UFES - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
Prof. Thomas W. Rauber

2º Trabalho de Algoritmos Numéricos (INF 09269) 2016/1

Data de entrega: veja `www.inf.ufes.br/~thomas`

Linguagem de Programação para Implementação
Octave/Matlab, preferencialmente código que seja compatível com os dois ambientes
Grupo de até dois alunos

Estudo Comparativo de Métodos de Interpolação Polinomiais Globais e Locais usando Splines Cúbicos.

Problema: Dado um conjunto de $(n+1)$ nós de interpolação

$$\{(x_i, y_i), i = 0, \ldots, n\}, \tag{1}$$

o objetivo é obter uma representação da suposta função $f(x)$ que deu origem a esses nós. Uma técnica é o uso de um polinômio

$$p_n(x) = \sum_{k=0}^{n} a_k x^k \tag{2}$$

de grau n, representado por $(n+1)$ coeficientes $a_0, \ldots, a_n$. Foram estudados três métodos para obter os coeficientes que levam ao mesmo resultado.

1. Resolução de um sistema linear com o uso da matriz de Vandermonde
2. Interpolação com polinômios de Lagrange
3. Interpolação com polinômios de Newton (uso de diferenças divididas)

Todas essas técnicas são globais, ou seja, usam em geral todos os $(n+1)$ nós para obter os coeficientes.

Ao contrário disso, uma interpolação local, por partes entre dois nós consecutivos $\{(x_i, y_i), (x_{i+1}, y_{i+1})\}$, usa somente um subconjunto dos nós e junta todas a funções locais $q_i(x), i = 0, \ldots, n-1$ geradas para definir a interpolação global. Neste trabalho devem-se usar Splines cúbicos para fazer a interpolação local. O resultado deve ser comparado com a interpolação global por uma das técnicas acima mencionada.

1 Splines Cúbicos

O conceito de interpolação pode ir além da interpolação de uma função. Isso significa que a curva representada pelo spline pode se movimentar livremente no plano x-y, ou até no espaço tridimensional x-y-z. Na Computação Gráfica, splines são frequentemente usados para fins de modelagem de curvas e superfícies.

1.1 Representação

Uma representação elegante de um spline usa um parâmetro $t \in [0, 1]$ para definir todos os pontos entre dois nós consecutivos. Nesse caso dois nós consecutivos são compostos por duas componentes, sendo $(x(t=0), y(t=0))$ o nó atual e $(x(t=1), y(t=1))$ o próximo nó no plano, e $(x(t=0), y(t=0), z(t=0))$, $(x(t=1), y(t=1), z(t=1))$ dois nós consecutivos no espaço[1]. Em seguida, somente curvas no espaço bidimensional são consideradas, e, ainda por cima, as curvas são funções (ou seja, não existem dois valores y distintos para o mesmo x).

[1] Uma representação paramétrica muito conhecida é a da circunferência unitária (centro em $(0, 0)$, raio um) $(x(t), y(t)) = (\cos t, \sin t), t \in [0, 2\pi]$.

Cada componente x e y é definida independentemente por um polinômio cúbico, portanto por $a + bt + ct^2 + dt^3$. Então o i-ésimo spline é definido como

$$q_i(t) = \begin{cases} x_i(t) = a_{x,i} + b_{x,i}t + c_{x,i}t^2 + d_{x,i}t^3 \\ y_i(t) = a_{y,i} + b_{y,i}t + c_{y,i}t^2 + d_{y,i}t^3 \end{cases} , \quad i = 0, \ldots, n-1, t \in [0, 1]. \tag{3}$$

Para a análise como obter os coeficientes para cada componente x e y, basta considerar x e y separadamente. Então cada componente $x_i(t)$ e $y_i(t)$ seja a spline e as suas primeiras duas derivadas definidos como

$$s_i(t) = a_i + b_i t + c_i t^2 + d_i t^3, \tag{4}$$

$$s_i'(t) = b_i + 2c_i t + 3d_i t^2, \tag{5}$$

$$s_i''(t) = 2c_i + 6d_i t, \qquad i = 0, \ldots, n-1, t \in [0, 1], \tag{6}$$

e cada componente dos $(n+1)$ nós de interpolação x_i e y_i é representado por

$$\{s_i, i = 0, \ldots, n\}.$$

Os nós formam o início e fim dos splines, então

$$\{s_i = s_i(t = 0), s_{i+1} = s_i(t = 1), i = 0, \ldots, n\}.$$

1.2 Restrições

Para obter os parâmetros a, b, c, d para cada um dos n splines da eq. (4), exploram-se as condições de cada nó de interpolação. No início do spline $(t = 0)$ e final do spline $(t = 1)$ tem-se para o valor

$$s_i(0) = a_i \tag{7}$$

$$s_i(1) = a_i + b_i + c_i + d_i, \tag{8}$$

e para a sua derivada

$$s_i'(0) := D_i = b_i \tag{9}$$

$$s_i'(1) := D_{i+1} = b_i + 2c_i + 3d_i, \tag{10}$$

onde a derivada no início do spline foi abreviada como D_i e no final como D_{i+1}.

Resolvendo (7) a (10) para a_i, b_i, c_i, d_i resulta em

$$a_i = s_i \tag{11}$$

$$b_i = D_i \tag{12}$$

$$c_i = 3(s_{i+1} - s_i) - 2D_i - D_{i+1} \tag{13}$$

$$d_i = 2(s_i - s_{i+1}) + D_i + D_{i+1}. \tag{14}$$

O primeiro nó s_0 é o início do primeiro spline $s_0(t)$, então

$$s_0 = s_i(0), \tag{15}$$

e o último nó s_n é o final do último spline $s_{n-1}(t)$, então

$$s_n = s_{n-1}(1). \tag{16}$$

O i-ésimo nó, a partir do segundo, e até o penúltimo é o final do $(i-1)$-ésimo spline s_{i-1} e o início do i-ésimo spline, então

$$s_i = s_{i-1}(1) \tag{17}$$

$$s_i = s_i(0). \tag{18}$$

As restrições (17) e (18) garantem que os pontos finais de um spline casem com os pontos iniciais do próximo spline, mas não garantem que nesses pontos tenha continuidade. Poderiam aparecer transições não suaves. Para forçar continuidade nesse pontos (que são os nós de interpolação), restrições adicionais, relacionadas às primeiras derivadas exigem que a primeira derivada no final do $(i-1)$-ésimo spline coincide com a primeira derivada do i-ésimo spline, portanto

$$s'_{i-1}(1) = s'_i(0). \tag{19}$$

Se, ainda a flexão da curva nos nós deve ser minimizada, a segunda derivada dos splines intermediários tem que coincidir, então

$$s''_{i-1}(1) = s''_i(0), \tag{20}$$

e, finalmente, nos nós inicial e final deve zerar como

$$s''_0(0) = 0 \tag{21}$$

$$s''_{n-1}(1) = 0. \tag{22}$$

Todas as restrições (7) a (22) formam $4n$ equações lineares para $4n$ desconhecidas $\{a_i, b_i, c_i, d_i\}, i = 0, \ldots, n-1$. Para calcular as derivadas $D_0, \ldots, D_n$, necessárias para obter as desconhecidas através das equações (11) a (14), monta-se o sistema linear de dimensão $(n+1)$ de forma triangular

$$\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix} = \begin{bmatrix} 3(s_1 - s_0) \\ 3(s_2 - s_0) \\ 3(s_3 - s_1) \\ \vdots \\ 3(s_{n-1} - s_{n-3}) \\ 3(s_n - s_{n-2}) \\ 3(s_n - s_{n-1}) \end{bmatrix}. \tag{23}$$

Note que para cada componente x e y, a resolução de (23) tem que ser feita, ou seja, duas vezes.

Agora para cada spline $q_i(t)$ com os coeficientes na mão, a curva paramétrica pode ser gerada e desenhada, variando o parâmetro t entre zero e um.

2 Estudo Comparativo

O erro de cada método de interpolação deve ser medido em relação ao valor exato da função analítica. Tem que ser definido a função $f(x)$ e a sua primitiva $F(x)$ para obter a área pela integral definida $I = \int_a^b f(x)dx = F(b) - F(a)$. Esta área é o valor de referência. Além disso, a função f deve ser usada para gerar os $(n+1)$ nós de interpolação (1). Defina pelo menos três funções diferentes para fazer os experimentos.

2.1 Integração da curve global

Para efetuar a integração numérica do polinômio definido em (2), devem ser usados os métodos do Trapézio repetida para calcular a área debaixo da curva I_{TR} e de Simpson 1/3 repetida para obter I_{SR}.

2.2 Integração dos splines

A área debaixo de todos os n splines $I_{\mathrm{Sp}} = \sum_{i=0}^{n-1} I_{\mathrm{Sp}_i}$ deve ser calculada pela soma das áreas individuais de cada spline. Use um método de integração numérica apropriado para um spline cúbico.

Cada método numérico deve ser finalmente comparado com o valor exato, calculando o módulo da diferença

$$E_. = |I - I_.|.$$

Referências

- Interpolação usando Splines
  `http://mathworld.wolfram.com/CubicSpline.html`

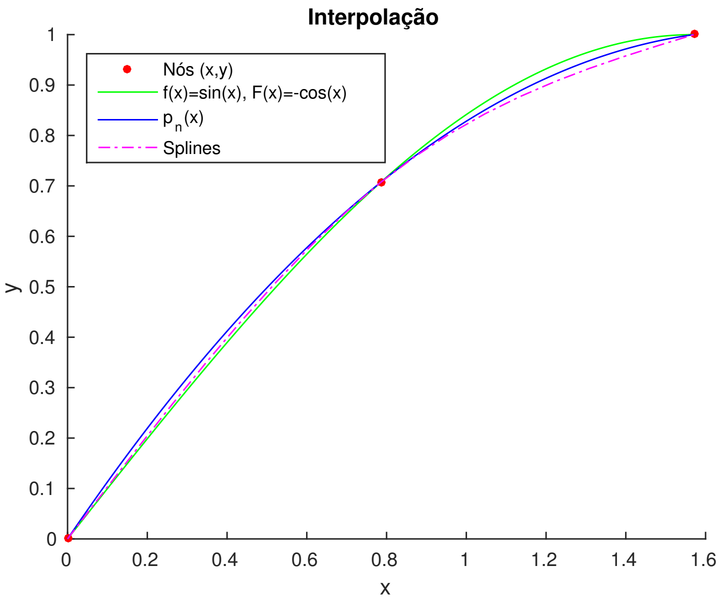

Figura 1: Gráfico comparativo que mostra a função analítica $f(x) = \sin(x)$ que deu origem aos nós de interpolação em $\{x_0 = 0, x_1 = \pi/4, x_2 = \pi/2\}$, a interpolação global usando um polinômio de grau $n = 2$, e dois splines cúbicos entre os nós consecutivos.

Elaboração: O resultado deve ser código em Octave que produz as tabelas e gráficos para o problema apresentado. Os alunos obrigatoriamente devem fornecer exemplos testados. O usuário (neste caso o professor) não deve ter o trabalho de digitar nada, além da linha de comando no Octave que executa o(s) programa(s) principa(l)(ais).

O produto final deve ser um arquivo no formato `zip` com a seguinte sintaxe: `aluno1+aluno2.zip`. O aluno se responsabiliza pelo envio e recepção correta. Em caso de problemas maiores de tráfego de rede (serviços UFES fora do ar), o aluno deve mandar novamente o arquivo original (encaminhamento da mensagem original) quando o serviço voltar. O arquivo `aluno1+aluno2.zip` deve conter uma **única** pasta com o nome `aluno1+aluno2`. Duas subpastas devem conter o código fonte e a documentação do projeto. O arquivo `aluno1+aluno2.zip` deve ser mandado como anexo **exclusivamente** copiando a hiperligação seguinte no browser ou cliente de E-mail:

`mailto:thomas.rauber@ufes.br?subject=Algoritmos%20Numericos%202016/2:%20Entrega%20de%20trabalho%20`

A documentação deve ser em forma de descrição de projeto com os, preferencialmente gerado por LaTeX, contendo os seguintes tópicos:

- Capa do Projeto
  - Título
  - Autoria
  - Data
  - Resumo
- Introdução
- Objetivos
- Metodologia
- Resultados e Avaliação
- Referências Bibliográficas

Última atualização: 1 de junho de 2016, 18:01

Bom trabalho!