Curvas Paramétricas

B-Splines

Renderização de Curvas

Conversão entre Curvas

Renderização de Superfícies

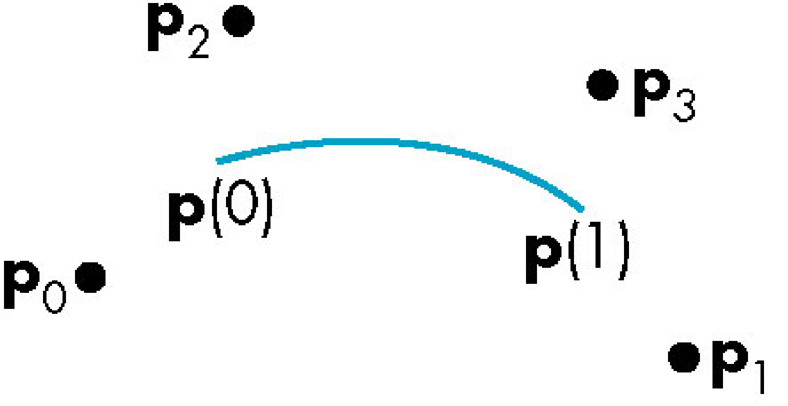
B-Splines

- <u>B</u>asis splines: usa os dados em $\mathbf{p}=[p_{i-2}\ p_{i-1}\ p_i\ p_{i-1}]^T$ para definir uma curva apenas entre p_{i-1} e p_i
- Permite impor um maior número de condições de continuidade para cada segmento
- Para cúbicas, temos continuidade da função e das derivadas primeira e segunda nas junções
- Custo é o triplo
- Para superfícies, representa 9 vezes mais trabalho

B-Spline Cúbica

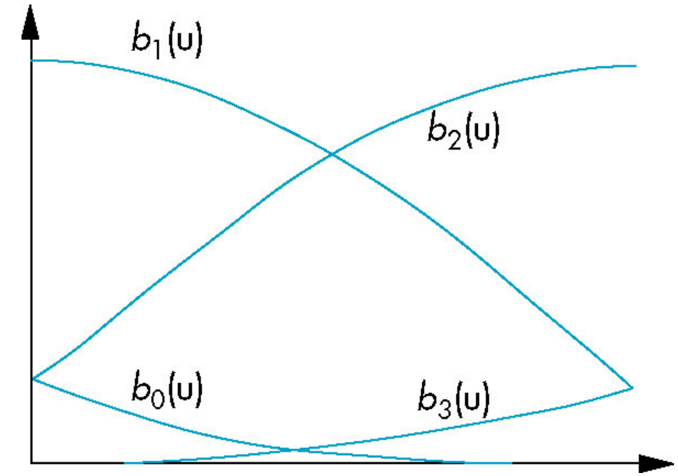
$$p(u) = \mathbf{u}^T\mathbf{M}_S\mathbf{p} = \mathbf{b}(u)^T\mathbf{p}$$

$$\mathbf{M}_S = \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

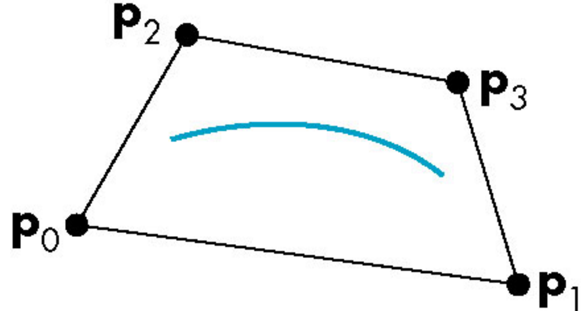


Funções de Mistura (Blending)

$$\mathbf{b}(u) = \frac{1}{6}\begin{bmatrix} (1-u)^3 \\ 4-6u^2+3u^3 \\ 1+3u+3u^2-3u^2 \\ u^3 \end{bmatrix}$$



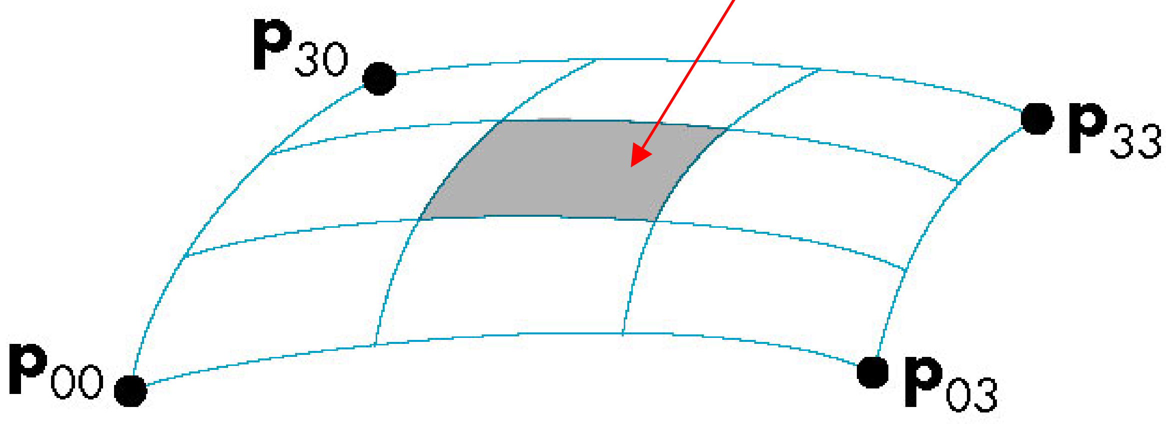
Propriedade da
Envoltória Convexa



Retalhos B-Spline

$$p(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3} b_i(u)\, b_j(v)\, p_{ij} = u^T \mathbf{M}_S \mathbf{P} \mathbf{M}_S^T v$$

definido apenas para 1/9 da região



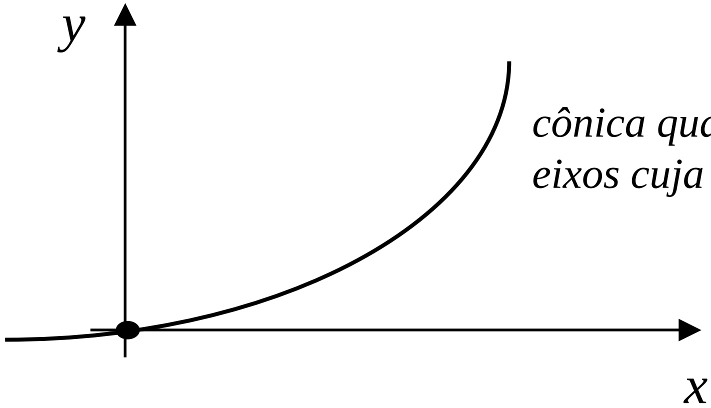
Generalizando Splines

- Podemos estender as splines para qualquer grau
- Dados e condições não precisam estar igualmente espaçados (*knots*)
  - Splines uniformes e não-uniformes
  - Pode haver *knots* repetidos
    - Podemos forçar a interpolação em alguns pontos

NURBS

- Nonuniform Rational B-Spline introduzem uma quarta variável w
  - Pode ser interpretada como peso para se dar maior importância para alguns pontos de controle
  - Também pode ser interpretada como a uso de coordenadas homogêneas

Cônicas



cônica qualquer escrita num sistema de eixos cuja origem é um ponto da cônica

$$ax^2 + bxy + cy^2 + dx + ey = 0$$

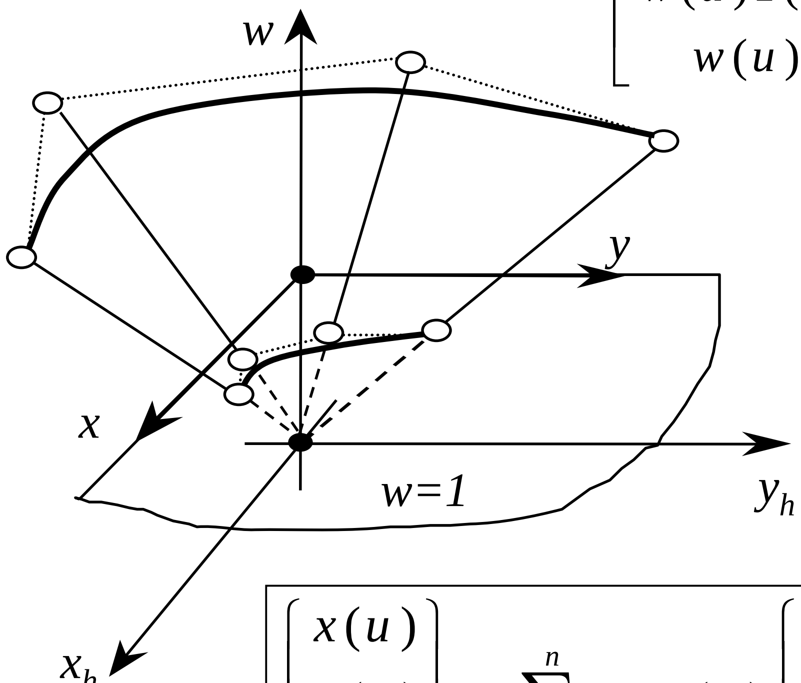
$$x = ty$$

$$at^2y^2 + bty^2 + cy^2 + dty + ey = 0$$

$$y = -\frac{dt + e}{at^2 + bt + c} \qquad x = -\frac{dt^2 + et}{at^2 + bt + c}$$

Qualquer cônica pode ser representada parametricamente como uma fração de polinômios quadráticos

NURBS
Non Uniform Rational B-Splines



$$\begin{bmatrix} w(u)x(u) \\ w(u)y(u) \\ w(u)z(u) \\ w(u) \end{bmatrix} = \sum_{i=0}^{n} N_{i,p}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix}$$

$$\begin{Bmatrix} x(u) \\ y(u) \\ z(u) \end{Bmatrix} = \sum_{i=0}^{n} \frac{w_i N_{i,p}(u)}{\sum_{k=0}^{n} w_k N_{k,p}(u)} \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}$$

$$\boxed{\begin{Bmatrix} x(u) \\ y(u) \\ z(u) \end{Bmatrix} = \sum_{i=0}^{n} R_{i,p}(u) \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \quad \textit{onde} \quad R_{i,p}(u) = \frac{w_i N_{i,p}(u)}{\sum_{k=0}^{n} w_k N_{k,p}(u)}}$$

Cônicas como NURBS

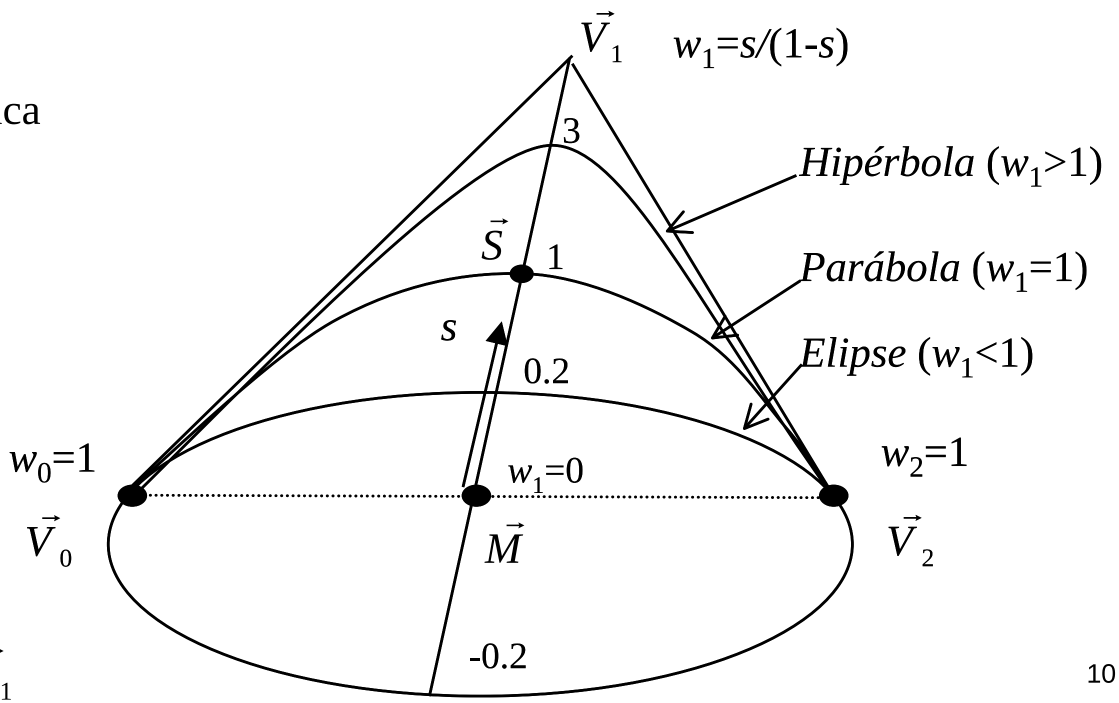
$$\vec{P}(u) = \frac{B_{0,2}(u)w_0\vec{V}_0 + B_{1,2}(u)w_1\vec{V}_1 + B_{2,2}(u)w_2\vec{V}_2}{B_{0,2}(u)w_0 + B_{1,2}(u)w_1 + B_{2,2}(u)w_2}$$

$$onde:$$

$$B_{i,2}(u) = N_{i,2}(u) \quad com \quad U = \{0,0,0,1,1,1\}$$

Faux et al.

w_0w_2/w_1 - determina a cônica



$\vec{P}(\tfrac{1}{2}) = \vec{S} \quad e$

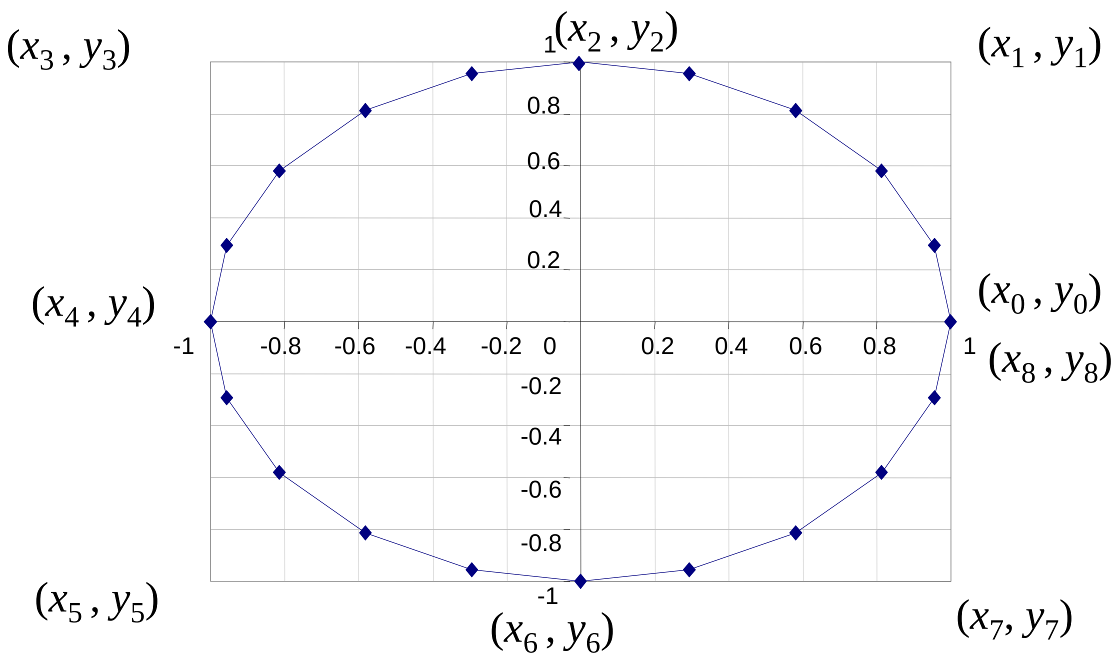
$\vec{S} = (1 - s)\vec{M} + s\vec{V}_1$

Círculo através de NURBS

$$\begin{Bmatrix} x(u) \\ y(u) \end{Bmatrix} = \sum_{i=0}^{8} R_{i,2}(u) \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \qquad onde \qquad R_{i,2}(u) = \frac{w_i N_{i,2}(u)}{\sum_{k=0}^{8} w_k N_{k,2}(u)}$$

$$\{w\} = \{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1\} \qquad \boldsymbol{U}=\{0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1\}$$

n=8
p=2
m=12



Avaliação de Polinômios

- Método mais simples de renderizar uma curva é avaliar o polinômio em vários pontos e formar uma linha poligonal aproximada
- Para superfícies pode-se formar uma malha aproximada de triângulos ou quadriláteros
- Uso da regra de Horner na avaliação

$$p(u)=c_0+u(c_1+u(c_2+uc_3))$$

- 3 multiplicações para cúbicas

Diferenças Finitas

Para $\{u_k\}$ igualmente espaçados, define-se *diferenças finitas :*

$$\Lambda^{(0)} p(u_k) = p(u_k)$$

$$\Lambda^{(1)} p(u_k) = p(u_{k+1}) - p(u_k)$$

$$\Lambda^{(m+1)} p(u_k) = \Delta^{(m)} p(u_{k+1}) - \Delta^{(m)} p(u_k)$$

Para um polinômio de grau n,
a n-ésima diferença finita é constante

Tabela de Diferenças Finitas

$p(u)=1+3u+2u^2+u^3$

t	0	1	2	3	4	5
$\mathbf{p}$	1	7	23	55	109	191
$\Delta^{(1)}\mathbf{p}$	6	16	32	54	82	
$\Delta^{(2)}\mathbf{p}$	10	16	22	28		
$\Delta^{(3)}\mathbf{p}$	6	6	6			

Obtendo os Próximos Valores

Iniciando no final da tabela, pode-se mover para cima gerando novos valores para o polinômio

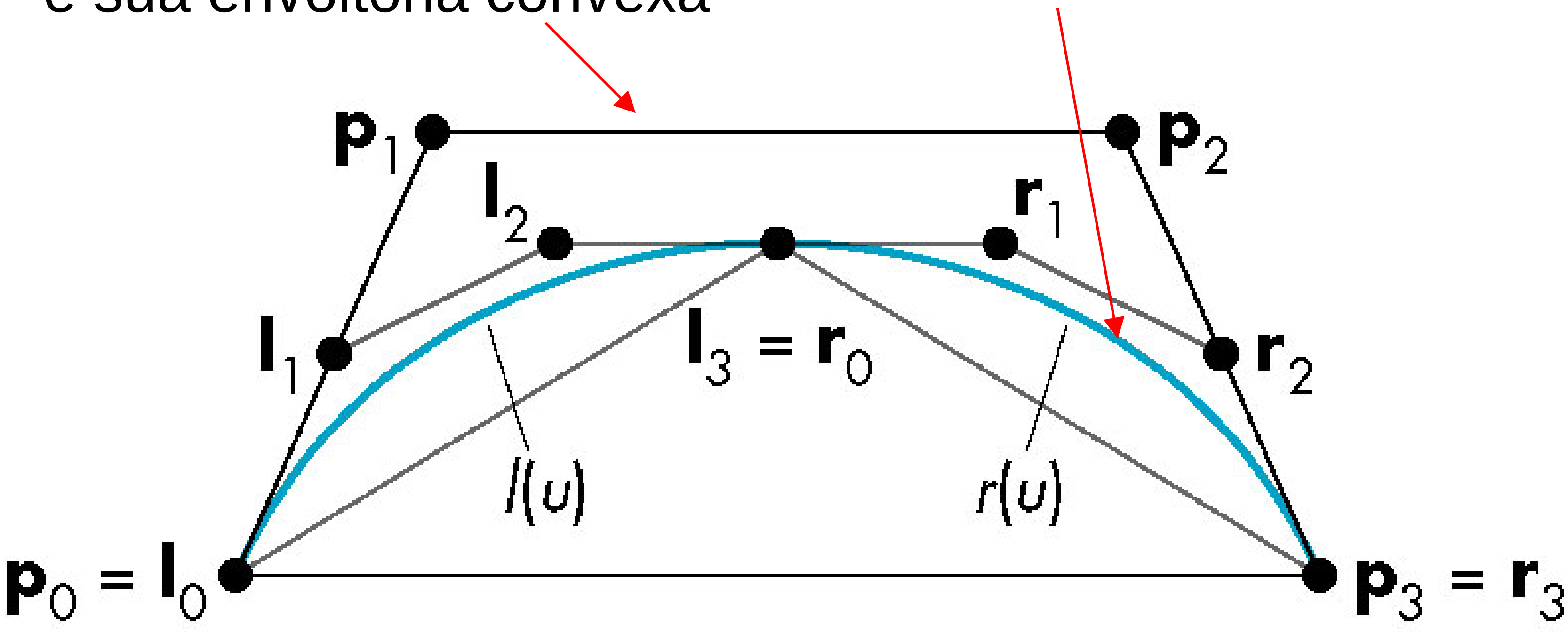
t	0	1	2	3	4	5
$\mathbf{p}$	1	7	23	55	109	191
$\Delta^{(1)}\mathbf{p}$	6	16	32	54	82	
$\Delta^{(2)}\mathbf{p}$	10	16	22	28		
$\Delta^{(3)}\mathbf{p}$	6	6	6			

Recursão deCasteljau

- Pode-se usar a propriedade da envoltória convexa das curvas de Bezier para se obter um método eficiente que não necessita de nenhuma avaliação da função
  - Usa apenas os valores nos pontos de controle
- Baseado na idéia de que “todo polinômio de Bezier pode ser dividido em outros cujos pontos de controle foram escolhidos adequadamente”

Dividindo uma Curva de Bezier

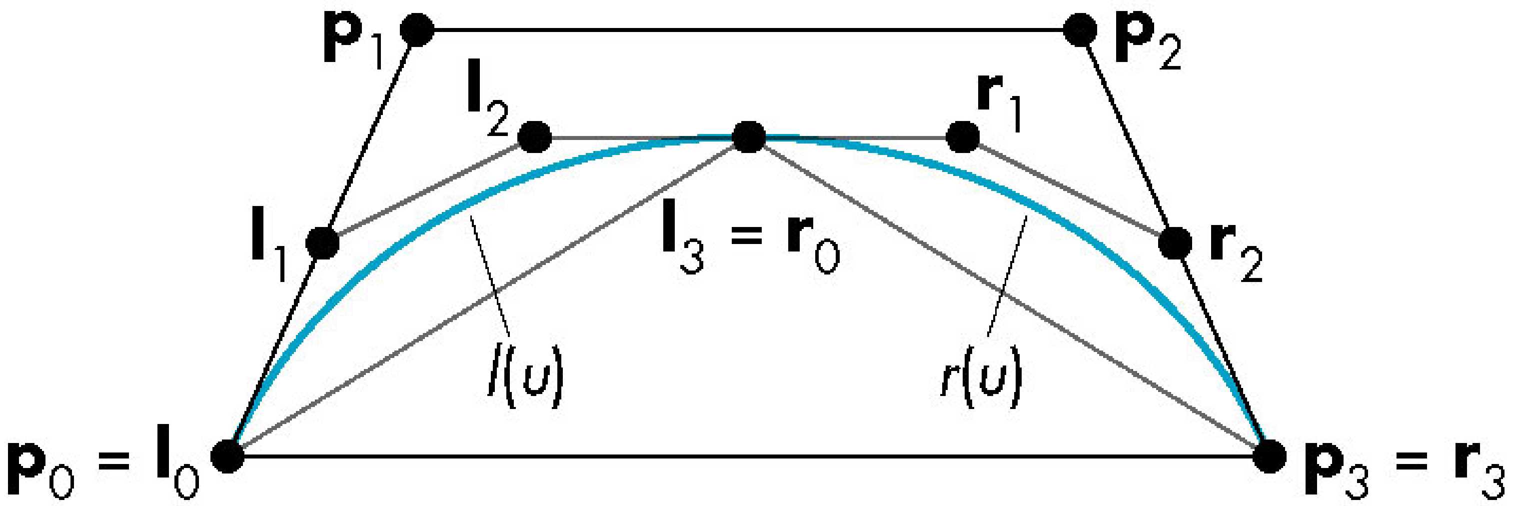
p_0, p_1 , p_2 , p_3 determinam um polinômio cúbico de Bezier e sua envoltória convexa



Considere a metade esquerda l(u) e a metade direita r(u)

l(u) e r(u)

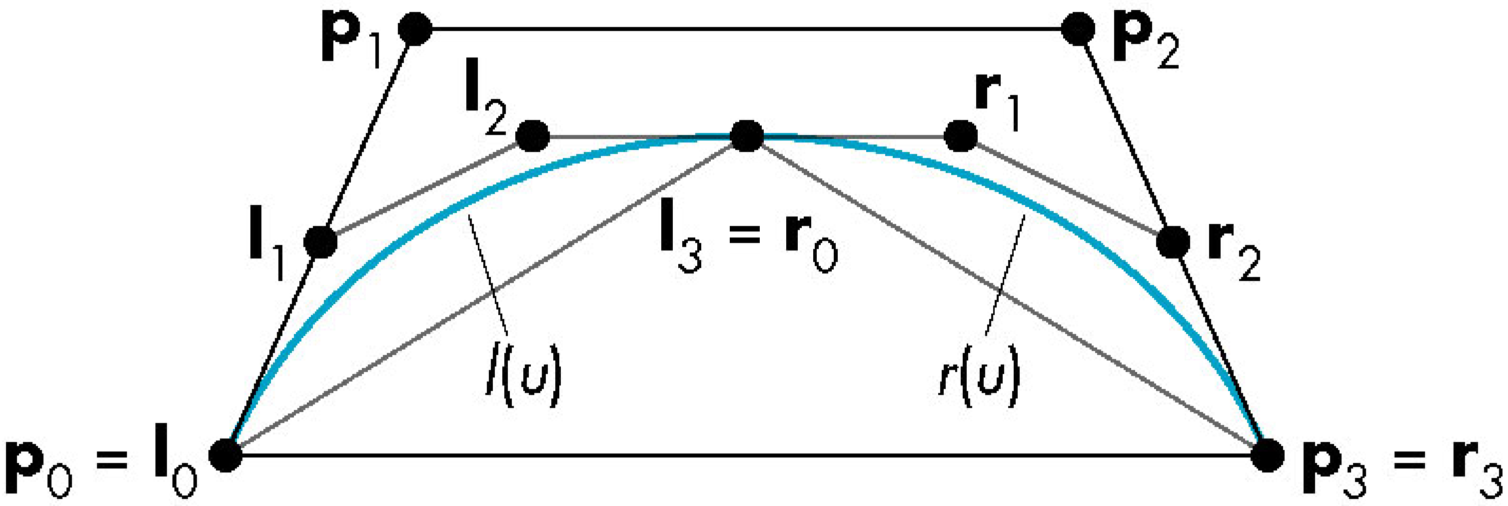
Como l(u) e r(u) são curvas de Bezier, devemos ser capazes de encontrar 2 conjunto de pontos $\{l_0, l_1, l_2, l_3\}$ e $\{r_0, r_1, r_2, r_3\}$ que as determinam



Envoltórias Convexas

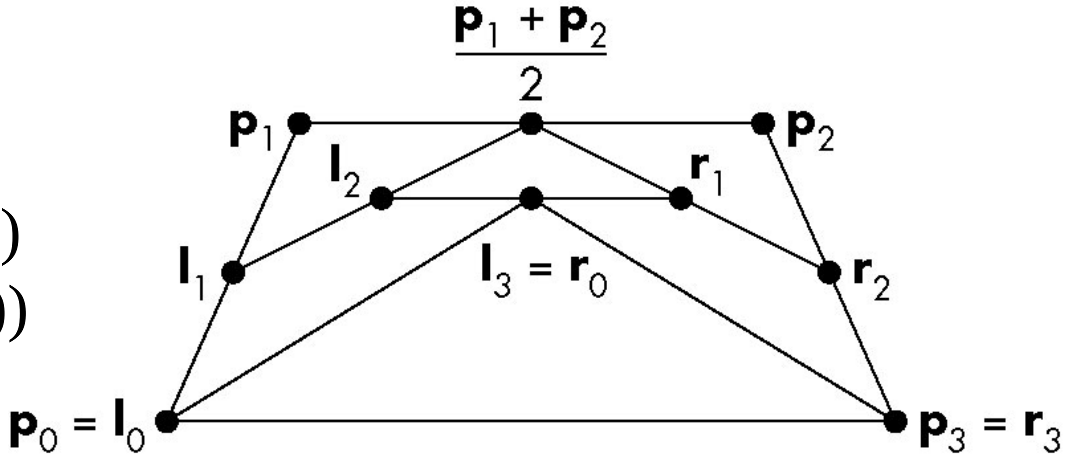
$\{l_0, l_1, l_2, l_3\}$ e $\{r_0, r_1, r_2, r_3\}$ possuem envoltórias convexas que são mais próximas de p(u) que a env. de $\{p_0, p_1, p_2, p_3\}$ Isto é conhecido como *propriedade* de *redução da variação*.

A linha poligonal de l_0 a l_3 $(= r_0)$ e de r_0 a r_3 é uma aproximação de p(u). Repetições recursivas desse método irão obter melhores aproximações.



Equações - Forma Eficiente

$l_0 = p_0$
$r_3 = p_3$
$l_1 = ½(p_0 + p_1)$
$r_1 = ½(p_2 + p_3)$
$l_2 = ½(l_1 + ½(p_1 + p_2))$
$r_1 = ½(r_2 + ½(p_1 + p_2))$
$l_3 = r_0 = ½(l_2 + r_1)$



Necessitam apenas *shifts* e somas !

Toda Curva é uma Curva de Bezier

- Pode-se renderizar um dado polinômio usando o método recursivo basta encontrar os pontos de controle de sua representação como uma curva de Bezier
- Suponha que $p(u)$ é dada pela curva interpolada a partir dos pontos $\mathbf{q}$

$$p(u)=\mathbf{u}^T\mathbf{M}_I\mathbf{q}$$

- Existem pontos de controle de Bezier $\mathbf{p}$ tal que

$$p(u)=\mathbf{u}^T\mathbf{M}_B\mathbf{p}$$

- Igualando e resolvendo, obtém-se $\mathbf{p}=\mathbf{M}_B^{-1}\mathbf{M}_I\,\mathbf{q}$

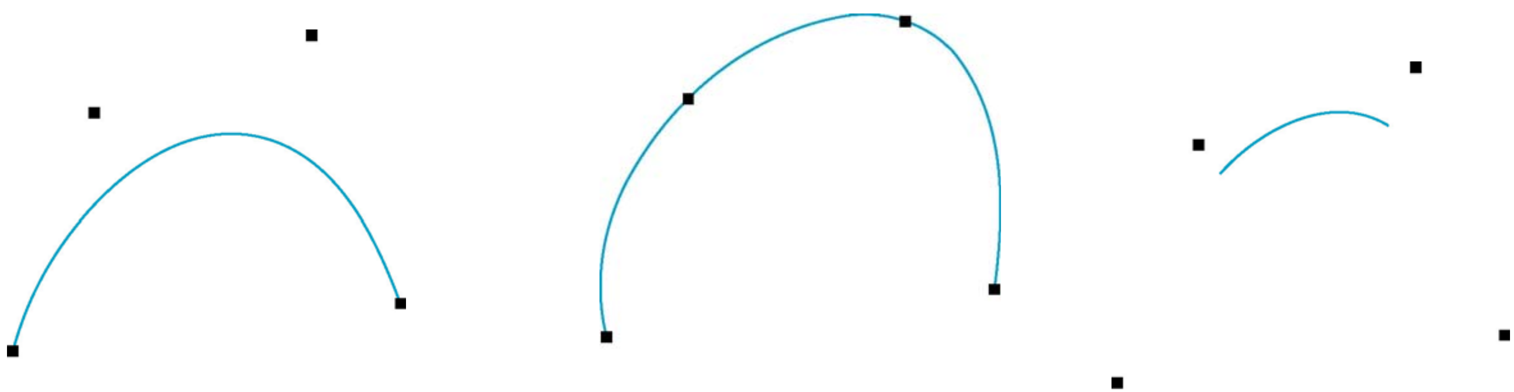
Matrizes de Conversão

Interpolação para Bezier $\mathbf{M}_B^{-1}\mathbf{M}_I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{5}{6} & 3 & -\frac{3}{2} & \frac{1}{3} \\ \frac{1}{3} & -\frac{3}{2} & 3 & -\frac{5}{6} \\ 0 & 0 & 0 & 1 \end{bmatrix}$

B-Spline para Bezier $\mathbf{M}_B^{-1}\mathbf{M}_S = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix}$

Exemplo

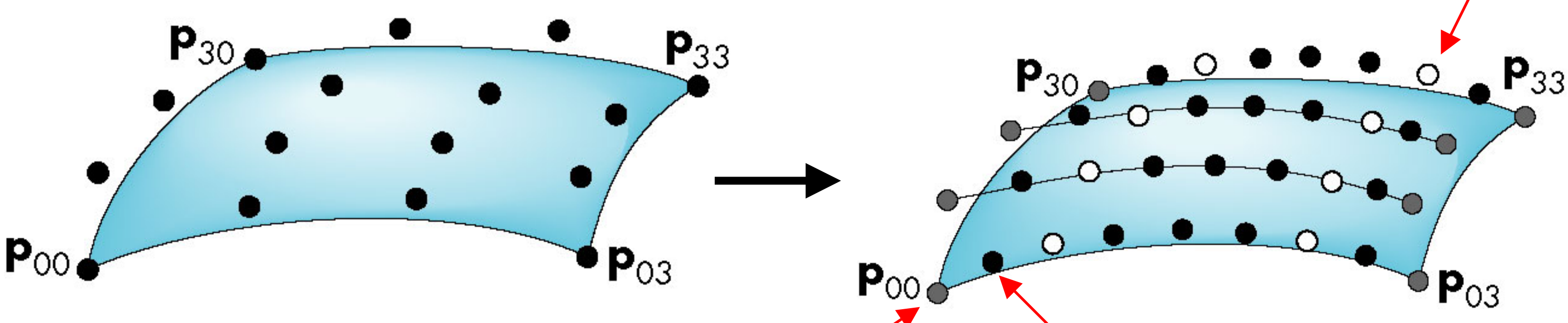
Estas três curvas foram totas geradas a partir do mesmos dados originais usando o método recursivo a partir da conversão dos pontos de controle para pontos de controle de Bezier



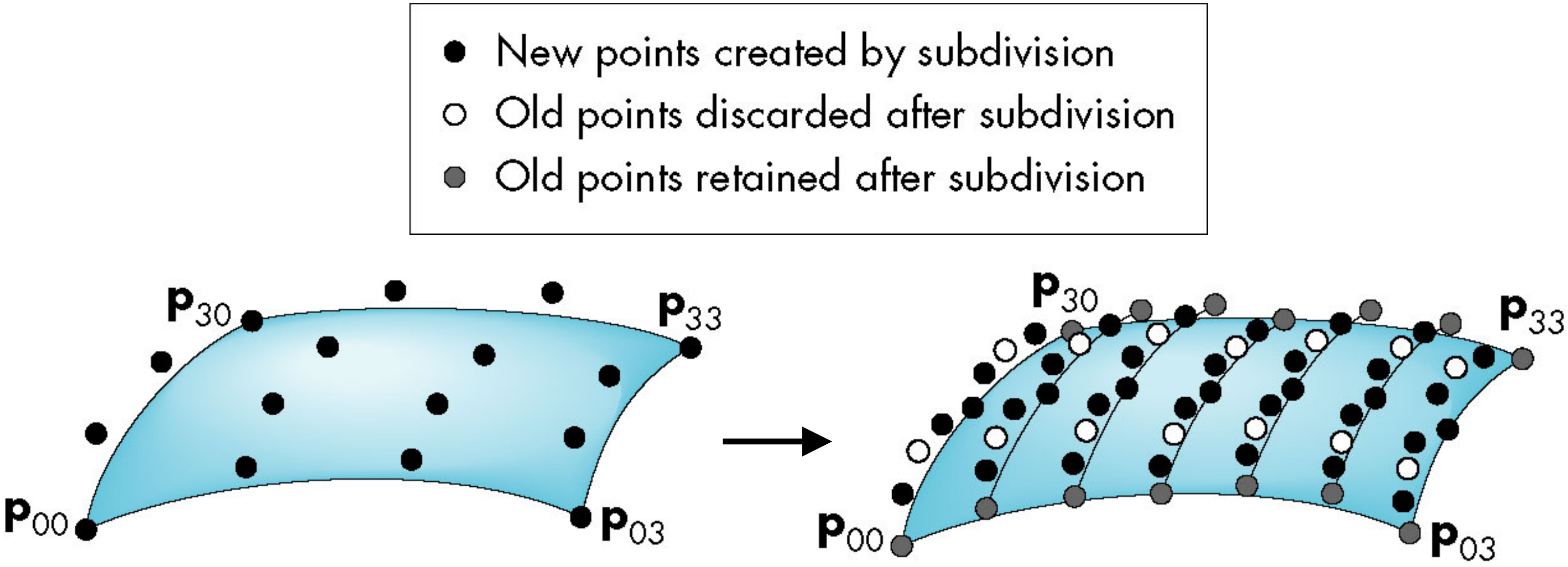
Bezier　　　Interpolating　　　B Spline

Superfícies

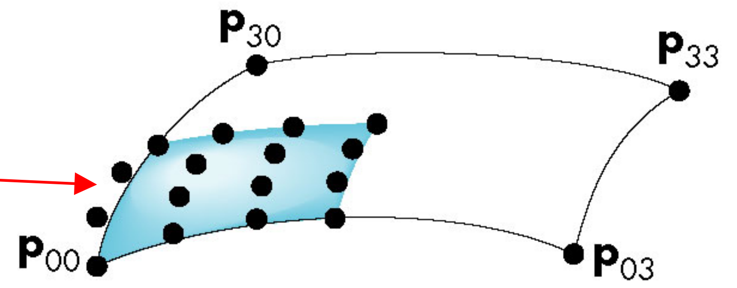
- Pode-se aplicar o método recursivo à superfícies basta lembrar que em um retalho de Bezier curvas com u (ou v) constantes são na verdade curvas de Bezier v (ou u)
- Primeiramente subdividimos em u
  - Processo cria novos pontos
  - Alguns pontos originais são discartados



Superfícies - Segunda Subdivisão



16 pontos finais para
1 dos 4 retalhos criados



Normais

- Para renderização necessitamos das normais se desejamos colorizar (shade)
  - Pode-se calcular a partir das eq. paramétricas

$$\mathbf{n} = \frac{\partial \mathbf{p}(u,v)}{\partial u} \times \frac{\partial \mathbf{p}(u,v)}{\partial v}$$

  - Pode-se utilizar os vértices dos cantos para se determinar