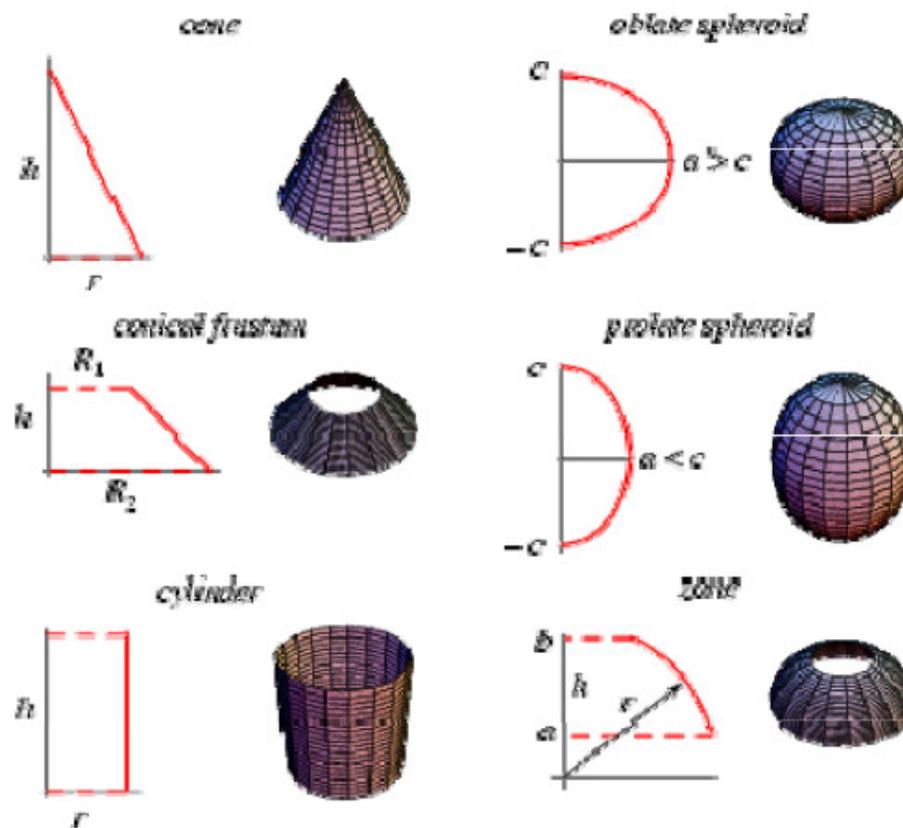


Superfície por caminho

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

Revolução

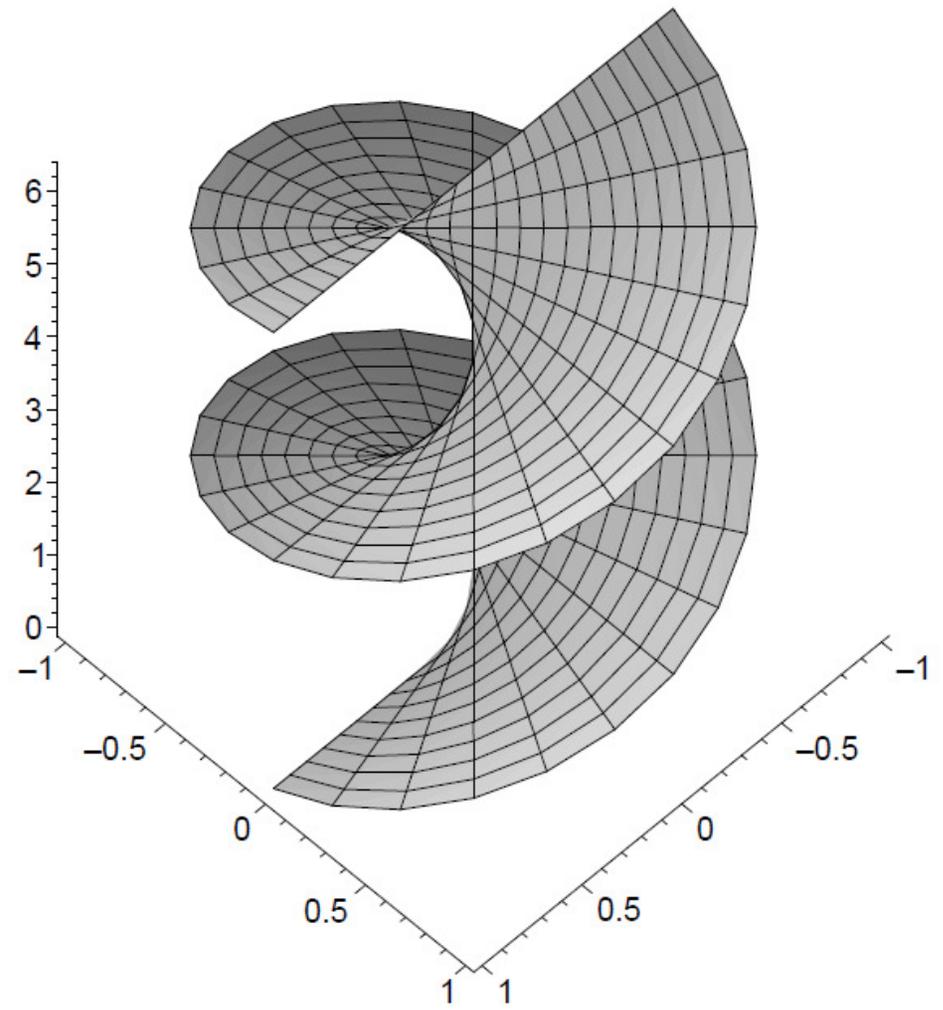
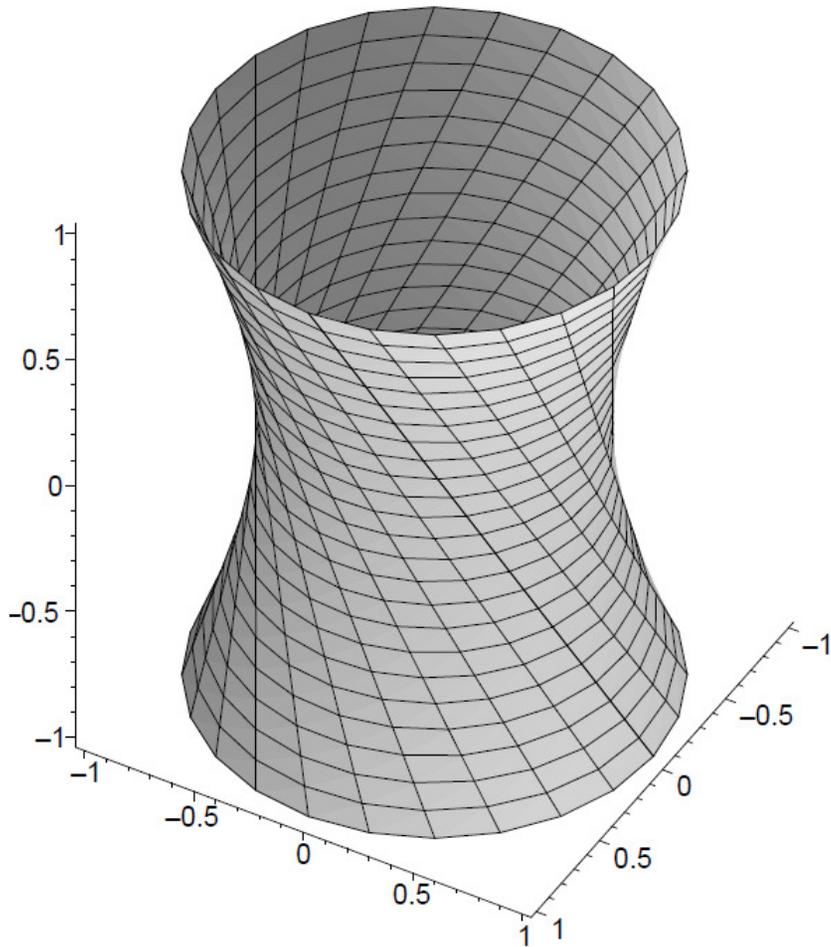
- Superfície criada pela rotação de uma curva sobre um plano em torno de uma linha reta (o eixo de rotação) que está sobre o mesmo plano.





Superfícies Regradas

Superfícies regradas são reuniões de retas.



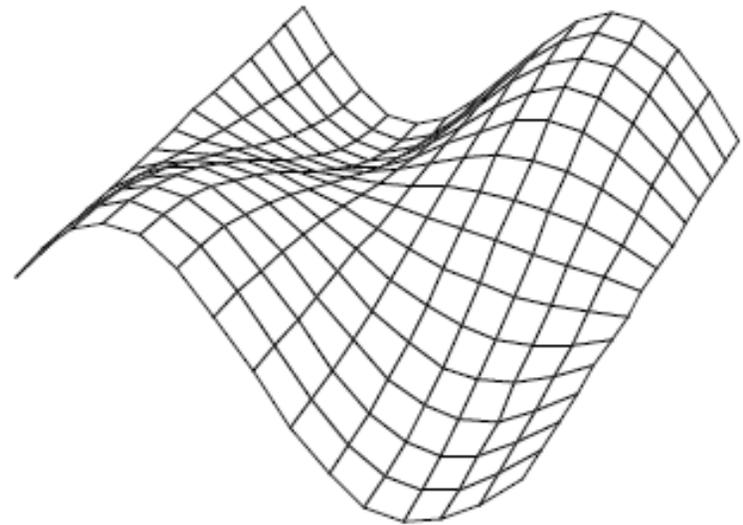
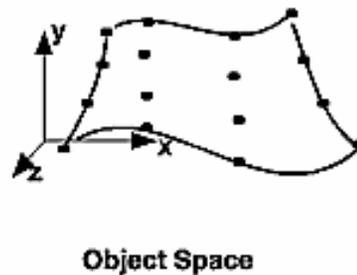
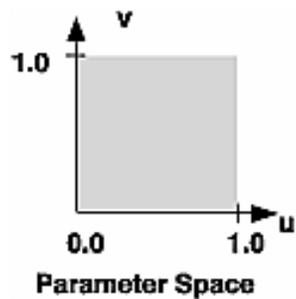
Por equações tri-dimensionais :

Superfícies Paramétricas

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

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

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

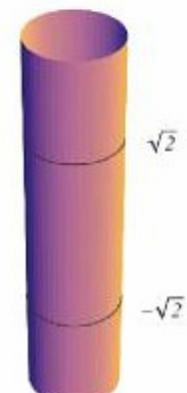
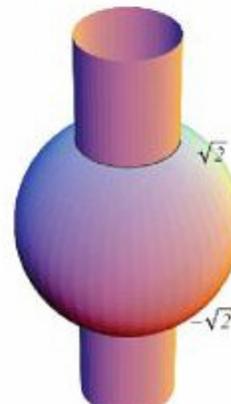


Superfícies Por equações tri-dimensionais :

representações não paramétricas



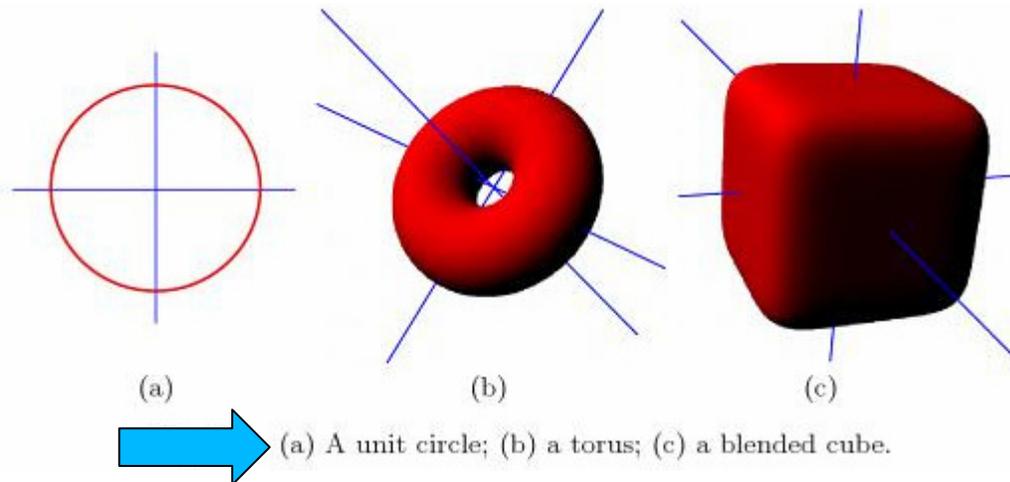
 (a) Whitney umbrella with-handle $x^2 - zy^2 = 0$; (b) Whitney umbrella without-handle $\{x^2 - zy^2 = 0\} - \{z < 0\}$.



exemplos

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

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

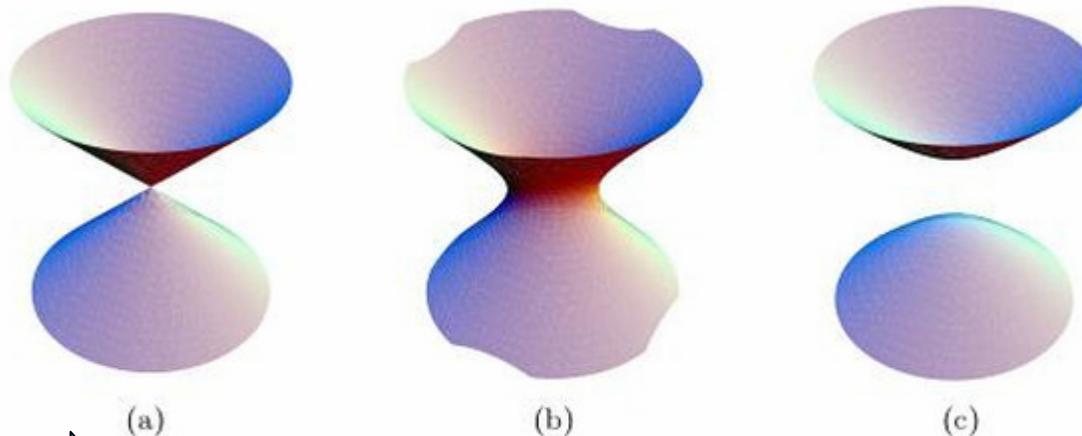


$$f(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} = \begin{pmatrix} \cos u (R + r \cos v) \\ \sin u (R + r \cos v) \\ r \sin v \end{pmatrix}$$

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

Representações paramétricas

Por equações tri-dimensionais :



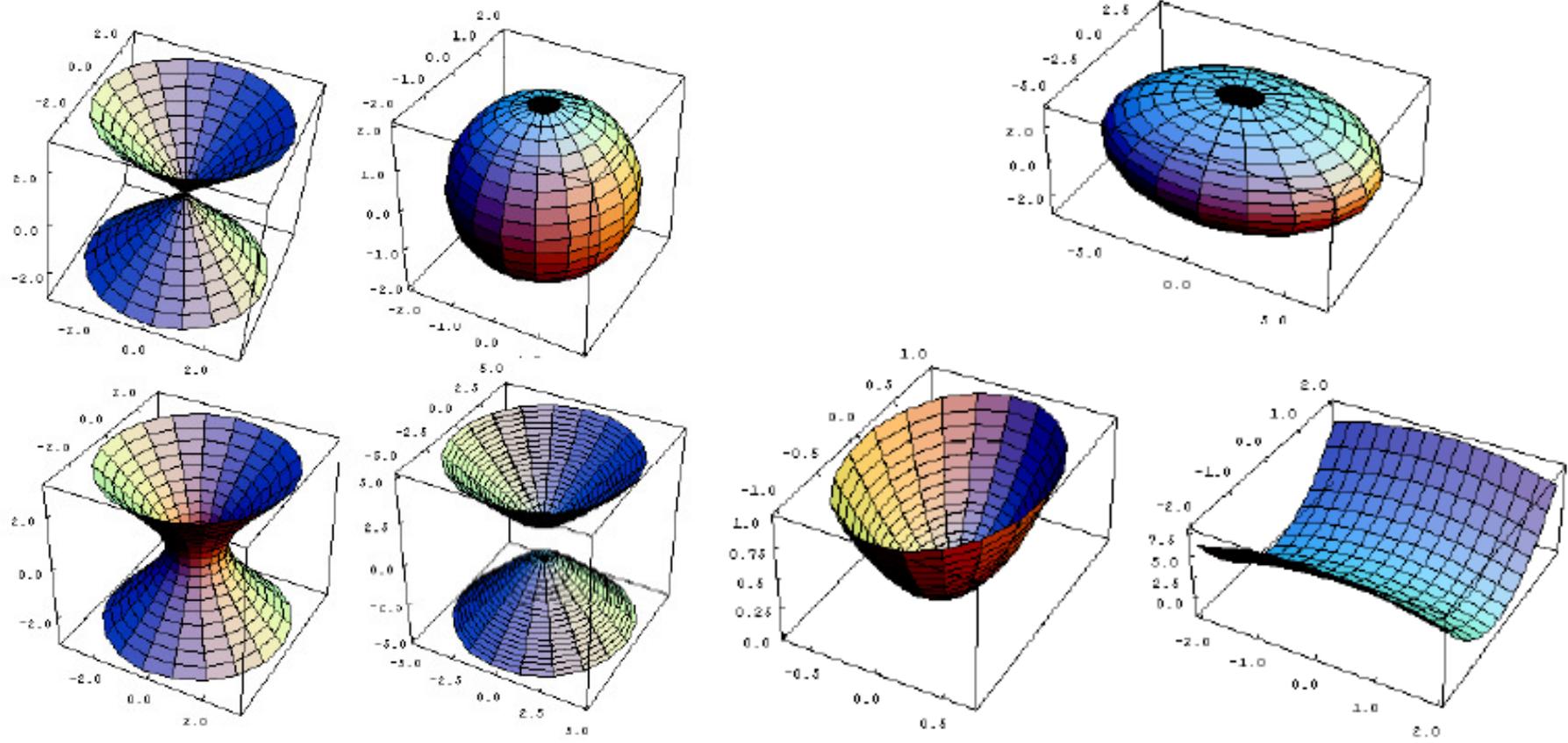
(a) Cone $x^2 + y^2 - z^2 = 0$; (b) hyperboloid of one sheet $x^2 + y^2 - z^2 = a^2$;
(c) hyperboloid of two sheets $x^2 + y^2 - z^2 = -a^2$.

Representações não paramétricas, implícita

Por equações tri-dimensionais :

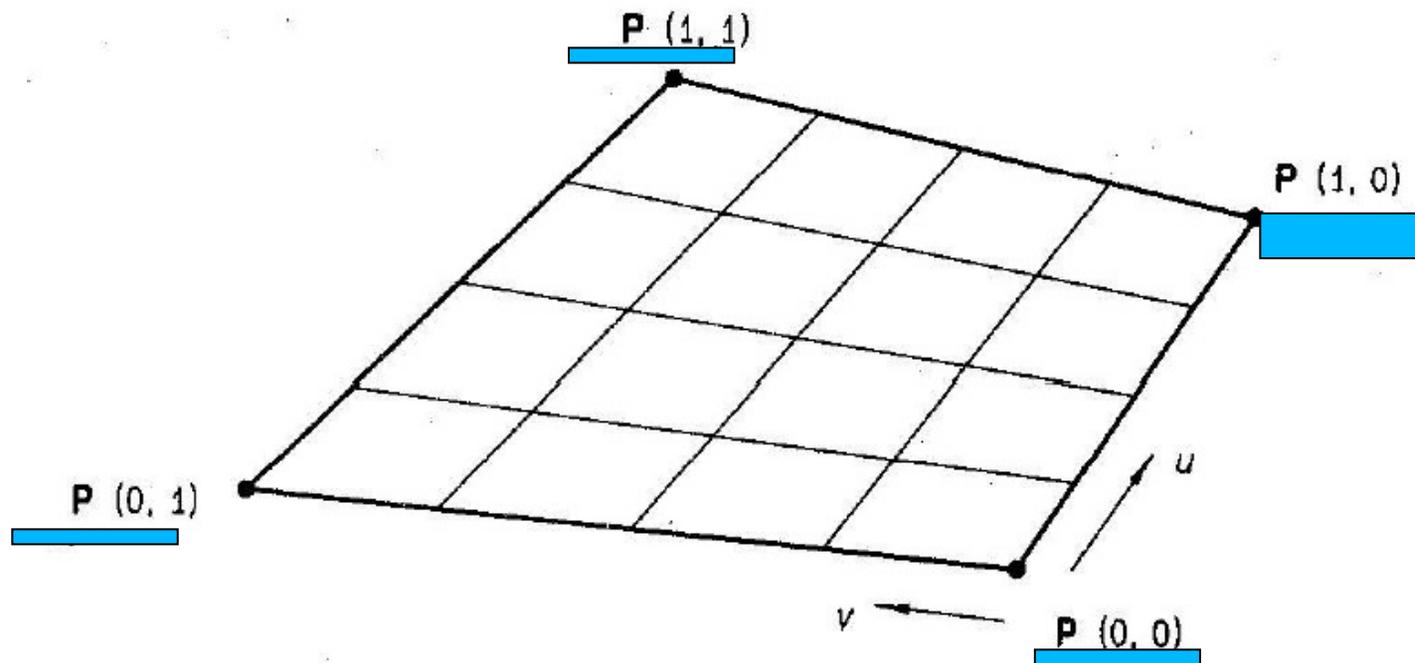
Quádricas

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



Geradas por interpolação

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



Definir pontos na superfície em termos de dois parâmetros (u, v)

Caso mais simples: interpolação bilinear

$$x(u,0) = (1-u)P_{0,0} + uP_{1,0}$$

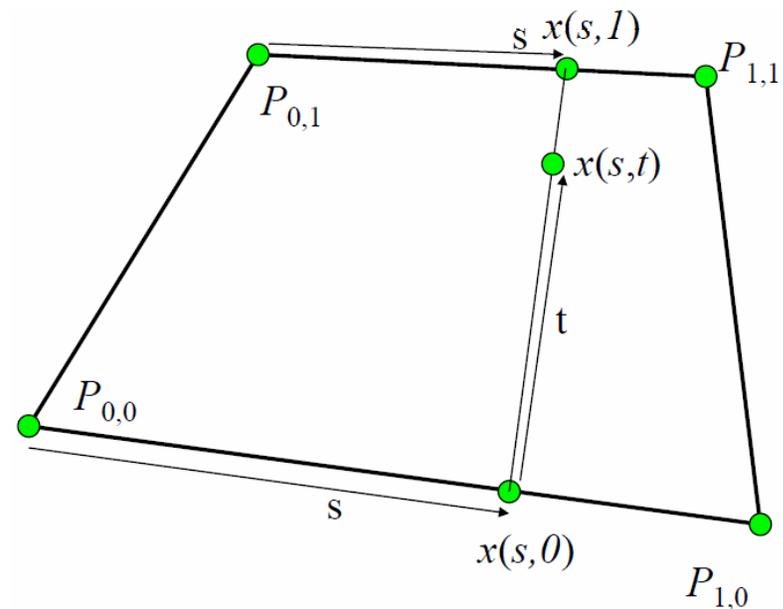
$$x(u,1) = (1-u)P_{0,1} + uP_{1,1}$$

$$x(u,v) = (1-v)x(u,0) + vx(u,1)$$

$$x(u,v) = \sum_{i=0}^1 \sum_{j=0}^1 P_{i,j} F_{i,u}(u) F_{j,v}(v)$$

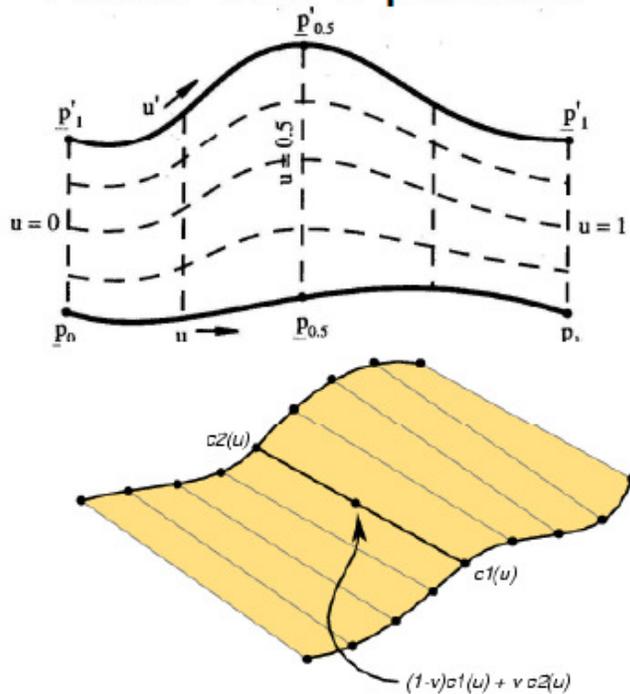
$$F_{0,u} = 1-u, \quad F_{1,u} = u$$

$$F_{0,v} = 1-v, \quad F_{1,v} = v$$



Lofting

- Contruída juntando 2 curvas por linhas retas entre pontos

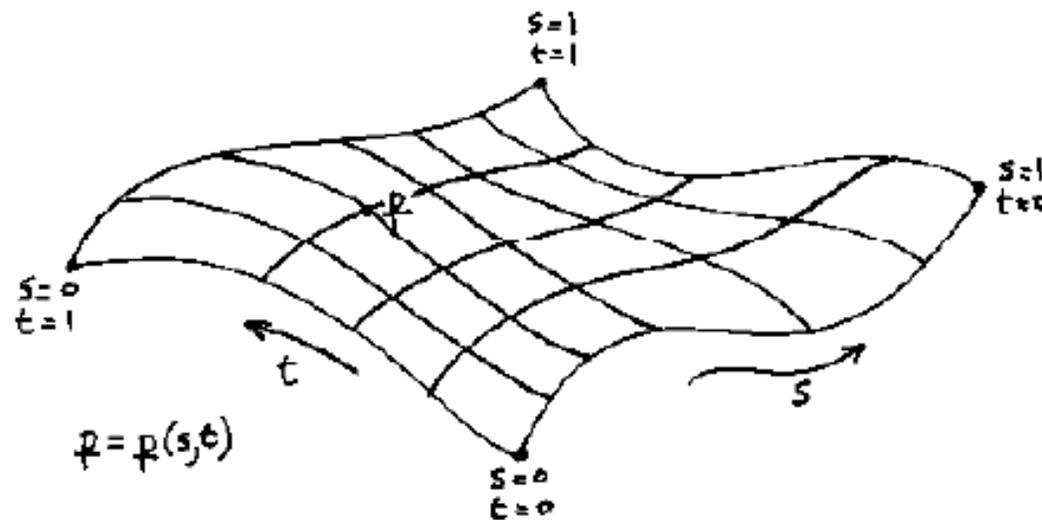


Duas curvas de Bézier podem ser utilizadas para formar uma superfície de Bézier.

$$\mathbf{x}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{i,j} B_i^n(u) B_j^m(v)$$

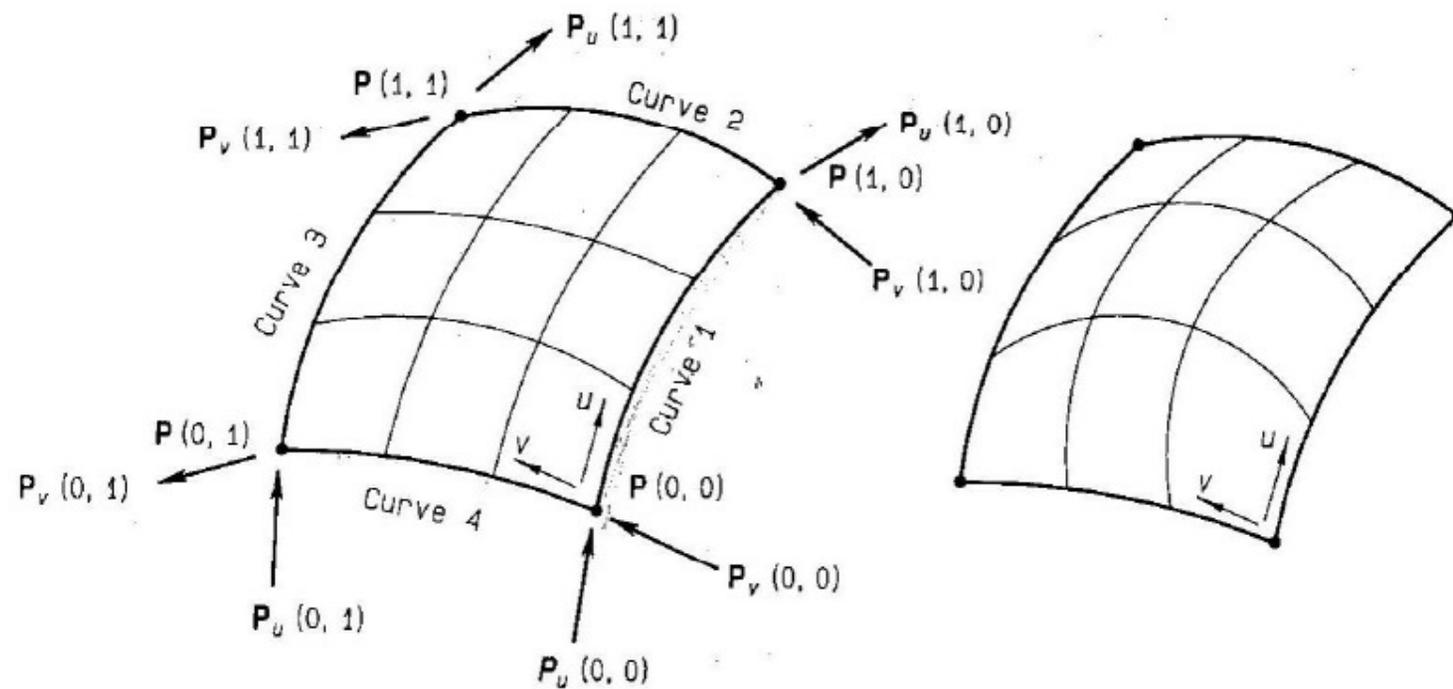
Superfície Linear de Coons

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



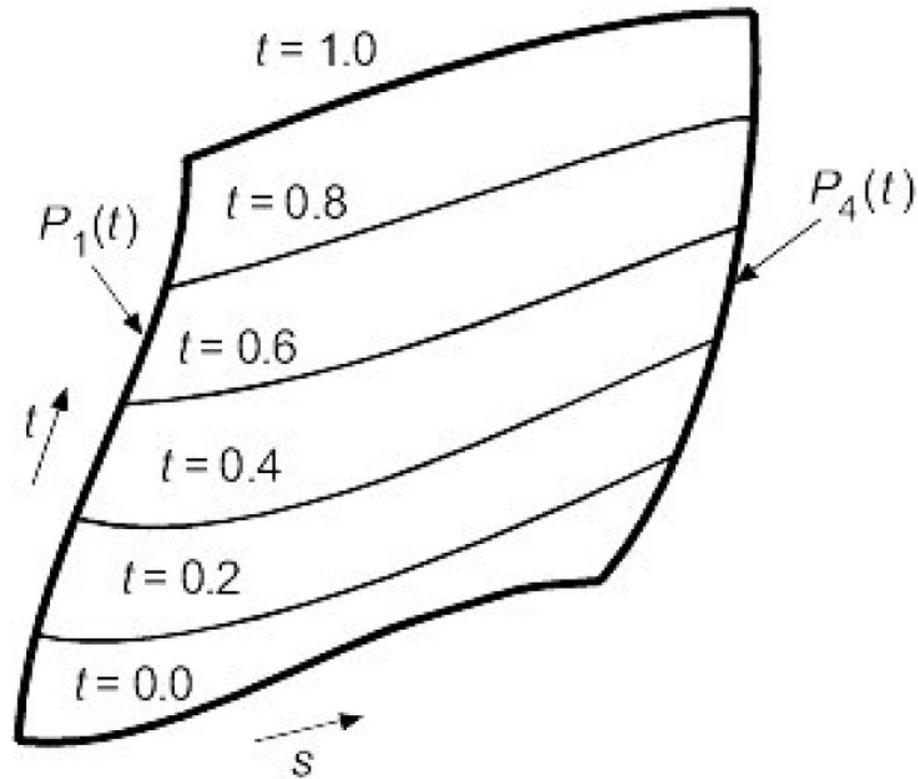
Patches

- Curvas cúbicas paramétricas como quatro curvas de fronteira



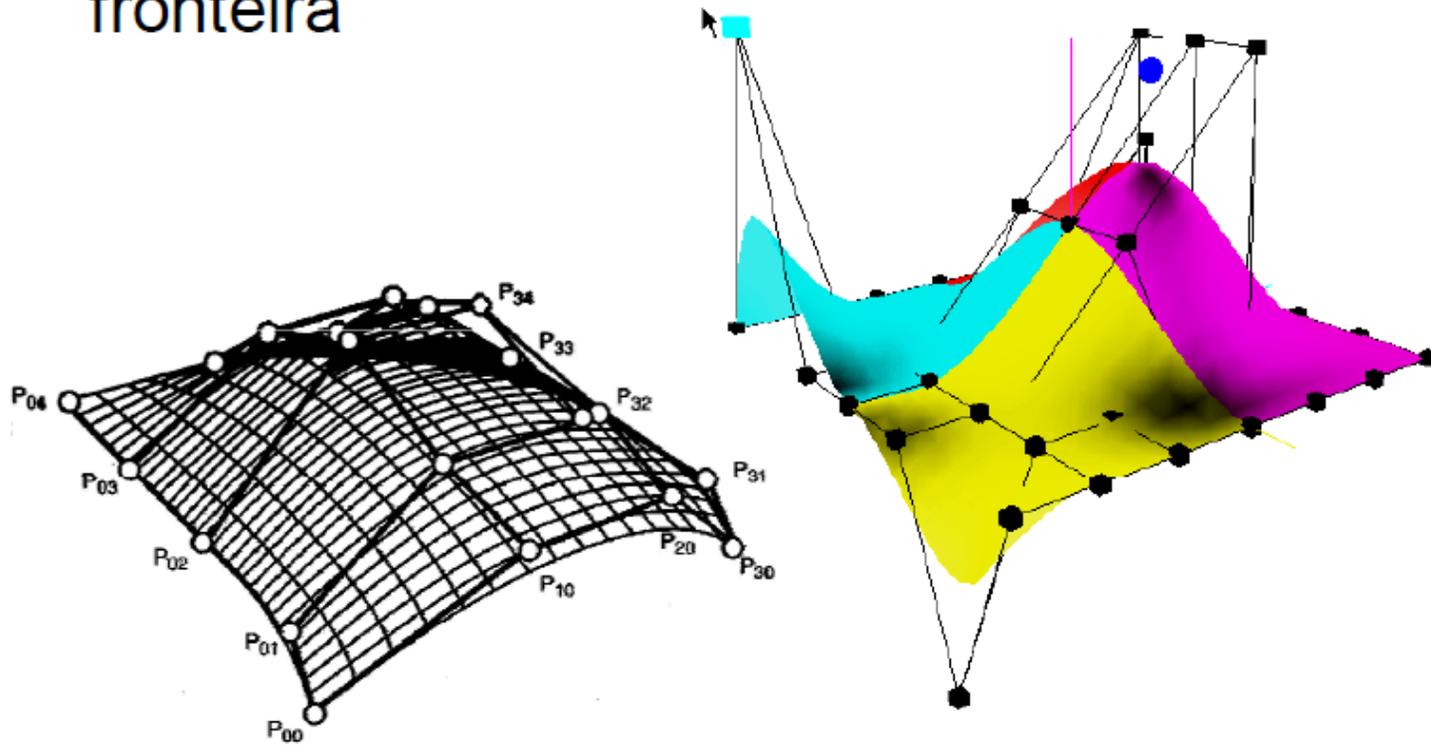
Remendos cúbicos

Superfície de Hermite



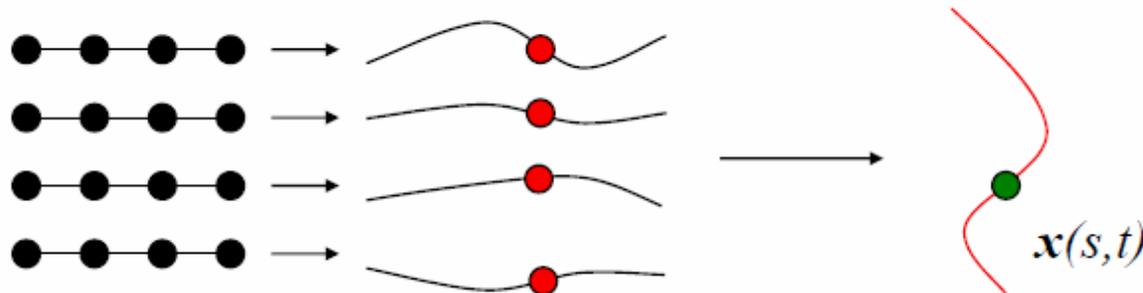
Superfície de Bezier

- Curvas Bezier como quatro curvas de fronteira



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 $\mathbf{x}(s,t)$



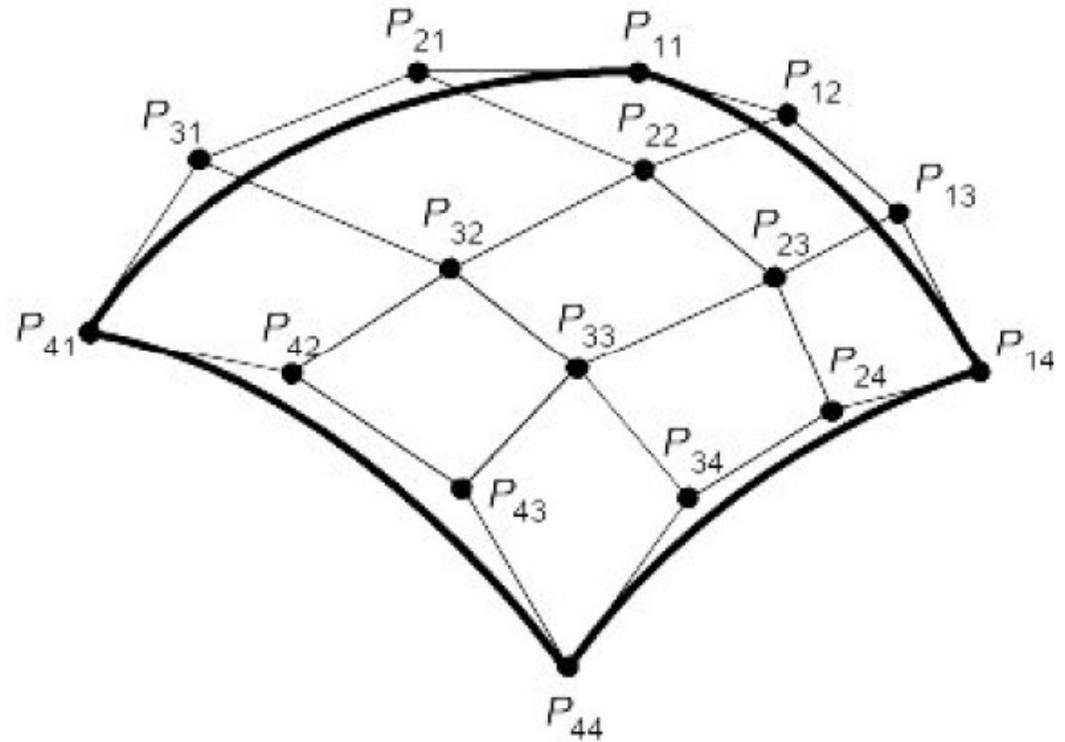
Superfície de Bézier

As equações para a superfície de Bézier podem ser obtidas da mesma forma que as de Hermite, resultando:

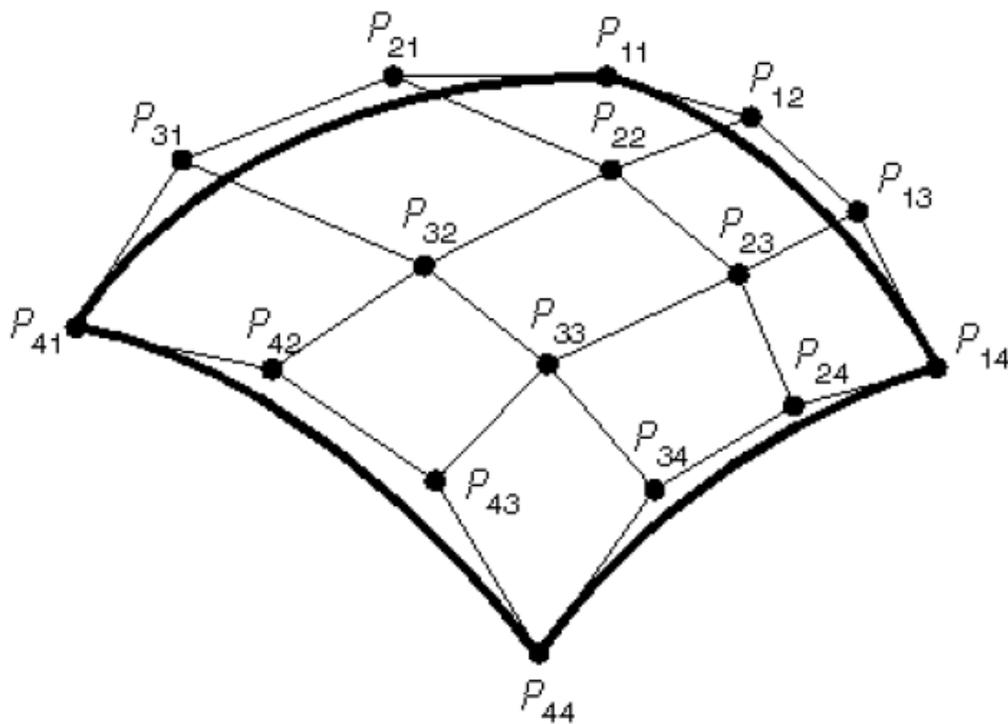
$$x(s, t) = S.M_B.G_{Bx}.M_B^T.T^T$$

$$y(s, t) = S.M_B.G_{By}.M_B^T.T^T$$

$$z(s, t) = S.M_B.G_{Bz}.M_B^T.T^T$$



A matriz geométrica tem 16 pontos de controlo.



- Superfície de Bézier é dada por

$$x(s, t) = S \cdot M \cdot G_x^T \cdot M^T \cdot T^T,$$

$$y(s, t) = S \cdot M \cdot G_y^T \cdot M^T \cdot T^T,$$

$$z(s, t) = S \cdot M \cdot G_z^T \cdot M^T \cdot T^T,$$

$$0 \leq s, t \leq 1$$

e é definida por 16 pontos de controle (matriz G)

Superfície de Bézier

Continuidade C^0 e G^0 é obtida fazendo coincidir os quatro pontos de controlo de fronteira: P_{14} , P_{24} , P_{34} , P_{44}

Para obter G^1 devem ser colineares:

P_{13} , P_{14} e P_{15}

P_{23} , P_{24} e P_{25}

P_{33} , P_{34} e P_{35}

P_{43} , P_{44} e P_{45}

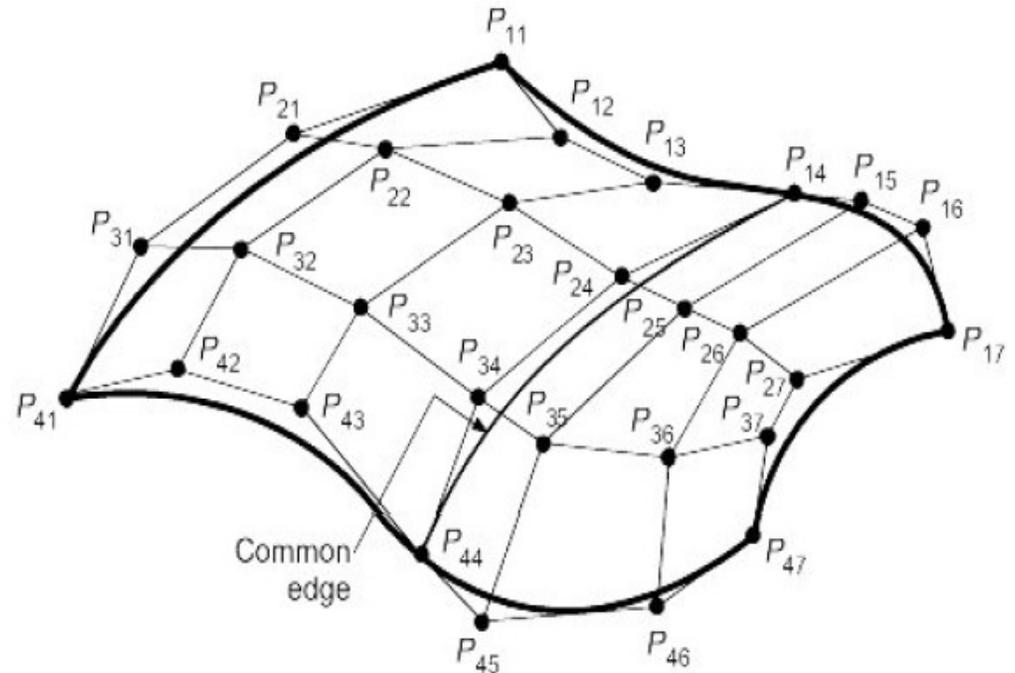
e

$$(P_{14} - P_{13}) / (P_{15} - P_{14}) = K$$

$$(P_{24} - P_{23}) / (P_{25} - P_{24}) = K$$

$$(P_{34} - P_{33}) / (P_{35} - P_{34}) = K$$

$$(P_{44} - P_{43}) / (P_{45} - P_{44}) = K$$

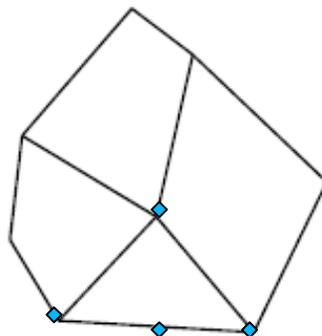


Malhas de Retalhos Bézier

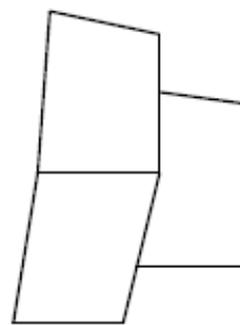
- São malhas compostas de diversos retalhos unidos ao longo de suas fronteiras
 - As arestas das grades de controle precisam se justapor perfeitamente
 - As grades precisam ser retangulares



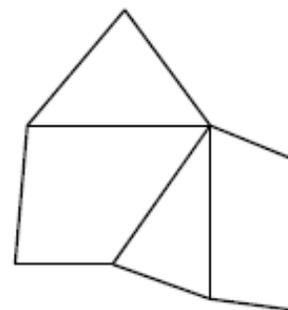
OK



OK

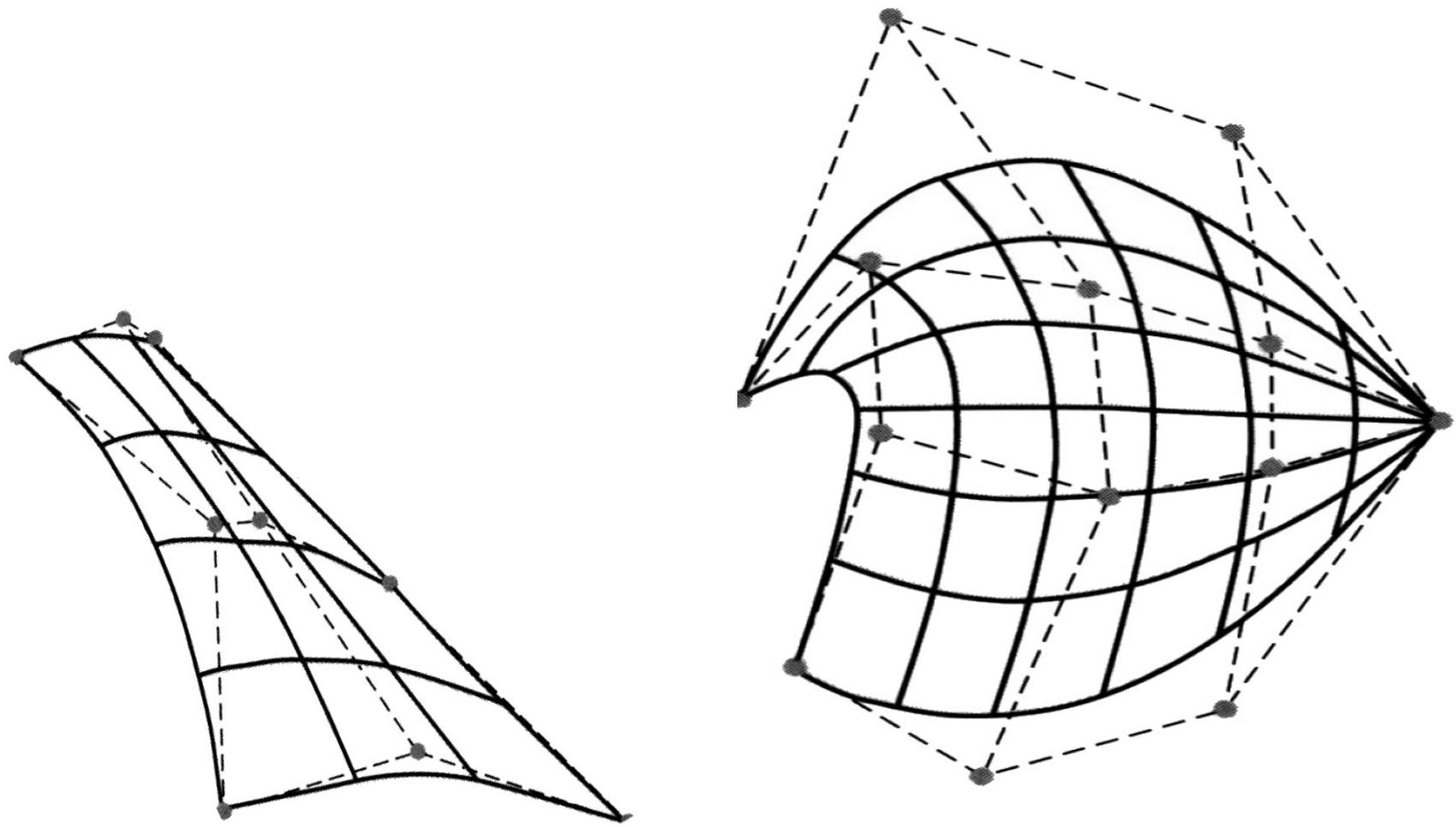


Não



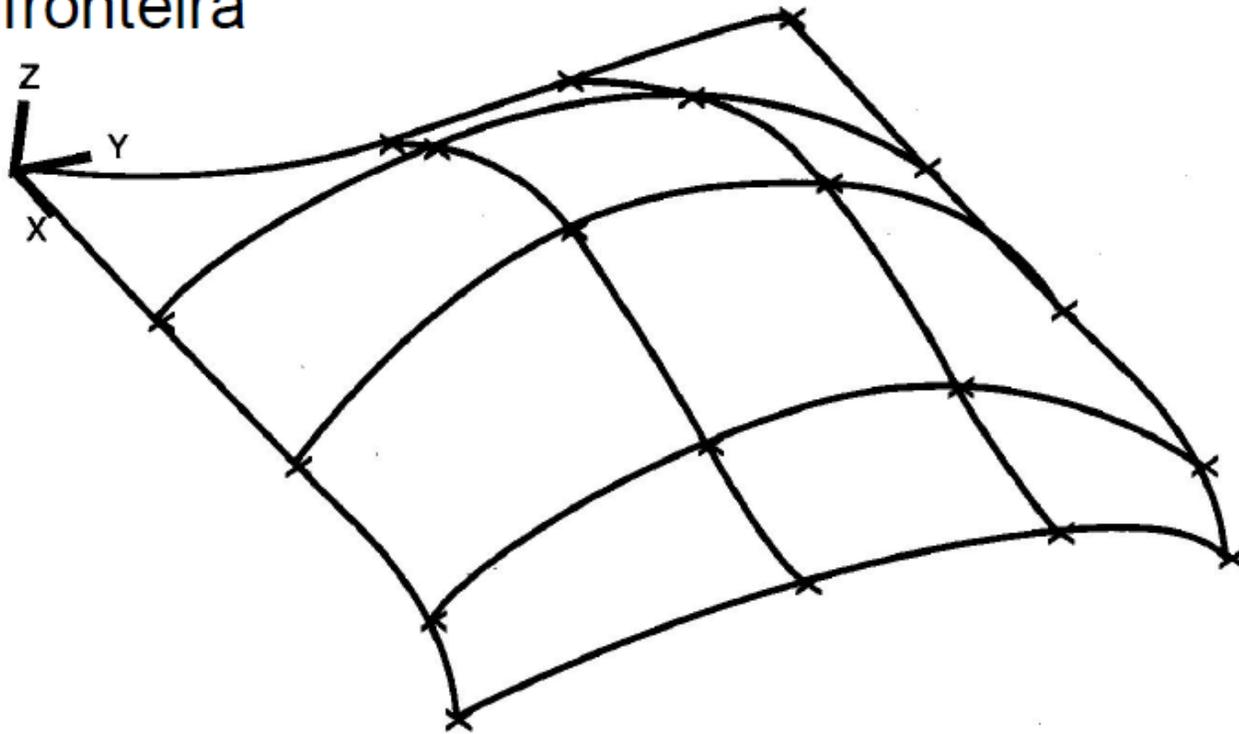
Não

A seguir dois exemplos de superfícies geradas com $m = 3, n = 3$ e $m = 4, n = 4$ pontos de controle.



Superfície B-Splines

- Curvas B-Spline como quatro curvas de fronteira



T-spline surface

- Superfície que pode ser considerada como uma NURBS na qual uma sequência de pontos de controle determina a superfície, que lembra a letra "T".
- Esse tipo de superfície facilita a fusão de pedaços .

Malha de polígono (mesh)

- Coleção de vértices e polígonos que definem a forma de um objeto poliédrico
- Malhas de triângulos ou quadriláteros
 - triangulação
- Bom para caixas, armários, construir exteriores
- Ruim para superfícies curvas

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