



# Computação Gráfica

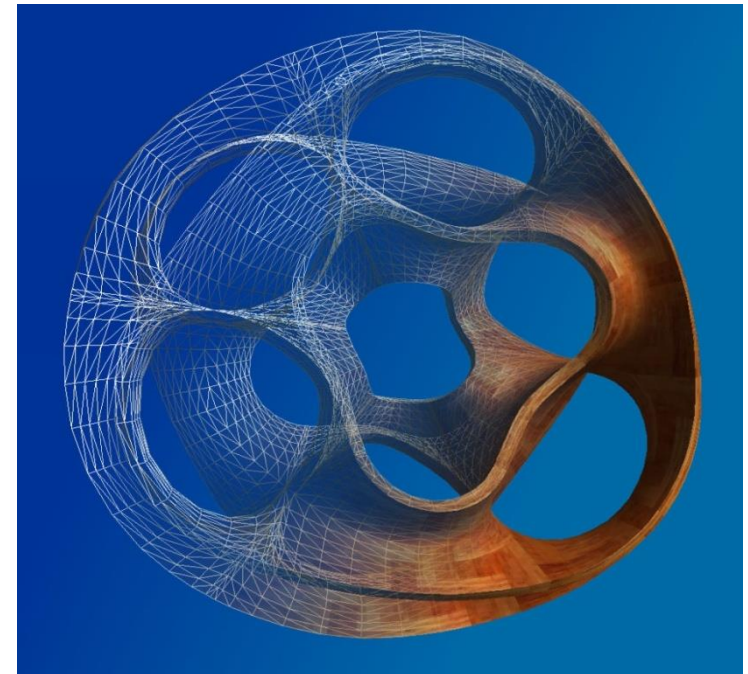
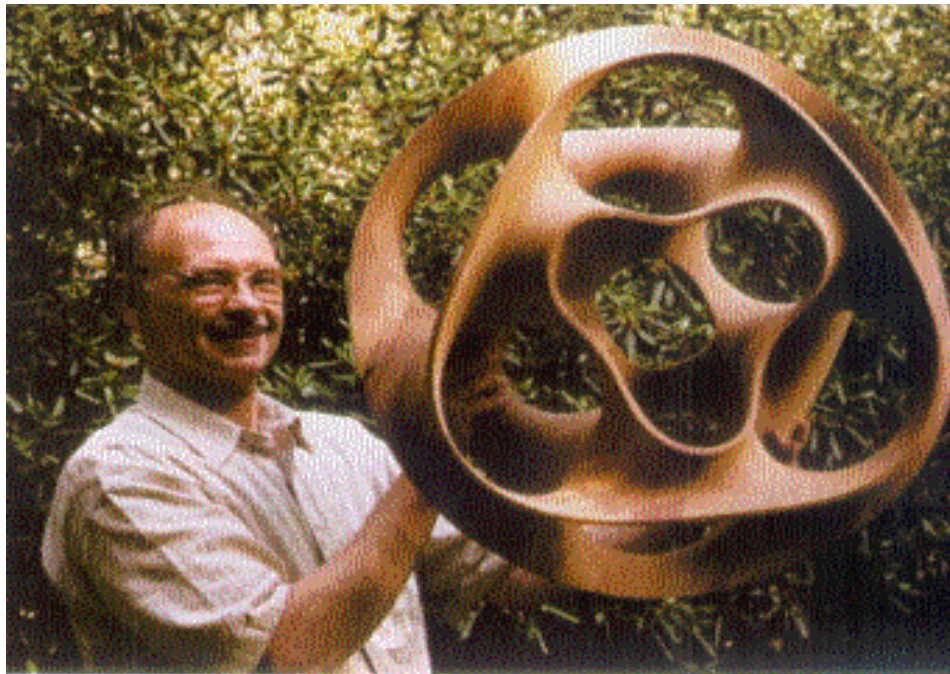
## Curva de Bézier

*Professor: Luciano Ferreira Silva, Dr.*



# Modelagem

- A modelagem na CG consiste em todo o processo de descrever um modelo, objeto ou cena, de forma que se possa desenhá-lo;





# Métodos de Modelagem Geométrica

- Podemos dividir as técnicas de modelagem em três formas:
  - ✓ modelagem manual, automática ou matemática.
- A modelagem automática é a mais sofisticada, mais rápida e poderosa.
  - ✓ Através de equipamentos especiais como scanners 3D, podemos obter o modelo tridimensional de quase tudo.



# Modelagem manual

- A modelagem manual é, sem dúvida, o método mais fácil, barato e antigo que utiliza basicamente as medidas de um modelo real e a intuição do modelador.



# Aproximação de superfícies e Malhas poligonais

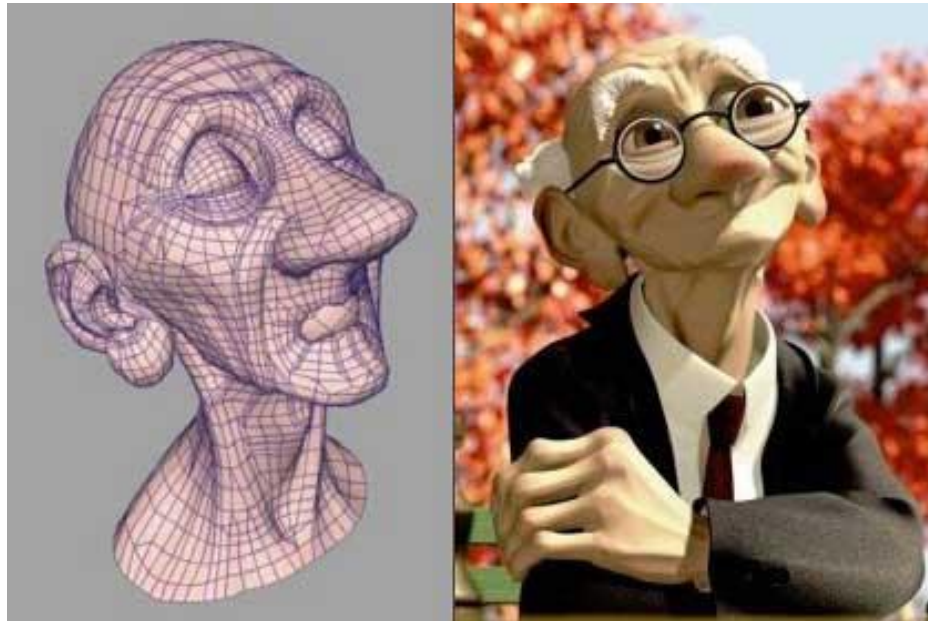
- Os métodos de aproximação empregam duas operações básicas:
  - ✓ Amostragem, que consiste em obter pontos na superfície para produzir os retalhos. Temos os métodos *uniformes* e *adaptativos*.
  - ✓ Estruturação, que envolve a colagem dos retalhos para formar uma malha.





# Malha de Polígonos

- Coleções de polígonos (ou faces) que, juntos, formam a “pele” ou “casca” do objeto;
  - ✓ B-Rep;
  - ✓ Forma rápida e prática para representar objetos





# Método Matemático

- O método matemático de modelagem usa uma descrição matemática e algoritmos para gerar um objeto.



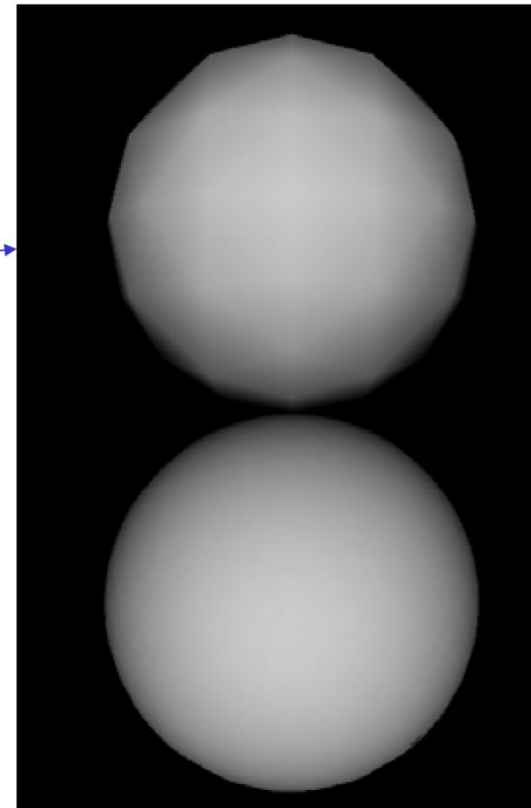
# Splines VRML

## Splines VRML (Extensão Cortona)

```
#VRML V2.0 utf8
Viewpoint { description "Initial view" position 0 0 9 }
NavigationInfo { type "EXAMINE" }

# No Spline
Transform {
  children Shape {
    appearance Appearance { material Material { } }
    geometry Sphere {
      radius 2
    }
  }
}

# Spline - Cortona Extension
Transform {
  translation 0 -4 0
  children Shape {
    appearance Appearance { material Material { } }
    geometry SplineSphere {
      radius 2
      distance 5
      quality [1 0.5]
    }
  }
}
```







# Método Matemático

- **Motivação: representar curvas suaves do mundo real.**
- **A representação por malha poligonal é uma aproximação:**
  - ✓ A curva é aproximada por uma sequência de segmentos lineares.
  - ✓ Grande quantidade de dados (vértices) para obter a curva com precisão.
  - ✓ Difícil manipulação para mudar a forma da curva, i.e. necessário posicionar vários pontos com precisão.



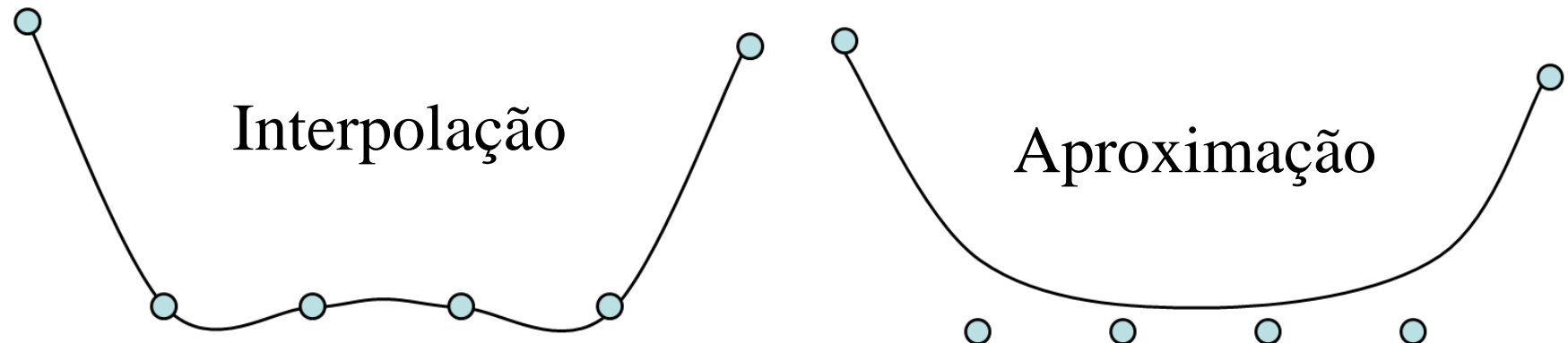
# Curvas de Bézier

- O conceito matemático foi desenvolvido pelo francês Pierre Bézier para a indústria automobilística (Renault), nos anos 60.
- Casteljau formulou, nesta época, sem contato com Bézier, um método de geração desta curva, de modo diferente de Bézier, para Citroen.
- Definições preliminares: Interpolação x Aproximação e *Representações de curvas;*



# Interpolação x Aproximação

- É natural querermos modelar uma curva suave que passa por um conjunto de pontos dados;
- Entretanto, o resultado nem sempre é o esperado (oscilações);
- É mais comum querermos curvas que “passem perto” dos pontos dados, isto é, *aproximações*;





# Representação de Curvas

- Existem basicamente 3 formas de representação para curvas:
  - ✓ a explícita,
  - ✓ a implícita;
  - ✓ e a paramétrica.



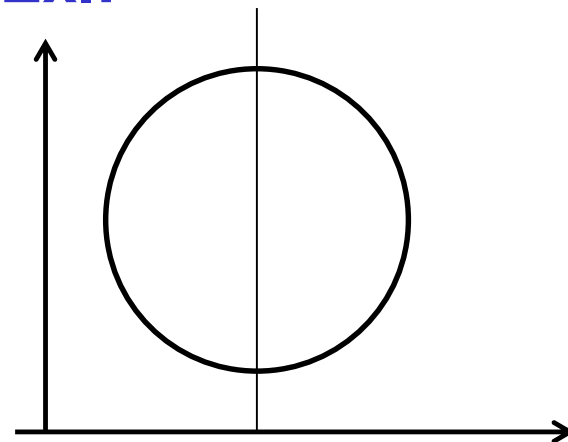
# Representação explícita

- Ocorre quando dado explicitamente uma das posições da coordenada se obtém um único valor para as demais posições,  $y = f(x)$  e  $z = g(x)$ . Ex.:

$$r: y = 2x - 1$$

$$s: y = x + 4$$

$$z = 2$$



- só existe um valor de  $y$  para cada valor de  $x$  ou  $z$
- Inviabiliza a representação de curvas fechadas





# Representação implícita

- As curvas são soluções de equações da seguinte forma  $f(x,y,z) = 0$ . *Ex.*

$$x^2 + y^2 - r^2 = 0 \text{ (Circunferência);}$$

$$x^2 + y^2 + z^2 - r^2 = 0 \text{ (Esfera)}$$

- Há dificuldade em estabelecer a direção da tangente no ponto de encontro de duas curvas;
- Não se apresenta distribuída uniformemente;
- Explícita e implícita dependem dos eixos



# Representação paramétrica

- O valor de cada variável espacial  $x$ ,  $y$  e  $z$  em função de uma variável independente  $t$ , que é chamada de parâmetro;

$$x = x(t); y = y(t); z = z(t);$$

- Esta representação é usada na curva de Bézier;



# Intuição sobre a curva de Bézier

- *“Como criar uma curva que começa em um ponto, termina em outro, e a sua forma depende de pontos de controle, que irão “puxar” ou “afastar” a curva de suas proximidades?”.*



# Intuição sobre a curva de Bézier

- Uma resposta plausível para esta questão seria que temos de construir uma função que estabeleça o “peso” que cada ponto de controlo em cada momento ao longo da curva.
- Funções utilizadas para esse efeito é o *Polinómio de Bernstein*.



# Polinómio de Bernstein

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

■ Onde:

✓  $\binom{n}{i}$  é o binómio de Newton dado por  $\binom{n}{i} = \frac{n!}{i!(n-i)!}$

- ✓  $i = 0, 1, 2, \dots, n$ , representa o índice do ponto de controlo;
- ✓ o parâmetro  $t \in [0, 1]$ .
- ✓ Define-se que  $0^0 = 1$ .





# Curvas de Bézier

- São dadas, em notação reduzida, por:

$$C(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i$$

- Com:  $C(t) = [x(t) \quad y(t) \quad z(t)]$

- Notação expandida:

$$\begin{cases} x(t) = \sum_{i=0}^n B_{i,n}(t) \cdot x_i \\ y(t) = \sum_{i=0}^n B_{i,n}(t) \cdot y_i \\ z(t) = \sum_{i=0}^n B_{i,n}(t) \cdot z_i \end{cases}$$



# Curvas de Bézier Lineares

- São de grau 1 ( $n = 1$ ), utilizam dois pontos de controle  $P_0$  e  $P_1$ , e são dadas pela equação

$$C(t) = (1 - t)P_0 + tP_1$$

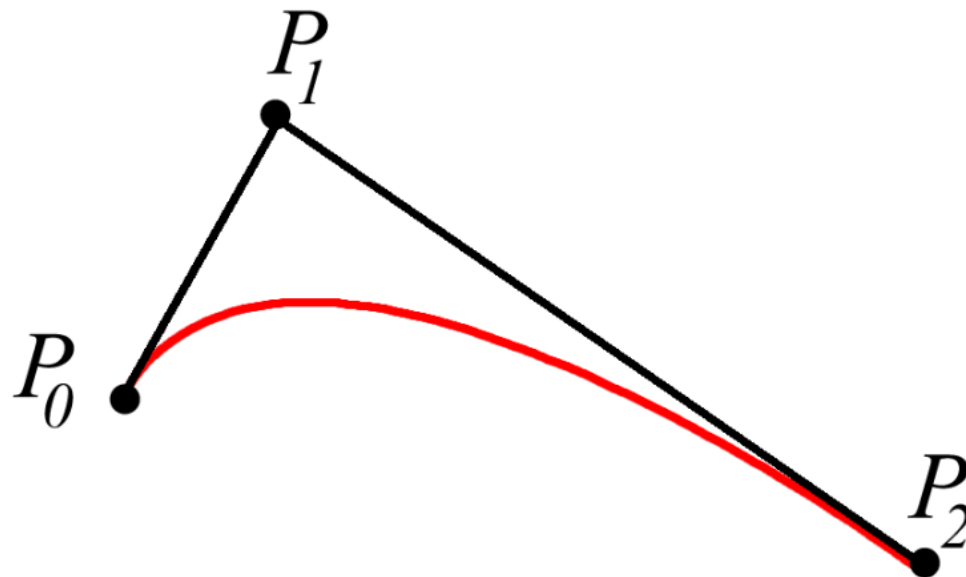




# Curvas de Bézier Quadráticas

- São de grau 2 ( $n = 2$ ), utilizam três pontos de controle  $P_0$ ,  $P_1$  e  $P_2$ , e são dadas pela equação

$$C(t) = (1 - t)^2 P_0 + 2t(1 - t)P_1 + t^2 P_2$$

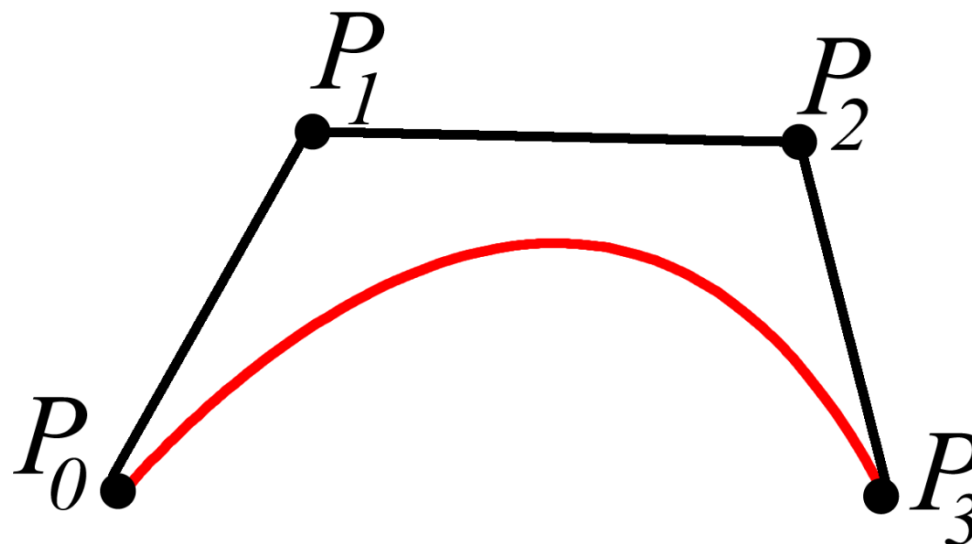




# Curvas de Bézier Cúbicas

- São de grau 3 ( $n = 3$ ), utilizam quatro pontos de controle  $P_0$ ,  $P_1$ ,  $P_2$  e  $P_3$ , e são dadas pela equação:

$$C(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$



- São as mais usadas.



# Notação Matricial

$$C(t) = [1 \quad t \quad t^2 \quad t^3] \times M_B \times \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

**Com**

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





# Observações

- A soma dos coeficientes que compõem sua equação matemática da curva de Bézier, os polinômios de Bernstein, é sempre igual a 1;
- Essa propriedade é chamada de *propriedade normalizante*;
- A curva gerada fica inteiramente dentro de uma figura convexa definida pelos pontos de controle  $P_i$ , denominado de *polígono convexo de Bézier*;
- Se o primeiro e o último ponto de controle de uma curva coincidem, então temos uma curva de Bézier fechada.



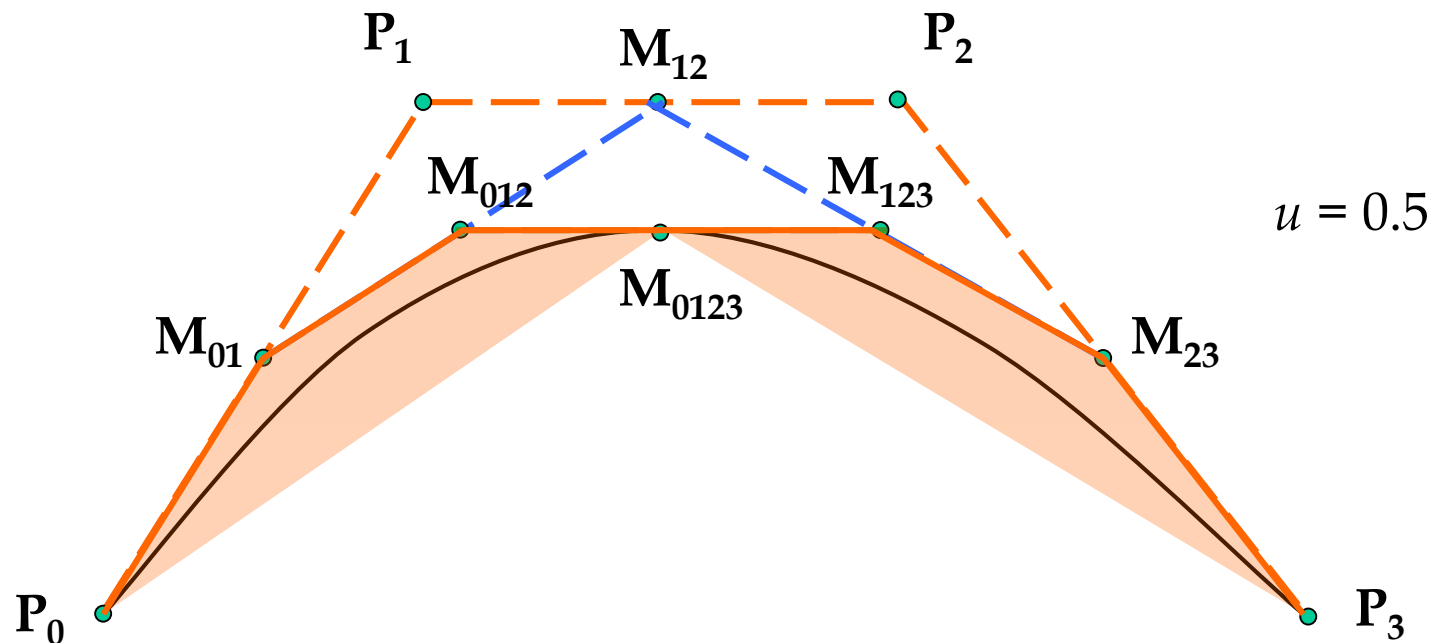
# Desenhando a curva

- Graficamente os pontos das curvas de Bézier podem ser obtidos pela simples variação do parâmetro  $t$ , de 0 a 1, na função paramétrica  $C(t)$ .
- No entanto para as curvas cúbicas tem-se um método que apresenta maior eficiência computacional, o *algoritmo de Casteljau*.



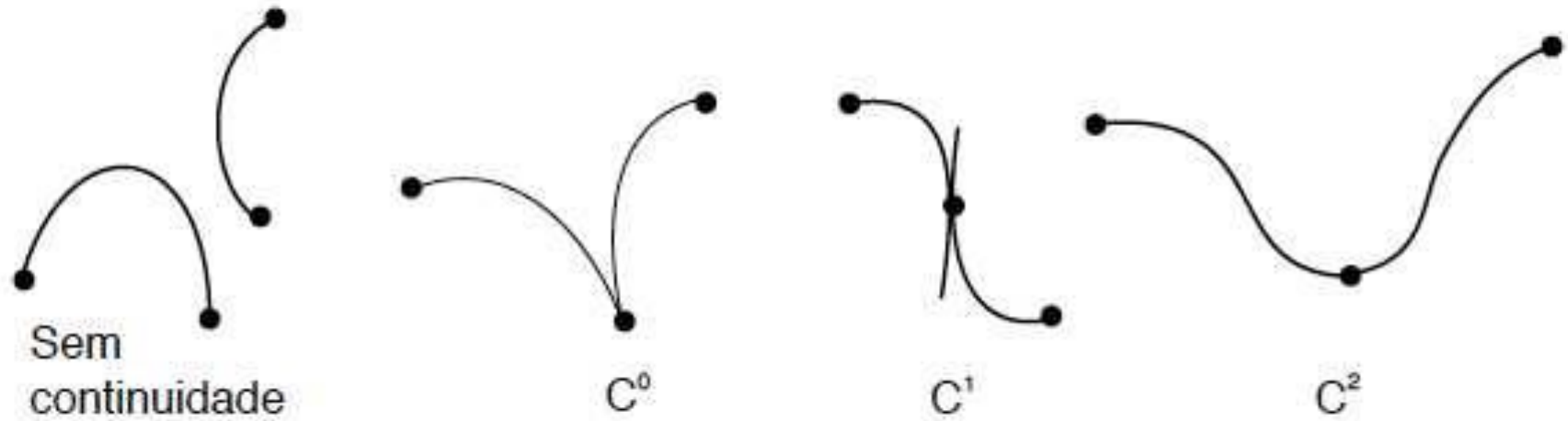
# Algoritmo de Casteljau

- Baseia-se na subdivisão recursiva dos segmentos medianos de reta criados a partir da união dos pontos de controle da curva





# Conectividade e continuidade das curvas de Bézier



## ■ Continuidade de ordem:

- ✓ 0 indica que as curvas se encontra no ponto,
- ✓ 1 que há continuidade na derivada primeira;
- ✓ 2 que há continuidade na derivada segunda;



# Quarto Trabalho

- **Neste trabalho você deve (INDIVIDUALMENTE):**
  1. Desenvolver um programa que construa graficamente a curva de Bézier por meio dos algoritmos: da **equação paramétrica** e de **Casteljau**.
  2. Construir um relatório que descreva a construção e os resultados de maneira comparativa.
  3. Apresentar o programa desenvolvido e entregar o relatório digital na sala do professor;
    - Obviamente você será arguido nesse momento.





# Quarto Trabalho

## ■ OBSERVAÇÕES:

- ✓ Objetivo do trabalho é que você veja os algoritmos trabalhando e os compare, desse modo, implemente de forma que isso aconteça;
- ✓ Trabalhos entregues sem defesa não serão aceitos (receberão nota zero (0)).
  - Dessa forma, **NÃO** adianta apenas fazer o trabalho e enviar por e-mail;
  - **Muita atenção:** não sou uma PJ ou PF precisando de software de rasterização, com isso em mente apresente com o intuito de:
    - Provar que você entende os algoritmos;
    - Provar que você realmente é o autor do programa.



# Quarto Trabalho

## ■ OBSERVAÇÕES (Cont...):

- ✓ Trabalhos entregues sem o relatório serão avaliados em cinquenta por cento (50%) da nota;
- ✓ Trabalhos entregues após a data final estabelecida não serão aceitos, a não ser com a apresentação de atestado médico ou declaração de serviço militar;

## ■ DICAS:

1. Escolha a linguagem de programação que você mais domina e que você entende que **não** será um fator limitante no desenvolvimento do trabalho;
2. Comece a desenvolver o trabalho hoje e não deixe para apresentar no último dia.



# Dúvidas

UFRR - Departamento de Ciência da Computação  
Computação Gráfica - Prof. Dr. Luciano F. Silva

