MAP0151 - Cálculo Numérico e Aplicações
Lista 8 (Correção)

5 questões $\times$ 2.0 pontos = 10.0 pontos.

(Questão 1) A tabela de diferenças divididas fica assim:

Ordem	0	1	2	3	4
$x_0 = 0$	3				
		-3			
$x_1 = 1$	0		4		
		9		2	
$x_2 = 3$	18		12		0
		45		2	
$x_3 = 4$	63		22		
		111			
$x_4 = 6$	285				

Logo, o polinômio interpolador na forma de Newton pedido fica

$$p_4(x) = 3 - 3x + 4x(x-1) + 2x(x-1)(x-3)$$

(Questão 2) A tabela de diferenças divididas fica assim:

Ordem	0	1	2
$x_0 = 1.0$	2.718		
		2.86	
$x_1 = 1.1$	3.004		1.5
		3.16	
$x_2 = 1.2$	3.320		

de modo que o polinômio interpolado na forma de Newton fica

$$\begin{aligned} p_2(x) &= 2.718 + 2.86(x-1.0) + 1.5(x-1.0)(x-1.1) \\ &= 2.718 + (x-1.0)[2.86 + (x-1.1)1.5] \end{aligned}$$

onde na última igualdade utilizamos o Método de Horner para escrever o polinômio interpolador na forma de Newton, o qual economiza cálculos das diferenças $x - x_i$ e revela a vantagem de escrevermos o polinômio interpolador na forma de Newton. Assim:

$$\begin{aligned} p_2(1.05) &= 2.718 + (1.05 - 1.0)[2.86 + (1.05 - 1.1)1.5] \\ &= 2.85725 \end{aligned}$$

Recordando que o erro cometido na interpolação de grau 2 de $f(x) = e^x$ é dado por

$$\begin{aligned} E_2(x) &= (x - x_0)(x - x_1)(x - x_2)\frac{f^{(3)}(\xi_x)}{3!} \\ &= \frac{1}{6}(x - 1.0)(x - 1.1)(x - 1.2)\exp(\xi_x) \end{aligned}$$

onde $\xi_x \in [1.0, 1.2]$, temos que

$$\begin{aligned} |E_2(1.05)| &= \frac{1}{6}|(1.05 - 1.0)(1.05 - 1.1)(1.05 - 1.2)||\exp(\xi_x)| \\ &= 6.25 * 10^{-5}|\exp(\xi_x)| \\ &\leq 6.25 * 10^{-5}|\exp(1.2)| \\ &= 6.25 * 10^{-5} * 3.320 \\ &= 2.075 * 10^{-4} \end{aligned}$$

pois $f^{(3)}(\xi_x) = \exp(\xi_x)$ é uma função crescente em $[1.0, 1.2]$.

(Questão 3) Vou utilizar a seguinte estratégia: vou tentar determinar o valor de k apenas utilizando a subtabela

$$\begin{array}{c|ccc} x & -1 & 0 & 1 \\ \hline y & 1 & 1 & 5 \end{array} \tag{1}$$

e depois encontrar o valor de α. Assim, se

$$p_3(x) = 3x^3 + kx^2 - x + 1$$

interpola

$$\begin{array}{c|cccc} x & -1 & 0 & 1 & 2 \\ \hline y & 1 & 1 & 5 & \alpha \end{array} \tag{2}$$

então p_3 também interpola (1). Logo, temos que

$$\begin{aligned} p_3(-1) &= k - 1 \\ &= 1 \\ p_3(0) &= 1 \\ p_3(1) &= k + 3 \\ &= 5 \end{aligned}$$

e deduzimos que $k = 2$. Então temos que $p_3(x) = 3x^3 + 2x^2 - x + 1$ e portanto

$$\begin{aligned} \alpha &= p_3(2) \\ &= 31. \end{aligned}$$

(Questão 4) O procedimento neste caso é determinar o polinômio interpolador p_2 da tabela dada e determinar o erro máximo E que cometemos em $x = 0.5$. Teremos então que $g(0.5) \in [p_2(0.5) - E, p_2(0.5) + E]$.

Construindo a tabela de diferenças divididas temos

Ordem	0	1	2
$x_0 = -1$	0.5		
		0.5	
$x_1 = 0$	1.0		-0.5
		-0.5	
$x_2 = 1$	0.5		

temos que $p_2(x) = 0.5 + 0.5(x + 1) - 0.5x(x + 1)$, de modo que

$$p_2(0.5) = 0.875.$$

Além disso, a expressão do erro cometido pela interpolação p_2 é dado por

$$\begin{aligned} |E_2(0.5)| &= |(0.5 + 1)0.5(0.5 - 1)| \frac{|g^{(3)}(\xi_x)|}{3!} \\ &\leq 0.1875 \end{aligned}$$

onde utilizamos que $|g^{(3)}(\xi_x)| \leq 3$ para qualquer que fosse o ponto $\xi_x \in [-1, 1]$. Então

$$\begin{aligned} g(0.5) &\in [p_2(0.5) - E, p_2(0.5) + E] \\ &= [0.875 - 0.1875, 0.875 + 0.1875] \\ &= [0.6875, 1.0625]. \end{aligned}$$

(Questão 5) Inicialmente, implementei uma função `monta_tab.sci` que, dada uma função $f : [a, b] \to \mathbb{R}$ e um tamanho de partição n, tabela a função nos pontos

$$x_j := a + jh, j = 0, 1, \ldots, n$$

onde $h = \frac{b-a}{n}$.

```
// monta_tab.sci

function T = monta_tab (f, a, b, n)
  T = zeros (2, n + 1);
  h = (b - a) / n;

  T(1,1) = a;
  T(2,1) = f(a);

  for i = 2:(n + 1) do
    T(1,i) = T(1,i - 1) + h;
    T(2,i) = f(T(1,i));
  end
endfunction
```

Em seguida, implementei uma função `interpola.sci` que, dada uma tabela T de função tabelada, devolve um vetor c com os coeficientes do polinômio interpolador de T.

```
// interpola.sci

function c = interpola (T)
  // importando as funcoes necessarias
  getf("gausspv.sci");

  [nl, nc] = size(T);

  // montando o sistema linear a ser resolvido
  A = zeros (nc, nc);

  for i = 1:nc do
    for j = 1:nc do
      A(j,i) = T(1,j)^(i - 1);
```

```
    end
  end

  b = T(2,:)';

  // resolvendo o sistema linear Ac = b;
  [p, B, c] = gausspv(A, b);
endfunction
```

Por fim, implementei uma função `avalia.sci` que, dado o vetor c de coeficientes de polinômio e um ponto x, devolve o valor daquele polinômio calculado em x.

```
// avalia.sci

function y = avalia (c, x)
  [nl, nc] = size(c);

  y = 0;
  for i = 1:nl do
    y = y + c(i) * x^(i - 1);
  end
endfunction
```

Por fim, fiz o que foi pedido no *script* `questao5.sce`.

```
// importando os metodos necessarios
getf("interpola.sci");
getf("monta_tab.sci");
getf("avalia.sci");

// definindo a funcao h
deff("y = h(x)", "y = 1 / (1 + x^2)");

// montando a tabela de interpolacao de h(x)
T = monta_tab (h, -5, 5, 10);

// encontrando os coeficientes do polinomio
// interpolador de T
p = interpola(T);
```

```
// PLOTANDO

// definindo um intervalo
x = [-5:0.001:5]';

[n, nc] = size(x);
H = zeros(n, 1);
P = zeros(n, 1);

for i = 1:n do
  H(i) = h(x(i));
  P(i) = avalia(p,x(i));
end

// plotando
plot2d(T(1,:)', T(2,:)', style=-2);
plot2d(x, [H P], leg="h(x)@p(x)");
```

O gráfico das funções h e de seu polinômio interpolador p_{10} podem ser vistas na Figura 1 (`questao5.pdf`).

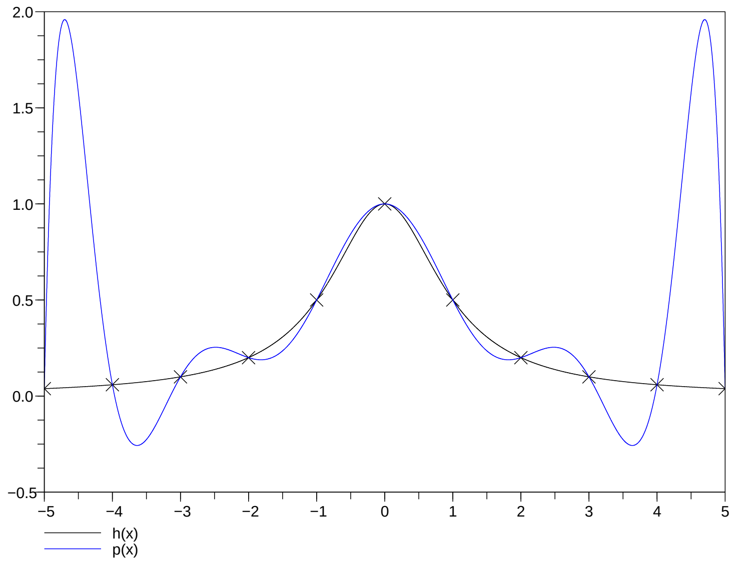

Figura 1: Função $h(x) = \frac{1}{1+x^2}$ *versus* seu polinômio interpolador $p_{10}(x)$, obtido a partir de 11 pontos equiespaçados do intervalo $[-5, 5]$.