Computação Gráfica I

Professor:

Anselmo Montenegro
www.ic.uff.br/~anselmo

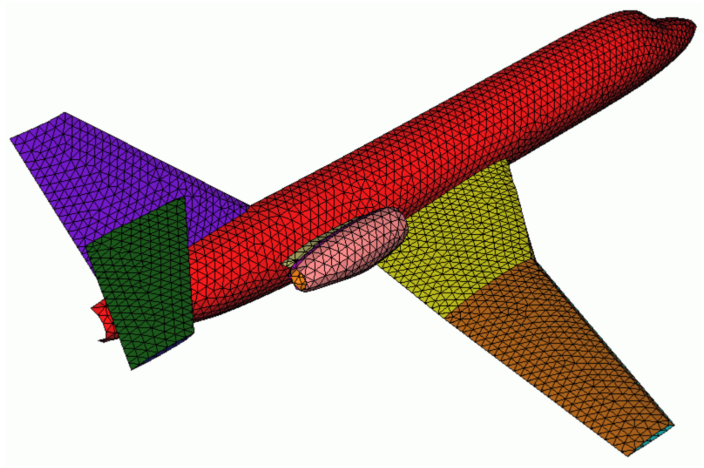
Conteúdo:

- Curvas interativas

Instituto de Computação - UFF

Construção interativa de curvas: *introdução*

- Curvas e superfícies são objetos matemáticos fundamentais na modelagem de objetos gráficos.



- Uma das representações mais utilizadas para curvas e superfícies é a baseada em funções paramétricas.

Construção interativa de curvas: *introdução*

- As formas de parametrização apresentadas anteriormente (ver aulas sobre objetos gráficos), apesar de poderosas, não são apropriados para interação.

- Um caso extremo é o de representações lineares por partes.

- Como vimos anteriormente, representações lineares podem requerer a descrição de um número muito grande de dados.

Construção interativa de curvas: *introdução*

- Considere, por exemplo, um projetista que precisa editar um forma muito complexa, alterando a posição de cada um dos pontos de uma malha densa.

- Esta forma de interação é muito pouco prática e ineficiente.

- Para contornar este problema é necessário *formas de representação mais compactas e que facilitem o processo de criação e modificação*.

<u>Construção interativa de curvas</u>: *curvas interativas*

- Um esquema de representação apropriado para interação é o baseado em tipo especial de representação paramétrica.

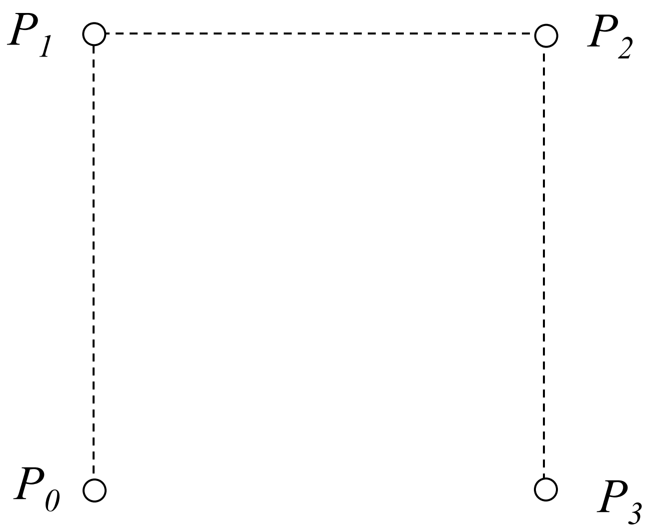
- Tal representação é caracterizada por dois conjuntos:

  - Um conjunto de pontos discretos denominados *pontos de controle*.
  - Um conjunto de *funções de base* ou *funções de mistura* (*blending functions*).

- Neste curso serão abordadas representações para curvas. Representação para superfícies são extensões das técnicas aqui apresentadas.

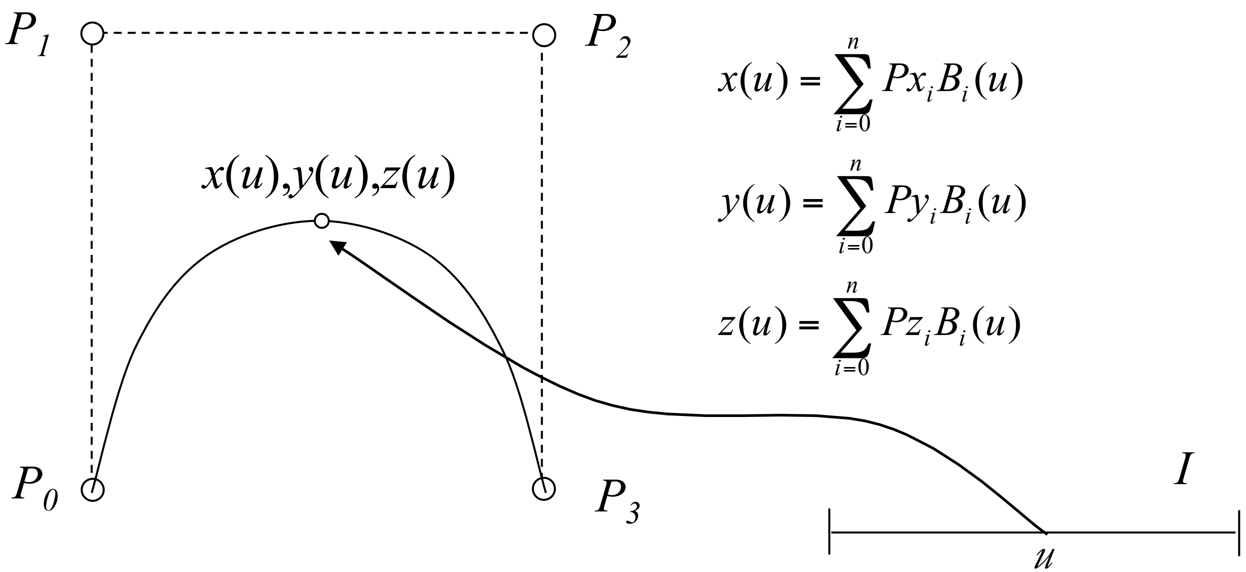
Construção interativa de curvas: *representação de curvas - definição*

- Considere um conjunto de pontos de controle P_0, P_1, P_2, P_3 onde $P_i, i=0,..,3$ e Px_i, Py_i, Pz_i indicam as coordenadas de cada ponto P_i.



Construção interativa de curvas: *representação de curvas - definição*

- Pode-se determinar um curva paramétrica $Q(u)$ através de uma expressão que *pondere* a contribuição das coordenadas de cada ponto de controle, para cada ponto associado a um valor de *u* no espaço de parâmetros.
- A ponderação é feita através da definição de uma função de base para cada ponto de controle.



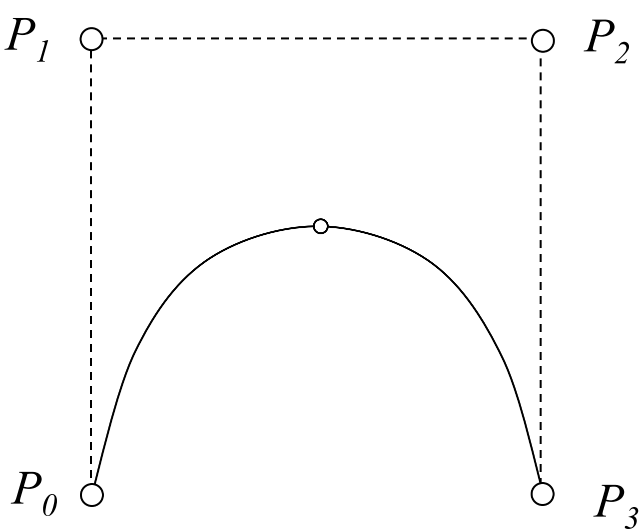
$$x(u) = \sum_{i=0}^{n} Px_i B_i(u)$$

$$y(u) = \sum_{i=0}^{n} Py_i B_i(u)$$

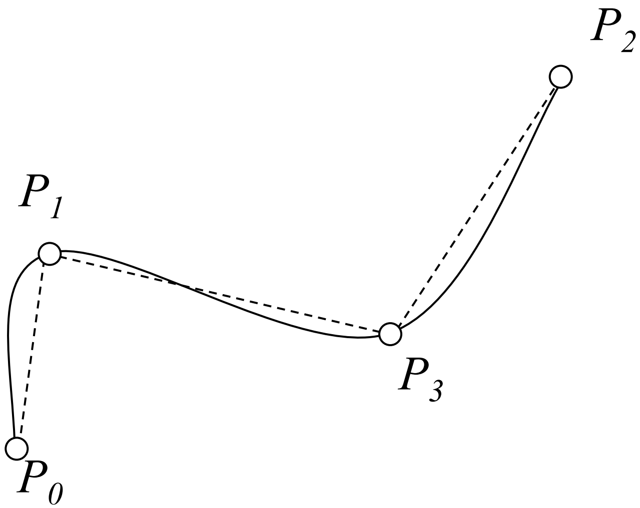
$$z(u) = \sum_{i=0}^{n} Pz_i B_i(u)$$

Construção interativa de curvas: *representação de curvas - propriedades*

- Diferentes funções de base possuem diferentes *propriedades*.
- Uma das propriedades diz respeito ao fato delas gerarem curvas que *interpolam* ou *aproximam* os pontos de controle.
- Outras propriedades estão relacionadas a *continuidade* e *localidade* da curva.



Curva interpolando os extremos e aproximando os pontos interiores do polígono de controle



Curva interpolante

Construção interativa de curvas: *propriedade de continuidade de curvas*

- Quando dois segmentos de curvas são juntados, a continuidade da curva é alterada dependendo da forma de junção:
  - Continuidade geométrica G^0 - dois segmentos são colados em um ponto comum.
  - Continuidade geométrica G^1- os vetores tangentes aos segmentos no ponto de contato tem *direções iguais*, mas magnitudes diferentes.
  - Continuidade C^1- a *direção e magnitude* dos vetores tangentes são iguais.
  - Continuidade C^n – a direção e magnitude das n-ésimas derivadas dos segmentos $\frac{d^n}{dt}\left[Q(u)\right]$ no ponto de contato dão iguais.

Construção interativa de curvas: *curvas cúbicas*

- Em Computação Gráfica é comum a utilização de *curvas cúbicas* pelos seguintes motivos:

  - São *poderosas* o suficiente para representar as mais diversas formas.

  - São *curvas verdadeiramente espaciais* e não planares (curvas quadráticas são determinadas por 3 pontos de controle e estão restritas a um plano no espaço).

  - Curvas de grau mais elevado introduzem custos e dificuldades adicionais.

Construção interativa de curvas: *curvas cúbicas*

- Um curva cúbica espacial pode ser descrita da seguinte forma:

$$x(u) = a_x u^3 + b_x u^2 + c_x u + dx$$

$$x(u) = a_y u^3 + b_y u^2 + c_y u + dx$$

$$z(u) = a_z u^3 + b_z u^2 + c_z u + dx$$

$$\text{onde } 0 \leq u \leq 1$$

- $x(u)$,$y(u)$ e $z(u)$ são polinômios cúbicos no parâmetro *u*.

Construção interativa de curvas: *Curvas de Bérzier*

- Uma *curva de Bérzier* é uma parametrização que utiliza a base de *Bérzier*, também denominadas funções de mistura de *Bernstein*.

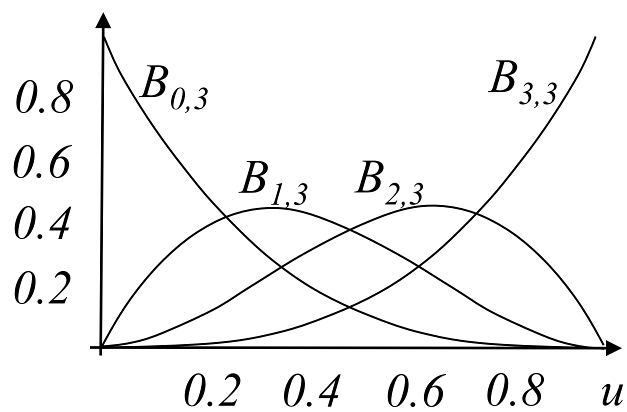
- A base de *Bérzier* de grau 3 é dada pelos seguintes termos:

$$B_{0,3} = (1-u)^3$$

$$B_{1,3} = 3u(1-u)^2$$

$$B_{2,3} = 3u^2(1-u)$$

$$B_{3,3} = u^3$$



Construção interativa de curvas: *Curvas de Bérzier – forma geral*

- O conjunto de funções de mistura para um polinômio cúbico é um caso particular das funções de base de grau *n*:

$$B_{i,n}(u) = C(n,i)u^i(1-u)^{n-i}$$

- De fato, uma curva de Bérzier de grau *n* tem a seguinte forma:

$$Q(u) = \sum_{i=0}^{n} P_i B_{i,n}(u)$$

- Note que para um curva de grau *n* são necessários *n+1* pontos de controle.

Construção interativa de curvas: *Curvas de Bérzier*

- O polinômio descrito pela base de Bérzier cúbica e quatro pontos de controle é dado por:

$$Q(u) = P_0(1-u)^3 + P_1 3u(1-u) + P_2 3u^2(1-u) + P_3 u^3$$

- Ou em forma matricial

$$Q(u) = UBP$$

$$= \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

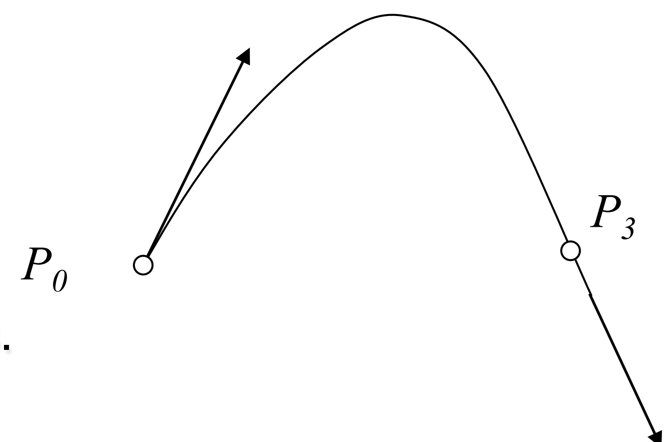
Construção interativa de curvas: *Curvas de Bérzier*

- A movimentação dos pontos de controle P_0, P_1, P_2 *e* P_3 altera a forma da curva.
- P_0 e P_3 são os extremos da curva.

- Derivando-se $Q(u)$ com relação a u é possível mostrar que

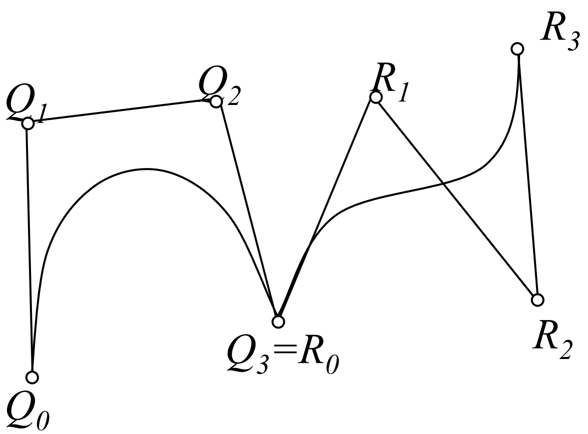
$$Q_u(0) = 3(P_1 - P_0)$$
$$Q_u(1) = 3(P_2 - P_3)$$

- $Q_u(0)$ e $Q_u(1)$ dão os vetores tangentes nos pontos extremos e P_1 e P_2 estão sobre eles.

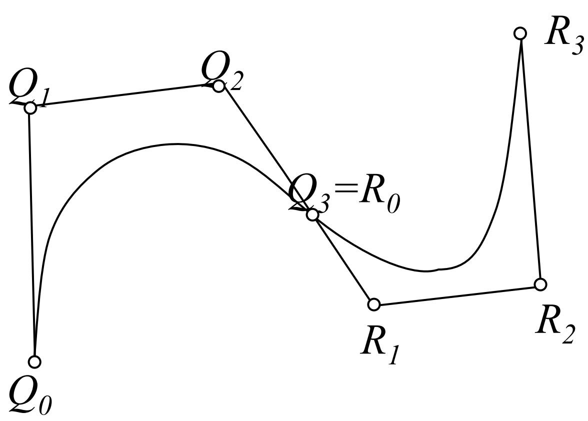


<u>Construção interativa de curvas</u>: *como juntar segmentos de curva de Bérzier*

- Segmentos de curva definidos por um conjunto de quatro pontos de controle podem ser conectados para gerar formas mais complexas.
- O resultado é uma *curva polinomial por partes*.
- A conexão requer restrições nos pontos de junção.



Continuidade G_0 - *o extremo do primeiro segmento coincide com o inicio do segundo*



Continuidade G_1 - *as arestas dos polígonos de controle no ponto de junção são colineares. Os vetores tangentes são iguais a menos de uma constante, isto é* $\boldsymbol{(Q_3-Q_2)=k(R_1-R_0)}$

Construção interativa de curvas: *como juntar segmentos de curva de Bérzier*

- Uma alternativa é aumentar o grau do polinômio.

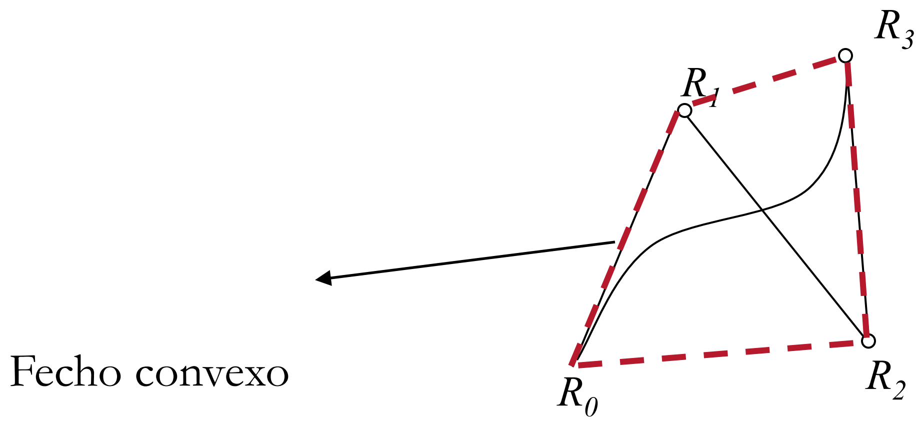
- Isto introduz uma maior complexidade matemática e computacional.

- É bem mais simples particionar uma curva em segmentos cúbicos.

<u>Construção interativa de curvas</u>: *sumário das propriedades da curva de Bézier*

- O grau do polinômio é um a menos que o número de pontos de controle. Em geral utilizamos polinômios de grau 3.

- A curva segue a forma do polígono de controle e está contida no *fecho convexo* de tal polígono. O fecho convexo pode ser entendido como a forma obtida passando-se uma fita elástica em torno dos pontos.



Construção interativa de curvas: *sumário das propriedades da curva de Bézier*

- A curva se restringe ao fecho convexo uma vez que as funções de base somam 1 (um) para todo valor de *u*.

- *Os pontos de controle não exercem controle local*. Mover um ponto de controle move toda a curva (As funções de base são diferentes de 0 em todo o domínio exceto em $u=0$ e $u=1$).

Construção interativa de curvas: *sumário das propriedades da curva de Bézier*

- Os vetores tangentes à curva nos pontos extremos coincidem com a primeira e última aresta do polígono de controle.

- A curva não oscila sobre nenhuma reta mais do que oscila o polígono de controle (*propriedade de minimização de variação*).

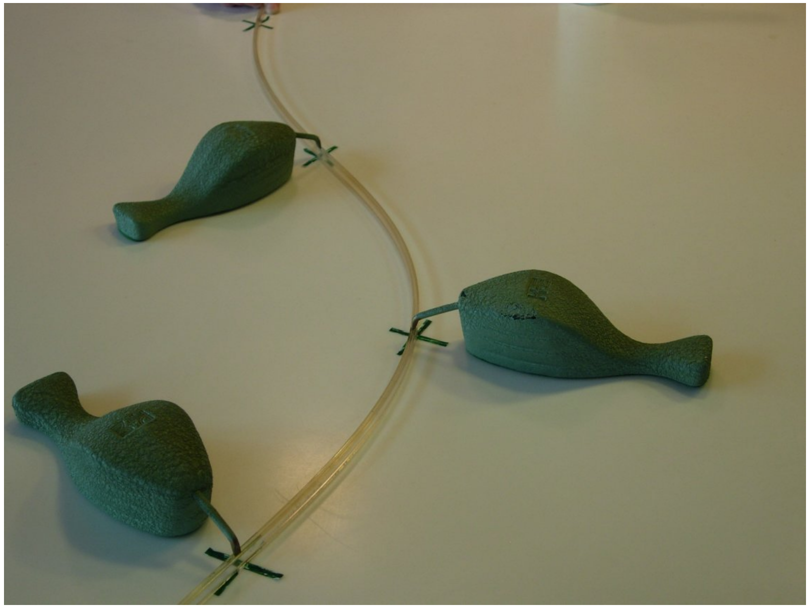
- A curva pode ser transformada por *transformações afins* (translações e rotações) definidas sobre os pontos de controle.

Construção interativa de curvas: *B-Splines*

- Curvas de Bézier possuem duas grandes desvantagens:
  - O controle exercido pelos pontos de *controle não é local*. A movimentação de um ponto altera toda curva, apesar de sua influência ser maior na vizinhança de tal ponto.
  - Não é possível definir uma curva de Bérzier cúbica para aproximar ou representar um conjunto de ***n*** pontos sem utilizar *múltiplos segmentos de curva*.
- Uma representação que não possui tais deficiências é a baseada em curvas denominadas *B-splines*.

Construção interativa de curvas: *B-Splines*

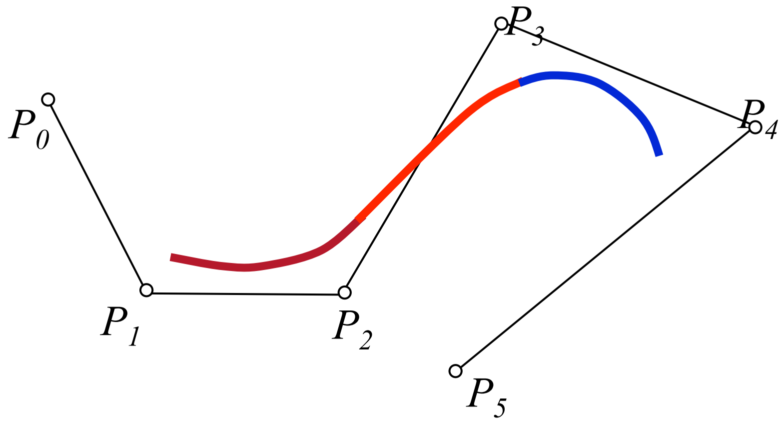
- Originalmente, uma *spline* é uma ferramenta de desenho.
- Consiste em uma tira de metal flexível usada para desenhar curvas fixando-se *pesos* a pontos denominados *nós*.



Construção interativa de curvas: *B-Splines*

- Assim como as curvas de Bézier, *B-Splines* não passam pelos pontos de controle.

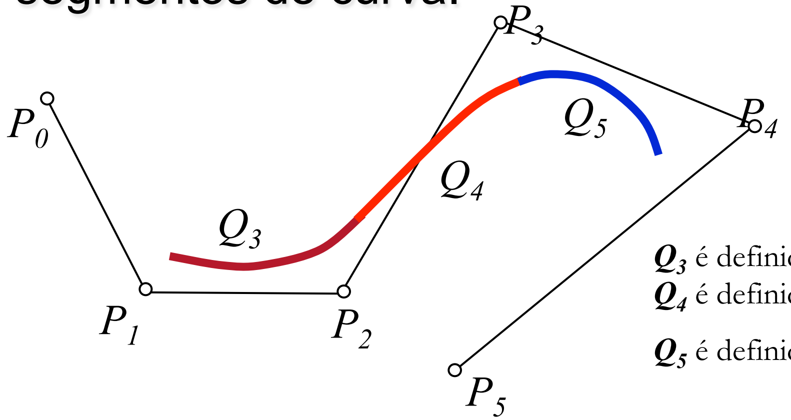
- Uma *B-Spline* é uma curva completa polinomial por partes consistindo de um conjunto de segmentos.



Construção interativa de curvas: *B-Splines cúbicas*

- Uma *B-spline* é composta por uma série de m-2 segmentos de curva $Q_3, Q_4, \ldots, Q_m$ definidos por $m+1$ pontos de controle $P_0, P_1, \ldots, P_m$, $m \geq 3$.

- Cada segmento de curva é definido por quatro pontos de controle e 4 funções de mistura. Cada ponto de controle influencia somente 4 segmentos de curva.



$\boldsymbol{Q_3}$ é definido por $\boldsymbol{P_0P_1P_2P_3}$ e ponderado por $\boldsymbol{B_0B_1B_2B_3}$
$\boldsymbol{Q_4}$ é definido por $P_1P_2P_3P_4$ e ponderado por $\boldsymbol{B_0B_1B_2B_3}$
$\boldsymbol{Q_5}$ é definido por $\boldsymbol{P_2P_3P_4P_5}$ e ponderado por $\boldsymbol{B_0B_1B_2B_3}$

- Para facilitar a notação iremos abordar apenas *B-Splines* cúbicas, apesar de existirem *B-Splines* de qualquer grau.

Construção interativa de curvas: *B-Splines cúbicas*

- A formulação de um segmento $Q_i(u)$ *B-spline cúbica* é a seguinte:

$$Q_i(u) = UB_sP = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$

$$Q_i(u) = \sum_{k=0}^{3} P_{i-3+k} B_{i-3+k}(u)$$

- Onde i é o número do segmento e k é o índice do ponto de controle correspondente ao segmento i.

- O valor de u para um único segmento está contido no intervalo $0 \leq u \leq 1$. Isto é, u é um parâmetro local que varia entre 0 e 1 para definir um único segmento.

Construção interativa de curvas: *B-Splines cúbicas*

- É possível definir todo o conjunto de segmentos através da expressão:

$$Q(u) = \sum_{k=0}^{m} P_i B_i(u)$$

- Nesse caso i é um índice global para os pontos de controle, assim como u é um parâmetro global.

Construção interativa de curvas: *B-Splines cúbicas*

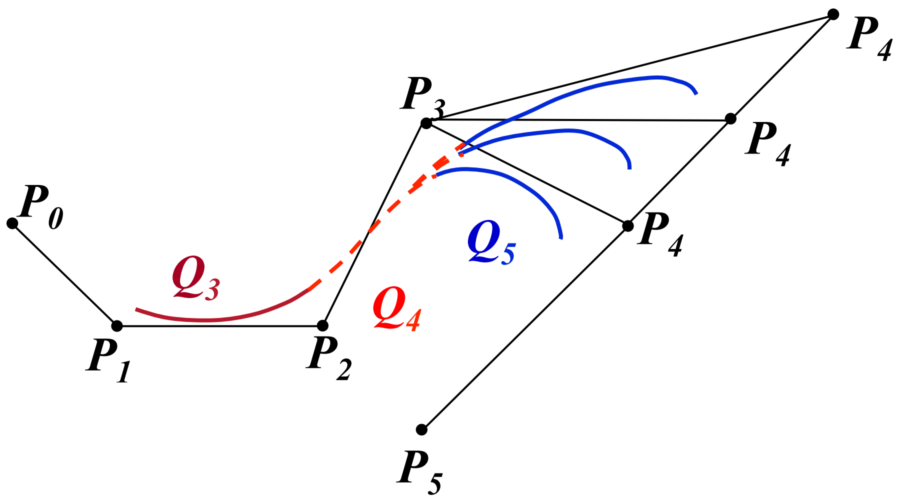
- Um conjunto de múltiplos segmentos de *Bérzier* também admite controle local ao custo da introdução de restrições para continuidade.

- *B-splines* não necessitam de restrições adicionais para controle local.

Construção interativa de curvas: *B-Splines cúbicas*

- A movimentação de um ponto de controle de uma *B-spline* afeta, no pior caso, somente um conjunto de 4 segmentos.

- Na figura abaixo a movimentação de P_4 afeta somente Q_4 e Q_5 mantendo Q_3 intacto.



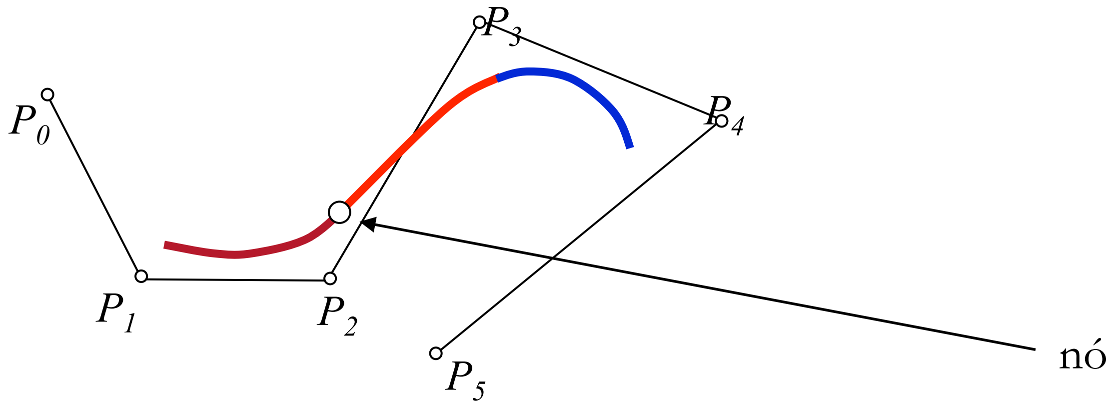
Construção interativa de curvas: *B-Splines cúbicas - continuidade*

- Um B-Spline exibe continuidade C^0(posicional), C^1 (da primeira derivada) e C^2 (da segunda derivada).

- O mecanismo que garante a continuidade C^2 está no compartilhamento dos pontos de controle por segmentos de curva adjacentes.

Construção interativa de curvas: *B-Splines cúbicas uniformes*

- Os pontos associados a junções de dois segmentos adjacentes são denominados *nós*.



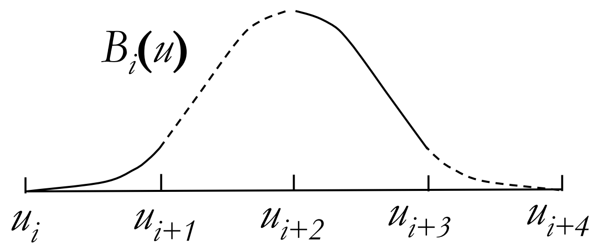
- O valor de u correspondente a um nó é denominado *valor do nó.*

- Uma *B-spline* é dita uniforme se valores dos nós são igualmente espaçados no espaço do parâmetro *u*.

Construção interativa de curvas: *B-Splines cúbicas uniformes – as funções de base*

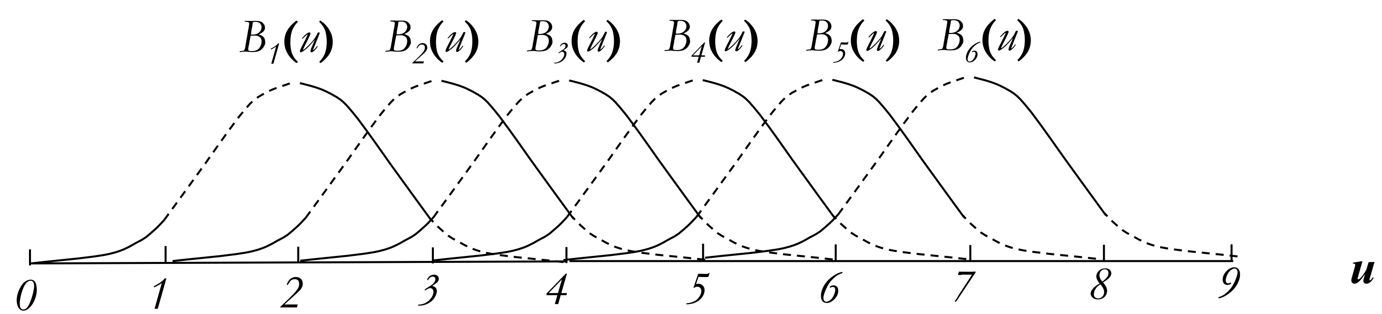
- Cada função da base de funções de uma *B-spline* é por si própria uma uma função cúbica composta de 4 segmentos.

- Uma função da base é diferente de zero em 4 intervalos consecutivos no espaço de parâmetros.



Construção interativa de curvas: *B-Splines cúbicas uniformes – as funções de base*

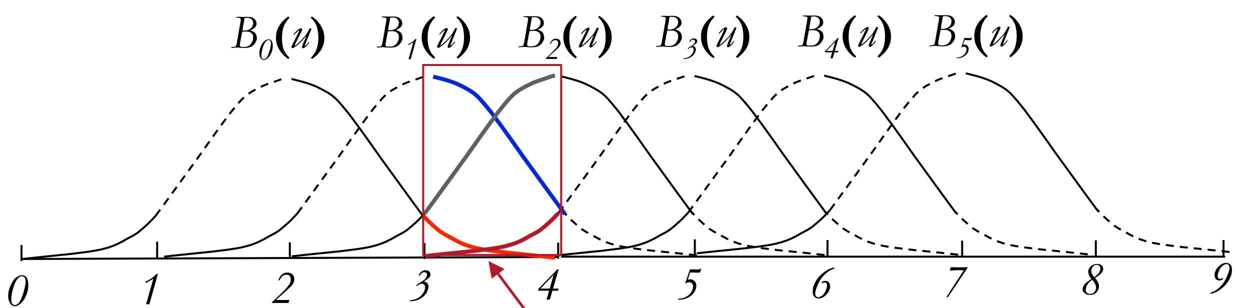
- No caso de *B-splines* uniformes os intervalos são de mesmo comprimento.
- Isto significa que os valores dos nós são *igualmente espaçados* e as funções da base são *cópias transladadas*.
- *Uma B-spline com m+1 pontos de controle possui m+5 nós*



Funções de base de uma *B-spline* uniforme com 6 pontos de controle

Construção interativa de curvas: *B-Splines cúbicas uniformes – as funções de base*

- Um segmento de curva está definido em um intervalo de u_i a u_{i+1}.



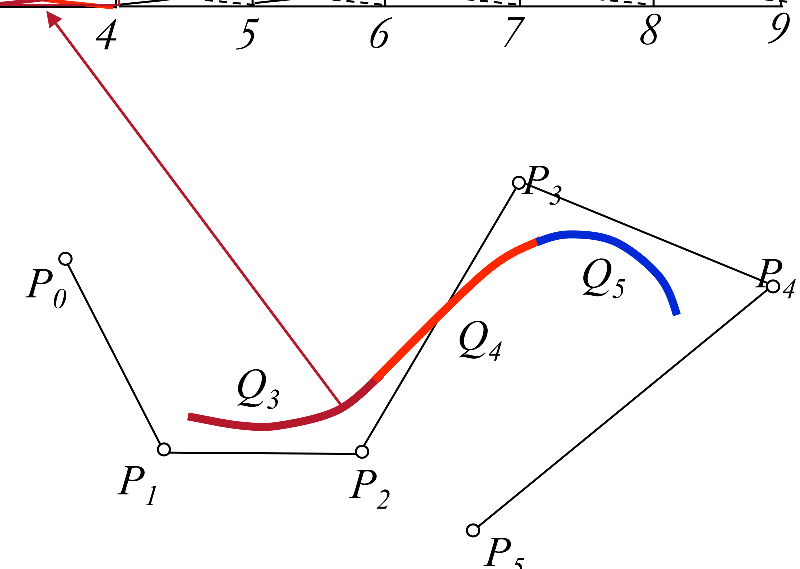
- De fato, avaliamos 4 bases no intervalo $u_i \leq u \leq u_{i+1}$

$$B_i = 1/6u^3$$

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

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

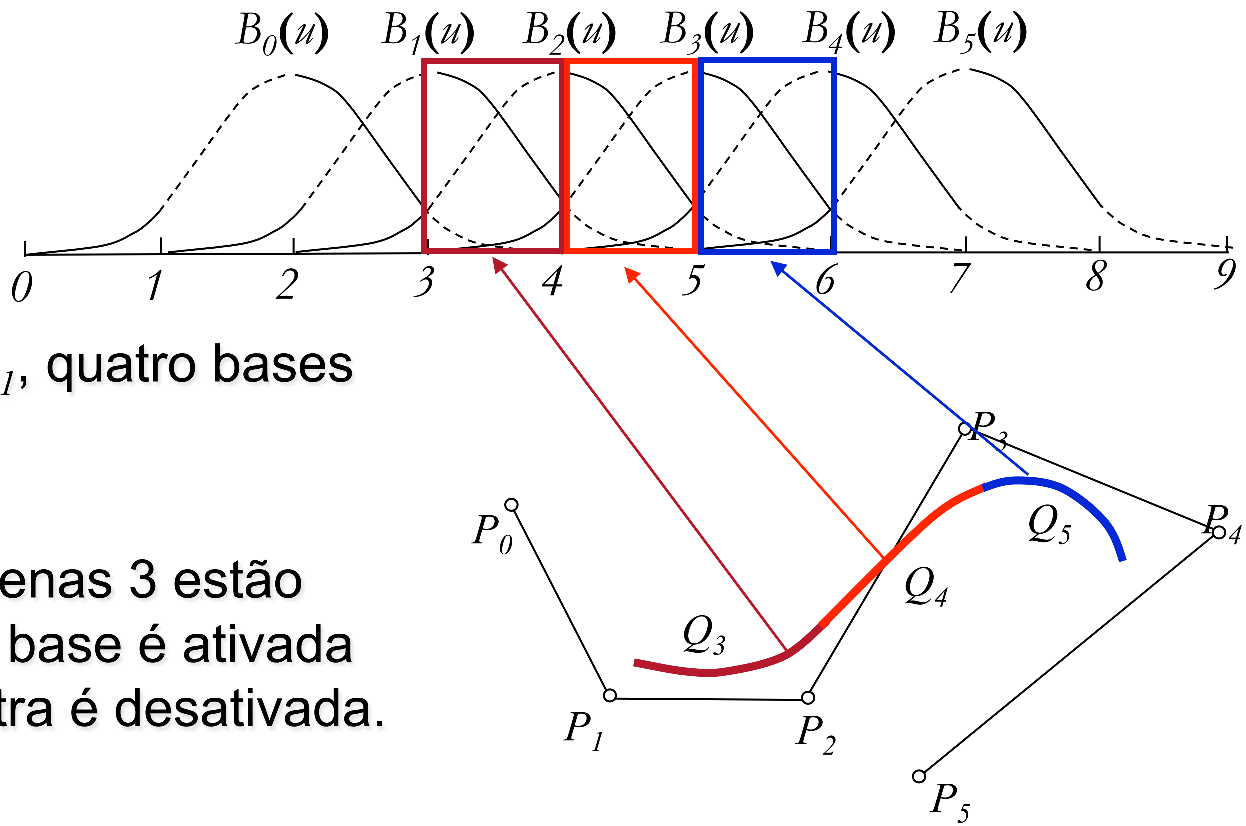
$$B_{i-3} = 1/6(1-u)^3$$



- Obs.: as equações acima não definem uma base e sim os pedaços das bases que definem um segmento da curva

Construção interativa de curvas: *B-Splines cúbicas uniformes – as funções de base*

- No exemplo abaixo, a curva soma a unidade no intervalo $3 \leq u \leq 6$

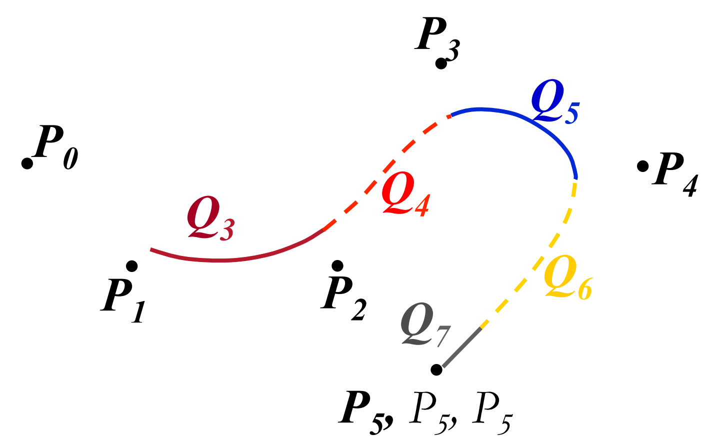


- Para $u_i < u < u_{i+1}$, quatro bases estão ativas.

- Para $u = u_i$ apenas 3 estão ativas, e uma base é ativada assim que outra é desativada.

Construção interativa de curvas: *B-Splines cúbicas uniformes – interpolando pontos*

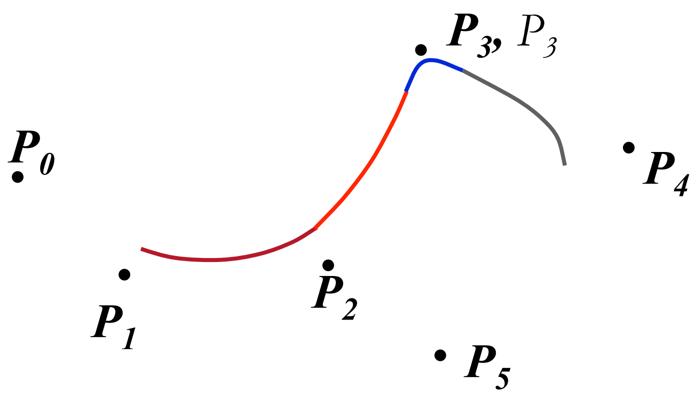
- No caso geral, uma B-spline *não interpola pontos de controle*.
- É possível fazer com que uma B-spline interpole um ponto de controle através da replicação de vértices.
- Intuitivamente, replicar um vértice faz com que a curva seja atraída para tal ponto de controle, já que seu peso na ponderação é maior.

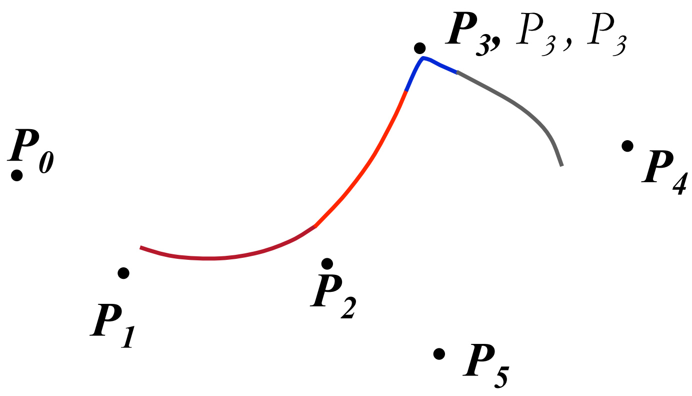


Construção interativa de curvas: *B-Splines cúbicas uniformes – interpolando pontos*

- O preço pago pelo uso da replicação de pontos é a perda da continuidade C^2 da curva.

- Exemplo da replicação em pontos de controle interiores:





Construção interativa de curvas: *B-Splines cúbicas não-uniformes*

- Um B-Spline é não-uniforme quando os intervalos paramétricos entre valores de nós sucessivos não são necessariamente iguais.

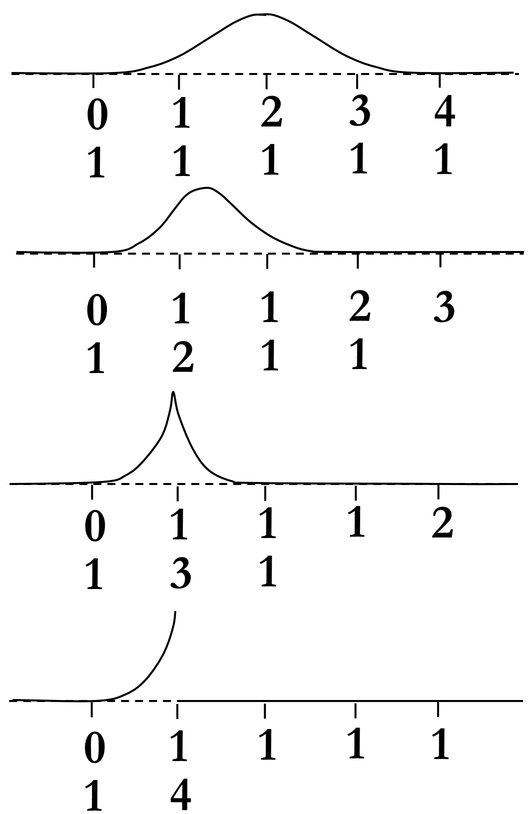
- Isto significa que as função da base não são translações umas das outras. São completamente distintas.

- O caso mais comum é aquele em que alguns intervalos tem comprimento zero.

- Isto é feito através da inserção de nós múltiplos.

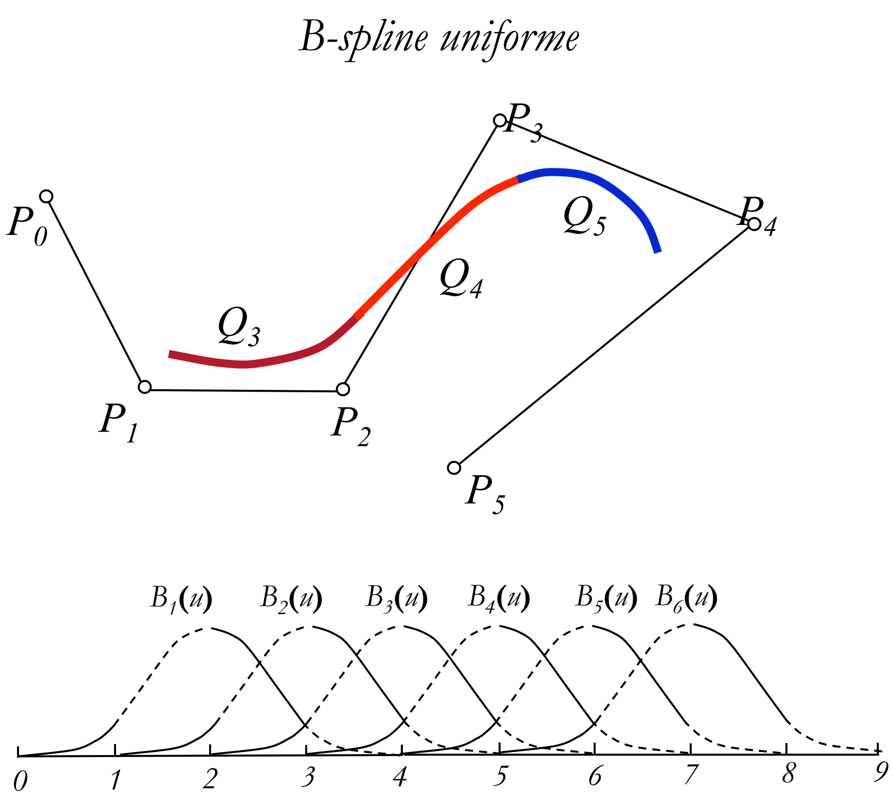
Construção interativa de curvas: *B-Splines cúbicas não-uniformes*

- O número de valores de nós idênticos é denominado *multiplicidade* do nó.
- Efeito da multiplicidade de nós em uma base B-spline cúbica.

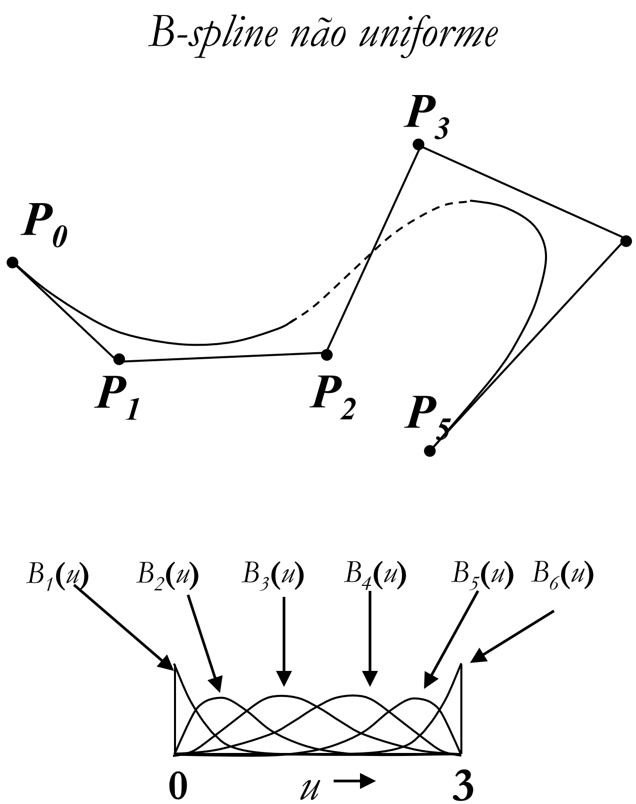


Construção interativa de curvas: *B-Splines cúbicas não-uniformes*

- Comparação de uma B-spline uniforme e uma não uniforme:



Nós:[0,1,2,3,4,5,6,7,8,9,10]



Nós:[0,0,0,0,1,2,3,3,3,3,3]

Construção interativa de curvas: *B-Splines - sumário*

- Sumário das propriedades de uma curva *B-spline*:
  - *Segue a forma do polígono de controle* e é restrita a permanecer no fecho convexo dos pontos de controle.
  - Exibe propriedade de *minimização de variação*.
  - Pode ser transformada através de uma transformação afim (translação + rotação) aplicando-se a transformação sobre os pontos de controle.
  - Exibe *controle local*.

Construção interativa de curvas: *NURBS*

- *NURBS* são *B-Splines* não uniformes dadas pela razão de de dois polinômios.

$$\frac{\sum_{i=0}^{n} P_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

Ordem da curva. A ordem é o grau do polinômio + 1.

- Os valores w_i associados a cada ponto de controle são pesos que podem ser vistos como parâmetros extras.

- Os w_i afetam a curva apenas localmente. A curva é atraída para um ponto P_i se o w_i correspondente aumenta e é afastada de P_i se w_i diminui.

- Os w_i podem ser compreendidos como parâmetros de acomplamento da curva aos pontos de controle.

Construção interativa de curvas: *Curvas e OpenGL*

- Como determinar as equações para *B-splines* não uniformes?
- Funções dependem dos intervalos entre os nós.

- Equações de *recorrência* pra *B-splines* cúbicas.

$$B_{i,1}(u) = \begin{cases} 1, u_i \leq u < u_{i+1} \\ 0, \text{caso contrário} \end{cases}$$

$$B_{i,2}(u) = \frac{u - u_i}{u_{i+1} - u_i} B_{i,1}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} B_{i+1,1}(u)$$

$$B_{i,3}(u) = \frac{u - u_i}{u_{i+2} - u_i} B_{i,2}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} B_{i+1,2}(u)$$

$$B_{i,4}(u) = \frac{u - u_i}{u_{i+3} - u_i} B_{i,3}(u) + \frac{u_{i+4} - u}{u_{i+4} - u_{i+1}} B_{i+1,3}(u)$$

Construção interativa de curvas: *Curvas e OpenGL*

- Equações de *recorrência* geral para *B-splines*

$$B_{i,1}(u) = \begin{cases} 1, u_i \leq u < u_{i+1} \\ 0, \text{caso contrário} \end{cases}$$

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

Construção interativa de curvas: *Curvas e OpenGL*

- OpenGL suporta curvas e superfícies através de mecanismos denominados *evaluators* (avaliadores).

- Os *evaluators* *calculam valores intermediários para os polinômios* que descrevem curvas e superfícies, dada uma seqüência de pontos de controle.

- Os *evaluators* da OpenGL são *avaliadores de Bézier*.

- *Evaluators* criam splines e superfícies baseadas em bases de (Bernstein-Bézier).

Construção interativa de curvas: *Curvas e OpenGL*

- Para utilizar um avaliador Bézier emprega-se a seguinte função:

```
void glMap1f(GLenum target, GLfloat u1, GLfloat u2, Glint
        stride, Glint order, const GLfloat *
    points);
```

Parâmetro	Significado
Target	Significado dos pontos de controle
u_1 e u_2	Intervalo para a variável de controle *u*
Stride	Quantos valores *float* existem entre cada elemento do vetor
Order	Quantidade de elementos no vetor de pontos de controle
Points	Apontador para primeira coordenada do primeiro ponto de controle

Construção interativa de curvas: *Curvas e OpenGL*

- Os avaliadores servem não somente para gerar curvas de coordenadas, mas também de cores, vetores normais, coordenadas de textura(assunto a ser visto futuramente), etc.

Constante	Significado
GL_MAP1_VERTEX_3	Coordenadas x,y e z
GL_MAP1_VERTEX_4	Coordenadas x,y,z e w (homogêneas)
GL_MAP1_INDEX	Índice de cor
GL_MAP1_COLOR_4	Cor com RGBA
GL_MAP1_NORMAL	Vetor normal
GL_MAP1_TEXTURE_COORD_1	Coordenadas de textura s.
GL_MAP1_TEXTURE_COORD_2	Coordenadas de textura s e t.
GL_MAP1_TEXTURE_COORD_3	Coordenadas de textura s,t e r.
GL_MAP1_TEXTURE_COORD_4	Coordenadas de textura s,t,r e q.

- Uma vez definido um avaliador, deve-se habilitá-lo através da função *glEnable* sobre alguma das constantes da tabela acima.

Construção interativa de curvas: *Curvas e OpenGL*

- A utilização da constante GL_MAP1_VERTEX indica que queremos apenas a *posição* de cada ponto intermediário.

- O procedimento de desenho é simples: basta percorrer os pontos intermediários desejados e *solicitar a avaliação da curva Bézier* por meio da seguinte função:

```
glEvalCoord1f(GLfloat u)
```

- O valor *u* indica o valor a ser passado para a Bézier normalmente 0(início) da curva e 1(final).

<u>Construção interativa de curvas</u>: *Curvas e OpenGL*

- Exemplo

```
#ifdef WIN32
#include <windows.h>   /* Inclui header padrão do Windows   */
#endif

#include <GL/gl.h>     /* Inclui header da biblioteca gl */
#include <GL/glut.h>   /* Inclui header da biblioteca glut  */

#define TOTAL 5

int prec = 10;  /* Total de pontos intermediários */

float pontos[5][3] = {{0.0,0.0,0.0},{0.3,0.8,0.0},{0.7,0.8,0.0},{1.0,0.0,0.0},{0.5,0.2,0.0}}; /* Pontos de controle */

/* Configura estados e parâmetros da OpenGL */
void Init(void)
{
 glClearColor (0.0, 0.0, 0.0, 0.0);   /* Seleciona a cor negra como cor de fundo                */

glMap1f(GL_MAP1_VERTEX_3,0.0,1.0,3,TOTAL,&pontos[0][0]); /* Define significado dos pontos de controle */

glEnable(GL_MAP1_VERTEX_3); /* Ativa geração de coordenadas */
}
```

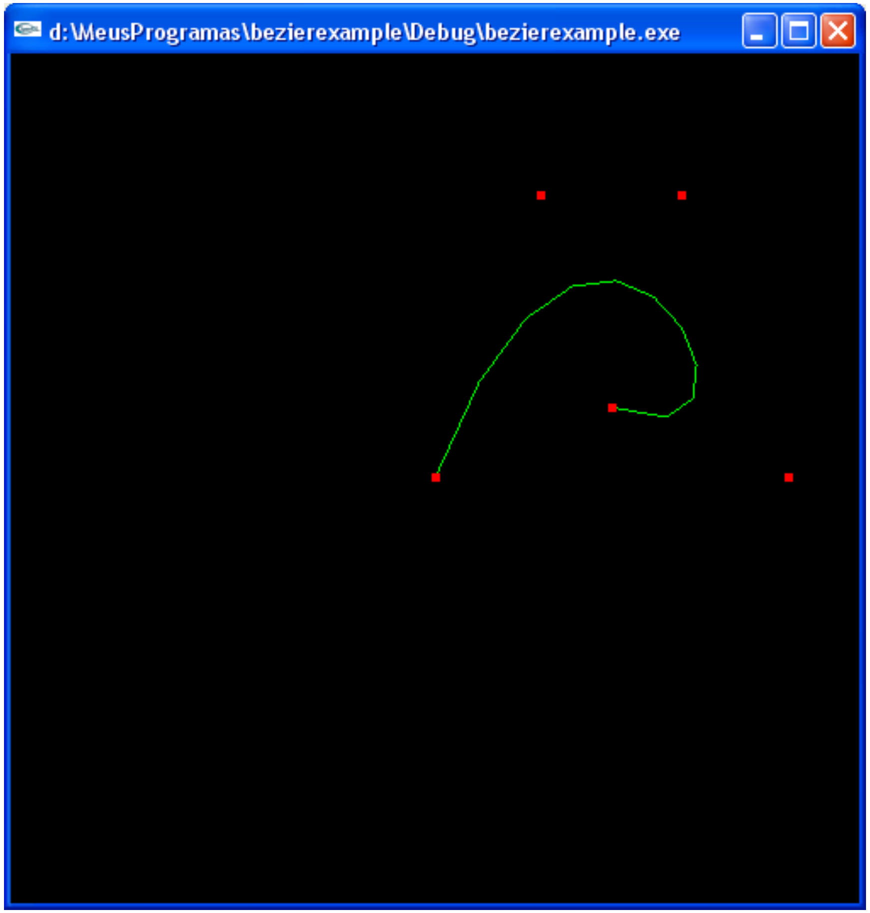
Construção interativa de curvas: *Curvas e OpenGL*

- Exemplo (continuação)

```
/* Função de desenho */
void Display (void)
{
 float delta =1.0/(float)prec;

  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); /* Limpa a tela e o buffer de profundidades */
  glColor3f(0,1,0); /* Especifica a cor vermelha */

  glBegin(GL_LINE_STRIP);  /* Desenha a curva */
    for (float u=0;u<=1.01;u+=delta)
                        glEvalCoord1f(u); /* invoca o avaliador para o parâmetro f */
  glEnd();

  glColor3f(1,0,0);  /* Especifica a cor vermelha */
  glPointSize(5); /* Especifica o tamanho do ponto */

  /* Desenha os pontos de controle */
  glBegin(GL_POINTS);
    for (int i=0;i<TOTAL;i++)
       glVertex3fv(pontos[i]);
  glEnd();

  glutSwapBuffers ( );/* Troca os buffers */}
```

Construção interativa de curvas: *Curvas e OpenGL*

- Resultados: poucos pontos intermediários



<u>Construção interativa de curvas</u>: *Curvas e OpenGL*

- Resultados: mais pontos intermediários