Análise Matemática II

Atividade 03 – Máquina para Derivação e Integração

Docente: Arménio Correia

Miguel Ângelo Rodrigues Ferreira nº 2020107016 - LEI

Pablo Oliveira Amaral nº 2020143935 - LEI

Paulo Henrique Figueira Pestana de Gouveia nº 2020121705 – LEI

Coimbra, 18 de junho de 2021

Índice

Índice....1
1 Introdução....2
2 Métodos numéricos para a derivação....3
2.1 Fórmulas....3
2.1.1 Fórmula de diferenças finitas em 2 pontos....3
2.1.2 Fórmula de diferenças finitas em 3 pontos....3
2.2 Funções em MATLAB....5
2.2.1 Derivada progressiva de diferenças finitas em 2 pontos....5
2.2.2 Derivada regressiva de diferenças finitas em 2 pontos....5
2.2.3 Derivada progressiva de diferenças finitas em 3 pontos....5
2.2.4 Derivada regressiva de diferenças finitas em 3 pontos....6
2.2.5 Derivada centrada de diferenças finitas em 3 pontos....6
2.2.6 Derivada centrada de diferenças finitas em 3 pontos....6
2.3 Derivação simbólica no MATLAB....7
2.3.1 Função diff()....7
3 Métodos numéricos para integração....8
3.1 Fórmulas....8
3.2 Algoritmos e Funções em MATLAB....9
3.3 Integração simbólica no MATLAB....10
3.3.1 Função int()....10
4 Exemplos de aplicação....12
4.1 Derivação....12
5 Conclusão....16

1 Introdução

Neste relatório vamos introduzir a noção de derivada e de integral, e ao mesmo tempo vamos mostrar como foi criada a máquina de derivação e integração, mostrando alguns exemplos práticos com a GUI do MATLAB. Dizemos que derivada é o limite da razão entre o acréscimo de uma função e o acréscimo da variável independente, quando tende para zero. Já a integral de uma função foi criada para determinar uma área sob uma curva no plano cartesiano.

2 Métodos numéricos para a derivação

Tendo em conta que a maioria das vezes o cálculo de uma derivada é lento em termos de algoritmo, surgiu a ideia de criar aproximações numéricas. Ao falarmos destas aproximações numéricas, somos introduzidos às diferentes fórmulas de diferenças finitas. Dentro destas fórmulas há as progressivas, regressivas e centradas.

2.1 Fórmulas

2.1.1 Fórmula de diferenças finitas em 2 pontos

Progressivas

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{h}$$

Regressivas

$$f'(x_k) = \frac{f(x_k) - f(x_k)}{h}$$

2.1.2 Fórmula de diferenças finitas em 3 pontos

Progressivas

$$\frac{-3f(x_k) + 4f(x_{k+1}) - f(x_{k+2})}{2h}$$

Regressivas

$$\frac{f(x_{k-2}) - 4f(x_{k-1}) + 3f(x_k)}{2h}$$

Centradas

$$\frac{f(x_{k+1}) - f(x_{k-1})}{h^2}$$

2ª Derivada

$$\frac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1})}{h^2}$$

2.1.3 Formula da derivada parcial

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x} f(x, y)$$

Dependendo se o cálculo esta a ser feito em ordem a x ou a y, a variável oposta de ver tratada como uma constante.

2.2 Funções em MATLAB

2.2.1 Derivada progressiva de diferenças finitas em 2 pontos

```
function [x,y,dydx] = NDerivadaFP2(f,a,b,h,y)
        x = a:h:b;
        n = length(x);
    if nargin == 5
        y = f(x);
    end
        dydx = zeros(1,n);
    for i = 1:n-1
        dydx(i) = (y(i+1)-y(i))/h;
    end

    dydx(n) = dydx(n-1);
    %dydx(n) = (y(n)-y(n-1))/h;
end
```

2.2.2 Derivada regressiva de diferenças finitas em 2 pontos

```
function [x,y,dydx]=NDerivadaFR2(f,a,b,h,y)
    x=a:h:b;
    n=length(x);
    if nargin==5
        y=f(x);
    end
    dydx=zeros(1,n);
    dydx(1)=(y(2)-y(1))/h;
    for i=2:n
        dydx(i)=(y(i)-y(i-1))/h;
    end
end
```

2.2.3 Derivada progressiva de diferenças finitas em 3 pontos

```
function [x,y,dydx]=NDerivadaFP3(f,a,b,h,y)
    x=a:h:b;
    n=length(x);
    if nargin==5
        y=f(x);
    end
        dydx=zeros(1,n);
    for i=1:n-2
        dydx(i)=( (-3)*y(i) + 4*y(i+1) - y(i+2) ) / (2*h);
    end
    dydx(n-1)=( y(n-3) - 4*y(n-2) + 3*y(n-1) )/(2*h);
    dydx(n)=( y(n-2) - 4*y(n-1) + 3*y(n) )/(2*h);
end
```

2.2.4 Derivada regressiva de diferenças finitas em 3 pontos

```
function [x,y,dydx]=NDerivadaFR3(f,a,b,h,y)
    x=a:h:b;
    n=length(x);
    if nargin==5
        y=f(x);
    end
    dydx=zeros(1,n);
    dydx(1)=( (-3)*y(1) + 4*y(2) - y(3) )/(2*h);
    dydx(2)=( (-3)*y(2) + 4*y(3) - y(4) )/(2*h);
    for i=3:n
        dydx(i)=( y(i-2) - 4*y(i-1) + 3*y(i) )/(2*h);
    end
end
```

2.2.5 Derivada centrada de diferenças finitas em 3 pontos

```
function [x,y,dydx]=NDerivada2O(f,a,b,h,y)
    x=a:h:b;
    n=length(x);
    if nargin==5
        y=f(x);
    end
    dydx=zeros(1,n);
    dydx(1)=(y(3)-2*y(2)+y(1))/(h^2);
    for i=2:n-1
        dydx(i)=(y(i+1)-2*y(i)+y(i-1))/(h^2);
    end
    dydx(n)=(y(n)-2*y(n-1)+y(n-2))/(h^2);
end
```

2.2.6 Derivada centrada de diferenças finitas em 3 pontos

```
function [x,y,dydx]=NDerivadaFC3(f,a,b,h,y)
    x=a:h:b;
    n=length(x);
    if nargin==5
        y=f(x);
    end
    dydx=zeros(1,n);
    dydx(1)=(y(2)-y(1))/h;
    for i=2:n-1
        dydx(i)=(y(i+1)-y(i-1))/(2*h);
    end
    dydx(n)=(y(n)-y(n-1))/h;
end
```

2.3 Derivação simbólica no MATLAB

2.3.1 Função diff()

Segundo a documentação do MATLAB, podemos ver que se fizermos Y=diff(x), calculamos a diferença entre os elementos adjacentes de "X" ao longo da primeira dimensão do array cujo tamanho é diferente de 1.

Ou seja, com o diff() podemos calcular derivadas de qualquer ordem.

```
f=str2sym('4*x^2+2*x^2')
diff(f)
```

f = $6\,x^2$

ans = $12\,x$

É também possível calcular derivadas de qualquer ordem, apenas é necessário especificar o número da ordem no segunda parâmetro da função. Em baixo estamos então a mostrar como podemos usar o diff() para calcular uma derivada de ordem 3.

```
f=str2sym('6*x^5+6*x^2')
diff(f)

diff(f,3)
```

f = $6\,x^5 + 6\,x^2$

ans = $30\,x^4 + 12\,x$

ans = $360\,x^2$

3 Métodos numéricos para integração

Apesar de existirem vários métodos numéricos para a integração, apenas a Regra de Trapézios e a Regra de Simpson serão mencionados, uma vez que serão utilizados no trabalho realizado.

- Regra dos Trapézios

Trata-se do caso mais simples da Fórmula de Newton-Cotes fechada. Neste caso, consideramos n= 1 e temos dois nós de integração.

- Regra de Simpson

A regra de Simpson é mais uma vez um exemplo da fórmula de Newton Cotes fechada, mas em vez de considerarmos a aproximação e cada subintervalo através de um polinómio interpolador do 1º grau, pensamos numa aproximação melhor considerando um polinómio de 2º grau, ou seja, uma parábola. Para isto precisamos então de um ponto médio para considerarmos a regra de integração simples.

3.1 Fórmulas

Regra dos Trapézios

$$T(f) = \frac{I_1(f) = (f(a) + f(b))(b-a)}{2}$$

Regra de Simpson

$$S(f) = I_2(f) = \frac{\big(f(a) + 4f(c) + f(b)\big)h}{3}$$

3.2 Algoritmos e Funções em MATLAB

Algoritmo de Trapézios

```
function T = RTrapezios(f,a,b,n)

h = (b-a)/n;
s = 0;
x=a;

for i=1:n-1
    x=x+h;
    s = s+f(x);
end

T = h/2*((f(a) +2*s + f(b)));
end
```

Algoritmo de Simpson

```
function S = RSimpson(f,a,b,n)
    h = (b-a)/n;
    x = a:h:b;
    s = 0;
    for i = 2:n-1
        if mod(i,2)==0
            s = s+2*f(x(i));
        else
            s = s+4*f(x(i));
        end
    end
    S = h/3*(f(a)+s+f(b));
end
```

3.3 Integração simbólica no MATLAB

3.3.1 Função int()

Ao pesquisar no site do MATLAB, podemos ver que:

F = int(expr) calcula o integral indefinido de expr e usa a variável de integração padrão determinada por symvar(expr,1). Se expr for uma constante a variável padrão é x.

```
syms x
expr = -2^x/(1+x^2)^2

F= int(expr)
```

expr =

$$-\frac{2^x}{(x^2+1)^2}$$

F =

$$\int \left(-\frac{2^x}{(x^2+1)^2}\right) \mathrm{d}x$$

Para calcular o integral indefinido de expr tendo em conta a variável simbólica escalar var, utiliza-se F = int(expr, var).

```
syms x z
f(x,z)=x/(1+z^2)

Fx= int(f, x)

Fz=int(f,z)

var=symvar(f,1)

F=int(f)
```

f(x, z) =

$$\frac{x}{z^2+1}$$

Fx(x, z) =

$$\frac{x^2}{2\,(z^2+1)}$$

Fz(x, z) = $x\,\mathrm{atan}(z)$

var = x

F(x, z) =

$$\frac{x^2}{2\,(z^2+1)}$$

F = int(expr, a, b) calcula o integral indefinido de expr de a até b e usa variável de integração determinada por symvar(expr,1). Se expr for uma constante a variável padrão é x.

```
syms x
expr = x*log(1+x)
F=int(expr,[0 1])

syms t
F= int(2*x, [sin(t) 1])
```

expr = $x\log(x+1)$

F =

$$\frac{1}{4}$$

F = $\cos(t)^2$

Por fim, para calcular o integral indefinido de expr tendo em conta a variável simbólica escalar var de a até b, F = int(expr,var,a,b).

```
syms x
f= cos(x)/sqrt(1+x^2)
Fint=int(f,x,[0 10])

Fvpa = vpa(Fint)
```

f =

$$\frac{\cos(x)}{\sqrt{x^2+1}}$$

Fint =

$$\int_0^{10} \frac{\cos(x)}{\sqrt{x^2+1}}\,dx$$

Fvpa =

0.375706282990797234784

4 Exemplos de aplicação

4.1 Derivação

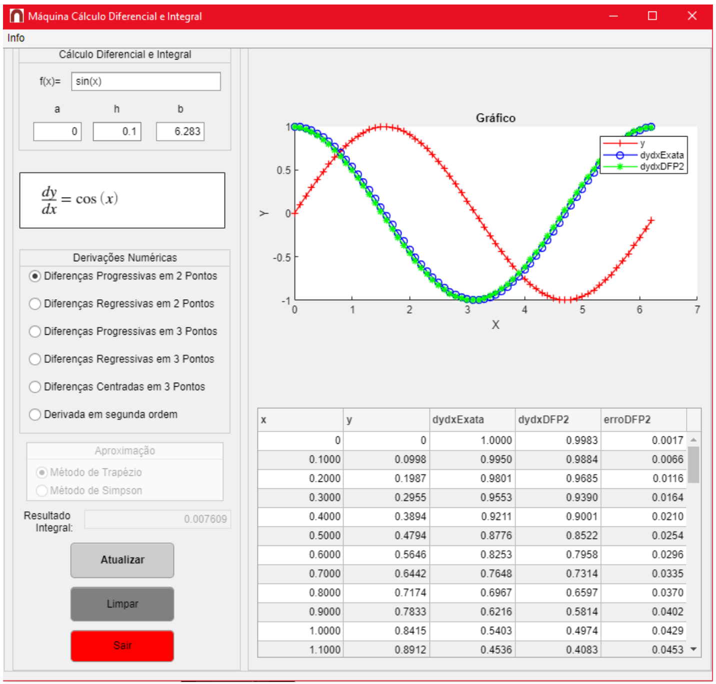

Figura 1-Progressivas em 2 pontos

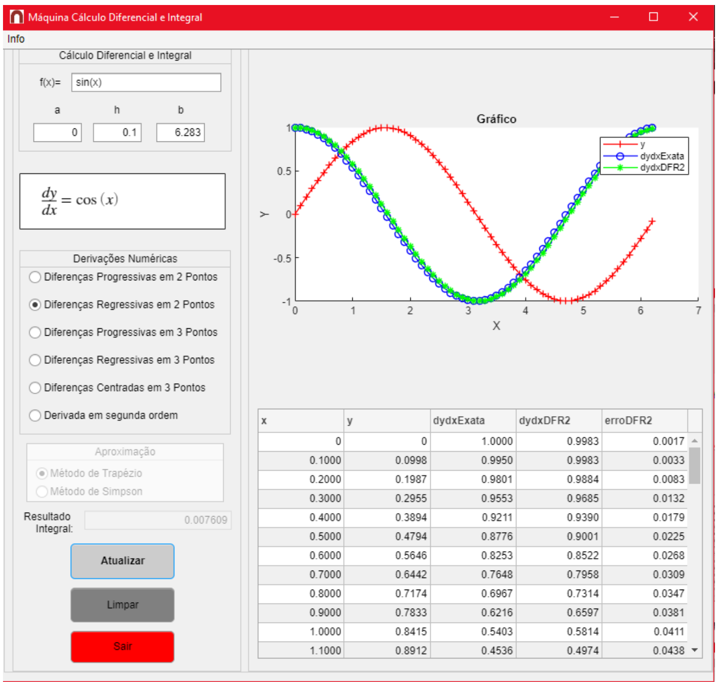

Figura 2-Regressiva em 2 pontos

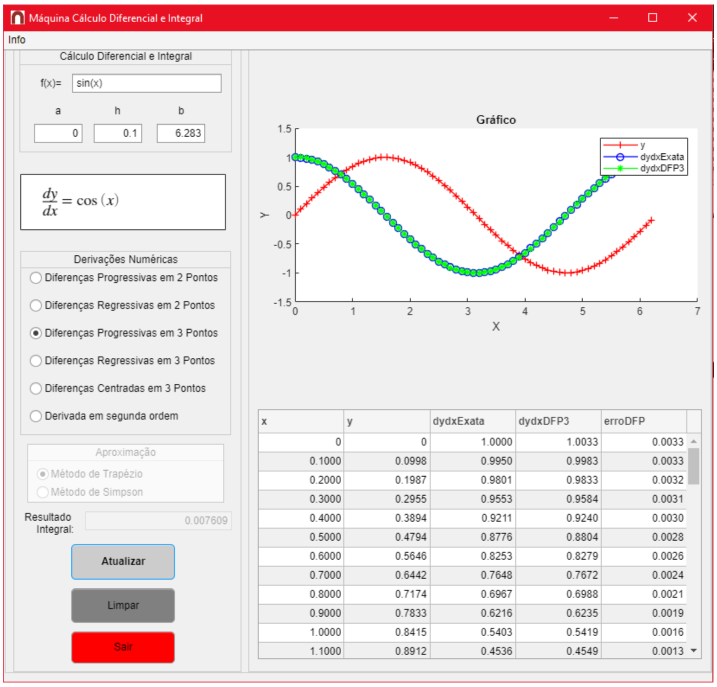

Figura 3-Progressivas em 3 pontos

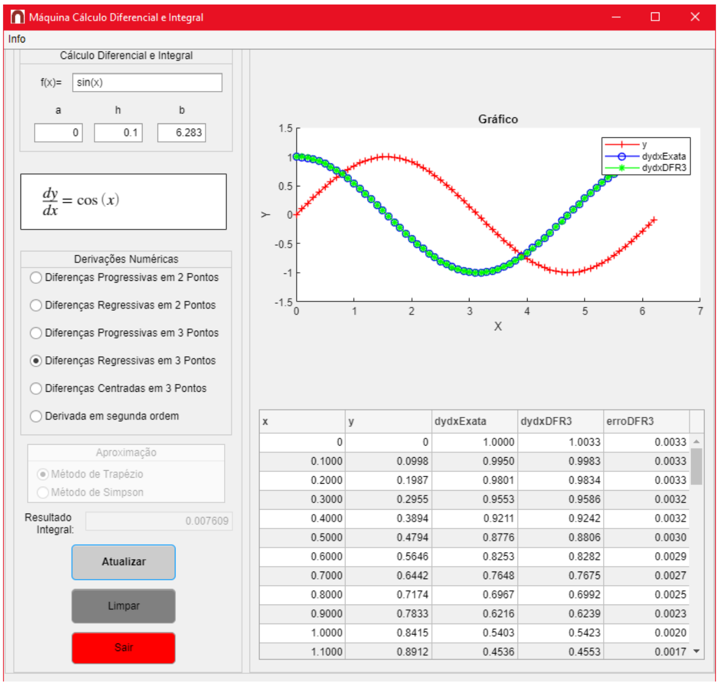

Figura 4-Regressivas em 3 pontos

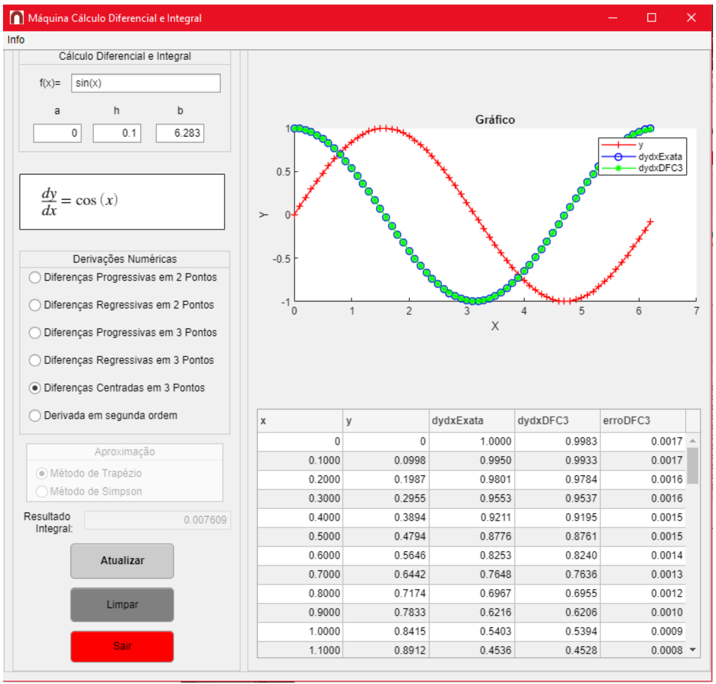

Figura 5-Centradas em 3 pontos

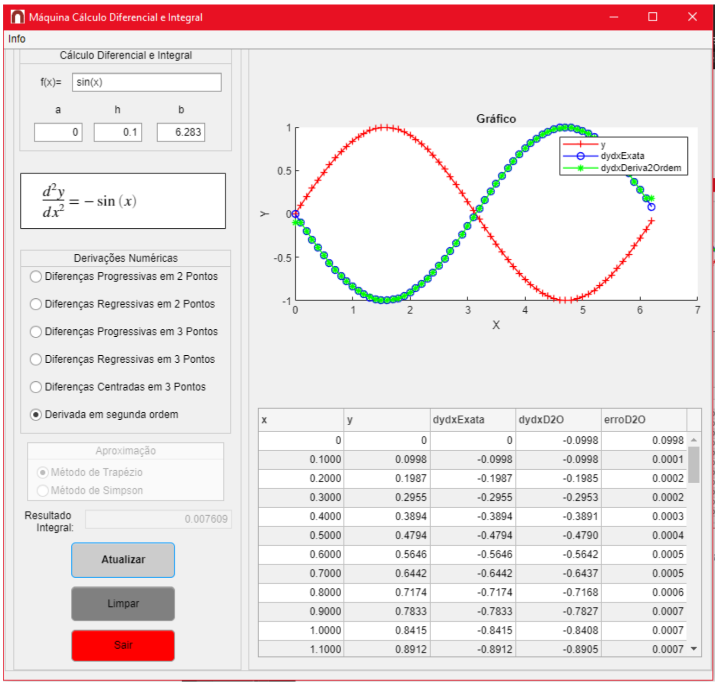

Figura 6-Derivadas de 2ª ordem

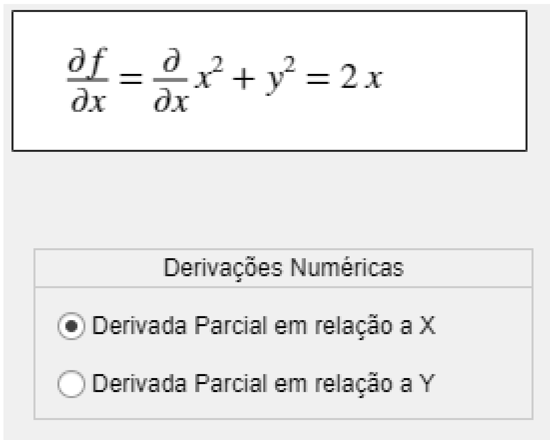

Figura 7-Derivada parcial em relação a X

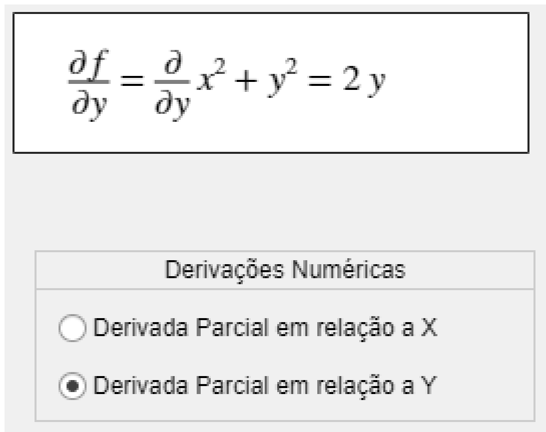

Figura 8-Derivada parcial em relação a X

5 Conclusão

Com este trabalho podemos concluir que os métodos numéricos têm diversas aplicações, quer para derivadas, quer para integrais, o que facilita a resolução de problemas mais densos de variadas áreas da ciência (e não só), onde, por vezes, necessitamos de calcular integrais analiticamente, algo que pode não ser possível

Como já visto anteriormente, verificamos que quanto maior for o número de subintervalos n, menor é o erro dos métodos / fórmulas. A introdução do tamanho de cada subintervalo h tem o efeito contrário: quanto menor o tamanho introduzido, menor o erro dos métodos / fórmulas.

Em relação às fórmulas da derivação numérica, a comparação encontrada foi que a melhor aproximação ao valor real se origina a partir das fórmulas que utilizam 3 pontos, comparativamente às que apenas utilizam 2 pontos (que muitas vezes apresentam o dobro do erro, ou mais).