Universidade Federal do Pampa
Ciência da Computação
Computação Gráfica
Prof. Alessandro Bof de Oliveira

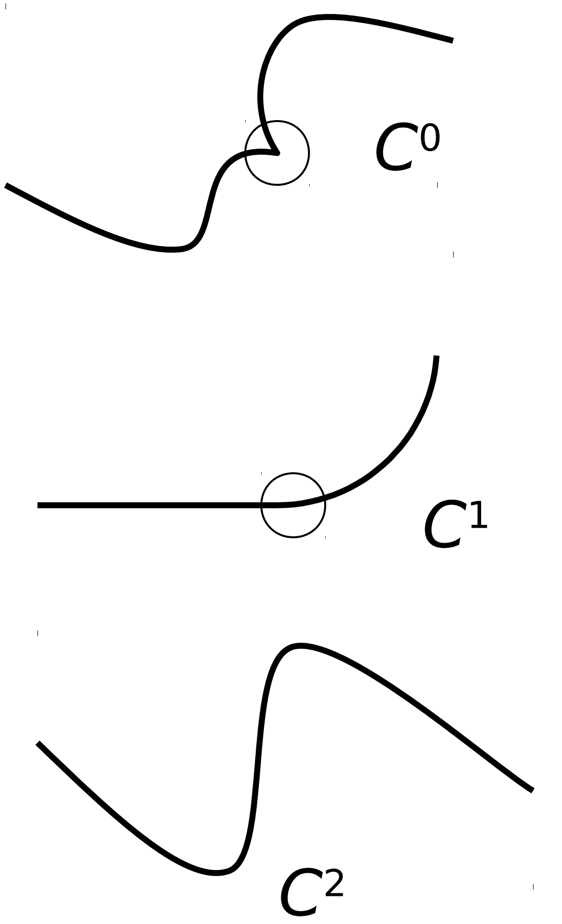
Computação Gráfica
Curvas

Curvas Paramétricas

- Normalmente, o resultado da modelagem é dado em forma paramétrica
- Permite que uma curva seja definida facilmente através de um parâmetro (que em geral varia entre 0 e 1).
- Permite descrever uma curva por trechos (segmentos).
- Uma curva em 3D é dada por
  - $C(u) = [C_x(u)\ C_y(u)\ C_z(u)]^T$

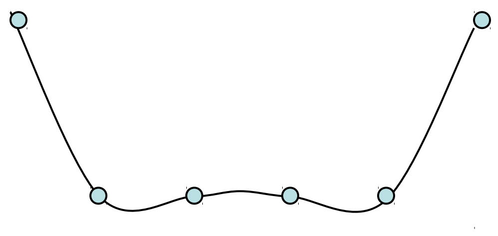
Continuidade

- Podemos definir a suavidade de uma curva algebricamente.
- Continuidade $C^0 \rightarrow$ funções paramétricas são contínuas, sem descontinuidades.
- Continuidade $C^1 \rightarrow$ funções paramétricas têm primeiras derivadas contínuas, isto é, tangentes variam suavemente
- Continuidade $C^2 \rightarrow$ funções paramétricas têm *segunda* derivadas contínuas

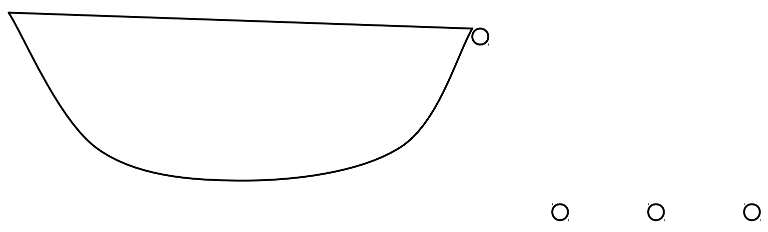


Interpolação x Aproximação

- Modelagem da curva através de um conjunto de pontos.
- Se a curva desejada passa obrigatoriamente pelos pontos temos uma interpolação.
- Porém se a curva passa perto dos pontos temos métodos de aproximação.



Interpolação

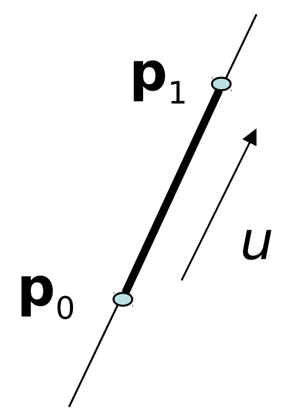


Aproximação

Algoritmo de De Casteljau

- Suponha que queiramos aproximar uma curva polinomial entre dois pontos $\mathbf{p}_0$ e $\mathbf{p}_1$ dados
- Uma solução natural é um segmento de reta que passa por $\mathbf{p}_0$ e $\mathbf{p}_1$ cuja parametrização pode ser escrita como:

  $\mathbf{p}(u) = (1 - u)\mathbf{p}_0 + u\,\mathbf{p}_1$
- Observe que os polinômios $(1 - u)$ e u somam 1 para qualquer valor de u.



Algoritmo de De Casteljau

- Para generalizar a idéia para três pontos $\mathbf{p}_0$, $\mathbf{p}_1$ e $\mathbf{p}_2$ consideramos primeiramente os segmentos de reta $\mathbf{p}_0\mathbf{p}_1$ e $\mathbf{p}_1\mathbf{p}_2$

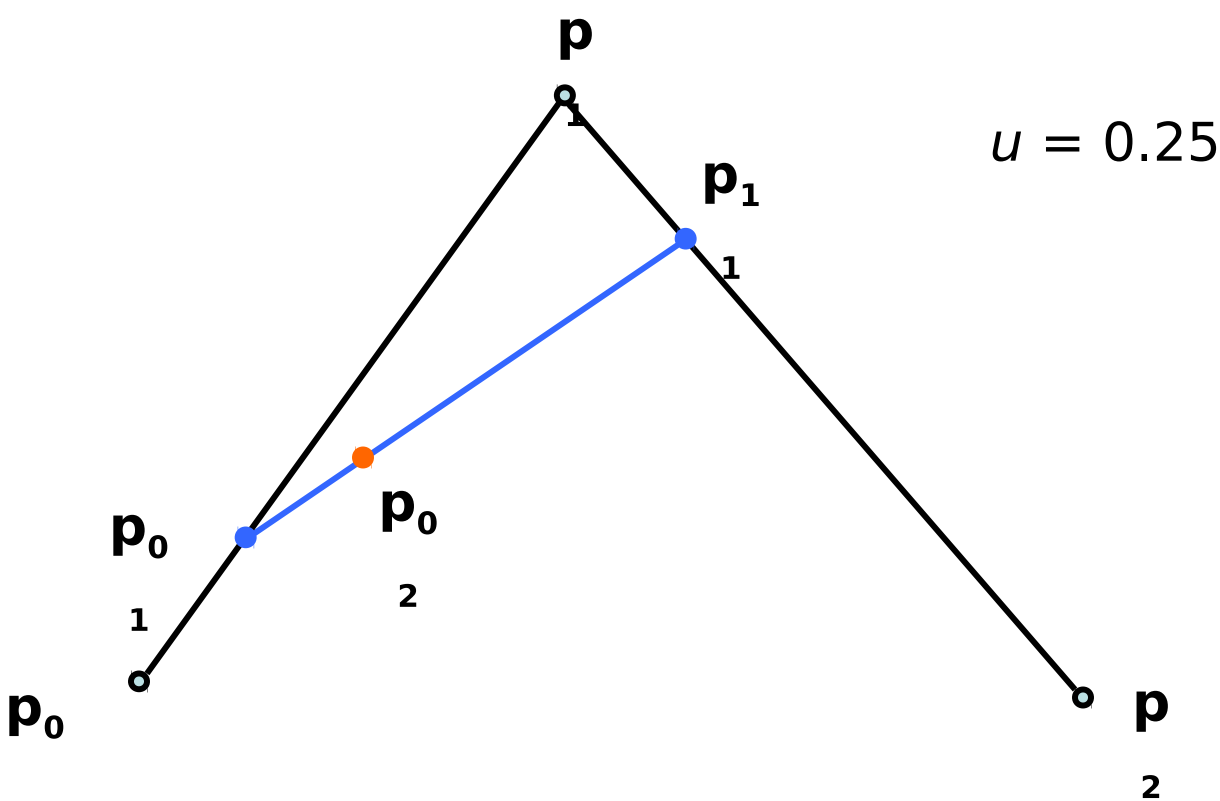
$$\mathbf{p}_{01}(u) = (1 - u)\mathbf{p}_0 + u\,\mathbf{p}_1$$

$$\mathbf{p}_{11}(u) = (1 - u)\mathbf{p}_1 + u\,\mathbf{p}_2$$

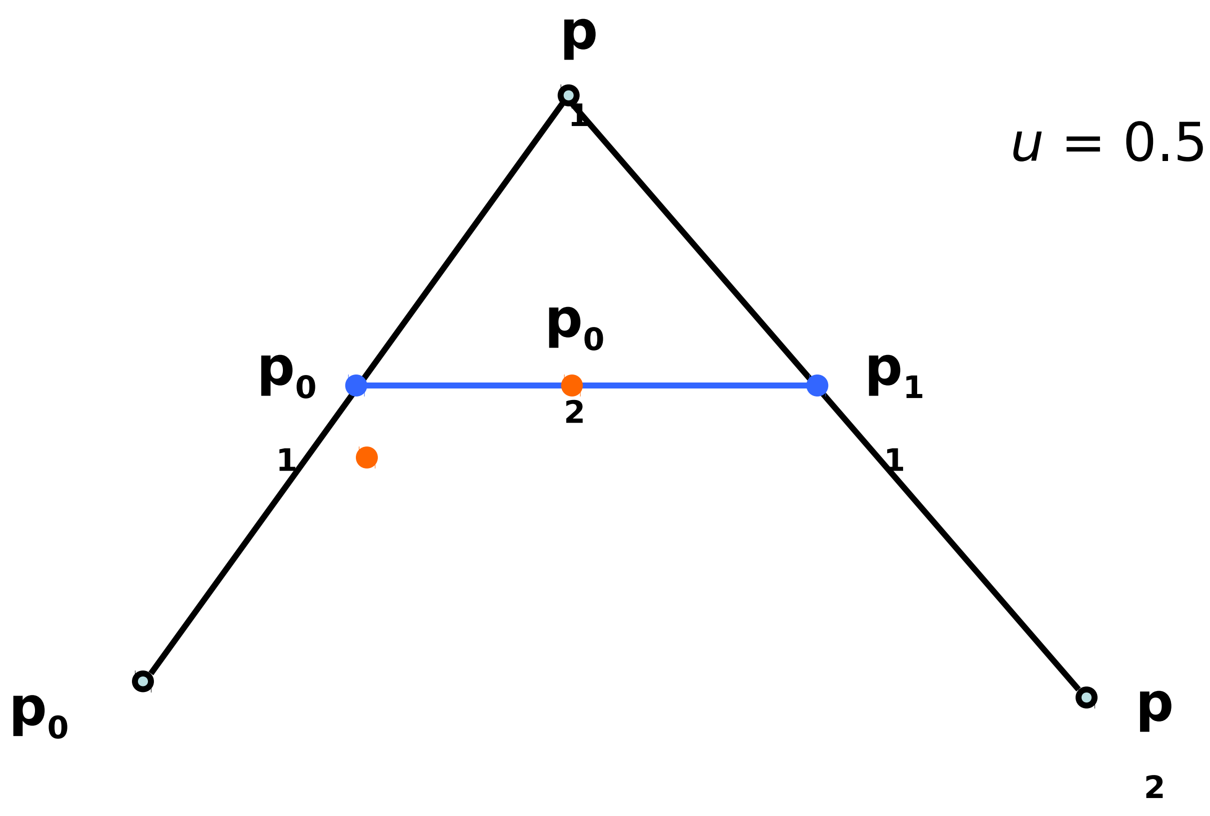
- Podemos agora realizar uma interpolação entre $\mathbf{p}_{01}(u)$ e $\mathbf{p}_{12}(u)$

$$\begin{aligned}\mathbf{p}_{02}(u) &= (1 - u)\mathbf{p}_{01}(u) + u\,\mathbf{p}_{11}(u)\\ &= (1 - u)^2\mathbf{p}_0 + 2\,u\,(1 - u)\mathbf{p}_1 + u^2\,\mathbf{p}_2\end{aligned}$$

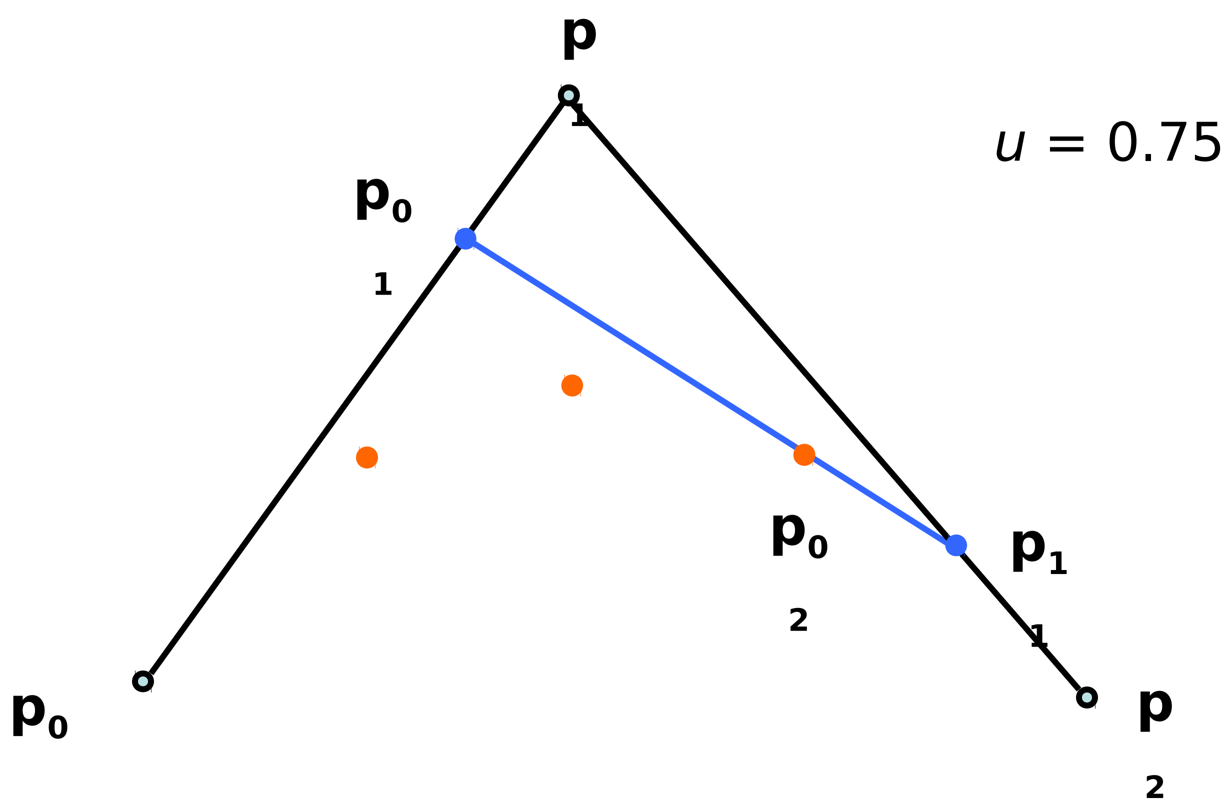
Algoritmo de De Casteljau



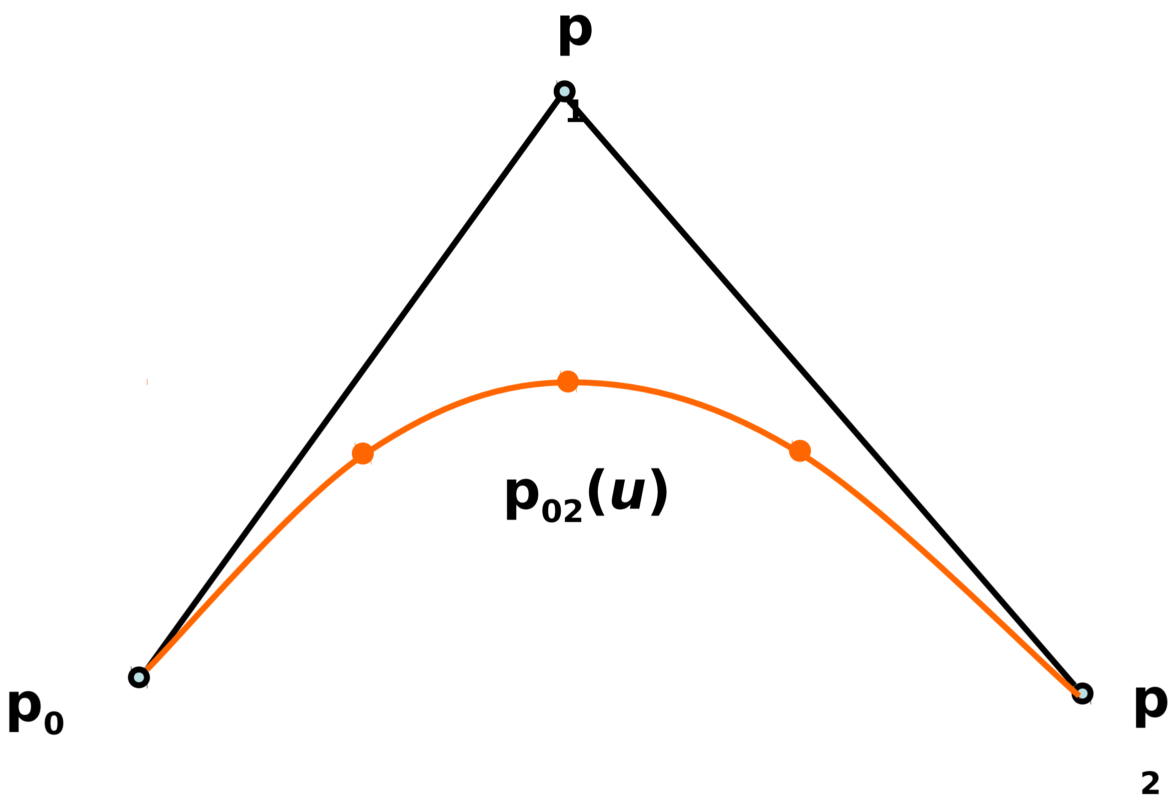
Algoritmo de De Casteljau



Algoritmo de De Casteljau



Algoritmo de De Casteljau



Algoritmo de De Casteljau

- A curva obtida pode ser entendida como a combinação dos pontos $\mathbf{p}_0$, $\mathbf{p}_1$ e $\mathbf{p}_2$ por intermédio de três funções quadráticas:
  - $b_{02}(u) = (1 - u)^2$
  - $b_{12}(u) = 2u\,(1 - u)$
  - $b_{22}(u) = u^2$
- Aplicando mais uma vez a ideia podemos definir uma cúbica por 4 pontos

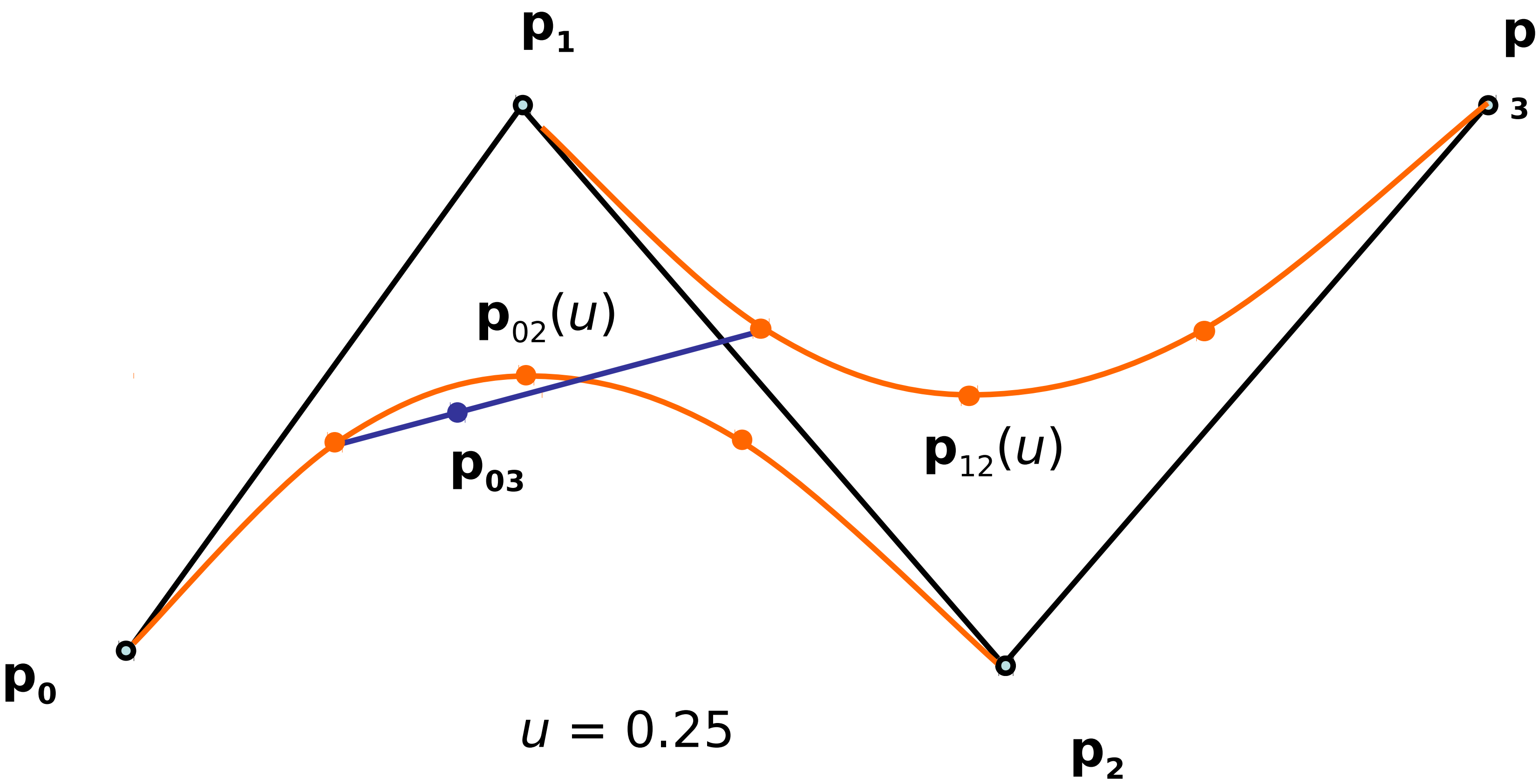
$$\mathbf{p}_{02}(u) = (1 - u)^2\,\mathbf{p}_0 + 2\,u\,(1 - u)\,\mathbf{p}_1 + u^2\,\mathbf{p}_2$$

$$\mathbf{p}_{12}(u) = (1 - u)^2\,\mathbf{p}_1 + 2\,u\,(1 - u)\,\mathbf{p}_2 + u^2\,\mathbf{p}_3$$

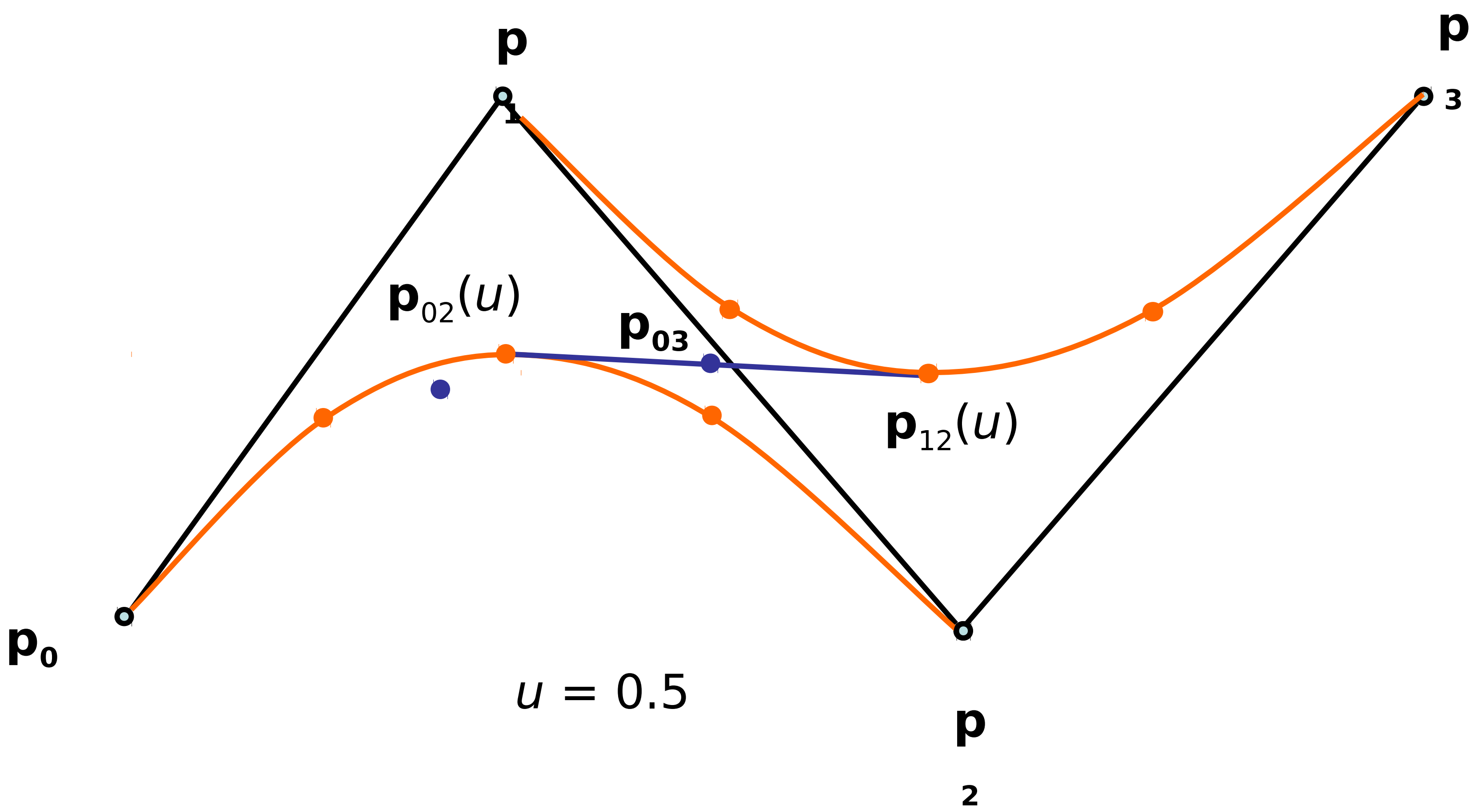
$$\mathbf{p}_{03}(u) = (1 - u)\,\mathbf{p}_{02}(u) + u\,\mathbf{p}_{12}(u)$$

$$= (1 - u)^3\,\mathbf{p}_0 + 3\,u\,(1 - u)^2\,\mathbf{p}_1 + 3\,u^2\,(1 - u)\,\mathbf{p}_2 + u^3\,\mathbf{p}_3$$

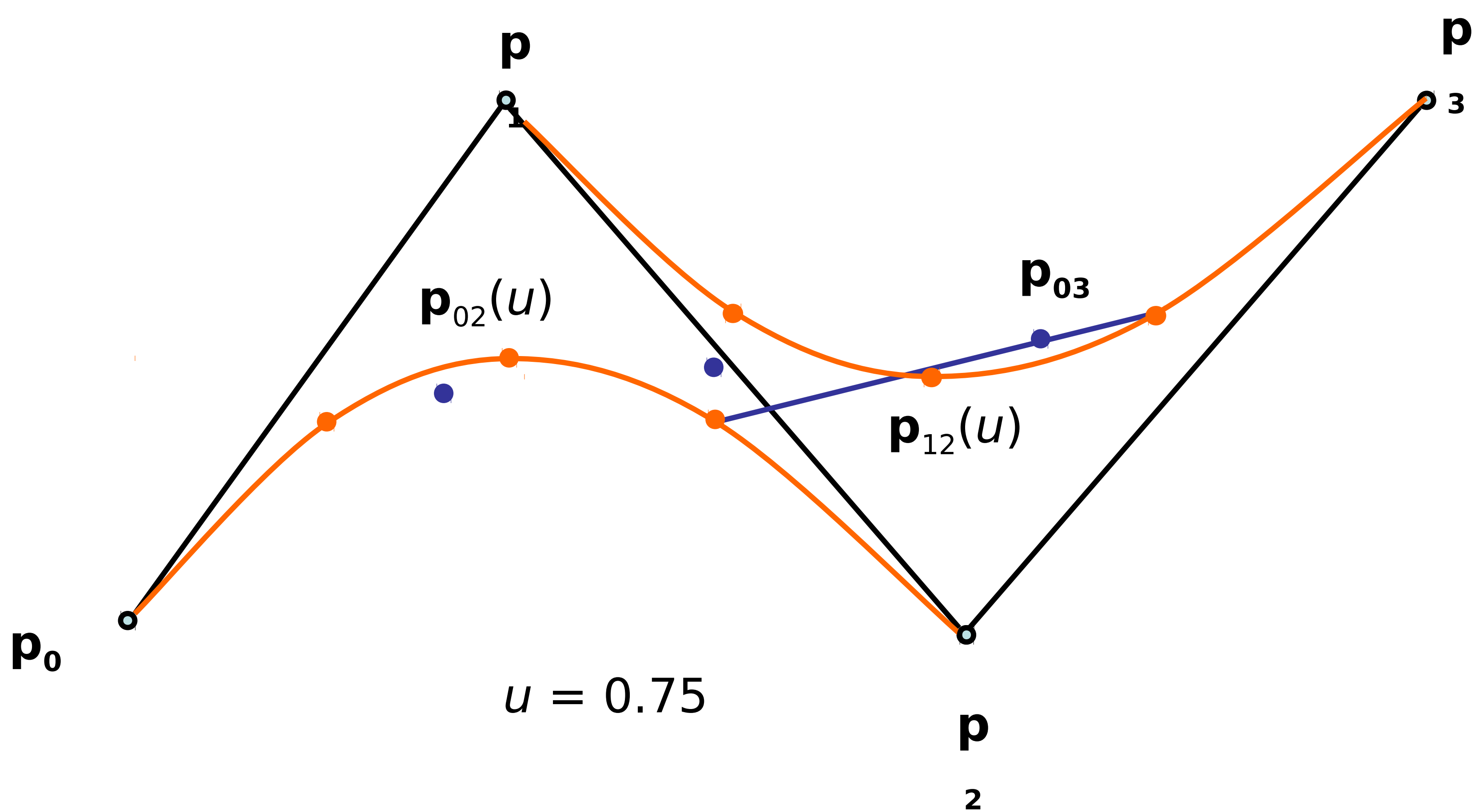
Algoritmo de De Casteljau



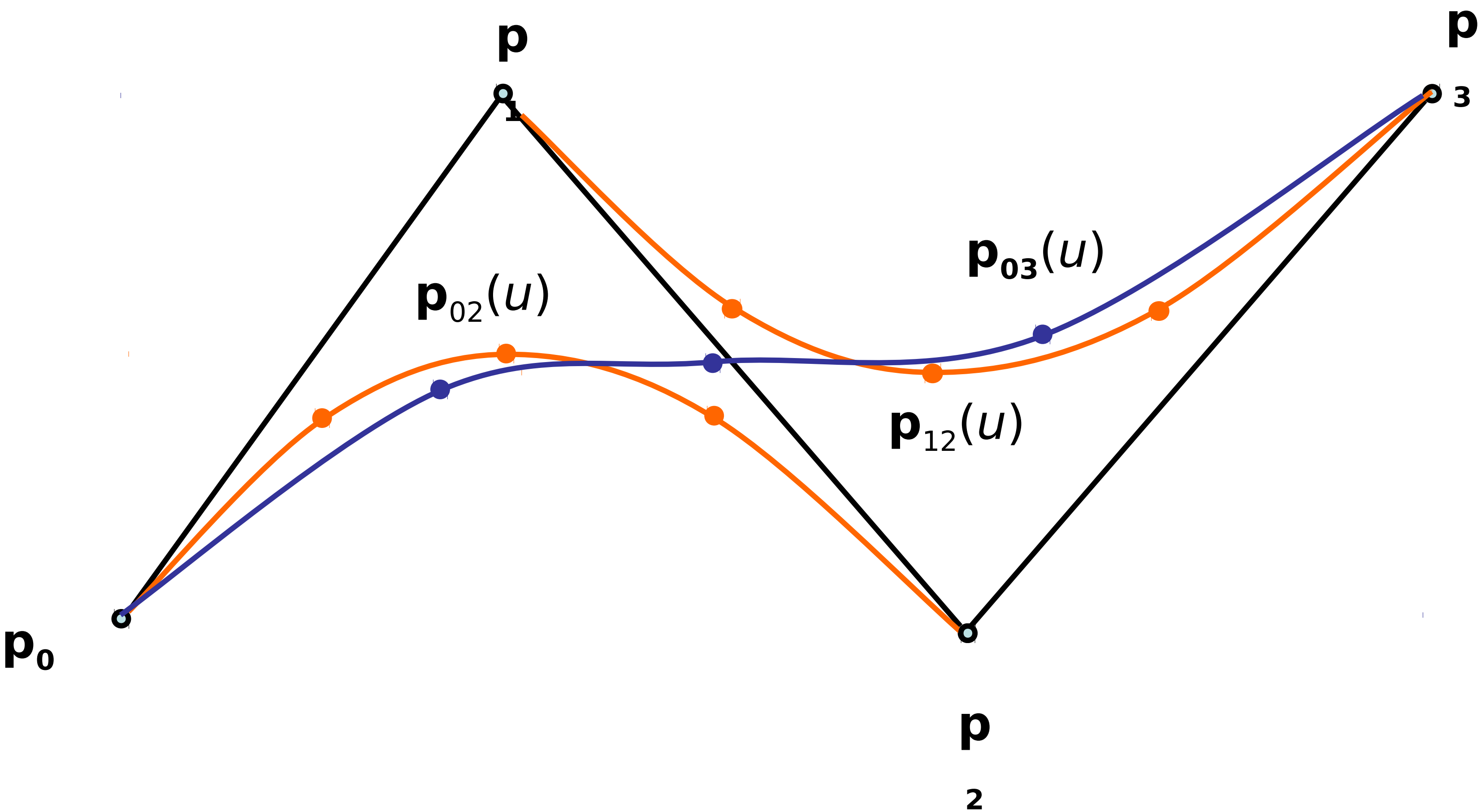
Algoritmo de De Casteljau



Algoritmo de De Casteljau



Algoritmo de De Casteljau



Algoritmo de De Casteljau

- Novamente temos uma curva dada pela soma de 4 funções de mistura (agora cúbicas), cada uma multiplicada por um dos 4 pontos
  - $b_{03}(u) = (1 - u)^3$
  - $b_{13}(u) = 3\,u\,(1 - u)^2$
  - $b_{23}(u) = 3\,u^2\,(1 - u)$
  - $b_{33}(u) = u^3$

- Uma curva de grau *n* pode ser construída desta forma e será expressa pela expressão geral:

$$p_{0n}(u) = \sum_{j=0}^{n} b_{jn}(u)\, p_j$$

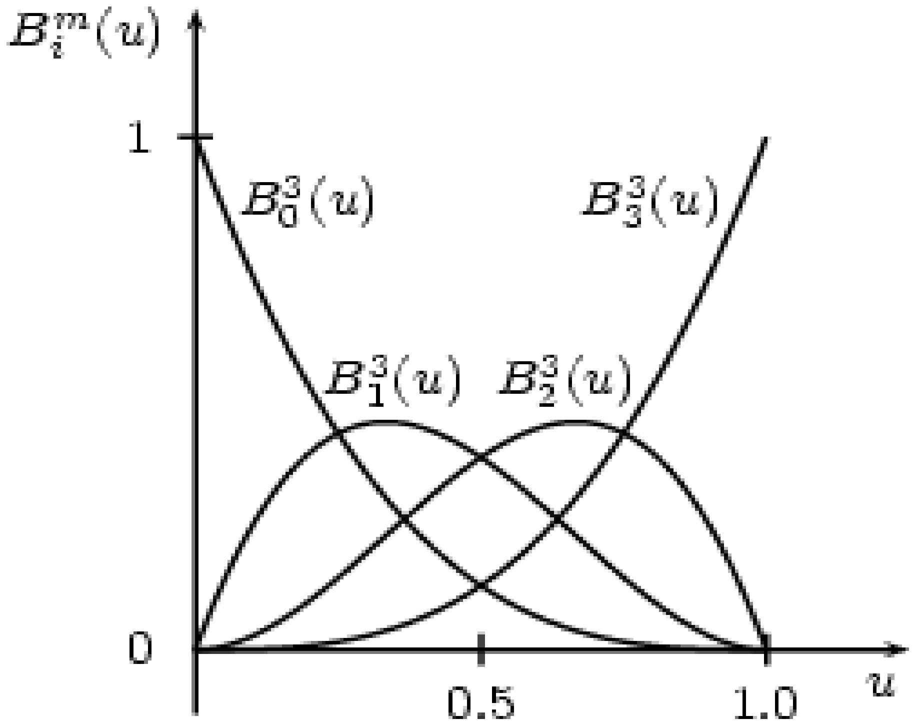
- Onde $b_{jn}(u)$ é denominado de polinômio de Bernstein.

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

Curvas de Bézier e Polinômios de Bernstein

- As curvas construídas pelo algoritmo de De Casteljau são conhecidas como *curvas de Bézier* e as funções de mistura são chamadas de *base Bézier* ou *polinômios de Bernstein*

Polinômios de Bernstein



Polinômios de Bernestein de terceiro grau (ou grau 3).

Forma Matricial da Base Bézier

- Podemos escrever a equação para uma curva de Bézier cúbica na forma

$$p(u) = p_{03}(u) = [1 \quad u \quad u^2 \quad u^3]\, M_B \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

onde M_B é a matriz de coeficientes da base Bézier

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

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 (n) de controle menos 1 (ou seja n-1)
- A curva de Bézier está contida no fecho convexo do polígono de controle
  - Os polinômios de Bernstein somam 1 para qualquer *u*
- A curva interpola o primeiro e último ponto do polígono de controle

Propriedades de Curva de Bézier

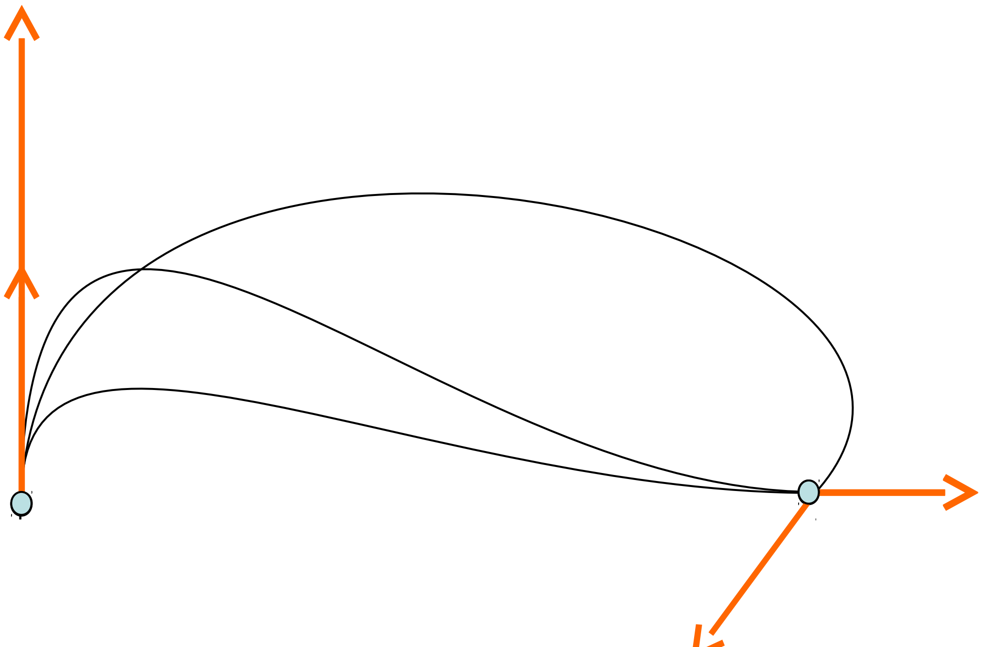
- As tangentes à curva em $\mathbf{p}_0$ e $\mathbf{p}_n$ têm a direção dos segmentos de reta $\mathbf{p}_0\mathbf{p}_1$ e $\mathbf{p}_{n-1}\mathbf{p}_n$, respectivamente
  - Para cúbicas, as derivadas são $3(\mathbf{p}_1 - \mathbf{p}_0)$ e $3(\mathbf{p}_2 - \mathbf{p}_3)$
- Qualquer linha reta intercepta a curva tantas ou menos vezes quanto intercepta o polígono de controle.

Desenhando Curvas Bézier

- Curva normalmente é aproximada por uma linha poligonal
- Pontos podem ser obtidos avaliando a curva em $u = u_1, u_2 \ldots u_k$
  - Avaliar os polinômios de Bernstein
  - Usar o algoritmo recursivo de De Casteljau
- Quantos pontos?
  - Mais pontos em regiões de alta curvatura
- Idéia: subdividir recursivamente a curva em trechos até que cada trecho seja aproximadamente “reto”

Curvas de Hermite

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



Curvas de Hermite

- No caso de cúbicas, temos o ponto inicial e final além dos vetores tangentes

$$p(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M_H \begin{bmatrix} p_0 \\ p_1 \\ p'(0) \\ p'(1) \end{bmatrix}$$

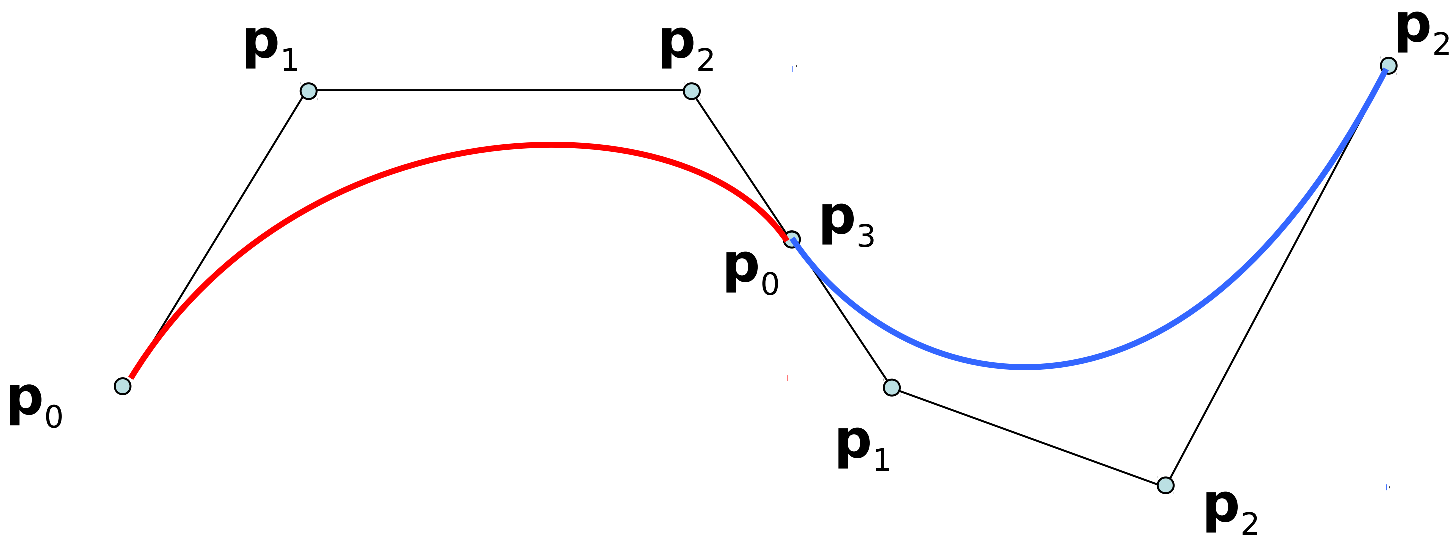
$$\text{onde } M_H = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Curvas Longas

- 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*
- Uma solução pode ser:
  - Emendar curvas polinomiais de grau baixo

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



Curvas Racionais

- Funções são razões
  - Avaliados em coordenadas homogêneas:

$$[x(t), y(t), z(t), w(t)] \rightarrow \left[\frac{x(t)}{w(t)}, \frac{y(t)}{w(t)}, \frac{z(t)}{w(t)}\right]$$

  - NURBS (Non-Uniform Rational B-Splines): *x(t)*, *y(t)*, *z(t)* e *w(t)* são B-splines não uniformes
- Vantagens:
  - Invariantes sob transformações perspectivas e portanto podem ser avaliadas no espaço da imagem

Parametrização de um Círculo

- Por exemplo, uma parametrização conhecida do círculo é dada por

$$x(u) = \frac{1-u^2}{1+u^2}$$

$$y(u) = \frac{2\mathrm{u}}{1+u^2}$$

- Podemos expressar essa parametrização em coordenadas homogêneas por:

$$x(u) = 1-u^2$$
$$y(u) = 2\mathrm{u}$$
$$w(u) = 1+u^2$$