UNIVERSIDADE DO MINHO

Licenciatura em Ciências da Computação

Análise Numérica

Duração: 2 horas 30 minutos — 28 de janeiro de 2017

EXAME FINAL (COM CONSULTA)

Deves escrever na tua folha de respostas todos os comandos executados no Matlab.

1. **a)** Escreve na forma de uma soma de potências de base 2 o maior número representável no formato duplo da norma IEEE.

   b) No Matlab, `>> (realmax/2)*2 == (2*realmax)/2` produz o valor lógico 0 (falso). Porquê?

2. No Matlab executa

   `>> f=inline('(1-cos(x).^2)./(sin(x).^2)'); k=1:10; f(10.^-k)`

   Porque é que alguns resultados se afastam tanto do valor exacto que é igual a 1?

3. Considera a equação $sin(x) = e^{-x}$.

   a) A equação dada tem alguma raíz negativa? Justifica.

   b) Usa um método à tua escolha para determinar a menor raíz positiva da mesma equação tão exatamente quanto possível (no Matlab usa `>> format long`).

4. **a)** Constrói a tabela das diferenças dividas para a função $log(x)$ nos nós $x_0 = 1$ e $x_i = x_{i-1} + 0.1, i = 1, \cdots, 3$. (nota: podes usar um código ("função") Matlab desenvolvido nas aulas).

   b) Para cada valor de $i = 1, \cdots, 3$, usa valores da tabela construída para escreveres, na forma dada pela fórmula interpoladora de Newton, as expressões dos polinómios de grau não superior a um que interpolam $log(x)$ nos nós x_{i-1} e x_i.

   c) Usa a fórmula interpoladora de Newton para escreveres a expressão do polinómio interpolador p_3 de $log(x)$ em todos os nós tabelados.

   d) Usa a função *fplot* do Matlab para encontrar um valor $M > 0$ tal que

   $$max_{x \in [1,1.3]} |(x-1)(x-1.1)(x-1.2)(x-1.3)| \leq M$$

   e, em seguida, determina um majorante para o erro $|log(x) - p_3(x)|$, qualquer que seja $x \in [1, 1.3]$ (nota: p_3 é o polinómio a que se refere a alínea anterior).

5. **a)** Calcula a aproximação do valor de $\int_{-1}^{1} e^{-x^2} dx$ dada por

   $$\frac{0.1}{3} \left[e^{-1} + 4 \sum_{i=1}^{m} e^{-x_{2i-1}^2} + 2 \sum_{i=1}^{m-1} e^{-x_{2i}^2} + e^{-1} \right]$$

   onde $x_k = -1 + 0.1k, \;\; k = 1, \cdots, 19$ (sugestão: usa o código Matlab desenvolvido nas aulas que implementa a regra anterior).

b) Sabendo que a derivada de quarta ordem da função e^{-x^2} não excede, em módulo, no intervalo $[-1, 1]$, o valor 12, determina um majorante para o erro cometido na aproximação calculada na alínea anterior.

c) Se quisermos garantir um erro de truncatura 16 vezes mais pequeno que o majorante anterior, que valor de h devemos usar? Justifica.

6. Para gerar uma certa matriz A e um certo vetor b, executa no Matlab

```
>> A=hilb(13); b=A*ones(13,1);
```

a) Usa a função *GaussElimPP* (das aulas) para aproximar a solução do sistema $Ax = b$ e escreve o resultado obtido na tua folha de respostas;

b) Por que é que a aproximação calculada tem um erro tão elevado?

questão	1a	1b	2	3a	3b	4a	4b	4c	4d	5a	5b	5c	6a	6b	Total
cotação	1,5	1,5	1,5	1,5	1,5	1	1	1,5	1,5	1,5	1,5	1,5	1,5	1,5	20

RESOLUÇÃO

1. a) No formato duplo da norma IEEE, um número expressa-se na forma

$$x = \pm\left(1.b_1 b_2 \cdots b_{52}\right)_2 \times 2^E$$

onde $b_i = 0$ ou $b_i = 1$, para cada $i = 1, \cdots, 52$, e $-1022 \leq E \leq 1023$. O maior número que se pode representar desta maneira ocorre com $b_i = 1$, para cada $i = 1, \cdots, 52$ e $E = 1023$, ou seja, o número

$$realmax = 2^{1023} + 2^{1022} + \cdots + 2^{971}.$$

b) No Matlab `>> (realmax/2)*2` produz o valor exato mas `>> (realmax*2)/2` produz Inf uma vez que, sendo realmax o maior número que se pode representar, ocorre "overflow" no produto realmax*2.

2. Obtêm-se os resultados

1.0000
1.0000
1.0000
1.0000
1.0000
1.0001
0.9992
0
0
0

À medida que x se aproxima de zero, $cos(x)$ tende para 1 e ocorre cancelamento subtrativo no cálculo de `(1-cos(x).^2)`, isto é o numerador é calculado com cada vez menos algarismos significativos corretos. Isto explica que o quociente que, em aritmética exata, é igual a 1 (para todos os valores de x que não anulam o denominador), se afasta do valor correto e acaba mesmo por dar zero no caso dos valores mais próximos de zero.

3. a) A equação dada não tem nenhuma raíz negativa porque os gráficos de e^{-x} e $sin(x)$ não se intersetam para valores $x < 0$. Com efeito, $e^{-x} > 1$ para $x < 0$ e $sin(x) < 1$ qualquer que seja x.

b) Por exemplo, se usarmos o método de Newton-Raphson, com $f(x) = sin(x) - e^{-x}$, a fórmula iterativa é

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{sin(x_k) - e^{-x_k}}{cos(x_k) + e^{-x_k}}$$

No Matlab, se inicializarmos as aproximações com `>> x=0`, a execução repetida de
`>> x=x-(sin(x)-exp(-x))/(cos(x)+exp(-x))`
produz (em format long) os seguintes resultados

0.5000
0.585643816966433
0.588529412626355
0.588532743977419
0.588532743981861
0.588532743981861

4. a) Executando no Matlab

```
>> x=1:0.1:1.3; TabDifDiv (x, log(x))
```

obtemos a tabela das diferenças divididas na parte triangular inferior da matriz

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0.0953 & 0.9531 & 0 & 0 \\ 0.1823 & 0.8701 & -0.4149 & 0 \\ 0.2624 & 0.8004 & -0.3484 & 0.2217 \end{array}$$

b) Para cada $i = 1, 2, 3$ o polinómio pedido é $p(x) = f(x_i) + (x - x_i)f[x_i, x_{i+1}]$. As diferenças divididas de primeira ordem $f[x_i, x_{i+1}]$ são dadas na segunda coluna da tabela anterior e tem-se

$$p(x) = 0 + 0.9531(x - 1)$$

nos nós $x_0 = 1$ e $x_1 = 1.1$,

$$p(x) = 0.0953 + 0.8701(x - 1.1)$$

nos nós $x_1 = 1.1$ e $x_2 = 1.2$, e

$$p(x) = 0.1823 + 0.8004(x - 1.2)$$

nos nós $x_2 = 1.2$ e $x_3 = 1.3$.

c) O polinómio é

$$p_3(x) = 0 + 0.9531(x - 1) - 0.4149(x - 1)(x - 1.1) + 0.2217(x - 1)(x - 1.1)(x - 1.2)$$

d) Do gráfico obtido no Matlab com

```
>> fplot('(x-1)*(x-1.1)*(x-1.2)*(x-1.3)',[1,1.3])
```

obtemos $M = 10^{-4}$ e da expressão geral do erro do polinómio interpolador

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\eta)}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n)$$

resulta neste caso, com $f(x) = log(x)$ e $f^{(iv)}(x) = -6/x^4$,

$$|f(x) - p_3(x)| \leq max_{x\in[1,1.3]} \frac{6}{4!x^4} \times 10^{-4} = 2.5 \times 10^{-5}.$$

5. a) Trata-se da regra composta de Simpson, com $h = 0.1$, para calcular $\int_{-1}^{1} e^{-x^2} dx$. Executando no Matlab

```
>> S=simpson('exp(-x.^2)',-1,1,20)
```

obtemos o resultado $S = 1.4936$.

b) O erro de truncatura da regra de Simpson composta é, no caso geral, dado por

$$-\frac{h^4}{180}(b-a)f^{(iv)}(\eta)$$

onde η é um ponto entre a e b. De acordo com o enunciado, no caso presente o erro (em valor absoluto) é majorado por

$$\frac{0.1^4}{180} \times 2 \times 12 = 1.3333\cdots \times 10^{-5}.$$

c) Uma vez que o erro de truncatura da regra de Simpson composta é proporcional a h^4 (ver a expressão dada na alínea anterior) e $(h/2)^4 = h^4/16$, basta dividir h por 2, no caso presente, usar $h = 0.05$.

6. a) No Matlab,

```
A=hilb(13); b=A*ones(13,1); x=GaussElimPP(A,b)
```

produz o resultado

1.0000
1.0000
1.0004
0.9931
1.0612
0.6693
2.1491
-1.6542
5.1185
-3.2430
3.7830
-0.0518
1.1743

b) A solução exata é $x_1 = x_2 = \cdots = x_{13} = 1$. Os erros no resultado obtido são devidos ao mau condicionamento do sistema. Com efeito o número de condição cond(A)=8.1661e+17 é muito grande.