Universidade do Minho Ano Lectivo de 2020/2021

Departamento de Matemática 1^o Semestre

Lic. em Ciências da Computação

Análise Numérica

Ficha de exercícios N^{o}4 - Interpolação polinomial

---

1. De uma certa função $y = f(x)$ conhecem-se os valores seguintes

x	-0.5	-1	0	3
y	-2.35	-6.7	-0.3	5.7

a) No Matlab execute

```
>>p3=@(x) x^3-3.1*x^2+2.3*x-0.3
```

para definir o polinómio $p_3(x) = x^3 - 3.1x^2 + 2.3x - 0.3$ e verifique que $p_3(x_i) = y_i$, para cada $i = 0, 1, 2, 3$. (nota: diz-se que p_3 é polinómio interpolador de f nos pontos dados; x_0, x_1, x_2, x_3 são os "nós de interpolação" e y_0, y_1, y_2, y_3 são os "valores nodais").

b) Execute $>>$ $fplot(p3, [-1, 4])$ para obter o gráfico do polinómio p_3 no intervalo $[-1,\ 4]$ e, em seguida, $>>$ *hold on*, $plot([-0.5,\ -1, 0, 3], [-2.35, -6.7, -0.3, 5.7],' o')$ para sobrepôr no mesmo gráfico os pontos $(x_i,\ y_i)$, $i = 0, 1, 2, 3$.

c) Existe outro polinómio, digamos q_3, de grau não superior a 3, tal que $q_3(x_i) = y_i$, para cada $i = 0, 1, 2, 3$? Porquê? E de grau superior a 3?

2. **a)** Para os nós x_i dados no exercício anterior, construa a respectiva matriz de Vandermonde $V = V(x_0, x_1, x_2, x_3)$ e verifique que se tem $\det(V) = \prod_{i,j=0,i>j}^{3} (x_j - x_i)$.

b) Sem resolver o sistema, diga qual é a solução do sistema $Va = y$, onde y é o vector (coluna) dos valores nodais dados no exercício anterior. Confirme a sua resposta, resolvendo o sistema.

3. Dada uma tabela de $n + 1$ pontos

x_0	x_1	$\dots$	x_n
y_0	y_1	$\dots$	y_n

a fórmula de Lagrange para calcular num ponto $x \neq x_i$ o valor do polinómio p_n, de grau não superior a n, tal que $p_n(x_i) = y_i$, para cada $i = 0, \cdots, n$, é

$$p_n(x) = y_0 \cdot L_0(x) + y_1 \cdot L_1(x) + \cdots + y_n \cdot L_n(x)$$

onde, para cada $i = 0, \cdots, n$,

$$L_i(x) = \frac{\prod_{j=0,j\neq i}^{n}(x - x_j)}{\prod_{j=0,j\neq i}^{n}(x_i - x_j)} = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

a) Use a fórmula interpoladora de Lagrange para determinar o polinómio interpolador de grau não superior a 3 para os pontos

0	1	2	4
1	-2	0	3

b) Determine o número de "flops" (operações aritméticas de ponto flutuante) necessárias para o cálculo de $p_n(x)$ pela fórmula de Lagrange.

c) No Matlab escreva uma função **px=poLagrange(xi,yi,x)** para implementar a fórmula interpoladora de Lagrange. Dados os vectores $xi = [x_0, x_1, \ldots, x_n]$ e $yi = [y_0,\ y_1, \ldots,\ y_n]$ e o ponto x, a função calcula o valor $px = p_n(x)$.

d) Use vectores xi (nós) e yi (valores nodais) à sua escolha (por exemplo, usando a função rand) e certifique-se de que para cada nó a função **poLagrange** produz o correspondente valor nodal.

4. **a)** No Matlab defina os nós e valores nodais com $>> xi = [1, 3, 10];\ yi = \log(xi)$ e, em seguida, execute

$$>> x = 1 : 0.1 : 10;\ \ for\ k = 1 : length(x),\ y(k) = poLagrange(xi,\ yi, x(k));\ end,\ plot(x, y, 'r')$$

para obter aproximadamente o gráfico do polinómio interpolador de log nos nós 1, 3 e 10.

b) Use a função $fplot$ para sobrepôr ao gráfico anterior o gráfico da função log no intervalo $[1, 10]$.

5. O erro do polinómio interpolador é dado por

$$f(x) - p_n(x) = W_n(x)\frac{f^{(n+1)}(\xi_x)}{(n+1)!}$$

onde $W_n(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$ é o polinómio nodal (de grau n+1) e ξ_x é um ponto que depende de x e que está no intervalo $\Omega = [\min\{x_0 \cdots, x_n, x\}, \max\{x_0 \cdots, x_n, x\}]$.

a) Use a expressão dada para encontrar um limite para o erro $\log(1.5) - p_1(1.5)$, onde p_1 é o polinómio interpolador de $\log(x)$ nos nós $xi = [1, 2]$.

b) Use a mesma expressão para encontrar limites para os erros $\log(1.5) - p_n(1.5)$, para $n = 2, 3, 4, 5, 6$, onde p_n é o polinómio interpolador de $\log(x)$ nos nós $xi = [1, 2, 3, ..., n+1]$.

6. **a)** Construa a tabela das diferenças divididas para os dados (xi, yi) com $xi = [0,\ 0.2,\ 0.4]$ e $yi = \sinh(xi)$. Use a fórmula de Newton com diferenças divididas para calcular $p_2(0.3)$, sendo p_2 o polinómio interpolador da função seno hiperbólico [no Matlab é $\sinh(x)$] nos nós dados.

b) Repita o exercício, acrescentando o nó 0.6 aos anteriores.

7. **a)** No Matlab escreva o código da função **T=TabDifDiv(xi,yi)** para calcular a tabela das diferenças divididas na forma de uma matriz triangular inferior T. Dados os vectores $xi = [x_0, x_1, \ldots, x_n]$ e $yi = [y_0,\ y_1, \ldots,\ y_n]$ a primeira coluna de T armazena os valores nodais, a segunda coluna armazena as diferenças divididas de 1^a ordem, etc. As diferenças a usar na fórmula interpoladora de Newton estão na diagonal principal de T.

b) No Matlab escreva o código da função **px=polNewton(xi,yi,x).** Dados os nós $xi = [x_0, x_1, \ldots, x_n]$ e os valores nodais $yi = [y_0,\ y_1, \ldots,\ y_n]$, usa a fórmula de Newton com diferenças divididas (invoca **TabDifDiv(xi,yi)**) para calcular o valor do polinómio interpolador no ponto x.

c) Para valores **xi, yi, x** à sua escolha, certifique-se de que as funções **poLagrange** e **polNewton** dão o mesmo resultado.