Lab 4: Interpolação de Polinômios

INF1608 – Análise Numérica

Leonardo Seperuelo Duarte

lduarte@tecgraf.puc-rio.br

Departamento de Informática, PUC-Rio

30 de março de 2024

1. Implemente as seguintes funções de interpolação de polinômios:

   (a) Implemente uma função que retorne n amostras espaçadas regularmente no intervalo $[a, b]$, onde $x_i[0] = a$, $x_i[n-1] = b$ e os demais valores são regularmente distribuídos no intervalo. A função deve calcular as amostras x_i preenchendo o vetor `xi`, pré-alocado, passado como parâmetro, seguindo o protótipo:

   ```
   void regular (int n, double a, double b, double* xi);
   ```

   (b) Implemente uma função que retorne as n amostras de Chebyshev para a aproximação de uma função qualquer, dentro do intervalo $[a, b]$.

   $$x_i = \frac{b-a}{2} \cos\frac{\beta\pi}{2n} + \frac{a+b}{2}, \quad \beta = 1, 3, 5, \cdots, 2n-1$$

   A função deve calcular as amostras x_i preenchendo o vetor `xi`, pré-alocado, passado como parâmetro, seguindo o protótipo:

   ```
   void chebyshev (int n, double a, double b, double* xi);
   ```

   (c) O polinômio interpolante por diferenças divididas de Newton pode ser expresso por:

   $$P_{n-1}(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \cdots + b_{n-1}(x - x_0)(x - x_1)\cdots(x - x_{n-2})$$

   onde:

   $$\begin{aligned} b_0 &= F[x_0 \cdots x_0] \\ b_1 &= F[x_0 \cdots x_1] \\ &\cdots \\ b_{n-1} &= F[x_0 \cdots x_{n-1}] \end{aligned}$$

   A expressão $F[x_i \cdots x_j]$ representa a diferença finita de Newton e é dada por:

   $$F[x_i \cdots x_j] = \begin{cases} f(x_i) & \text{se } i = j \\ \frac{F[x_{i+1} \cdots x_j] - F[x_i \cdots x_{j-1}]}{x_j - x_i} & \text{se } i < j \end{cases}$$

   Escreva uma função para calcular os n coeficientes b_i. Pode-se usar uma implementação recursiva simples.

A função deve receber as abscissas das amostras x_i e a função que se deseja interpolar, e deve preencher o vetor `bi`, pré-alocado, com os coeficientes calculados:

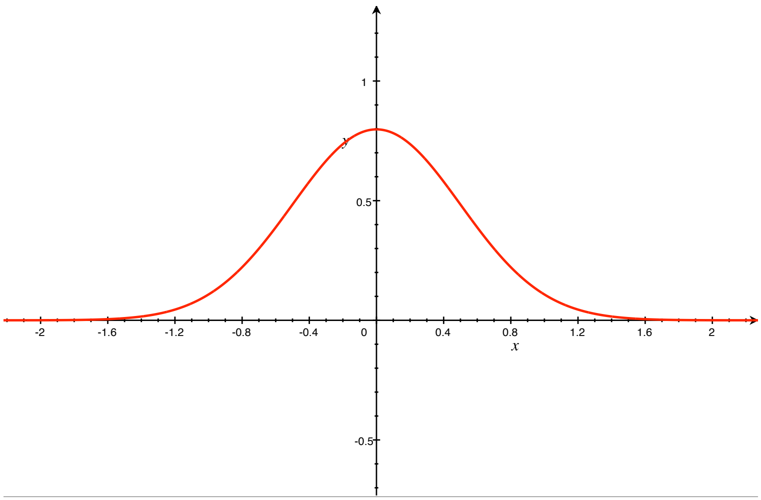
```
void coeficientes (int n, double* xi, double (*f) (double), double* bi);
```

(d) Escreva uma função para avaliar o polinômio interpolante de Newton em um ponto x dado. A função recebe como parâmetros as amostras x_i, os coeficientes b_i e o valor x onde o polinômio deve ser avaliado, e deve retornar o valor do polinômio no ponto x, seguindo o protótipo:

```
double avalia (int n, double* xi, double* bi, double x);
```

2. Para testar sua implementação, escreva um código cliente que ache o polinômio interpolante da função de distribuição normal:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Por exemplo, com $\mu = 0.0$ e $\sigma = 0.5$, no intervalo $[-2, 2]$, compare os polinômios interpolantes usando diferentes números de amostras, regularmente espaçadas e segundo Chebyshev. Avalie os polinômios em diferentes pontos no intervalo $[-2, 2]$ e compare a diferença em relação à função original. A amostragem de Chebyshev de fato diminui o erro absoluto máximo?

Agrupe os protótipos das funções em um módulo "interp.h" e as implementações em um módulo "interp.c". Escreva o teste em um outro módulo "main.c".

Entrega: O código fonte deste trabalho (isto é, os arquivos "interp.h", "interp.c" e "main.c", sem compressão) deve ser enviado via página da disciplina no EAD. O sistema receberá trabalhos com atraso (com perda de 1 ponto na avaliação) até 1 hora após o fim da aula.