UNIVERSIDADE DO MINHO

Licenciatura em Ciências da Computação

Análise Numérica

Duração: 2 horas e 30 minutos	14 de janeiro de 2020
Teste 2	

<u>Deves escrever na tua folha de respostas todos os comandos executados no Matlab.</u>

1. a) A fórmula iterativa
$$x^{(k+1)} = x^{(k)}(2 - bx^{(k)})$$
pode ser usada para calcular o inverso de um número $b \neq 0$. Usa-a para tentar calcular o inverso do número $b = 254$ começando com a aproximação inicial $x^{(0)} = 0.5$. Escreve na folha de respostas os resultados obtidos nas 5 primeiras iterações e comenta-os.

   b) Mostra que se $b \in]1, 2[$ então com $x^{(0)} = 0.5$ a sucessão de aproximações produzidas pela fórmula iterativa anterior converge.

   c) Podemos dizer que, nas condições da alínea anterior, a convergência é rápida? Porquê?

   d) Tendo em conta que
$$254 = 1.984375 \times 2^7,$$
usa o resultado da alínea b) para inverter o número 1.984375 tão exatamente quanto o Matlab te permitir. Determina em seguida a correspondente aproximação para o inverso de 254.

2. Considera o seguinte integral definido
$$I = \int_{-1}^{1} e^{-x^2/2} dx.$$

   a) A regra simples de Simpson aproxima o valor de I pelo valor de
$$\int_{-1}^{1} p_2(x) dx.$$
O que representa $p_2(x)$ na expressão anterior?

   b) Usa os códigos Matlab, desenvolvidos nas aulas, que implementam as regras compostas dos trapézios e de Simpson, para calcular aproximações de I, com n = 10, e escreve-as na folha de respostas.

   c) Qual dos valores calculados se espera que seja melhor aproximação para o valor de I? Justifica.

d) Usa a expressão do erro da regra composta de Simpson para determinar o valor de n que garante um erro de truncatura inferior a 10^{-8}. Sabendo que a derivada de quarta ordem de $f(x) = e^{-x^2/2}$ é

$$f^{(IV)}(x) = e^{-x^2/2}(x^4 - 6x^2 + 3),$$

no Matlab usa a função fplot para obteres o gráfico de $f^{(IV)}(x)$ e extrai deste gráfico informação relevante para responderes à questão.

3. Considera o seguinte sistema de equações lineares, com n equações e n incógnitas, cuja matriz é tridiagonal (uma matriz diz-se tridiagonal quando todas as entradas fora da "banda" formada pela três diagonais centrais são nulas)

$$\begin{bmatrix} a_1 & c_1 & & & \\ d_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & d_{n-2} & a_{n-1} & c_{n-1} \\ & & & d_{n-1} & a_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_{n-1} \\ b_n \end{bmatrix}.$$

a) No seguinte código Matlab, que implementa o método de eliminação de Gauss para resolver este sistema, faltam instruções nos lugares assinalados com ?????????? Na tua folha de respostas escreve o código completo, acrescentando as instruções em falta.

```
function x=Tridiagonal(a,c,d,b)
n=length(b);
for i=1:n-1
     m= ??????????
     a(i+1)=a(i+1)+m*c(i);
     b(i+1)=??????????
end
x(n)=b(n) / ??????????
for i=n-1:-1:1
     x(i)= ??????????
end
```

b) Para a resolução de um sistema tridiagonal, que vantagens oferece este código relativamente à *function* GaussElim desenvolvida nas aulas?

c) No Matlab, executa sucessivamente

```
>> c=sqrt(2); A=diag([eps 2 2 2])+c*diag(ones(3,1),-1)+c*diag(ones(3,1),1)
>> b=A*ones(4,1);
>> x=A\b, x2=GaussElim(A,b)
```

Na folha de resposta excreve as soluções x e x2 obtidas. Como explicas a diferença entre x e $x2$?

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

RESOLUÇÃO

1. a) Começamos por definir, no Matlab, o valor de b, a aproximação inicial para o inverso de b e a função iteradora $\phi(x) = x(2 - bx)$ (neste caso, função de x e também de b)

```
>> b=254; x=0.5; fi=inline('x*(2-b*x)')
```

Para simplificar o código, não usamos índices para as aproximações e limitamo-nos a calcular cada uma delas simplesmente executando repetidamente o mesmo comando com o novo valor de x

```
>> x=fi(b,x)

x =

  -62.5000

>> x=fi(b,x)

x =

  -9.9231e+05

>> x=fi(b,x)

x =

  -2.5011e+14

>> x=fi(b,x)

x =

  -1.5889e+31

>> x=fi(b,x)

x =

  -6.4124e+64
```

Destes resultados pode concluir-se que, começando com $x^{(0)} = 0.5$, a sequência $x^{(k+1)} = \phi(x^{(k)})$ diverge e, portanto, não produz o resultado desejado (inverso de $b = 254$).

b) Os erros nas iterações $x^{(k)}$ e $x^{(k+1)}$ estão relacionados pela expressão seguinte

$$x^{(k+1)} - \frac{1}{b} = \phi'(\theta^{(k+1)}) \left(x^{(k)} - \frac{1}{b} \right)$$

onde

$$\phi'(\theta^{(k+1)}) = 2\left(1 - b\theta^{(k+1)}\right)$$

e $\theta^{(k+1)}$ é um ponto que está entre $x^{(k)}$ e $1/b$.
Com $x^{(0)} = \frac{1}{2} < 1/b$, por ser $b < 2$, é

$$\frac{1}{2} < \theta^{(1)} < 1/b$$

e, por ser $b > 1$,

$$\frac{1}{2} < b\theta^{(1)} < 1.$$

Daqui resulta

$$0 < 1 - b\theta^{(1)} < \frac{1}{2}$$

e

$$0 < \phi'(\theta^{(1)}) < 1$$

o que mostra que

$$\left|x^{(1)} - \frac{1}{b}\right| < \left|x^{(0)} - \frac{1}{b}\right|$$

e

$$\frac{1}{2} < x^{(1)} < 1/b.$$

Partindo de

$$\frac{1}{2} < \theta^{(k+1)} < 1/b$$

deduz-se, seguindo os mesmos passos, que

$$0 < \phi'(\theta^{(k+1)}) < 1$$

o que mostra que os erros convergem para zero.

c) Como se escreveu antes, em cada iteração o erro é o produto do erro da iteração anterior pelo valor de $\phi'(\theta^{(k+1)})$. Ora, de

$$\phi'(\theta^{(k+1)}) = 2\left(1 - b\theta^{(k+1)}\right)$$

conclui-se que, à medida que nos aproximamos da solução $1/b$ esta derivada tende para zero e a convergência é rápida.

d)
```
>> fi=inline('x*(2-b*x)'); x=0.5; b=1.984375; format long e
>> x=fi(b,x)

x =

     5.039062500000000e-01

>> x=fi(b,x)

x =

     5.039370059967041e-01
```

```
>> x=fi(b,x)

x =

     5.039370078740157e-01

>> x=fi(b,x)

x =

     5.039370078740157e-01
```

O inverso do número 254 será aproximado pelo resultado anterior multiplicado por 2^{-7}:

```
>> x*2^-7

ans =

     3.937007874015748e-03
```

2. a) $p_2(x)$ é o polinómio, de grau não superior a 2, que interpola a função integranda nos nós $x_0 = -1, x_1 = 0$ e $x_2 = 1$.

   b)
```
>>T= trapezios('exp(-x.^2/2)',-1,1,10)

T =

    1.7072

>> S=simpson('exp(-x.^2/2)',-1,1,10)

S =

    1.7113
```

   c) Uma vez que as regras dos trapézios e de Simpson têm graus 1 e 3, respetivamente, espera-se que S seja melhor aproximação do que T.

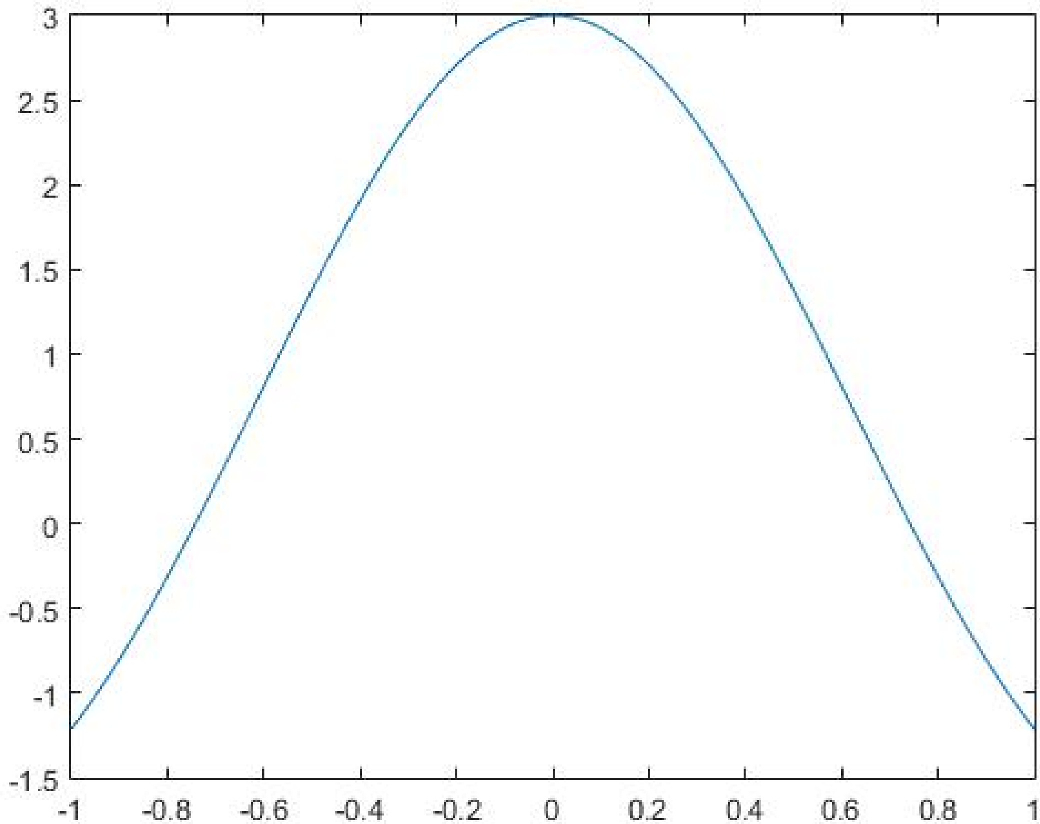
   d) A expressão geral do erro de truncatura da regra composta de Simpson é

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

onde η é um ponto que está entre a e b. Com

```
>> fplot('exp(-x^2/2)'*(x^4-6*x^2+3)',[-1,1])
```

obtem-se o gráfico



e podemos garantir que $|f^{(iv)}(x)| \leq 3$ no intervalo de integração. Uma vez que $h = (b-a)/n = 2/n$, vamos determinar o valor de n a partir de

$$\frac{\left(\frac{2}{n}\right)^4}{180} \times 2 \times 3 < 10^{-8}.$$

Resulta

$$n > \frac{2}{(30 \times 10^{-8})^{1/4}}$$

e de

```
>> 2/(30*1e-8)^0.25

ans =

   85.4574
```

conclui-se que $n = 86$ garante um erro inferior a 10^{-8}.

3. a)
```
function x=Tridiagonal(a,c,d,b)
n=length(b);
for i=1:n-1
    m=-d(i)/a(i);
    a(i+1)=a(i+1)+m*c(i);
    b(i+1)=b(i+1)+m*b(i);
```

```
end
x(n)=b(n)/a(n)
for i=n-1:-1:1
    x(i)=(b(i)-c(i)*x(i+1))/a(i);
end
```

b) Observe-se que a neste caso a matriz está armazenada em 3 vetores, a, b e c, o que corresponde a armazenar um total de $3n-2$ entradas. O código GaussElim usa um array de duas dimensões requerendo, portanto, o armazenamento de n^2 entradas, sendo que apenas $3n-2$ são diferentes de zero. Tal acarreta um grande desperdício de espaço de memória. Também ao nível do tempo de execução haverá diferenças significativas entre os dois códigos. Enquanto que a função Tridiagonal requer um número de operações da ordem de n, na função GaussElim tal número é $\mathcal{O}(n^3)$.

c)
```
>> x=A\b, x2=GaussElim(A,b)

x =

    1.0000
    1.0000
    1.0000
    1.0000


x2 =

    2.0000
    1.0000
    1.0000
    1.0000
```

O resultado x2 produzido com GaussElim não está correto. Esta função implementa o método de eliminação de Gauss sem pivotação; no primeiro passo de redução (eliminação de d_1), o multiplicador é muito grande o que torna o método numericamente instável.