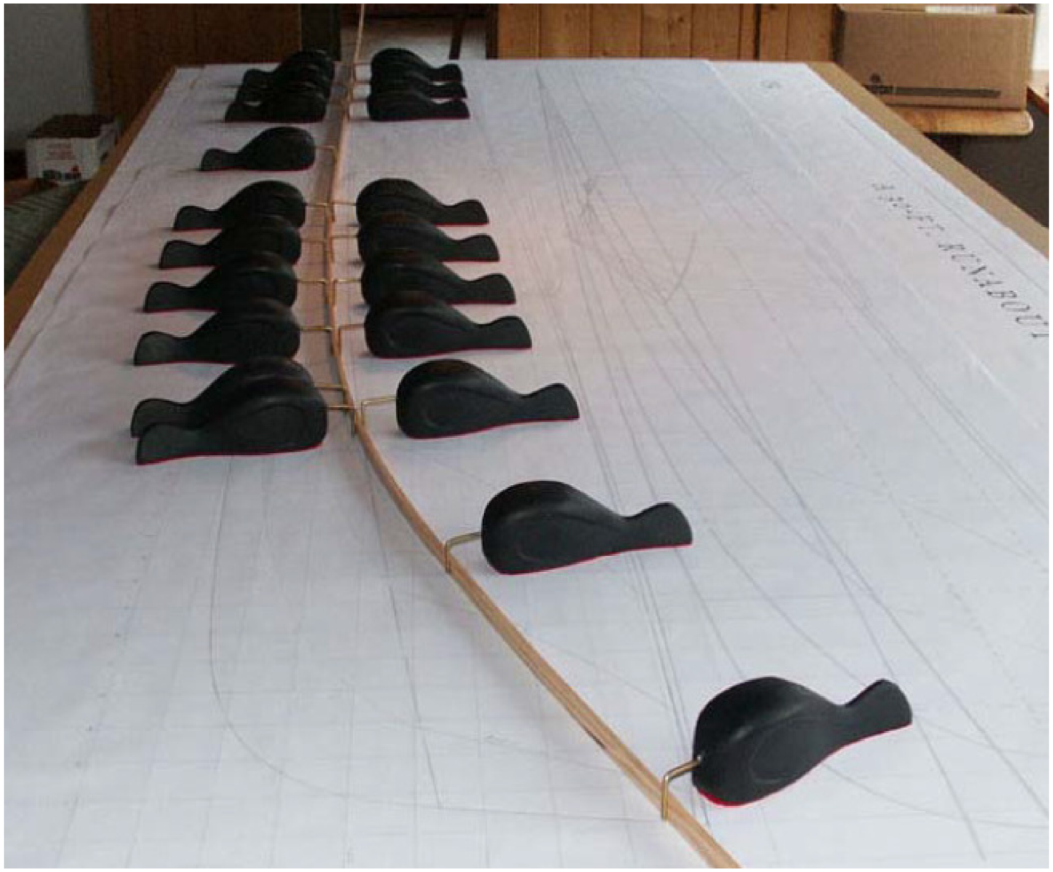
aula 7



Spline física

Curvas Splines

2016/2 – IC / UFF

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

Curvas Splines

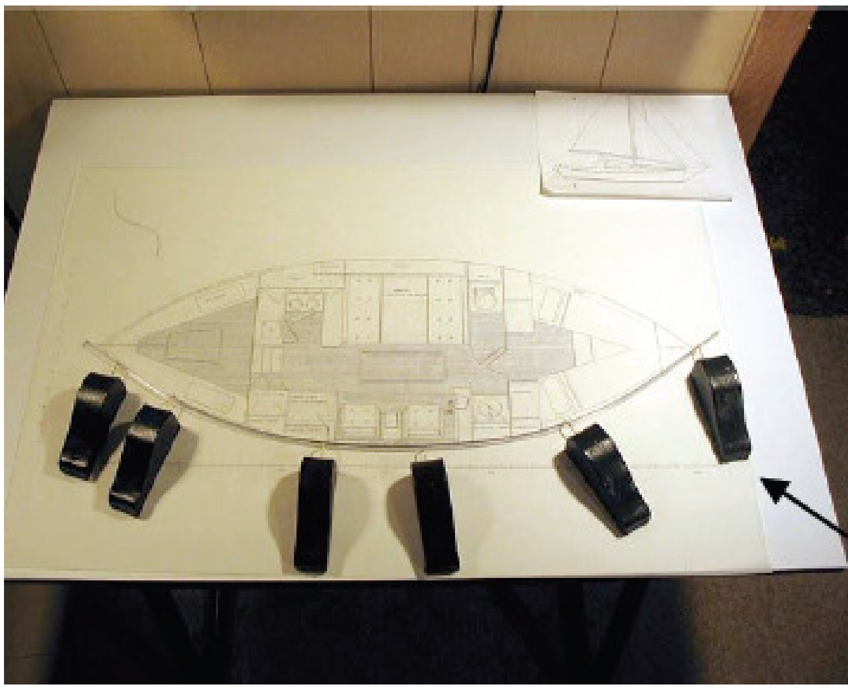
Com maior suavidade que as anteriores (tem **curvatura continuas**) e são conectadas formando curvas mais complexas.

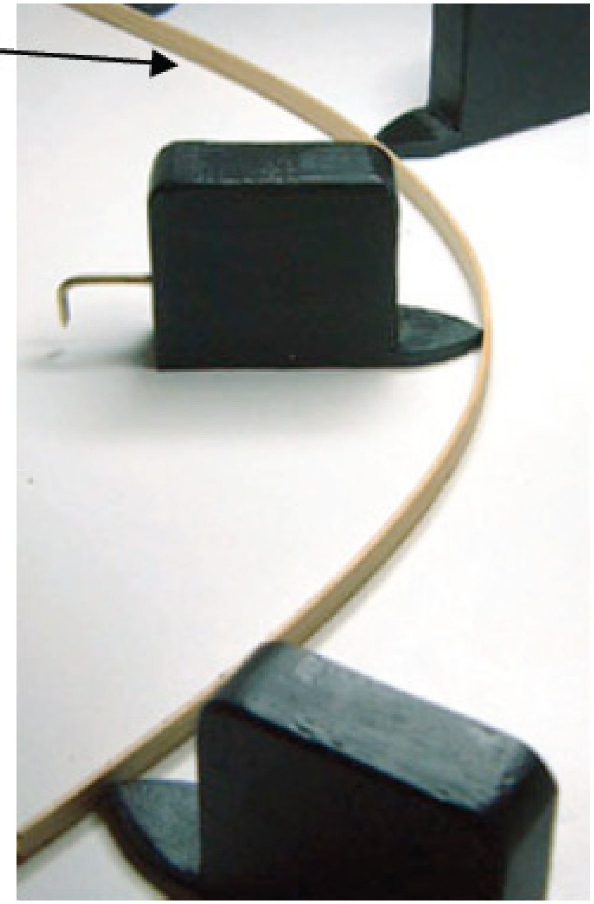
Spline é uma curva polinomial definida por partes

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

Metal flexível com continuidade de curvatura: C^2

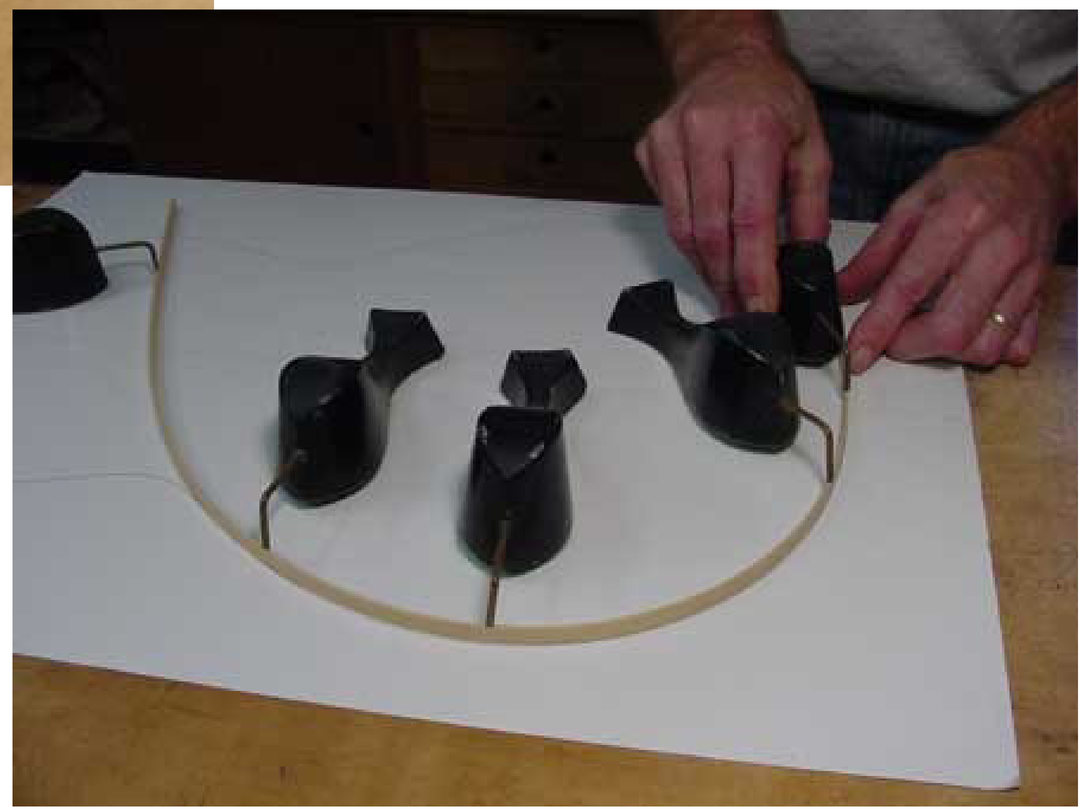
Construindo barcos





Pesos que dão forma = “ducks”

Exemplo de como são usadas



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}

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 involve inverter uma matriz de

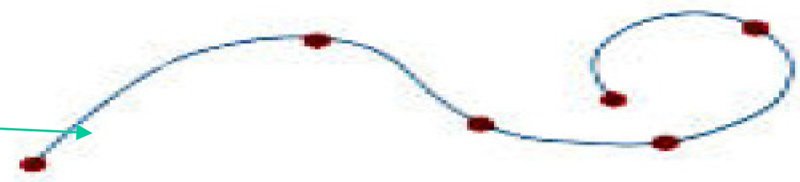
(n+1)x(n+1) pontos

Interpolação com splines cubicas

Dado um conjunto de N+1 coordenadas de pontos (P_0,P_1,P_2,..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 <= u <= 1)

Interpolação por Splines Cúbicas



Usa a teoria das vigas esbeltas

E é apresentada na seção 11.5

de ***Álgebra Linear***

com Aplicações

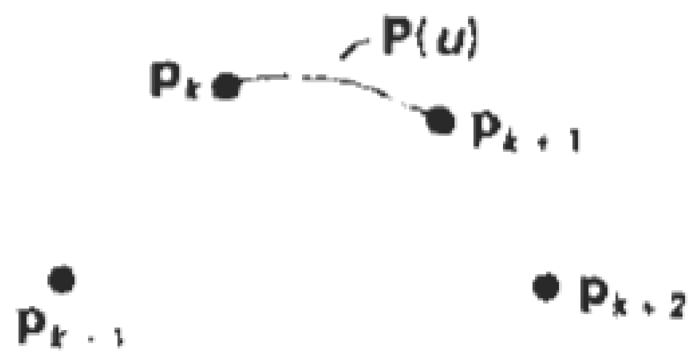
A. Anton e C. Rorres,

Bookman, 2001

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

Mas "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 ***Pk-1 , Pk , Pk+1 , Pk+2*** são os pontos de controle

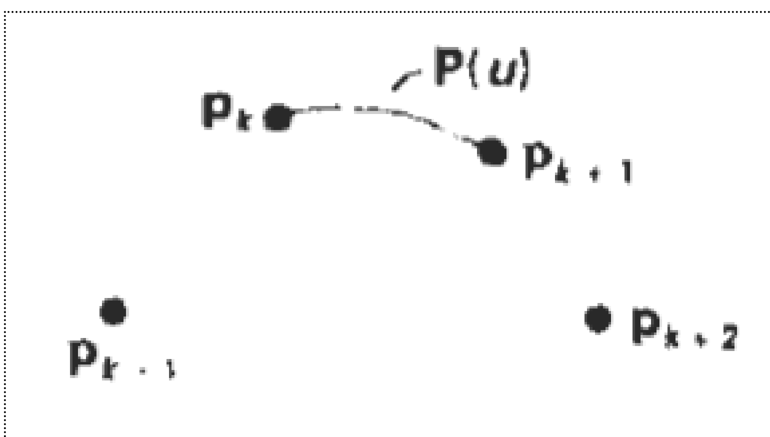
"Cardinal Spline"

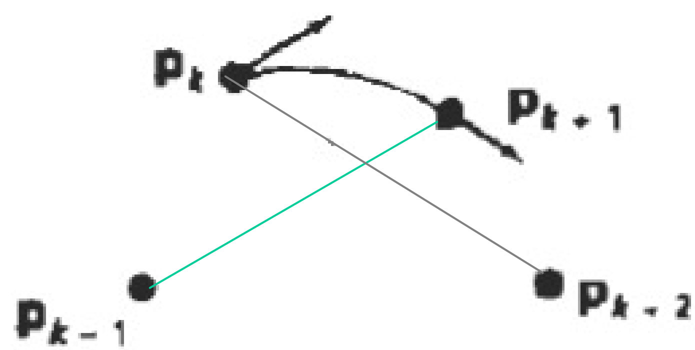
Especificada por 4 pontos de controle consecutivos:

Os 2 do meio definem o inicio e o fim da curva, e os 2 extremos ajudam a definir as inclinações da mesma nas extremidades, usando também o ponto seguinte:

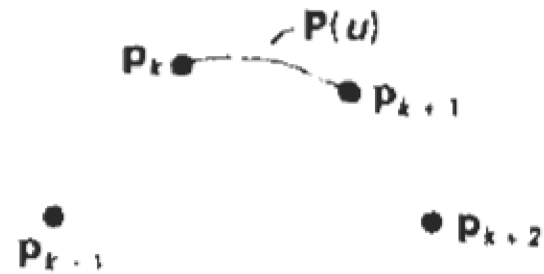
P(u) é a curva

$P_{k-1}, P_k, P_{k+1}, P_{k+2}$ => ***Os pontos de controle***





“**Cardinal** Spline”



P(u) é a curva

P_{k-1}, P_k, P_{k+1}, P_{k+2} => pontos de controle

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

$$\mathbf{P}(0) = \mathbf{p}_k$$

$$\mathbf{P}(1) = \mathbf{p}_{k+1}$$

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

$$\mathbf{P}'(1) = \frac{1}{2}(1 - t)(\mathbf{p}_{k+2} - \mathbf{p}_k)$$

$t < 0$ (Looser Curve) $t > 0$ (Tighter Curve)

"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:

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

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

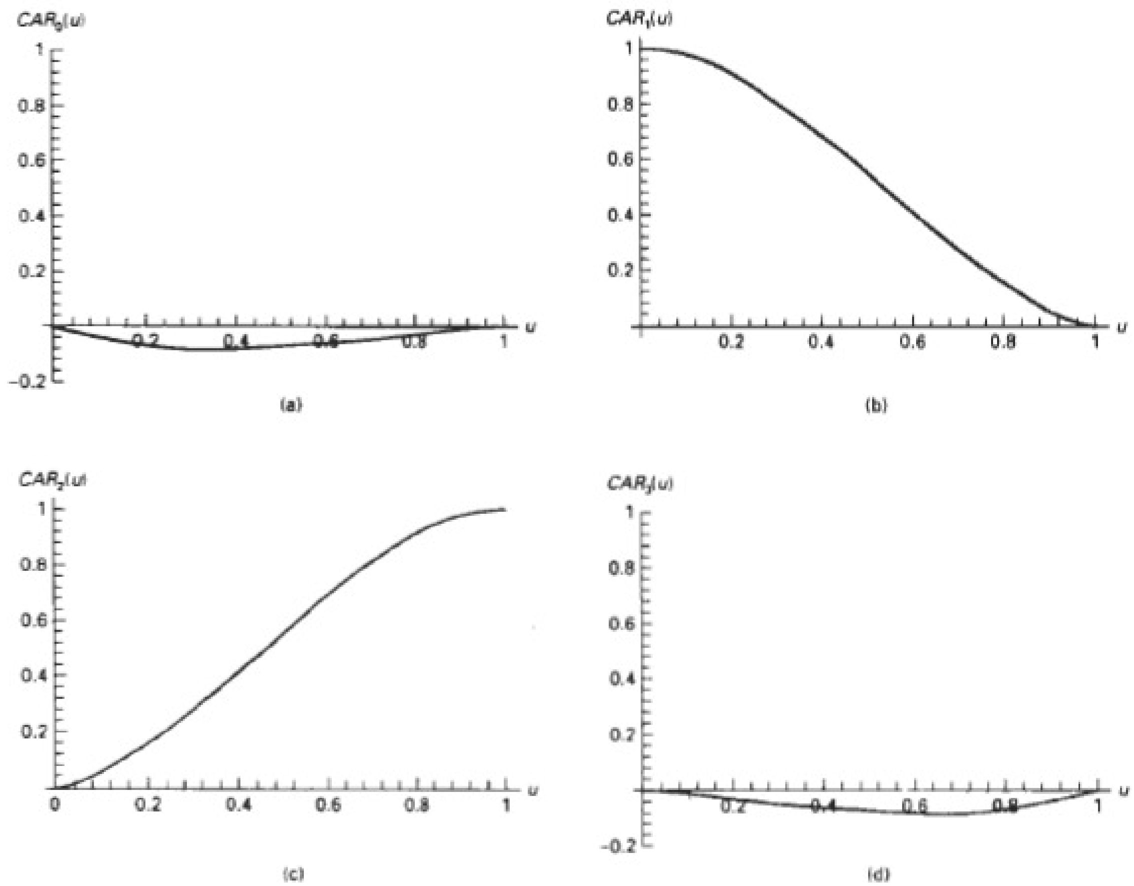
$$\mathbf{M}_C = \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \mathbf{P}(u) &= \mathbf{p}_{k-1}(-su^3 + 2su^2 - su) + \mathbf{p}_k[(2-s)u^3 + (s-3)u^2 + 1] \\ &\quad + \mathbf{p}_{k+1}[(s-2)u^3 + (3-2s)u^2 + su] + \mathbf{p}_{k+2}(su^3 - su^2) \\ &= \mathbf{p}_{k-1}CAR_0(u) + \mathbf{p}_k CAR_1(u) + \mathbf{p}_{k+1}CAR_2(u) + \mathbf{p}_{k+2}CAR_3(u) \end{aligned}$$

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

“Cardinal Spline”

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



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

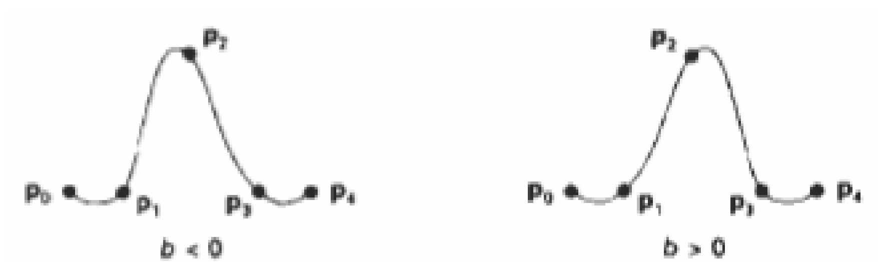
Computer Graphics C version de D. Hearne e M.P. Baker , p. 345-357

Outras Splines

As Korchanek-Bartel splines, são uma classe de Cardinais que além do parametro de tensão incluem mais dois:

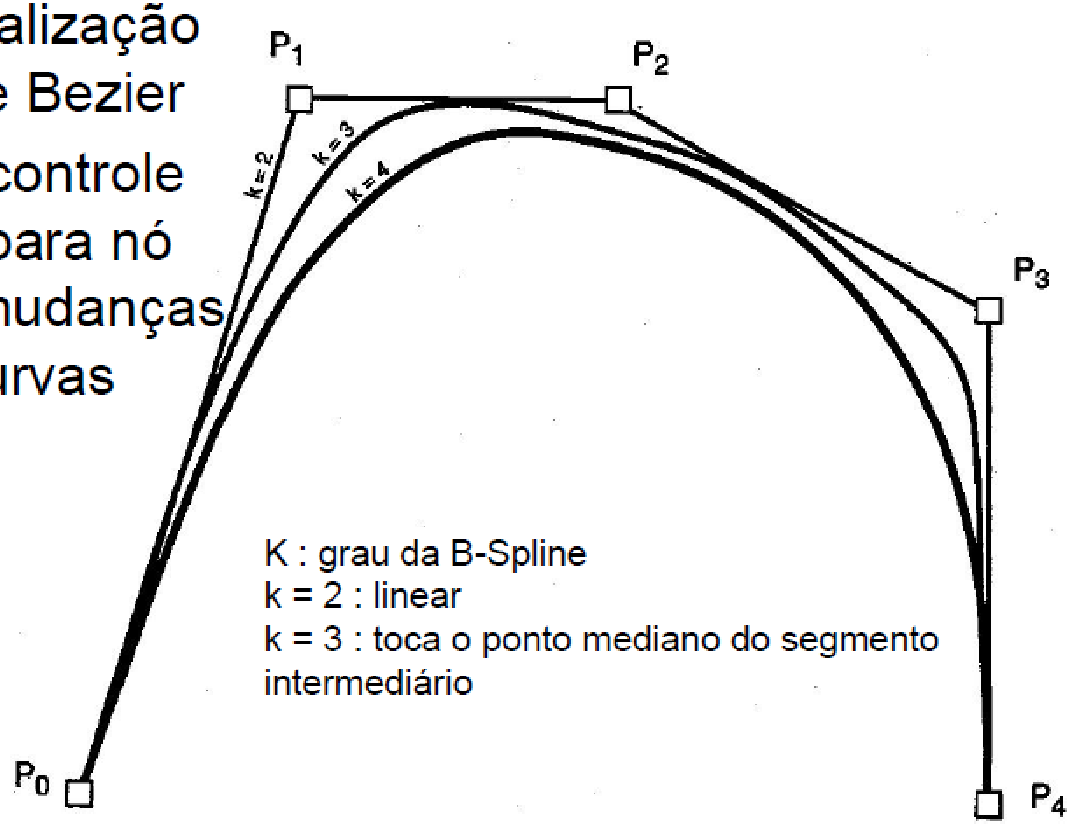
b – bias e c – continuidade

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 para nó permitem mudanças locais às curvas



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

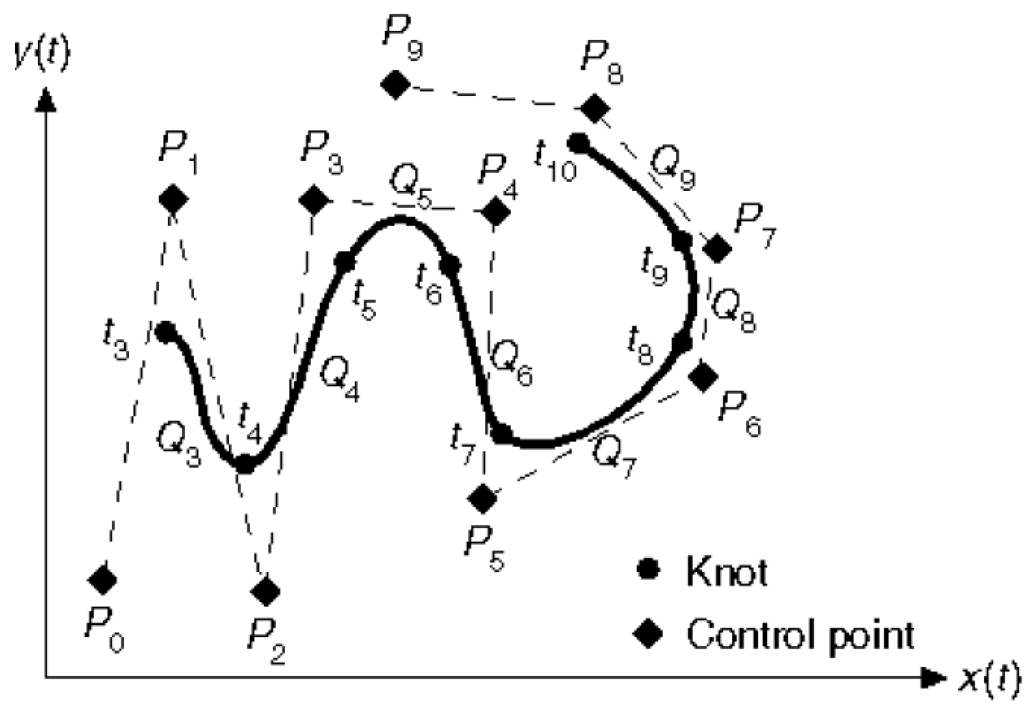
B-spline ou **basis spline**

Caracteristicas.

- 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 $\boldsymbol{p_i}$ pontos de controle .

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

Nós ≠ pontos de controle



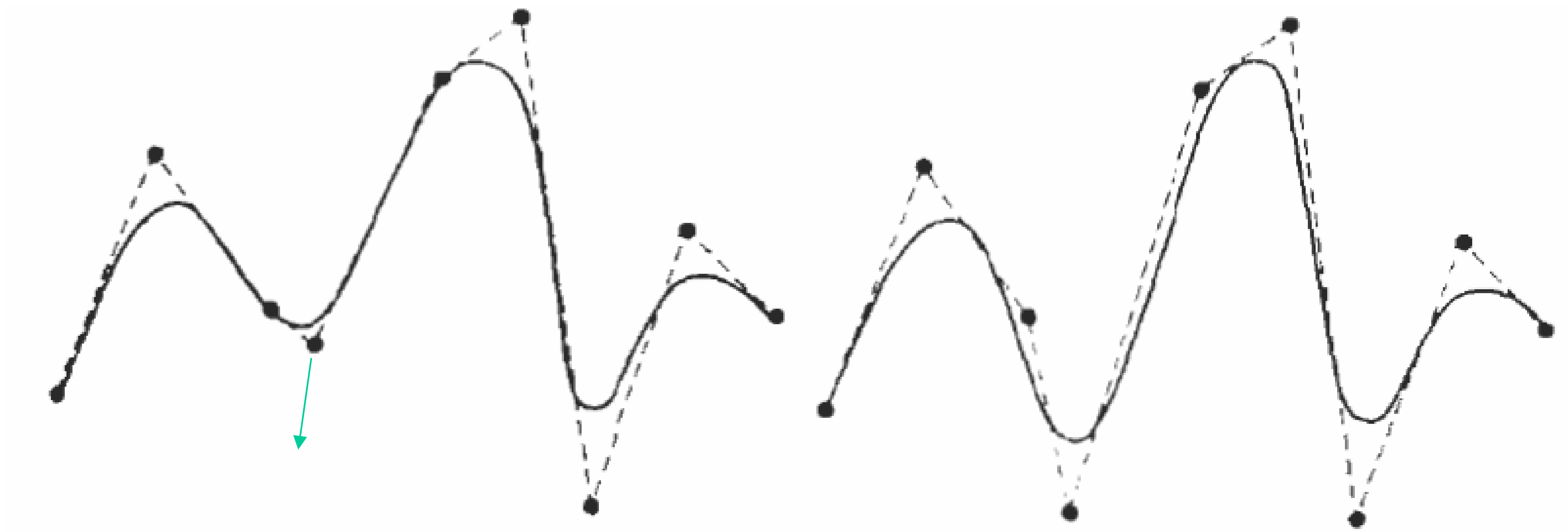
Nós:

No espaço paramétrico global temos **nós** ou **knots** que representam os valores de u onde os segmentos Qi 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 Qi 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 knots (u_0 -> u_{m+k})

Número de nós: nº de pts de cntrl + ordem da curva

As funções de interpolação são definidas de maneira recursiva em um intervalo d:

$$B_{k,1}(u) = \begin{cases} 1, & \text{if } u_k \leq u < u_{k+1} \\ 0, & \text{otherwise} \end{cases}$$

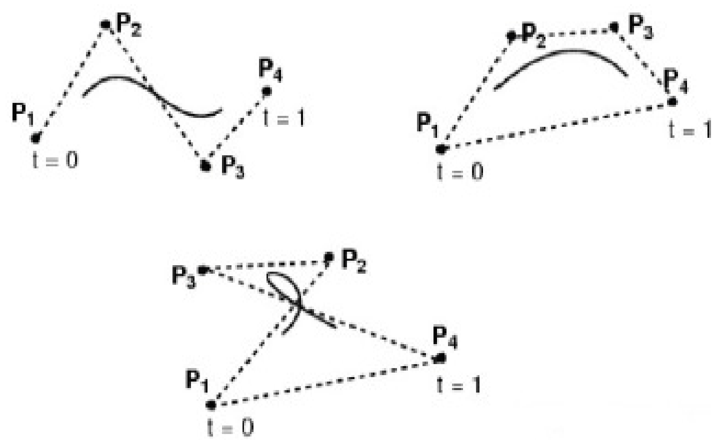
$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d-1} - u_k} B_{k,d-1}(u) + \frac{u_{k+d} - u}{u_{k+d} - u_{k+1}} B_{k+1,d-1}(u)$$

$$\sum_{k=0}^{n} B_{k,d}(u) = 1$$

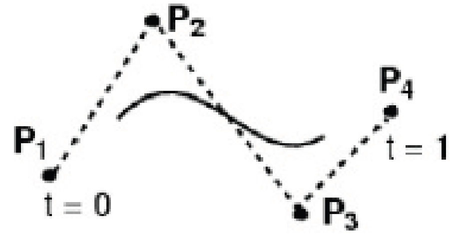
- Chamada Uniforme B-splines se têm *knots* que são eqüidistantes uns dos outros em função do parametro u.
- Muito usadas em CG são as Cúbicas com ***m+1*** pontos de controle , $P_0, P_1, P_2, \ldots, P_m$ onde, $m \geq 3$

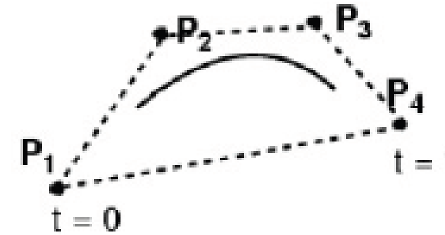
- Polinômios Cúbicos
  - $f(u) = a_x u^3 + b_x u^2 + c_x u + d_x$
  - u: (0 <= u <= 1)

- 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.



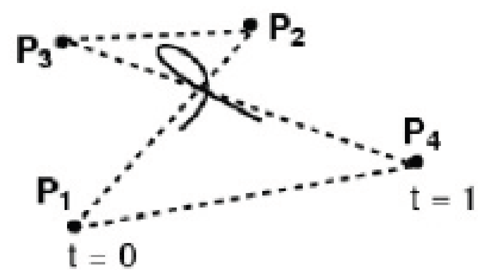
Porque são melhores? Porque fornecem **suporte local** e mais controle (ao contrário da Bézier que por mínima mudança nos pontos de controle, gera uma nova curva)





Genericamente:

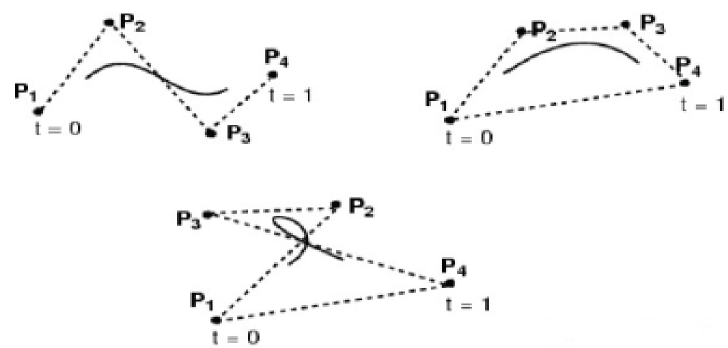
– Para m+1 pontos de controle
  • M>=3 P0,P1,..,Pn
– Teremos curvas com m-2 segmentos
  • Q3,Q4,..,Qm



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 P0 e Pn)

Knots são os pontos de junção entre os segmentos de curva



B-Splines Uniformes

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

Spline controlada por 4 pontos

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

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

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

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

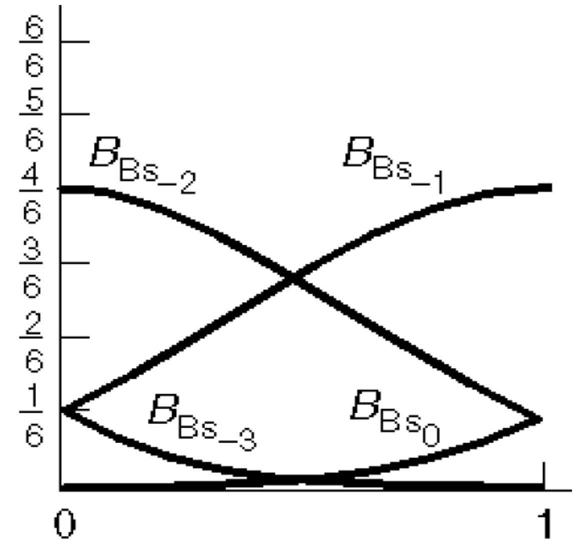
$$B_{i-3} = \frac{1}{6}(1-u)^3 \quad \text{com} \quad 0 \leq u \leq 1$$

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

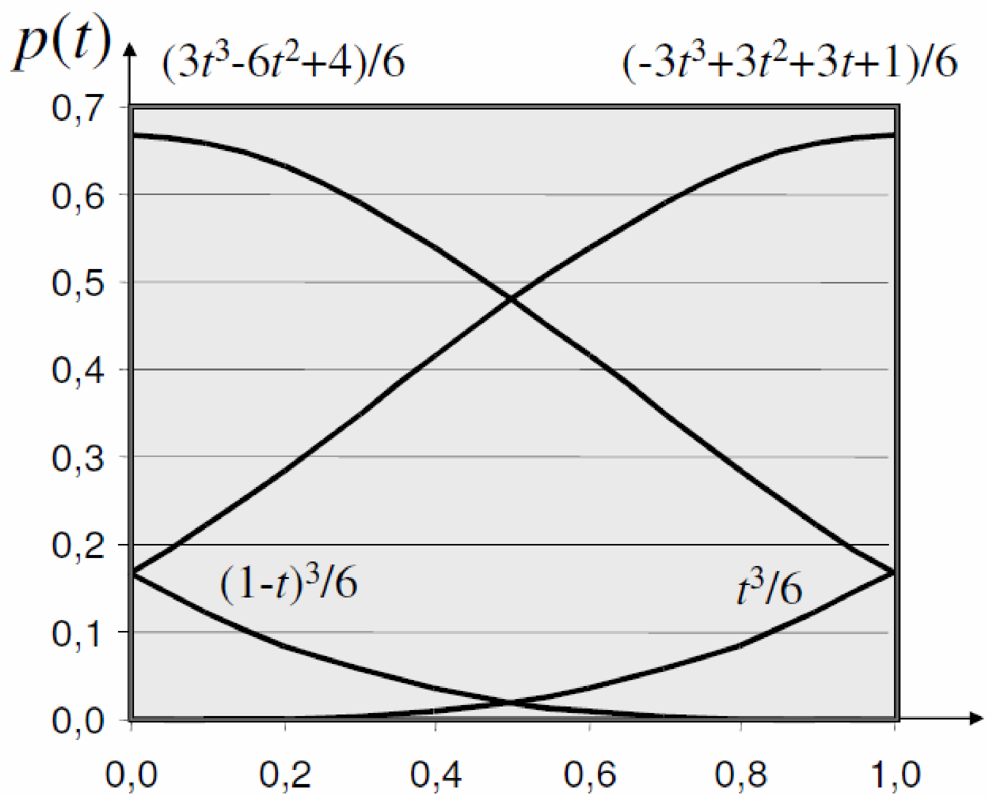


B-Splines Uniformes

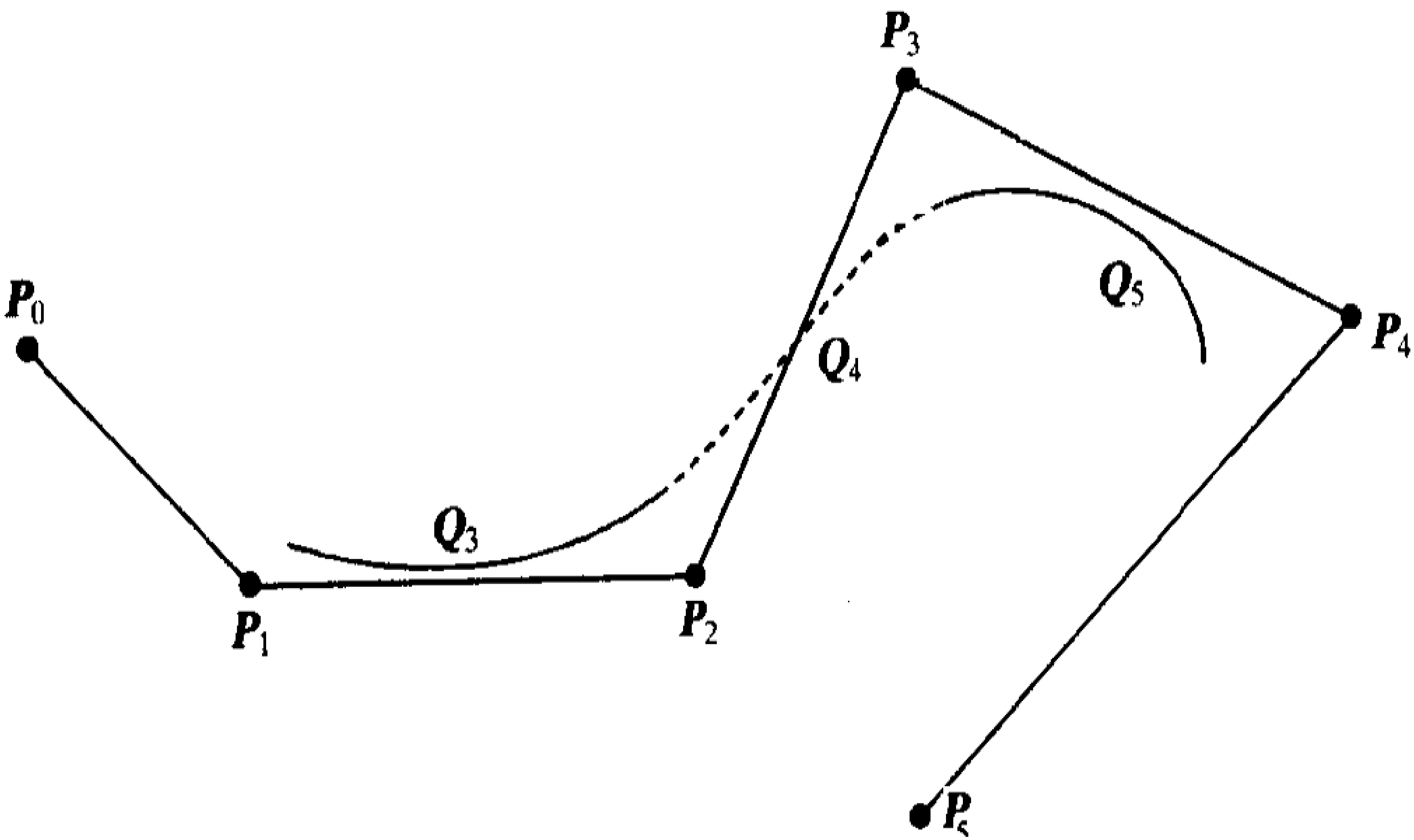
$$Q_{BS}(t) = \frac{(1-t)^3}{6} P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6} P_{i-2} +$$

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

$0 <= t <= 1$

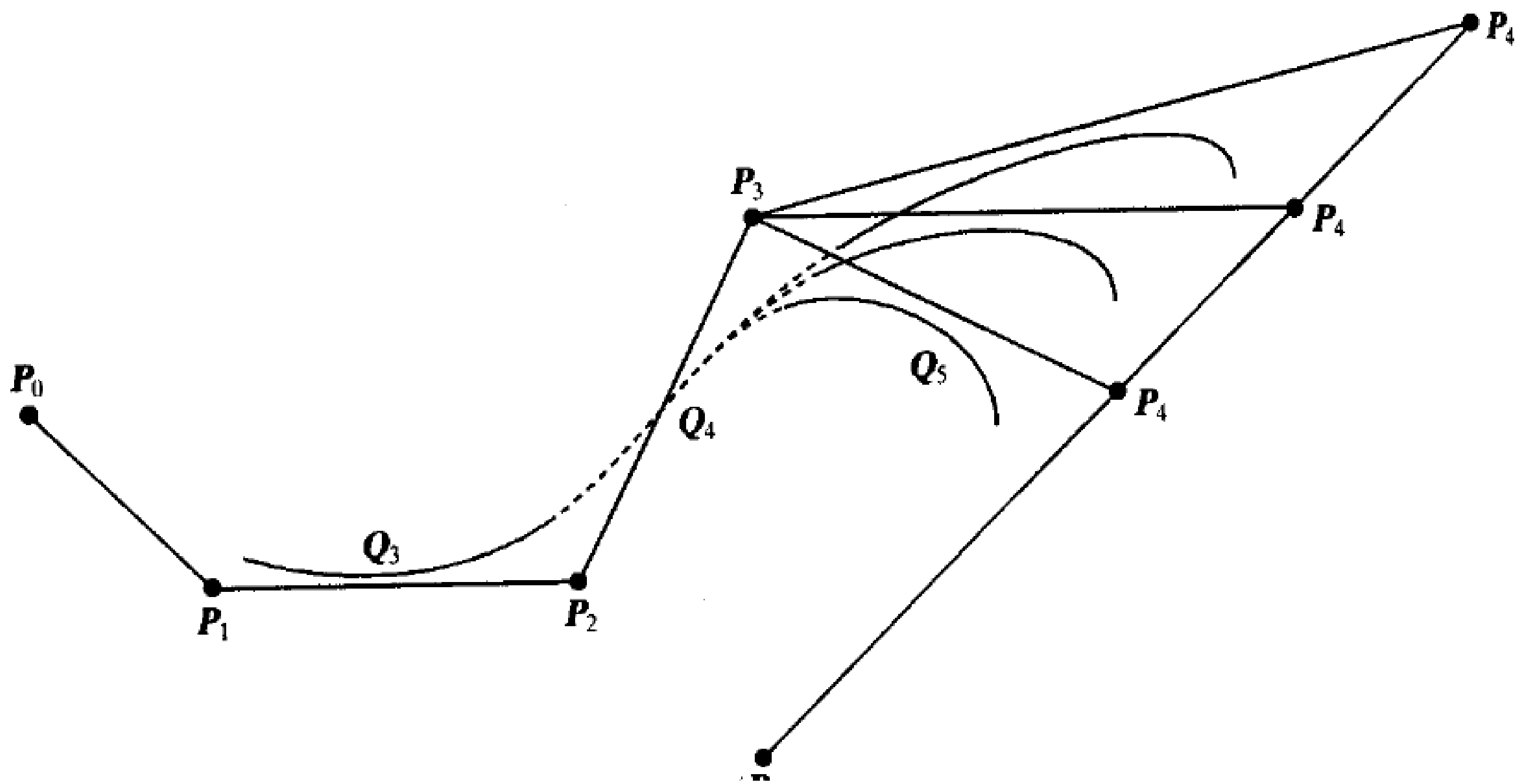


Unido 3 curvas B-Splines



Exemplo de controle local:

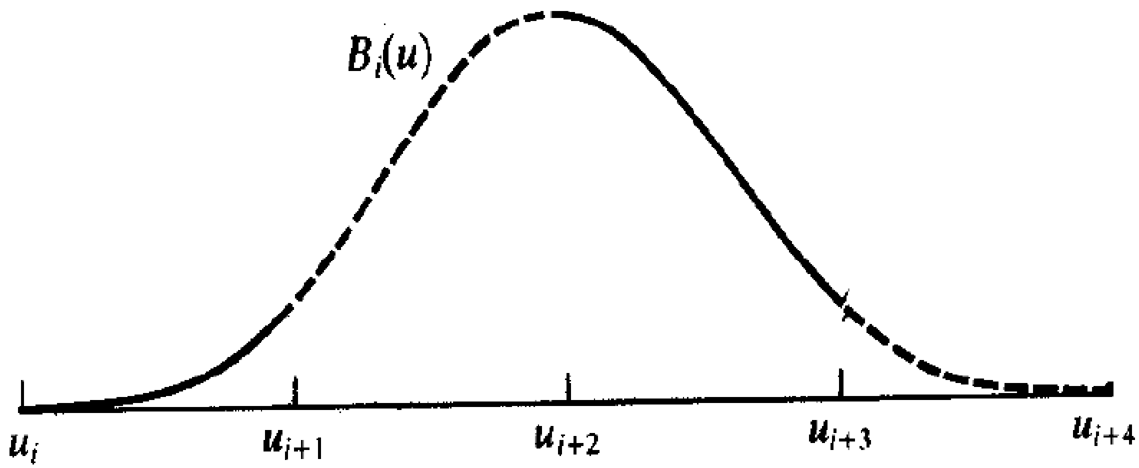
Alterando o penúltimo ponto, não se altera o trecho inicial e só parte do trecho intermediário



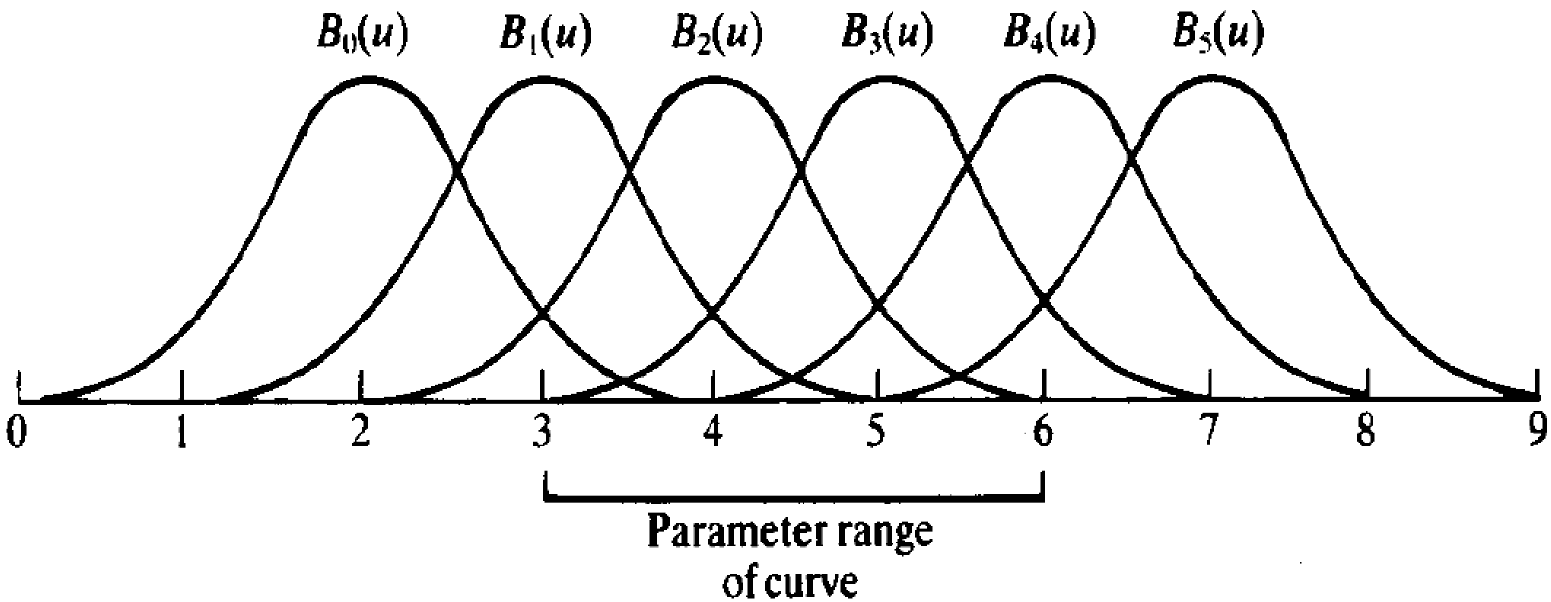
Ao ser controlada por 4 pontos, só se aproxima dos 2 centrais

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).

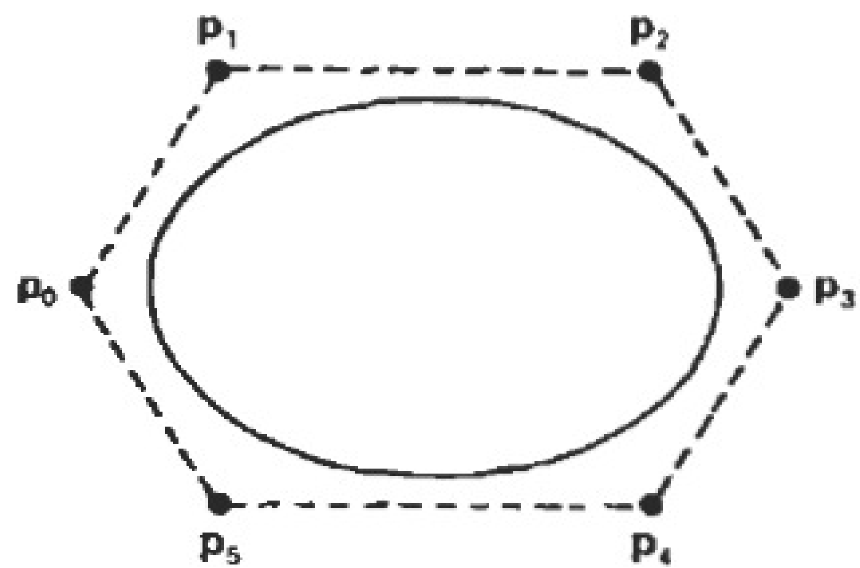
Número total de nós: 10

Uma curva B-Spline é calculada por:

Para criar uma curva spline fechada:

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

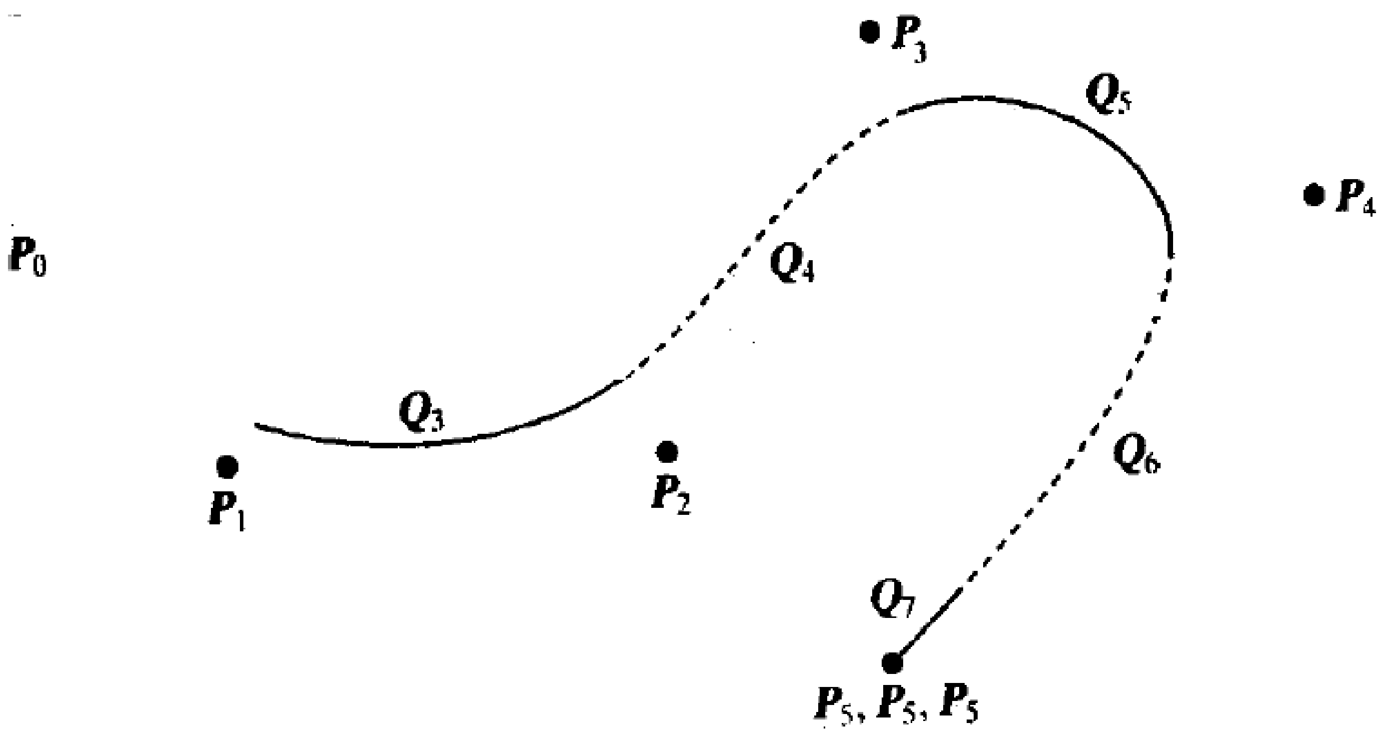
P_0, P_1, P_2, P_3 .... $.P_m, P_0, P_1, P_2$



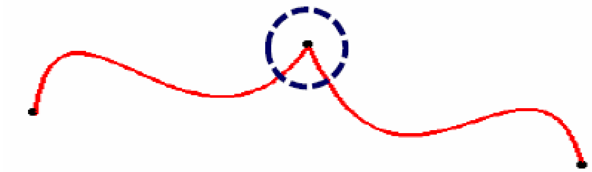
Spline com pontos controle coincidentes seguidos =>
Ela acaba por passar pelo ponto

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

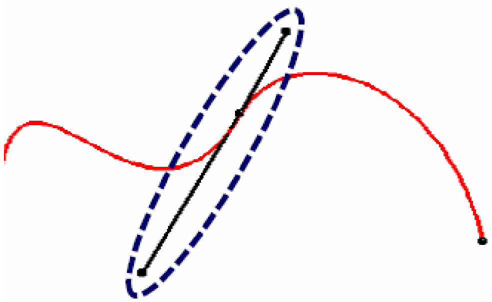
Q_7 ($7 \leq u \leq 8$) determinado por $P_4P_5P_5P_5$. Em u=8 interpola P_5



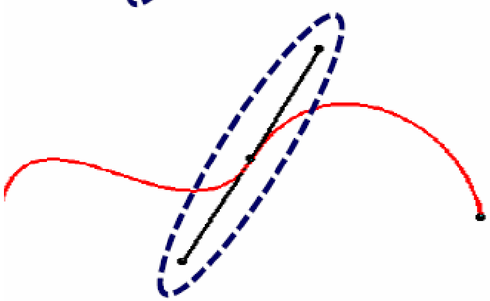
Lembrando o significado de continuidade



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

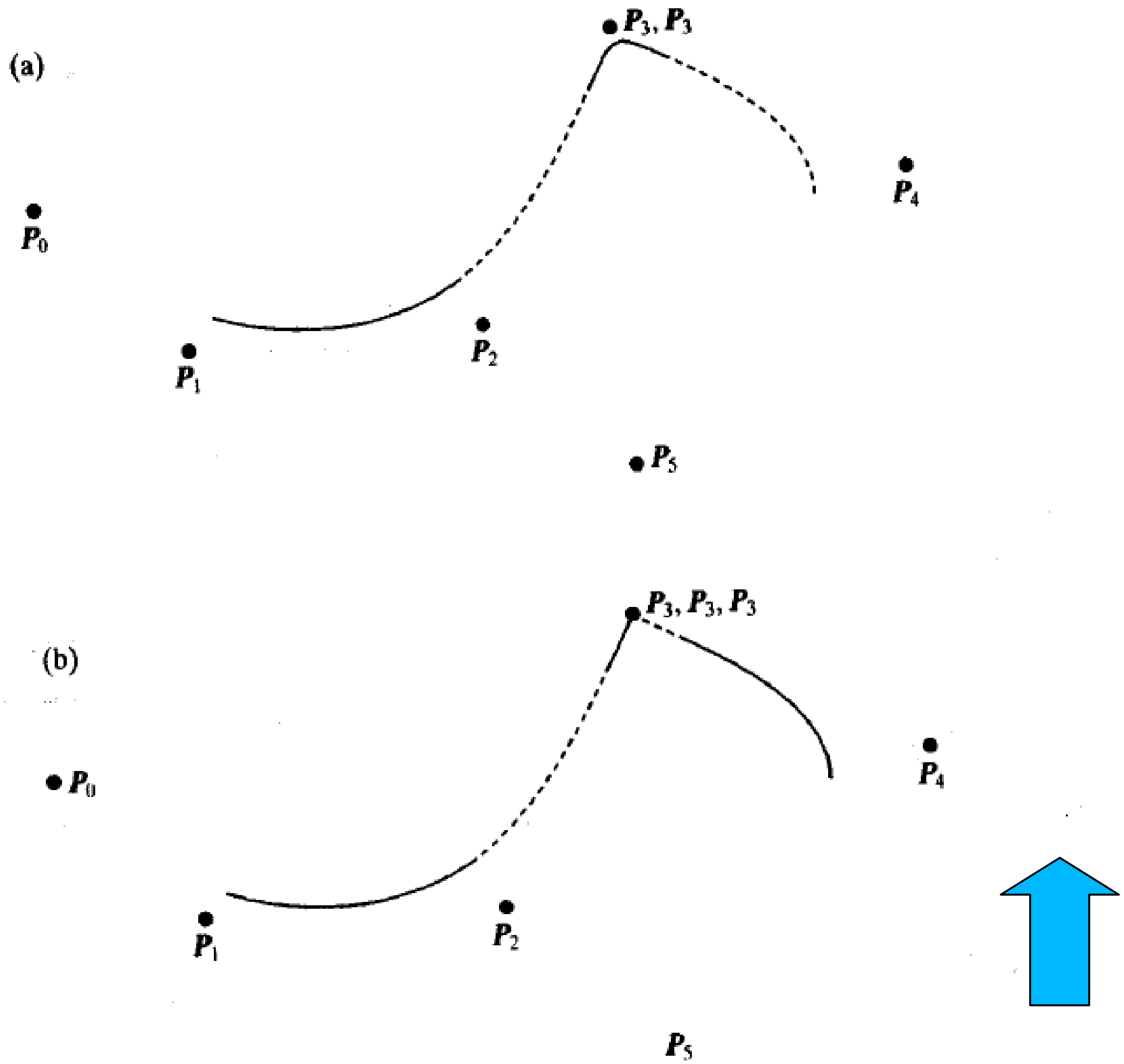


C^1: A direção dos vetores tangentes no ponto de junção é igual.



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

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



Perda de continuidade

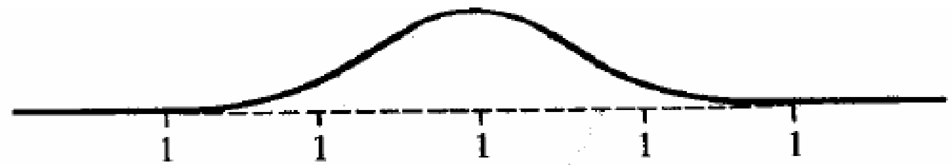
a) ponto duplo -G1

b) Ponto triplo - G0

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

a) Multiplicidade 1:

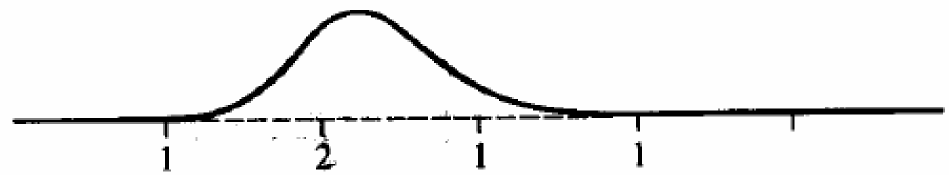
[0, 1, 2, 3, 4]



(a)

b) Multiplicidade 2:

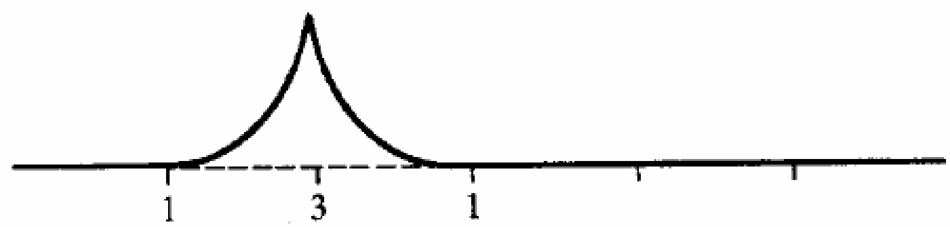
[0, 1, 1, 2, 3]



(b)

c) Multiplicidade 3:

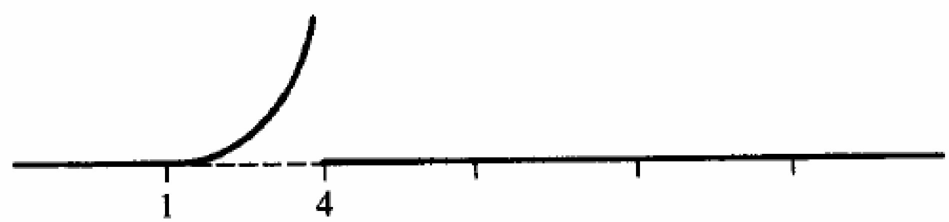
[0, 1, 1, 1, 2]



(c)

d) Multiplicidade 4:

[0, 1, 1, 1, 1]



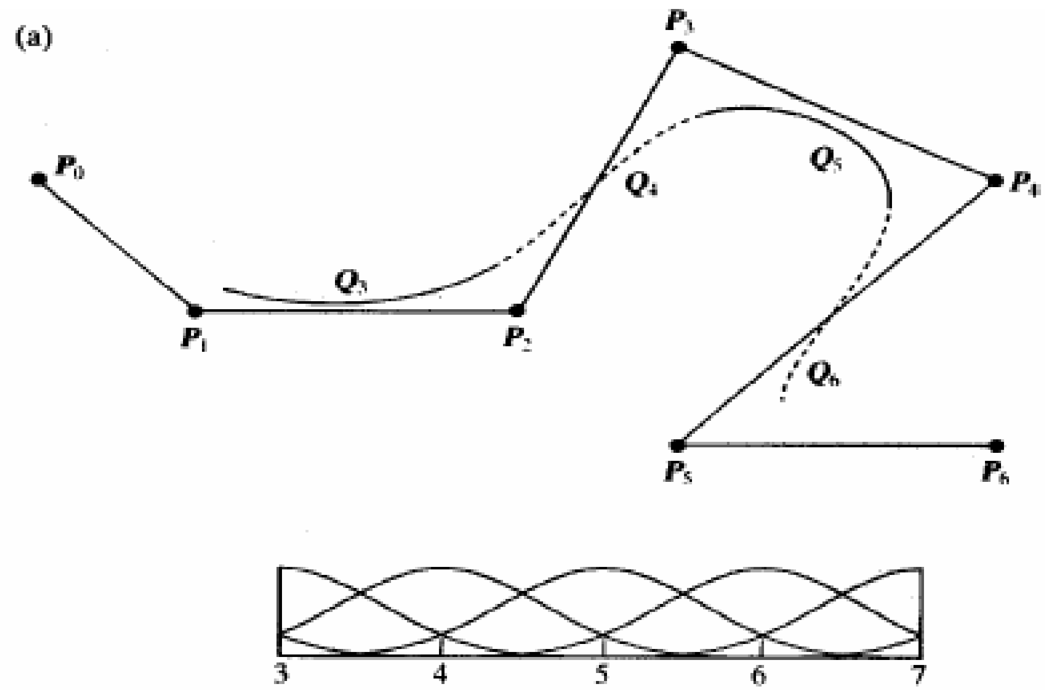
Propriedades

- 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.

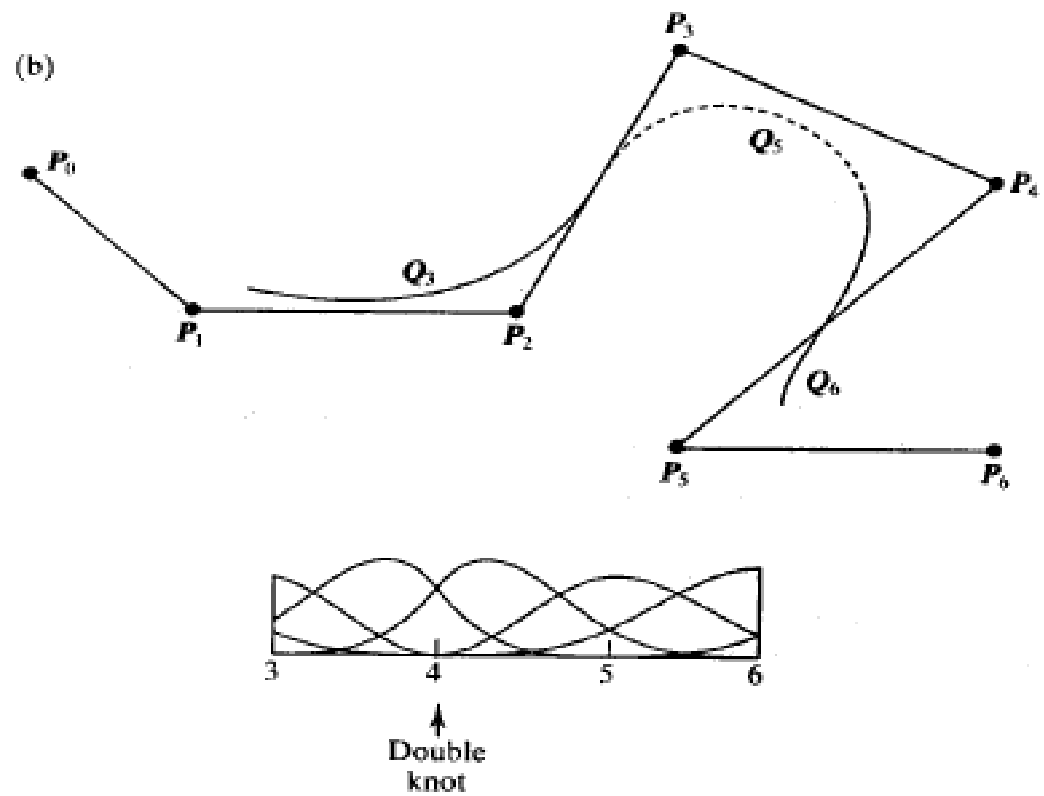
Spline => propriedades

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.

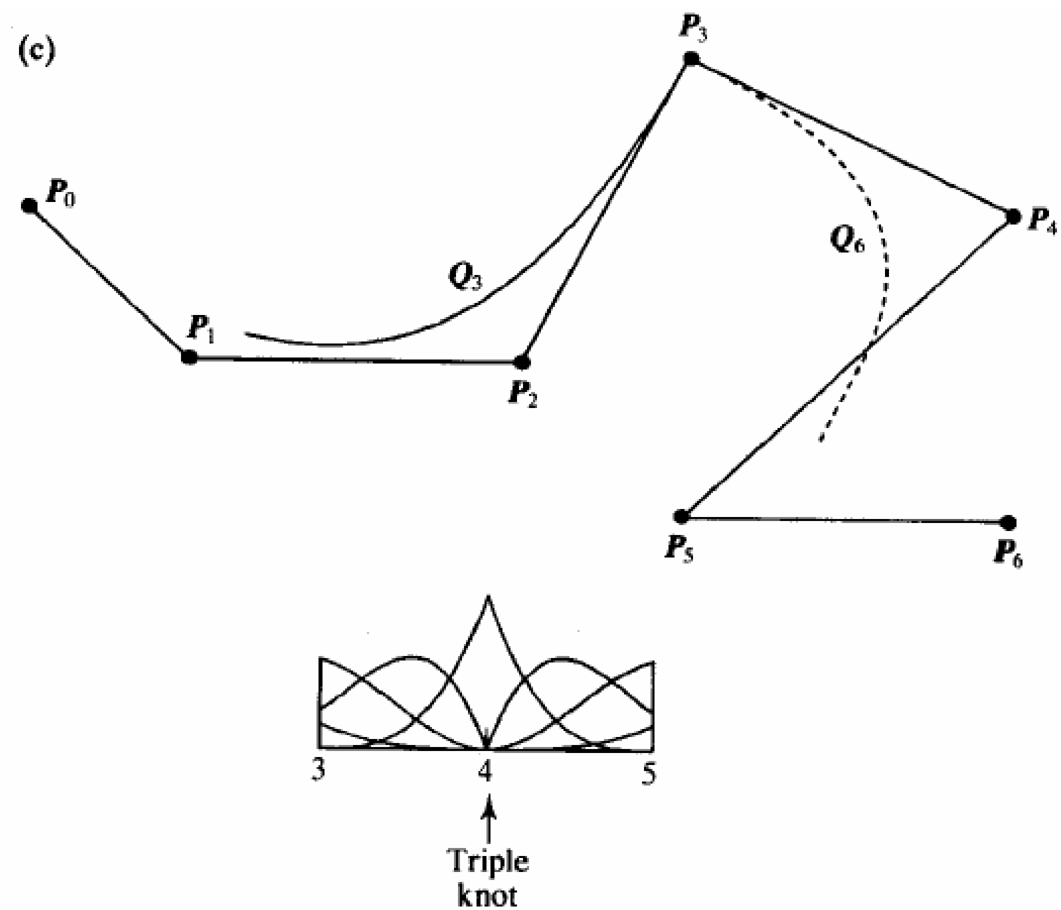


Spline com pontos controle coincidentes seguidos => perda nivel de continuidade



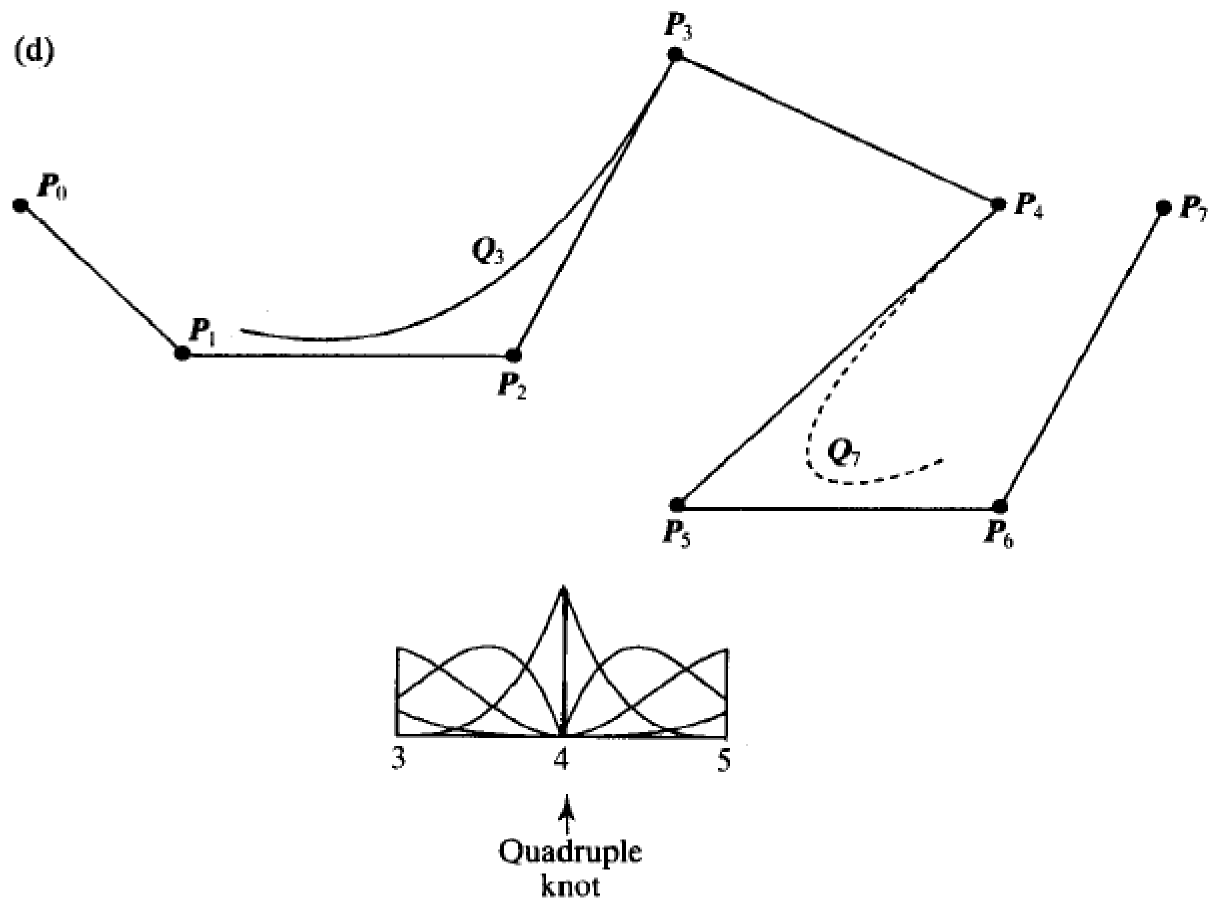
(b) Knot vector is [0,1,2,3,4,4,5,6,7,8,9]. Q_4 shrinks to zero.

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.

B-Splines Não-Uniformes

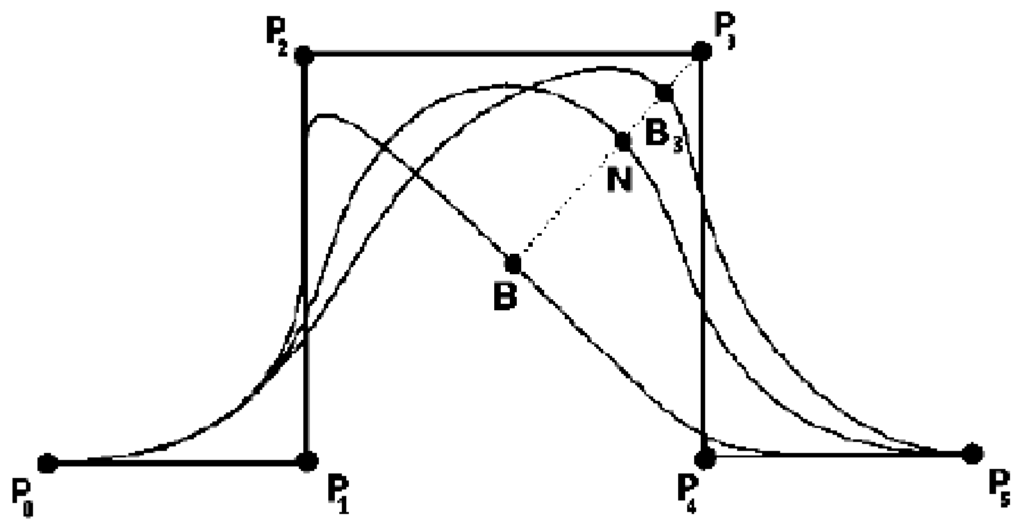
- O intervalo entre valores da variável paramétrica não é necessariamente uniforme…

- 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

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

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 homogéneas.

Curva no espaço homogéneo:

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

Para no Trabalho incluir uma :

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

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

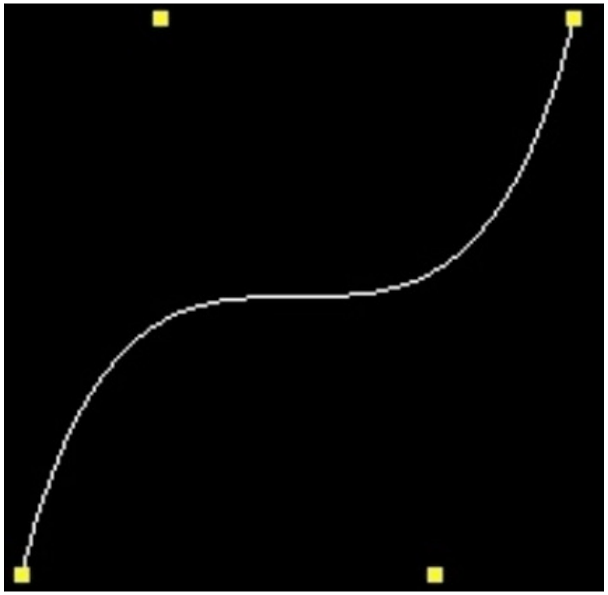
$$\frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{i-1} + \frac{t^3}{6} P_i$$

$$0 <= t <= 1$$

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++;
}
```



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