

INF1608 – Análise Numérica

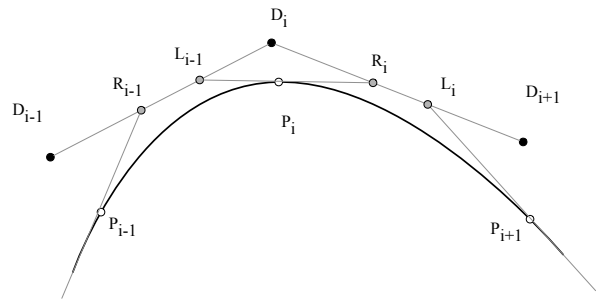
Projeto: Interpolação por Spline Cúbica

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

Descrição

Em diversas aplicações, precisamos determinar uma curva suave que interpola um conjunto de n pontos P_i , $i = 0, 1, \dots, n-1$. Em geral, utiliza-se *spline cúbica*, isto é, uma interpolação por partes onde cada tramo é representado por uma cúbica. Nos pontos interpolados, garante-se continuidade de derivada de primeira e segunda ordem (continuidade de tangente e curvatura). *Splines cúbicas* são muito utilizadas em modelagem, robótica e computação gráfica.

Este trabalho descreve um procedimento para, dado um conjunto de pontos P_i , determinar a spline cúbica. Uma spline é determinada por seus pontos de controle D_i , como ilustra a figura. De posse dos pontos de controle, podemos determinar os pontos R_i e L_i usados no traçado das curvas. A determinação dos pontos de controle recai na solução do seguinte sistema linear:



Pontos de controle D_i de uma spline

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & a_0 \\ a_1 & b_1 & c_1 & 0 & \cdots & 0 \\ \cdots & & & & & \\ 0 & \cdots & 0 & a_{n-2} & b_{n-2} & c_{n-2} \\ c_{n-1} & 0 & \cdots & 0 & a_{n-1} & b_{n-1} \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ \vdots \\ D_{n-2} \\ D_{n-1} \end{bmatrix} = \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{bmatrix}$$

Considerando curvas planas, cada ponto é dado pelas coordenadas (x, y) ; portanto, o mesmo sistema tem que ser resolvido para as coordenadas x e y . Os coeficientes a_i , b_i e c_i são dados por:

$$\begin{aligned} a_i &= (1 - \delta_i)(1 - \lambda_i) \\ b_i &= (1 - \delta_i)\lambda_i + \delta_i(1 - \mu_i) \\ c_i &= \delta_i\mu_i \end{aligned}$$

onde:

$$\delta_i = \frac{h_i}{h_i + h_{i+1}}$$

Os valores h_i , para $i = 0, 1, \dots, n$ (note que devem existir $n + 1$ valores de h), representam os espaçamentos entre os nós da spline. Se considerarmos uma spline com espaçamento *uniforme*, podemos adotar $h_i = 1$ para todos os espaçamentos entre nós internos. No entanto, splines não uniformes tendem a ter melhor ajuste. Uma heurística boa para spline não uniforme é adotar o espaçamento igual à distância entre os pontos a serem interpolados: $h_i = \|P_i P_{i-1}\|$. Se a spline for aberta, fazemos $h_0 = h_n = 0$. Se a spline for fechada, isto é, existe um tramo que interpola o último ponto ao primeiro, fazemos: $h_0 = h_n = \|\overline{P_{n-1} P_0}\|$.

Os coeficientes λ_i e μ_i , para $i = 1, \dots, n - 2$, são dados por:

$$\lambda_i = \frac{\gamma_{i-1}h_{i-1} + h_i}{\gamma_{i-1}h_{i-1} + h_i + \gamma_i h_{i+1}}$$

$$\mu_i = \frac{\gamma_i h_i}{\gamma_i h_i + h_{i+1} + \gamma_{i+1} h_{i+2}}$$

onde:

$$\gamma_i = \frac{2(h_i + h_{i+1})}{\nu_i h_i h_{i+1} + 2(h_i + h_{i+1})}, \quad i = 0, \dots, n - 1$$

Os coeficientes ν_i representam as tensões nos nós. Se fazemos $\nu_i = 0$, temos a interpolação normal de splines cúbicas; se aumentamos o valor de ν_i , tendemos a criar um corner no ponto de interpolação correspondente.

Para spline aberta, temos:

$$\lambda_0 = \lambda_{n-1} = 1$$

$$\mu_0 = \mu_{n-1} = 0$$

Com isso, a_0, c_0, a_{n-1} e c_{n-1} valem zero, e ficamos com um sistema tridiagonal simples. Para spline fechada, os valores de λ e μ devem corresponder ao do tramo interligando o último ponto ao primeiro:

$$\lambda_0 = \frac{\gamma_{n-1}h_{n-1} + h_0}{\gamma_{n-1}h_{n-1} + h_0 + \gamma_0 h_1}$$

$$\lambda_{n-1} = \frac{\gamma_{n-2}h_{n-2} + h_{n-1}}{\gamma_{n-2}h_{n-2} + h_{n-1} + \gamma_{n-1}h_0}$$

$$\mu_0 = \frac{\gamma_0 h_0}{\gamma_0 h_0 + h_1 + \gamma_1 h_2}$$

$$\mu_{n-1} = \frac{\gamma_{n-1}h_{n-1}}{\gamma_{n-1}h_{n-1} + h_n + \gamma_0 h_1}$$

Dados os pontos de controle D_i , podemos calcular os pontos que definem a spline em cada tramo: R_i e L_i .

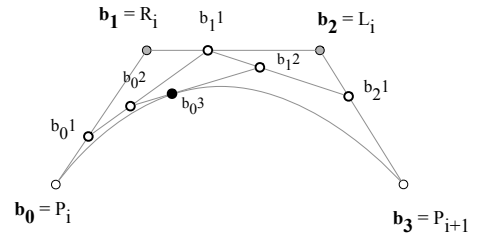
$$R_i = (1 - \mu_i)D_i + \mu_i D_{i+1}$$

$$L_i = (1 - \lambda_{i+1})D_i + \lambda_{i+1} D_{i+1}$$

Note portanto o significado geométrico dos parâmetros μ_i e λ_i : eles definem, respectivamente, a posição relativa dos pontos R_i e L_i no segmento que liga os dois pontos de controle, D_i e D_{i+1} , associados ao tramo em questão.

Os pontos $b_0 \equiv P_i$, $b_1 \equiv R_i$, $b_2 \equiv L_i$ e $b_3 \equiv P_{i+1}$ são então usados para determinar pontos $P(t)$, com $t \in [0, 1]$, ao longo da cúbica de cada tramo. Para tanto, podemos usar o algoritmo de Casteljau, ilustrado na figura, dado por:

$$P(t) = b_0^3(t)$$



Algoritmo de Casteljau.

onde:

$$b_i^k(t) = (1 - t)b_i^{k-1}(t) + t b_{i+1}^{k-1}(t)$$

Tarefa

O objetivo deste trabalho é implementar um algoritmo para interpolação de pontos por spline cúbica: dado um conjunto de n pontos P_i , determinar a spline cúbica interpolante. Este trabalho pode ser implementado considerando diferentes etapas:

1. Determinar a spline *uniforme* que interpola os pontos dados. Como sugestão de implementação, podemos pensar em definir um tipo que representa uma spline, que é instanciado com o conjunto de pontos a serem interpolados. Podemos então prever uma função para calcular a spline interpolante e uma função para calcular um ponto na spline. Esta última função pode receber dois parâmetros, i e t , onde i indica o tramo da spline e $t \in [0, 1]$, o parâmetro dentro deste tramo.
2. Determinar a spline *não uniforme* que interpola os pontos dados.
3. Experimentar a variação da tensão ν_i associada aos pontos. Avaliar os efeitos na spline resultante.
4. Considerar também spline fechadas (com o último ponto sendo conectado ao primeiro)
5. Investigar a resolução do sistema linear do problema (sistema tridiagonal ou próximo de tridiagonal, para splines fechadas) em complexidade $O(n)$.

Análise

Ao desenvolver seu trabalho e testá-lo, procure, baseado em experimentos computacionais, responder as seguintes perguntas:

- Dado um conjunto de pontos, a spline não uniforme resulta em uma curva interpolante de melhor qualidade, se comparada com a spline uniforme?
- Qual o efeito da variação das tensões?
- Seu método trata corretamente splines abertas e fechadas?
- O desempenho do método é significativamente melhorado com resolvedores lineares dos sistemas (quase) diagonais?
- Sua implementação é genérica para tratar qualquer conjunto de pontos a serem interpolados?