18 / 20

ISCTE INSTITUTO UNIVERSITÁRIO DE LISBOA

Integração e Derivação Numérica – 2º Trabalho

Trabalho elaborado no âmbito da Unidade Curricular de Tópicos de Matemática I do 1º ano da Licenciatura de Ciência de Dados

Realizado por:

105289 – André Plancha Fernandes – Turma CDA2

104753 – Daniel Lapas Ramalhete – Turma CDA2

105188 – João Gabriel Braga Madeira – Turma CDA2

104781 – Martim Costa Montalvão – Turma CDA2

Dezembro de 2021

Grupo I

5 / 5

a)

Para este exercício foi pedido para calcular numericamente a integral, com um erro inferior a 0,001, e utilizar o método do ponto médio.

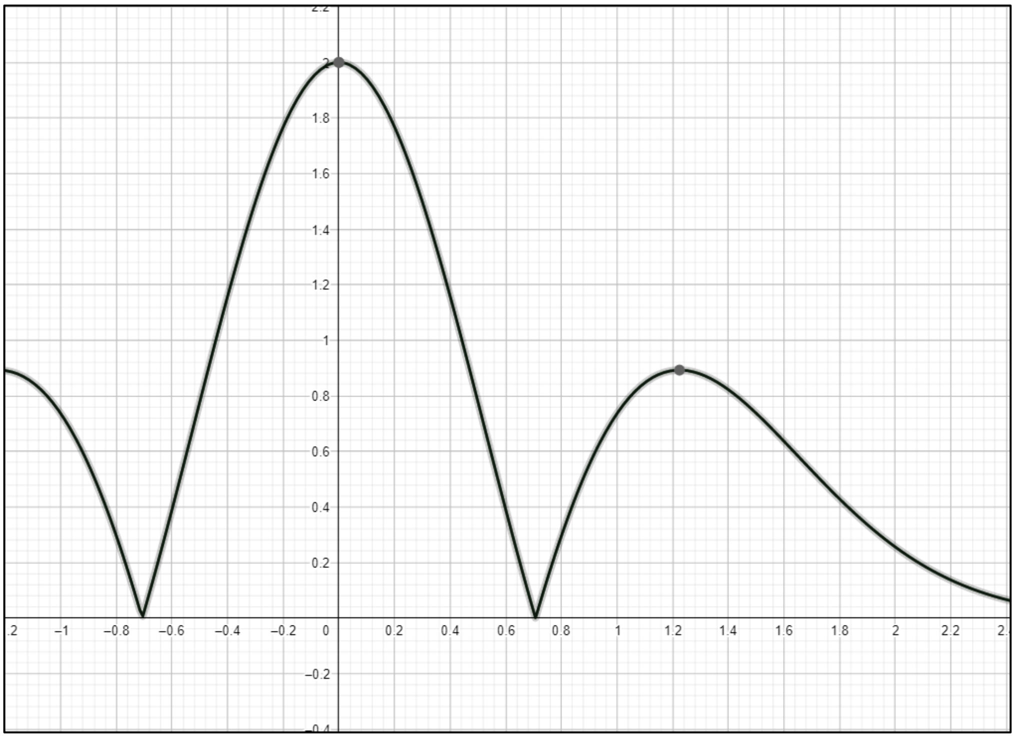
Primeiramente, descobrimos a primeira e segunda derivada de f(x):

f(x) = e^{-x^2}

f '(x) = $\frac{-2x}{e^{x^2}}$ ✓

f ''(x) = $\frac{4x^2-2}{e^{x^2}}$ ✓

Depois descobrimos qual o máximo de |f ''(x)| no intervalo [0 , 1]:



✓

Gráfico 1 - Módulo da Segunda Derivada de f(x)

Aqui concluímos que o máximo é de 2 ✓ no intervalo [0 , 1], sendo por isso possível calcular o valor de n da expressão:

$$\varepsilon_{mpm} = \alpha\frac{(b-a)^3}{24n^2} = n > \sqrt{\alpha\frac{(b-a)^3}{0{,}024}}$$ ✓

Sendo que o resultado da expressão será de n > 9,12, logo para a resolução deste exercício iremos usar n = 10

Por fim, temos tudo necessário para a utilização do método do ponto médio.

Usámos como código no Editor:

```
function s = IntMPM (f, a, b, n)

      dx = (b-a)./n;
      dx2 = dx./2;
      x = a:dx:(b-dx);
      s = dx .* sum(f(x+dx2));
end
```

E como código no command window:

```
>> x = sym('x');
f=@(x)(exp(1).^(-(x.^2)));
a = 0;
b = 1;
n = 10;
resultado = IntMPM (f, a, b, n)
```

Que deu como resultado deste exercício:

```
resultado =

    0.7471
```

Concluindo assim esta alínea.

4 / 5

b)

Nesta alínea foi pedida para calcular a integral dada utilizando o polinómio obtido pela fórmula de MacLaurin até aos termos da 3º ordem da função f(x).

Primeiro de tudo descobrimos a primeira, segunda e terceira derivada de f(x), sendo elas:

f(x) = e^{-x^2}

f ’(x) = $\frac{-2x}{e^{x^2}}$

f ’’(x) = $\frac{4x^2 - 2}{e^{x^2}}$

f ’’’(x) = $\frac{12x - 8x^3}{e^{x^2}}$

Depois calculámos cada expressão usando o valor x = 0, obtendo os seguintes resultados:

f(0) = 1

f ’(0) = 0

f ’’(0) = -2 ✓

f ’’’(0) = 0

Sendo assim possível utilizar a fórmula de MacLaurin com a seguinte expressão:

$$f(a)+f'(a)(x-a)+\frac{f''(a)}{2}(x-a)^2+\frac{f'''(a)}{6}(x-a)^3$$

Desta forma, obtivemos o polinómio: $1-x^2$ ✓

Por fim, com todos os dados necessários, calculámos a integral utilizando o seguinte código no Editor:

```
function s = IntMPM (f, a, b, n)

      dx = (b-a)./n;
      dx2 = dx./2;
      x = a:dx:(b-dx);
      s = dx .* sum(f(x+dx2));
end
```

A ideia aqui era calcular o integral analiticamente, uma vez que a fórmula de MacLaurin dá origem a um polinómio — que sabemos sempre integrar exatamente.

E como código no Command Window:

```
>> x = sym('x');
f=@(x)(1-(x.^2));
a = 0;
b = 1;
n = 10;
resultado = IntMPM (f, a, b, n)
```

Que deu como resultado deste exercício:

```
resultado =

    0.6675
```

Neste grupo foi pedido a resolução de duas alíneas, ambas para calcular a integral indicada no enunciado. Na alínea a) foi pedido para a calcular numericamente, com um erro inferior a 0,001, e utilizar o método do ponto médio, dando como resultado 0,7471. Já na alínea b), foi pedido para a calcular utilizando o polinómio obtido através da fórmula de MacLaurin até aos termos de 3º ordem da função dada, obtendo o resultado 0,6675.

O gráfico 1 não permite concluir isto, porque é o gráfico de f''(x)

Assim, podemos concluir que apesar de calcularem a mesma integral apresentam resultados diferentes, isto é explicado, uma vez que a resolução realizada na alínea a) representa uma maior área no plano cartesiano do que a realizada na alínea b), como se pode observar nos gráficos 1 e 2, este último está apresentado abaixo deste comentário.

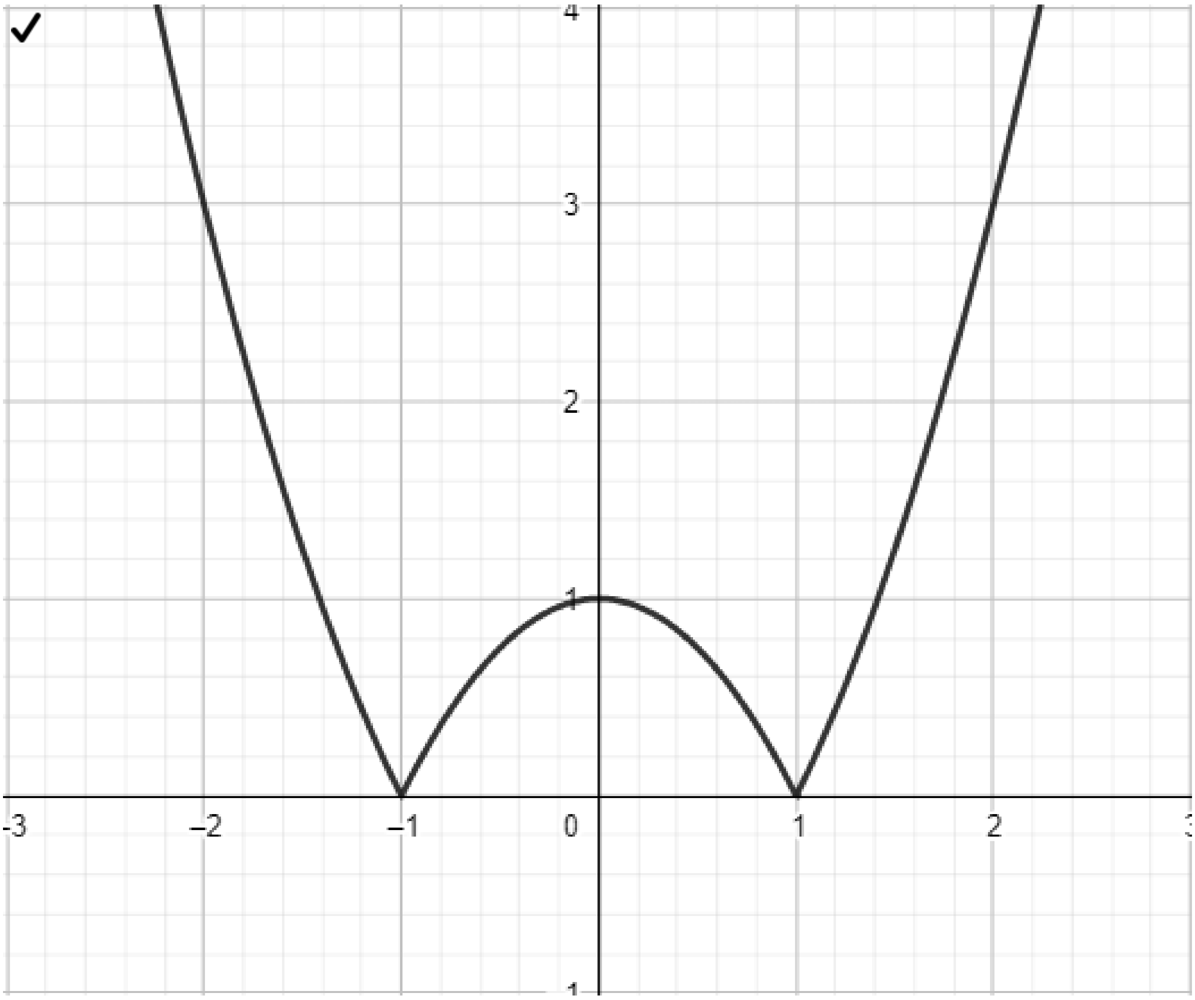


Gráfico 2 - Módulo do Polinómio obtido pelo MacLaurin

Concluindo assim esta alínea e o Grupo I.

Grupo II

No grupo II a tabela com todos os resultados vai estar no fim, em conjunto com o código que contém a organização e exposição dos resultados na tabela.

O ficheiro que contém f(x), f.m, encontra-se se seguida.

```
function y = f(x)
    y = atan(sin(x));
end
```

3.5 / 4

a) O método dos trapézios revela-se ser o seguinte:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[f(x_0) + f(x_n) + 2\sum_{i=1}^{n-1}(f(x_i)]$$

, sendo $\Delta x = (b-a)/n$ e $x_i = a + i\,\Delta x$. No nosso caso, $a = 0, b = x, f(x) = \arctan(\sin(x)), n \geq \sqrt{\frac{K*(a-b)^3}{12*|E_T|}}$. Este n equivale ao número de trapézios que o programa irá calcular, e é o inteiro calculado diretamente do erro, $|E_T|$ que o método produz. Neste caso, $|E_T| = 0.001$. O nosso K é calculado através da visualização direta do máximo do gráfico $y = f''(x)$.

```
n = ceil((sqrt((K * (deltaX^3))/(12*maxerror))));
```

```
X = 0:pi/100:2*pi;
Y = f(X);
dY = gradient(Y(:))./gradient(X(:));
dYY = gradient(dY(:))./gradient(X(:));
plot(X, dYY);
```

No programa vai ser calculado $g(x)$ para cada x presente na tabela. Para isso vai ser usado um ciclo `for`. Na tabela vai ser também mostrado o n e $|E_T|$ calculado para cada x.

De seguida encontra-se o ficheiro A.m:

```
function ret = A(xs, maxerror)

    format long;
    K = 0.9;
    a = 0;
    results = [];
    erros = [];
    Ns = [];
    for b = double(xs)
        deltaX = b - a;
        n = ceil((sqrt((K * (deltaX^3))/(12*maxerror))));
```

<— Porque escolheram K=0.9? Não encontro o cálculo que o justifica nem o gráfico que produziram em plot(X,dYY)

```
        x = a:deltaX/n:b;
        x = x(:);
        y = f(x);
        y = y(:);
        result = (ones(1,n) .* (deltaX/n)) * (y(1:end-1, :) + y(2:end, :))/2;
        results = [results result];
        erroAbsolutoMenorQue = (K * (deltaX^3))/(12*(n^2));
        erros = [erros erroAbsolutoMenorQue];
        Ns = [Ns n];
    end
    ret = [double(results); erros ;Ns];
end
```

4 / 4

b)

O método referido no enunciado aparenta ser o seguinte:

$$g'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2(x_i - x_{i-1})}$$

. Quanto mais perto $x_i - x_{i-1}$ estiver de 0, maior a precisão do resultado com o real. Para calcular $x_i - x_{i-1}$ entre os pontos, podemos usar a função `diff`. Como os pontos inicial e final não contêm anteriores e posterior, respetivamente, usa-se a formula sem a necessidade de ter tais.

De seguida encontra-se o ficheiro B.m:

```
function ret = B(xs, gxs)
    hs = diff(double(xs));
    ret = ones(size(xs));
    ret(1) = (gxs(2) - gxs(1) ) / hs(1);
    ret(end) = (gxs(end) - gxs(end-1)) / hs(end);
    ret(2:end-1) = (gxs(3:end) - gxs(1:end-2) )./(2*hs(2:end));
end
```

1.5 / 2

c)

O teorema fundamental do cálculo descreve que se g(x) estiver definido em $[a, b]$, e $g(x) = \int_a^x f(t)dt$, então $\forall x \in [a, b]$: $g'(x) = f(x)$. Sendo $f(x) = arctan(sin(x))$ e a = 0, $g'(x) = f(x) \equiv g''(x) = f'(x)$.

De seguida encontra-se o ficheiro C.m:

```
function ret = C(xs)
    fxs = f(xs);
    dgdxExato = fxs;
    syms x
```

```
    dfdx = diff(f(x));
    dg2dx2Exato = double(subs(dfdx, x, xs));
    ret = [dgdxExato; dg2dx2Exato];
end
```

análise dos resultados)

De seguida encontra-se code.m, o ficheiro que constrói a tabela, e a própria.

```
clear;
K = 0.9;
maxerror = 0.001;
h = 0.0000000000000000001;
xs = sym([]);
for i = 1 : 11
    xs(i) = (i-1) * (pi/5);
end
strXs = string(xs);
xs = double(xs);
format long;
t1 = A(xs, maxerror);
t2 = B(xs, t1(1, :));
t3 = B(t1(1, :), t2(1, :));
t4 = C(xs);
T = array2table( ...
    [t1 ; t2 ; t4(1, :); t3; t4(2, :)], 'VariableNames', strXs,'RowNames',{
...
        'g(x)',
        'Erro Absoluto Menor Que',
        'nº de trapézios',
        'dg/dx calculado' ,
        'dg/dx exato',
        'd^2g/dx^2 calculado' ,
        'd^2g/dx^2 exato'
        });

uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...
     'RowName',T.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0,
1, 1]);
```

✓

	0	pi/5	(2*pi)/5	(3*pi)/5	(4*pi)/5	pi	(6*pi)/5	(7*pi)/5	(8*pi)/5	(9*pi)/5	2*pi
g(x)	0	0.1804	0.6005	1.0888	1.5089	1.6899	1.5092	1.0891	0.6010	0.1809	5.3993e-17
Erro Absoluto Menor Que	NaN	7.4415e-04	8.8065e-04	9.4953e-04	9.7195e-04	9.6854e-04	9.8106e-04	9.9705e-04	9.9179e-04	9.9073e-04	9.9120e-04
nº de trapézios	0	5	13	23	35	49	64	80	98	117	137
dg/dx calculado	0.2872	0.4779	0.7228	0.7229	0.4784	1.7914e-04	-0.4781	-0.7227	-0.7228	-0.4782	-0.2879
dg/dx exato	0	0.5314	0.7603	0.7603	0.5314	1.2246e-16	-0.5314	-0.7603	-0.7603	-0.5314	-2.4493e-16
d^2g/dx^2 calculado	1.0570	0.5186	0.2509	-0.2909	-1.9970	2.6460	0.8605	0.2506	-0.2910	-1.2022	-1.0525
d^2g/dx^2 exato	1	0.6013	0.1623	-0.1623	-0.6013	-1	-0.6013	-0.1623	0.1623	0.6013	1

✓ ✓ ✓ ✓ <— algo errado ✓

Os resultados da a) mostram que para um domínio maior entre a e b, maior será necess o número de trapézios de forma a manter um erro baixo. ✓

(Tabela espectacular!)

Os resultados da b), comparado com os resultados da c), mostram uma grande imprecisão. Isto deve-se ao $x_i - x_{i-1}$ (aqui sempre igual a $\frac{\pi}{5}$) estar muito longe de 0.

Mas a imprecisão é maior nos extremos do intervalo, pois a fórmula utilizada para calcular as derivadas nesses dois pontos é de ordem menor, e portanto menos precisa.