Capítulo 8

Curvas e Superfícies em Computação Gráfica

Curvas e superfícies tridimensionais (espaciais) desempenham um papel importante:

- na engenharia, projeto e manufatura de uma ampla gama de produtos, como automóveis, cascos de navios, fuselagem e asas de aviões, lâminas de propulsão, sapatos, garrafas, edificações, etc.
- na descrição e interpretação de fenômenos físicos em áreas como geologia, física e medicina.
- **Projeto Geométrico Apoiado por Computador** (*Computer Aided Geometric Design*): nova disciplina que incorpora modelos matemáticos e computacionais desenvolvidos para apoiar os processos de engenharia, projeto e manufatura.

Frequentemente, superfícies são descritas por uma malha de curvas definidas em planos ortogonais (Figura 8.1). As curvas podem ser obtidas através da digitalização de um modelo físico ou desenho, e posterior ajuste de uma curva matemática aos pontos digitalizados, ou geradas a partir de formulações criadas especificamente para a definição de curvas no espaço.

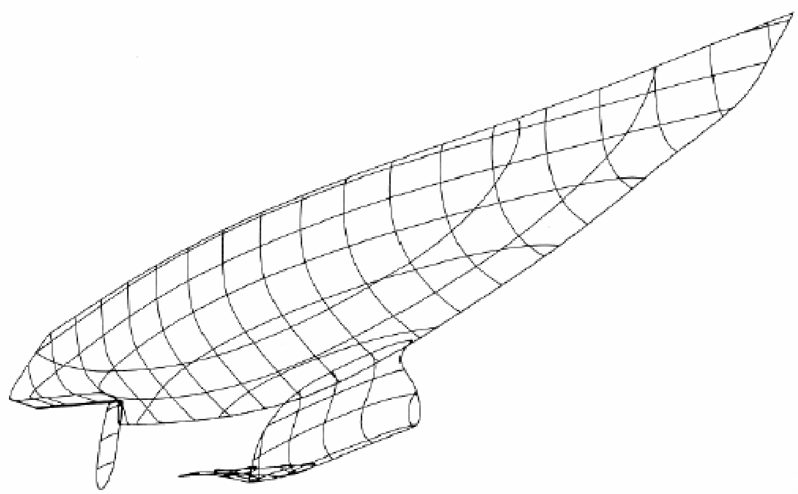

Figura 8.1: Representação de superfícies através de malhas.

8.1 Representação de Curvas

Existem duas formas alternativas de representação:

- por um conjunto de pontos: nesse caso, uma representação visual adequada pode ser gerada conectando-se os pontos por segmentos de reta, se estes forem espaçados adequadamente (Figura ??). Observe que nas regiões da curva onde a curvatura é grande, a proximação por segmentos de reta é particularmente ruim (Figura ??(a)). Uma maneira de melhorar a representação é aumentar a densidade dos pontos nessas regiões (Figura ??(b)).

- analítica: ou seja, através de formulações matemáticas, o que apresenta várias vantagens em relação à representação anterior:
- precisão,
- armazenagem compacta,
- facilidade de cálculo (exato) de pontos intermediários (em uma representação por pontos pontos intermediários precisam ser determinados por interpolação),
- facilidade para calcular propriedades da curva como inclinação e curvatura (em uma representação por pontos tais propriedades precisam ser calculadas por diferenciação numérica, um procedimento que é pouco preciso),
- facilidade para desenhar as curvas,
- facilidade para fazer alterações contínuas no formato da curva de forma a atender requisitos de projeto ("design").

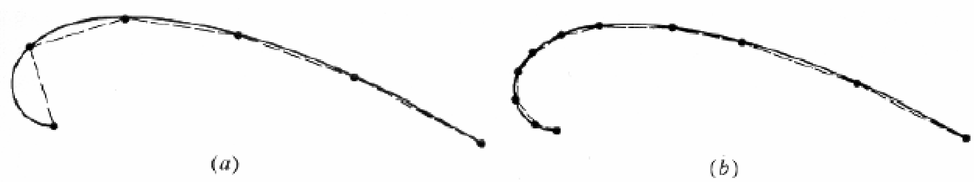

Figura 8.2: Aproximação de curvas por segmentos de reta conectados.

8.2 Curve Fitting x Curve Fairing

8.2.1 Ajuste de curvas (curve fitting)

Frequentemente, deseja-se obter representações analíticas para curvas definidas originalmente por conjuntos de pontos - por exemplo, pontos em uma curva ou superfície real foram digitalizados. Do ponto de vista matemático, esse é um problema de **interpolação**. Uma curva que **ajusta** (*fit*) os pontos dados passa por todos esses pontos. Uma técnica usual de ajuste de curvas são as **splines cúbicas**, uma estratégia de aproximação polinomial por partes.

8.2.2 Aproximação de curvas (curve fairing)

Se os pontos dados são aproximações para valores desconhecidos, por exemplo, são pontos coletados ou obtidos em medidas experimentais, então o que se deseja é uma curva que mostre a tendência dos dados. Em geral, a curva obtida pode não passar por nenhum dos pontos dados, e diz-se que a curva **aproxima** (*fair*) os dados. Alternativamente, pode-se desejar gerar uma descrição matemática de uma curva no espaço sem qualquer conhecimento prévio da forma da curva. Para esse caso, técnicas usuais são as representações de Bézier e B-splines (ambas são estratégias de **aproximação de curvas**).

8.3 Representações Paramétricas e Não Paramétricas (explícita e implícita)

Para uma curva plana, a forma não paramétrica explícita é dada por

$$y = f(x) \tag{8.1}$$

por exemplo, a equação de uma linha reta é dada por

$$y = mx + b \tag{8.2}$$

Desta forma, obtém-se um valor de y para cada valor de x dado. Consequentemente, curvas fechadas, ou com valores múltiplos, como um círculo, não podem ser representadas explicitamente. Essa limitação não existe no caso de representações implícitas, na forma

$$f(x, y) = 0 \tag{8.3}$$

A equação implícita de segundo grau genérica:

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \tag{8.4}$$

engloba uma variedade de curvas bi-dimensionais denominadas **seções cônicas**. Os três tipos de seções cônicas são a **parábola**, a **hipérbole** e a **elipse** (um círculo é um caso especial de uma elipse) (Figura ??).

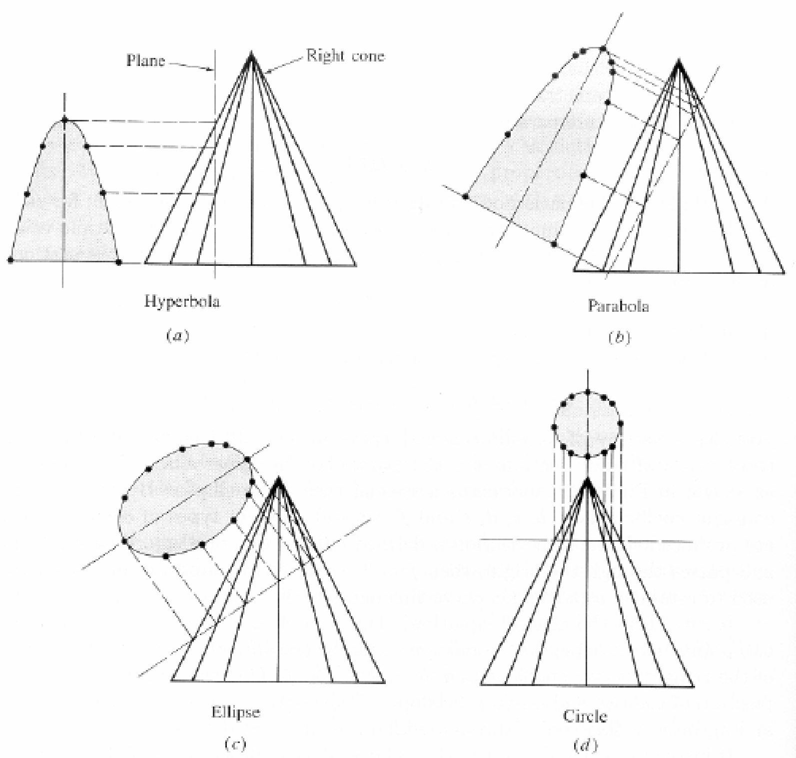

Figura 8.3: Aproximação de curvas por segmentos de reta conectados.

Dependendo dos valores de a, b, c, d, e e f, diferentes tipos de seções cônicas são produzidas. Se a seção cônica é definida em relação a um sistema de coordenadas local e passa pela origem, então $f = 0$.

8.3.1 Limitações das representações não paramétricas

- Ambas (explícita e implícita) são dependentes do sistema de coordenadas, cuja escolha afeta a facilidade de uso. Por exemplo, surgem problemas se no sistema de coordenadas escolhido for necessária uma inclinação infinita como condição de contorno, pois essa inclinação não pode ser usada diretamente como uma condição de contorno numérica. Ou o sistema de coordenadas é alterado, ou a inclinação infinita será representada numericamente por um valor muito grande, positivo ou negativo.

- pontos em uma curva calculados a partir de incrementos uniformes em x ou y não estão distribuídos uniformemente ao longo da curva, o que afeta a qualidade e precisão de uma representação gráfica. Estas limitações são superadas pelo uso de representações paramétricas.

Na forma paramétrica, cada coordenada de um ponto em uma curva é representada como uma função de um único parâmetro, sendo que a posição de um ponto na curva é fixada pelo valor do parâmetro. Para uma curva 2D que usa t como parâmetro, as coordenadas cartesianas de um ponto na curva são:

$$\begin{aligned} x &= x(t) \\ y &= y(t) \end{aligned} \tag{8.5}$$

O vetor que define a posição de um ponto na curva é portanto dado por

$$P(t) = [\, x(t) \quad y(t)\,] \tag{8.6}$$

A derivada, ou vetor tangente da curva é dada por

$$P'(t) = [\, x'(t) \quad y'(t)\,] \tag{8.7}$$

A inclinação da curva é dada por

$$\frac{dx}{dy} = \frac{\frac{dx}{dt}}{\frac{dy}{dt}} = \frac{x'(t)}{y'(t)} \tag{8.8}$$

Observações

- A forma paramétrica é adequada para representar curvas fechadas e com valores múltiplos. A forma não paramétrica pode ser obtida eliminando-se o parâmetro para obter uma única equação em x e y.
- Uma vez que um ponto na curva é especificado por um único valor de parâmetro, a forma paramétrica é independente do sistema de coordenadas. Os extremos e o comprimento da curva são fixos pelo intervalo de variação do parâmetro, frequentemente normalizado para , por conveniência. Como as curvas são independentes do sistema de coordenadas, elas são facilmente manipuladas usando as transformações geométricas afins (ver Capítulo 5).
- Determinar um ponto em uma curva, i.e., determinar o valor de y, dado x, é trivial no caso da representação explícita. No caso da paramétrica, é necessário obter o valor do parâmetro t, a partir de x, e a seguir usar este valor para obter y. (Para equações paramétricas mais complexas, uma técnica iterativa pode ser mais conveniente).
- Ambas as formas de representação têm vantagens e desvantagens em situações específicas!

Exemplos:

1. **segmento de reta**

   Para 2 vetores que especificam as posições iniciais P1 e P2, uma representação paramétrica do segmento é dada por

   Como $P(t)$ é um vetor de posição, cada um de seus componentes tem uma representação paramétrica $x(t)$ e $y(t)$ entre $P1$ e $P2$.

2. **Círculo no primeiro quadrante**

   Uma comparação das representações visuais obtidas a partir das formas paramétrica e não-paramétrica para um círculo no primeiro quadrante é apresentada na Figura 8.4 (observe que a representação paramétrica para uma curva não é única!)

   Os pontos na circunferência mostrados nas Figuras 8.4(b) e 8.4(c) foram obtidos com intervalos constantes dos parâmetros q e t, respectivamente. A primeira representação produz comprimentos de arcos idênticos ao longo da circunferência, ao contrário da segunda (a qual, entretanto, tem aparência visual melhor que a mostrada em 8.4(a), gerada pela representação explícita). Por outro lado, o cálculo das funções trigonométricas é computacionalmente caro, de maneira que a segunda forma de representação paramétrica representa um meio termo entre a forma explícita e a forma paramétrica padrão.

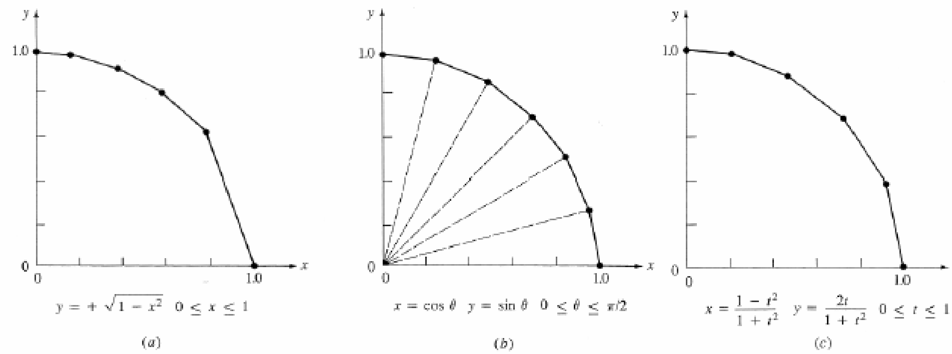

Figura 8.4: Pontos de uma circunferência.

8.4 Curvas de Bézier

Técnicas de aproximação de curvas são muito usadas em ambientes de projeto (CAD) interativos, por serem mais intuitivas do que as técnicas de ajuste. Um método adequado para o design de curvas e superfícies de forma-livre em ambientes interativos foi desenvolvido por Pierre Bézier. Uma curva de Bézier é determinada por um polígono de definição (Figuras 8.5 e ??).

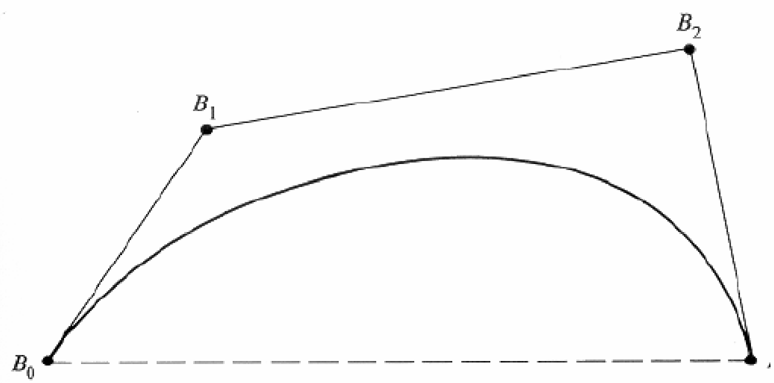

Figura 8.5: Representação de uma curva de Bézier definida pelos pontos B_0, B_1, B_2 e B_3.

Matematicamente, uma curva de Bézier paramétrica é definida como

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \qquad 0 \le t \le 1 \tag{8.9}$$

onde as funções de blending, ou funções base de Bézier são

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{8.10}$$

com

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{8.11}$$

$J_{n,i}(t)$ é a i-ésima função base de Bernstein de ordem n, onde n, o grau da função-base, e portanto do segmento de curva polinomial, é um menos o número de pontos do polígono de definição. Os vértices do polígono são numerados de 0 a n (como na Figura 8.5). A Figura 8.6 mostra as funções de blending

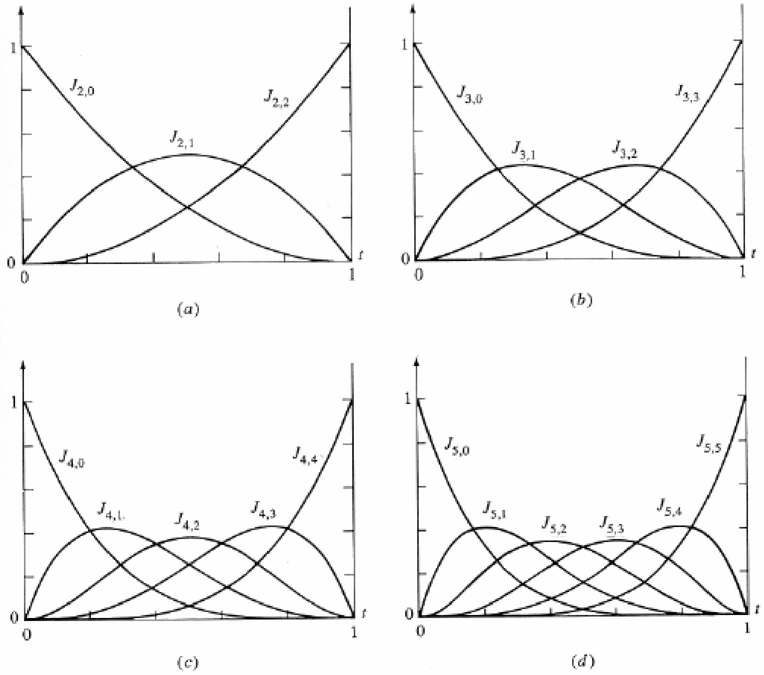

Figura 8.6: Funções de blending para vários valores de n.

para vários valores de n. Observe a simetria das funções. Examinando-se as Equações 8.9 a 8.11 para o primeiro ponto na curva, i.e., em $t = 0$, verifica-se que

$$\begin{array}{ll} J_{n,0}(0) = 1 & i = 0 \\ J_{n,i}(0) = 0 & i \neq 0 \end{array} \tag{8.12}$$

Portanto,

$$P(0) = B_0 J_{n,0}(0) = B_0 \tag{8.13}$$

e o primeiro ponto na curva coincide com o primeiro ponto do polígono de controle. Verificação análoga pode ser feita para o último ponto, i.e., em t = 1. As funções base da Figura 8.6 ilustram estes resultados.

Além disso, pode-se mostrar que para qualquer valor do parâmetro t, o somatório das funções base é 1, i.e.,

$$\sum_{i=0}^{n} J_{n,i} = 1 \tag{8.14}$$

A equação para uma curva de Bézier pode ser expressa na forma matricial como

$$P(t) = TNG = FG \tag{8.15}$$

onde $F = J_{n,0} J_{n,1} \cdots J_{n,n}$ e $G^r = B_0 B_1 \cdots B_n$.

As matrizes específicas para valores pequenos de n (n = 3, 4) são de particular interesse. Para qualquer valor de n, a matriz [N] é simétrica em relação à diagonal principal e o canto triangular inferior direito contém apenas zeros.

Em geral, uma forma complexa não pode ser modelada por uma única curva, mas por várias curvas que são conectadas em seus pontos extremos. Ao criar as junções, o projetista, em geral, deseja controlar a continuidade nos pontos de junção. Continuidade de ordem 0 significa que as 2 curvas se encontram; continuidade de primeira ordem exige que as curvas sejam tangentes no ponto de junção, e continuidade de segunda ordem exige que as curvaturas sejam as mesmas.

A formulação de Bézier apresenta várias propriedades interessantes:

- As funções-base são reais.
- O grau do polinômio que define uma curva é um menos o número de pontos do polígono de definição.
- A forma da curva geralmente acompanha a forma do polígono de definição (na verdade é uma versão "suavizada" da forma do polígono (v. Fig. 3.3)). Assim, para desenhar uma curva, basta definir o polígono e depois ajustar os pontos que forem necessários para aproximar melhor a forma desejada. Isso torna a formulação adequada para o design interativo. Um projetista experiente consegue obter a forma desejada depois de 2 ou 3 interações com um sistema computacional.
- O primeiro e último pontos da curva coincidem com o primeiro e último pontos do polígono de definição. Em situações práticas, em geral é desejável ter controle direto sobre os pontos extremos da curva.
- Os vetores tangentes nos extremos da curva têm a mesma direção que o primeiro e o último segmentos do polígono de definição, respectivamente.
- A curva está contida no fecho convexo do polígono de definição, i.e., no maior polígono convexo que pode ser obtido com os vértices do polígono de definição (Figura 8.5). Uma consequência simples deste fato é que um polígono plano sempre gera uma curva plana.

  A importância desta propriedade está principalmente no contexto de verificação de interferência. Suponha que desejamos saber se 2 curvas se interceptam. Por exemplo, cada uma representa o caminho a ser percorrido pelo braço de um robô, e nosso objetivo é garantir que os caminhos não se interceptam, evitando colisões. Ao invés de tentar calcular uma possível intersecão, podemos fazer um teste muito mais barato: envolver o polígono de controle no menor retângulo que o contém e cujas arestas são paralelas a um sistema de coordenadas (minmax boxes). O retângulo contém o polígono de controle e, pela propriedade do fecho convexo, contém também a curva. Verficar se os 2 retângulos se interceptam é um teste trivial, e fica simples checar a não-intersecção. Se houver intersecção dos retângulos, mais verificações precisam ser feitas nas curvas. A possibilidade de detectar não-interferência de forma rápida é muito importante, pois na prática é freqüente que tais testes precisem ser feitos milhares de vezes para cada objeto.
- A curva exibe a propriedade da variação decrescente (variation diminishing property). Isto significa, basicamente, que a curva não oscila em relação a qualquer linha reta com mais frequência que o polígono de definição. Algumas representações matemáticas têm a tendência de amplificar, ao invés de suavizar, quaisquer irregularidades de formato esboçadas pelos pontos de definição, enquanto que outras, como as curvas de Bézier, sempre suavizam os pontos de controle. Assim, a curva nunca cruza uma linha reta arbitrária mais vezes que a sequência de segmentos que conectam os pontos de controle (v. Fig.)
- A curva é invariante sob transformações afins. Transformações afins estão disponíveis em qualquer sistema de CAD, pois é essencial reposicionar, escalar, etc. os objetos. Esta propriedade garante que os 2 procedimentos abaixo produzem os mesmos resultados: a) primeiro calcula um ponto na curva, e depois aplica a ele uma transformação afim; e b) primeiro, aplica uma transformação afim ao polígono de definição, e depois gera o ponto na curva.

Uma consequência prática: suponha que traçamos uma curva cúbica calculando 100 pontos sobre ela; e que agora queremos desenhar a mesma curva depois de uma rotação. Podemos aplicar a rotação a cada um dos 100 pontos, e desenhar os pontos resultantes, ou aplicar a rotação a cada um dos 4 pontos do polígono de controle, calcular novamente os 100 pontos e traçá-los. A primeira estratégia requer que a rotação seja aplicada 100 vezes, e a segunda requer a sua aplicação apenas 4 vezes! É interessante observar que as curvas de Bézier não são invariantes sob transformações projetivas.

Bibliografia

Fonte: Caps. 4,5 Rogers & Adams - Mathematical Elements for Computer Graphics