**Modelagem Geométrica**

**Bibliografia**

FOLEY, James. Computer Graphics: Principles and Pratice in C. 2 Ed.

STROUD, Ian. Boundary Representation modeling techniques.

**Fundamentos matemáticos**

**Matriz**: rows x cols  
**Translação**: Somar nas coordenadas  
**Rotação**: Multiplicar com senos e cossenos  
**Escalar**: Multiplicar pela identidade \* S  
**Reflexão**  
**Cisalhamento**

**Modelo**

Uma abstração do real.  
Mundo Físico -> Modelo Matemático -> Representação -> Implementação  
deve der capaz de criar, representar (em um conjunto mínimo de dados) e analisar/simular um objeto.

**Como se obtém modelos geométricos?**

Através de sistemas gráficos. São classificados em 2D, 2,5D e 3D

**Classificação Modelos Geométricos manufaturados rígidos**

* Sólidos
* Não sólidos
* Superfícies

**Objetos 3D**

Superfícies, sólidos, objetos tridimensionais não-sólidos e objetos e fenômesos naturais

**Modelagem de Sólidos**

B-rep e CSG

* Diferenças com superfícies
* Uma coleção de superfícies não necessariamente forma um sólido.
* Interessante para verificar se algo está dentro ou fora e para colorização automática de objetos com recursos de trabsparência e/ou translucência
* Representação não ambígua dos objetos

**Modelagem de Superfícies**

Bezier, polinômios naturais, B-spline e NURBS.

* Superfícies não tem espessura nem volume

**Sistemas 2,5D**

Extensão direta dos programas 2D para “acomodar” a informação 3D

**Representação de sólidos**

**Wireframes**

Ambíguo, usa linhas pra representar o encontro das faces.  
Não serve para objetos muito complexos que tem muitas arestas.

**Curvas**

Para engenharia e descrição de fenômenos físicos

**Representação de curvas**

1. **Conjunto de pontos**: representa-se uma curva por um conjunto de pontos, geralmente medidas experimentais em coordenadas cartesianas (René Descartes - 1637) espaçadas adequadamente.
   * Em regiões de grande cobertura, aumentar a densidade de pontos
2. **Representação analítica**: uma ou mais equações.
   * Vantagem sobre 1: Precisão, armazenamento compacto, facilidade de cálculo exato de novos pontos intermediários, facilidade de calcular propriedades das curvas como **inclinação** (primeira derivada) e **curvatura** (segunda derivada) - representação por pontos seria derivação numérica, facilidade dpara fazer alterações no formato da curva de forma a atender requisitos do projeto.
   * Ajuste de curvas (*curve fitting*): Representação analítica para conjunto de pontos. Ex.: Digitalizados.
     + Problema de interpolação
     + A curva que ajusta os pontos passa por todos estes pontos
     + Técnica usual: splines cúbicas (aproximação polinomial por partes)

**Aproximação de curvas (*curve fairing*)**

Os pontos dados são aproximações para valores desconhecidos (ex. obtidos em medidas experimentais). Deseja-se uma curva que mosra a **tendência** dos dados, a curva pode não passar por nenum dos pontos dados. Aproxima (*fair*) os dados.

* Técnicas usuais: Bèzier e B-spline

2.1 **Representação paramétrica e não paramétrica (explícita/implícita)**

* Não paramétrica explícita: y=f(x)
  + Ex.: reta, parábola, 1/4 de círculo, polinômios
  + Fáceis de integrar, derivar, combinar, avaliar valor em um ponto.
  + Um valor de y para cada valor de x -> curvas fechadas ou valores múltiplos não podem ser representados explicitamente.
* Não paramétrica implícita: f(x,y) = 0
  + Ex.: Equação impĺicita de 2º grau genérica: ax² + bx + cy² + dx + ey + f = 0 (engloba as cônicas 2D: parábola, hipérbole, elipse)

**Limitações das não paramétricas**

* Dependentes de sistema de coordenadas
* Pontos a pardir de incrementos uniformes de x e y não estão uniformemente distribuídos ao longo da curva -> qualidade baixa de representação gráfica
* Paramétrica: curva representada em função de um parâmetro
  + Ex.: curva 2D com parãmetro t as coordenadas são: x=x(t) e y=y(t)
    - Vetor com a posição do ponto p(t) = [x(t), y(t)]
    - Derivada ou valor tangente: p'(t) = [x'(t), y'(t)]
    - Inclinação: dx/dy = (dx/dt)/(dy/dt) = x'(t)/y'(t)
  + Adequada para curvas fechadas e de múltiplos valores
  + Indepente do sistema de coordenadas
  + Os extremos e o comprimento da curva são fixos pelo intervalo da variação do parâmetro (em geral normalizado)
  + Determinar y dado x é trivial na representação explícita. Na paramétrica, é necessário obter o parâmetro t a partir de x.
  + Ex.: segmento de reta (P(t) = (1-t)P1 + P2), círculo no 1º quadrante (x=cos(t), y=sen(t)) - ou versão mais barata

**Bézier**

Em computação gráfica/MoG usa-se em geral Bézier cúbica -> 4 pontos de controle - passa para o primeiro e o último e uda os outros para construir as tangentes.

Curva paramétrica de Bézier é definida como:

P(t) = sum i=0..n BiJni(t) 0 <= t <= 1

* J = funções que combinam a influência de todos os pontos (*blending functions*)
* B = representa cada um dos n+1 pontos de controle

**Polinômios de Bernstein**

https://i.imgur.com/k6dy41i.png

PS: coeficientes binomiais

1. Jni(t) >= 0 para 0 <= t <= 1
2. sum i=0..n Jni(t) = 1 para 0 <= t <= 1 (propriedade normalizante, força a curva a ficar dentro da figura convexa (Convex Hull) definida pelos pontos extremos e de controle.

Curvas fechadas: pontos extremos são coincidentes.

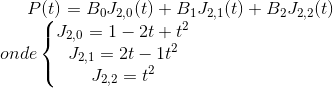
Observação: Curvas de Bézier tem controle global. Isso implica que movendo-se um só ponto é modificada toda a forma da curva.

**Soluções**: Aumentar bastante o número de pontos. Porém expressões podem se tornar complexa devido aos polinômios de ordem superior.

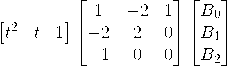
**Alternativa**: Quando muitos pontos são necessários:

* Fazer a conexão de vários segmentos de curvas de graus menores
* Para que duas curvas tenham a mesma inclinação no ponto de junção (derivadas contínuas) deve-se ter 3 pontos de controle em uma mesma reta.
* Continuidade no ponto de junção
  + De ordem 0 -> 2 curvas se econtram (C^0)
  + De 1ª ordem -> Exige que tenham tangentes comuns no ponto (C^1)
  + De 2ª ordem -> Exige que tenham curvaturas iguais (C^2) (derivadas de segunda ordem iguais).
  + De n ordem -> Derivadas de ordem n iguais

Supondo 3 pontos B0, B1, B2 (Polinômios de grau n=2)

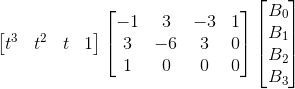


Forma mais compacta: P(t) = T Mb Gb. Onde T é a matriz da potência dos parâmetros, Mb é a matriz de Bézier e Gb são as condições geométricas (influência nos pontos)



Para 4 pontos de controle (grau n=3)

https://i.imgur.com/ZeJ6ojy.png



* Funções bases são reais
* A forma da curva geralmente acompanha o polígono de definição (é uma versão “suavizada” do mesmo)
* Para desenhar a curva, define-se o polígono
* Curva contida no fecho convexo do polígono
* Invariante às transformaçoes **afins** (translação, escala, rotação)
* Não são invariantes às transformações projetivas.

**Splines**

Schoemberg (1969)  
Régua Flexível: curvas livres suaves C^2 (curvaturas contínuas)  
Expressão matemática: spline cúbica natural -> altera um ponto, altera toda a curva -> controle global  
Curvaturas continuas são importantes (geralmente) -> usar preferencialmente splines cúbicas

Forma geral da curva B-Spline (Semelhante a Bézier):

P(t)=sum[i=0..n]BiNik(t)

Onde K controla a continuidade da curva e n é o nº de pontos de controle.  
Ex.: Nik representa as funções de grau (K-1) e curvas de continuidade C^(K-2)

Cada uma das funções Nik(t) é definida de forma recursiva:

Nik(t) = 1 se ti <= t < t[i+1] senão 0  
Nik(t) = (t-ti)/(t[i+k-1] - ti) Ni[k-1](t) + (t[i+k] - t)/(t[i+k] - t[i+1]) N[i+1][k-1](t)

Onde t0 <= t1 <= … <= t[n+k] são os nós da parametrização.

Propriedades

* P é um polinômio de grau K-1 em cada intervalo [ti, t[i+1]], i = 0,…,n+k-1
* A derivada de ordem k-2 é contínua em [t0, t[n+k]]
* sum[i=0…n] Nik(t) = 1 para todo t pertencente a [t0, t[n+k]] e 0 <= k <= n+1.
* Nik(t)>=0 para t pertencente a [t0, t[n+k]] e 1 <= k <= n+1
* A curva está contida no fecho convexo do polígono de controle
* A curva é invariante, quando submetida a transformações AFINS
* Cada ponto de controle é associado a uma única função BASE, influenciando a forma da cuva apenas em uma região.
* Quando K=n+1, a curva B-spline coincide com a Bézier
* A curva ão oscila com mais frequência em relaão a qualquer reta que seu polígono de controle

**Tipos de conjuntos de nós**

* **Não periódicos**: Nós extremos tem multiplicidade k e os demais são igualmente espaçados. Usados para gerar curvas abertas.
* **Uniforme periódico**: Os nós são igualmente espaçados. Ex.: [-2,0,2,4,6] delta t = 2
* **Não uniforme**: Os nós **não** são igualmente espaçados e podem ter qualquer multiplicidade. Ex.: [0.2,0.4,0.5,0.6,0.8,1.0]

**Para interpolações quadráticas (K=3)**

N[i,3](t) = (t - ti)/(t[i+2]-ti) \* N[i,2] + (t[i+3] - t)/(t[i+3]-t[i-1]) \* N[i+1,2](t) - recursivo, só ver a fórmula genérica para k.

**Catmull-row Splines**

Curva passa por todos os pontos de controle (não é comum para uma spline).

Ex.: 4 pontos de controle  
q(t) = 1/2[t^3 t^2 t 1] \* M \* vetor\_pontos  
com M:

-1 3 -3 1

2 -5 4 -1

-1 0 1 0

0 2 0 0

* Contínua em C^1 (tangentes)
* Não é contínua em C^2 (curvatura)
* Pode ter partes fora do convex hull

**Curvas Racionais**

Razão de 2 polinômios.  
Importância: invariantes às transformações projetivas.

* Sob projeção perspectiva, continuam sendo racionais
* Invariância: camadas **homogêneas** = ponto no espaço 3D -> elemento do espaço 4D homogêneo (wx,wy,wz,w) onde w é a coordenada homogênea ou peso. w > 0
* Projeção do ponto de controle Bi^w do 4D para o 3D é dado: Bi = Bi^w/w[i] => Bi^w = w[i]B[i]

**Superfícies**

**Interpolação Bilinear**

* Duas interpolações lineares

**Fronteiras definidas para curvas**

* Lofting vertical ou horizontal
* Interpola com uma reta pontos de uma curva para outra
* [Modelagem Geométrica](https://hackmd.io/@TkzUwcZzSLedNoRz5_GTMA/rJnJfy5HE?type=view#Modelagem-Geom%C3%A9trica)
* [Objetos 3D](https://hackmd.io/@TkzUwcZzSLedNoRz5_GTMA/rJnJfy5HE?type=view#Objetos-3D)
* [Curvas](https://hackmd.io/@TkzUwcZzSLedNoRz5_GTMA/rJnJfy5HE?type=view#Curvas)
* [Superfícies](https://hackmd.io/@TkzUwcZzSLedNoRz5_GTMA/rJnJfy5HE?type=view#Superf%C3%ADcies)