Algoritmos Numéricos DI/CT/UFES

Interpolação polinomial por partes via splines cúbicas

1 Objetivos

O objetivo deste trabalho é implementar a interpolação polinomial usando o método de splines cúbicas naturais.

2 Interpolação polinomial e a interpolação por partes

Dado um conjunto de $(n+1)$ pontos no plano C=((x_0, y_0), (x_0, y_0), ... (x_n, y_n)) com $x_0 < x_1 < x_2 ... < x_{n-1} < x_n$, há diversas formas de se fazer interpolação polinomial.

Uma delas das formas é aquela onde o objetivo é construir um único polinômio $p(x)$ que passe por todos os pontos. Sabe se que, dado um conjunto de $(n+1)$ pontos, é sempre possível obter um polinômio de grau n que satisfaça às $(n+1)$ restrições de interpolação $(p_n(x)_i = y_i)$.

Uma outra forma, bastante empregada, quando há uma quantide grande de pontos, é fazer a interpolação por partes.

3 Interpolação polinomial por partes

Na interpolação polinomial por partes, em cada intervalo (x_i, x_{i+1}) é construído um polinômio. O grau dos polinômios usados na interpolação por partes podem ser de grau 1 (interpolação via retas) ou de grau maior, por exemplo, de grau 3.

Splines cúbicas É uma das formas de interpolação por partes mais empregadas. Neste casos, os polinômios interpoladores são polinômios de grau 3. Os polinômios são construídos de tal forma que as curvas (as splines) são concatenadas formando uma função contínua e sem mudanças bruscas de inclinação e de curvatura nos pontos de junção, isto é, nos pontos internos do domínio $(x_1, x_2, ..., x_{n-1})$

A descrição e explicação de como obter as splines cúbicas, para um conjunto de pontos no plano, pode ser vista com detalhes nas páginas de 149 a 155, do livro Algoritmos Numéricos de Frederico Campos, 2^a edição e, também, nas páginas de 426 a 435 (paginas referentes à 5^a edição, do livro tradizido) no livro de Chapara e Canale (estou deixando uma cópia desta última referência).

4 Implementações

Implementar as splines cúbicas naturais que interpolam um conjunto de $(n+1)$ pontos no plano C=((x_0, y_0), (x_0, y_0), ... (x_n, y_n)) com $x_0 < x_1 < x_2 ... < x_{n-1} < x_n$ dispostos em uma configuração qualquer (não necessariamente igualmente espaçados) e para um dado valor $x = z$ fornecido (com $x_0 < z < x_n$ e $z \neq x_i$) calcular a imagem deste ponto pelas splines (ou seja, obter $s_i(z)$).

Os dados de entrada serão: a quantidade de pontos de interpolação, os pontos de interpolação (x_i, y_i), o ponto z e a quantidade m para os quais se vai calcular a imagem pelas splines.
Todos devem ser fornecidos diretamente via teclado.

A implementação do programa deve ser tal que calcule $s_i(z)$ para um dado ponto z e, também, gere as imagens das splines $s_i(x)$ para um conjunto de m pontos (com $m >> n$) em $D = [x_0, x_n]$, pontos igualmente espaçados, visando fazer, posteriormente, o gráfico das splines em D.

Observações:
1) Não é preciso gerar o gráfico de forma automatizada pelo código, a menos, que prefiram faze-lo assim. De qualquer forma, mesmo se gráfico for feito automaticamente, o conjunto de m valores de x e suas respectivas imagens pelas splines, deve ser exibido na tela).
2) Para a implementação, podem ser usadas quaisquer funções disponíveis na linguagem que forem usar para desenvolver o programa menos, é claro, a função (se houver) que calcule diretamente as splines.

5 Dados de entrada e valores de saída

Entrada:
O usúario deve fornecer os dados na seguinte ordem:
quantidade de pontos
(em uma nova linha) a lista de valores de x, separados por espaço
(em uma nova linha) a lista de valores de y, separados por espaço
(em uma nova linha) o valor de z
(em uma nova linha) o valor de m

Exemplo de uma entrada:
Para os pontos:
$C = ((x_0 = -1.0, y_0 = 1.0), (x_1 = 0, y_1 = 3.0), (x_2 = 1.5, y_2 = 0.3), (x_3 = 2.0, y_3 = 2.6))$
$z = 0.77$ e
$m = 31$ a entrada seria:

```
4
-1.0  0    1.5  2.0
1.0  3.0  0.3  2.6
0.77
31
```

Saída:
Deve ser exibido na tela:
em uma linha, o valor de z e sua imagem $s_i(z)$ (em uma linha isolada)
nas linhas seguintes (ou em uma única linha) os valores de x_i e suas respectivas imagens $s_i(x)$ no conjunto de m pontos, igualmente espaçados em $D = [a = x_0, b = x_n]$ (neste casos, observar que os pontos x_0 e x_n são os limites inferior e superior do domínio). Para exibir na tela estes últimos pontos, isto é, os valores de $s_i(x)$ no conjunto de m pontos pode ser usar o formato que for mais conveniente para a realização do gráfico. Inclusive, se preferirem, estes pontos podem, também, ser gravados em um arquivo chamado grafico.txt, no formato que for mais conveniente para a realização do gráfico.

6 Execuções

O código deve ser geral, capaz de obter as splines para um conjunto qualquer de pontos no plano (de acordo com o que foi definido no enunciado). Para fazer o relatório, deve executar o código para resolver os seguintes problemas:

1. **Problema 1**
   Obter as splines interpoladoras dos seguintes dados:

   ```
   5
   1.0  1.3  2.0  3.0  3.5
   0.5  0.2  0.8  1.7  1.3
   1.1
   31
   ```

   e fazer o gráfico das splines em $D = [a = 0.0, b = 3.5]$ usando $m = 31$ pontos (contando com $x_0 = 0.0$ e $b = 3.5$)

2. **Problema 2**
   Obter as splines interpoladoras dos dados:

   ```
   9
   0.0  1.25  2.5   3.75 5.0  6.25  7.5  8.75 10.0
   2.0  3.0   4.3   4.5  4.0  3.4   3.1  3.6  2.4
   3.0
   51
   ```

   e fazer o gráfico das splines em $D = [x_0 = 0.0, x_9 = 10]$ usando $m = 51$ pontos.

7 Relatório

1. Apresentar uma síntese do trabalho (objetivos gerais e o que foi feito).
2. Apresentar, para cada um dos problemas dados, as soluções e os gráficos solicitados.
3. Código: explicar como rodar o programa e imprimir, no final do relatório, uma cópia do código.

8 Condições de entrega

8.1 Grupo:

Este trabalho deverá ser realizado em grupos de, no máximo, **3 alunos**. Não serão aceitos grupos com mais componentes. OBS: Escreva os nomes dos componentes no código fonte do seu programa.

8.2 O que entregar:

- Envie o código.
- Envie o relatório (com o código impresso no final) e os resultados das execuções solicitadas.