Análise Matemática II | Engenharia Informática

Trabalho Prático Nº3

Máquina para Derivação e Integração

André Amando Rodrigues Lopes a2019139754
Nuno Gabriel Tavares Honório a2019126457
Rafaela Oliveira Carvalho a2019127935
Samuel Pires Tavares a2019126468

2019/2020

Índice

Índice de Imagens

1. Introdução

Para aproximar o valor de derivada num ponto temos que aplicar uma das fórmulas de diferenças finitas. Com este trabalho, para além de termos de aplicar estas fórmulas, implementámos em Matlab funções para estas, bem como para as fórmulas da Integração Numérica implementadas na atividade 01, tentando ao máximo melhorá-las. Com esta atividade desenvolvemos também a derivação numérica e simbólica em Matlab.

O professor Arménio Correia propôs a seguinte atividade:

Frequentemente somos confrontados com a necessidade de determinar valores da derivada de uma função num conjunto de pontos, conhecendo apenas o valor da função nesses pontos.

Questão: Como aproximar o valor da derivada num ponto?
Resposta: Aplicar uma das fórmulas de diferenças finitas: progressivas, regressivas ou centradas.

Seja f uma função definida em $[a, b]$ e suficientemente regular, conhecida num conjunto de pontos da partição uniforme $a = x_0 < x1 < \cdots < x_n = b$

Fórmulas de diferenças finitas em 2 pontos:

Progressivas » $f'(x_k) := \dfrac{f(x_{k+1}) - f(x_k)}{h}$

Regressivas » $f'(x_k) := \dfrac{f(x_k) - f(x_{k-1})}{h}$

Fórmulas de diferenças finitas em 3 pontos:

Progressivas » $f'(x_k) := \dfrac{-3f(x_k) + 4f(x_{k+1}) - f(x_{k+2})}{2h}$

Regressivas » $f'(x_k) := \dfrac{f(x_{k-2}) - 4f(x_{k-1}) + 3f(x_k)}{2h}$

Centradas » $f'(x_k) := \dfrac{f(x_{k+1}) - f(x_{k-1})}{2h}$

2ª derivada » $f''(x_k) := \dfrac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1})}{h^2}$

2. Métodos Numéricos para Derivação

2.1. Formulas

Fórmulas de diferenças finitas em 2 pontos:

Progressivas:

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

Regressivas:

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

Fórmulas 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:

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

2ª Derivada:

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

2.2. Algoritmo/Função

Algoritmo:

DFProgressivas_2

Input: f,a,b,h,y Output: x,y,dydx

x <- Inicializar vetor de a até b com um salto de h

n <- Comprimento de x

Se nº de argumentos de entrada = 4

y <- f(x)

Fim Se

dydx <- Vetor de zero de 1 até n

Equanto i=1 ≤ n-1

dydx(i) <- (y (i+1) – y (i)) / h

Fim Enquanto

dydx(n) <- (y(n) – y(n-1)) / h

Função:

```
function [x,y,dydx]=DI_DFProgressivas_2(f,a,b,h,y)
x=a:h:b;
n=length(x);
if nargin==4
    y=f(x);
end;
dydx=zeros(1,n);
for i=1:n-1
    dydx(i)=(y(i+1)-y(i))/h;
end;
dydx(n)=(y(n)-y(n-1))/h;
```

Algoritmo:

DFRegressivas_2

Input: f,a,b,h,y

Output: x,y,dydx

x <- Inicializar vetor de a até b com um salto de h

n <- Comprimento de x

Se nº de argumentos de entrada = 4

y <- f(x)

Fim Se

dydx <- Vetor de zero de 1 até n

dydx(1) <- (y(2) – y(1)) / h

Equanto i=2 ≤ n

dydx(i) <- (y (i) – y (i-1)) / h

Fim Enquanto

Função:

```
function [x,y,dydx]=DI_DFRegressivas_2(f,a,b,h,y)
x=a:h:b;
n=length(x);
if nargin==4
    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
```

Algoritmo:

DFProgressivas_3
Input: f,a,b,h,y
Output: x,y,dydx
x <- Inicializar vetor de a até b com um salto de h
n <- Comprimento de x

Se nº de argumentos de entrada = 4
y <- f(x)
Fim Se

Equanto i=1 ≤ n-2

dydx(i) <- ((-3) * y(i) + 4*y(i+1) – y(i+2)) / (2*h)

Fim Enquanto

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-1)) / (2*h)

Função:

```
function [x,y,dydx]=DI_DFProgressivas_3(f,a,b,h,y)
x=a:h:b;
n=length(x);
if nargin==4
    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);
```

Algoritmo:

DFRegressivas_3
Input: f,a,b,h,y
Output: x,y,dydx
x <- Inicializar vetor de a até b com um salto de h
n <- Comprimento de x

Se nº de argumentos de entrada = 4
    y <- f(x)
Fim Se

dydx <- Vetor de zero de 1 até 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)

Equanto i=3 ≤ n

    dydx(i) <- (y(i-2) - 4 * y(i-1) + 3*yi)) / (2*h)
Fim Enquanto

Função:

```
function [x,y,dydx]=DI_DFRegressivas_3(f,a,b,h,y)
x=a:h:b;
n=length(x);
if nargin==4
    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;
```

Algoritmo:

DFCentradas_3
Input: f,a,b,h,y
Output: x,y,dydx
x <- Inicializar vetor de a até b com um salto de h
n <- Comprimento de x
Se nº de argumentos de entrada = 4
    y <- f(x)
Fim Se

dydx <- Vetor de zero de 1 até n
dydx(1) <- (y(2) – Y(1)) / (2*h)
Equanto i=2 ≤ n-1

    dydx(i) <- (y(i+1) - y(i-1)) / (2*h)

Fim Enquanto

dydx(n) <- (y(n) – y(n-2)) / (2*h)

Função:

```
function [x,y,dydx]=DI_DFCentradas_3(f,a,b,h,y)
x=a:h:b;
n=length(x);
if nargin==4
    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;
```

Algoritmo:

DFDerivadas_2
Input: f,a,b,h,y
Output: x,y,dydx
x <- Inicializar vetor de a até b com um salto de h
n <- Comprimento de x
Se nº de argumentos de entrada = 4
    y <- f(x)
Fim Se

dydx <- Vetor de zero de 1 até n
dydx(1) <- (y(2) – Y(1)) / (2*h)
Equanto i=2 ≤ n-1

    dydx(i) <- (y(i+1) - y(i-1)) / (2*h)

Fim Enquanto

dydx(n) <- (y(n) – y(n-2)) / (2*h)

Função:

```
function [x,y,dydx]=DI_DFDerivada2(f,a,b,h,y)
x=a:h:b;
```

```
n=length(x);
if nargin==4
    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);
```

3. Derivação Simbólica no Matlab

3.1. Diff

Funciona também com a função diff mas neste caso calcula as diferenças entre elementos adjacentes de X ao longo de um array cujo tamanho não é igual a 1. Pode ser representada das seguintes maneiras:

Y = diff(X)   Y = diff(X, n)   Y = diff(X, n, dim)

4. Métodos Numéricos para integração

4.1. Formulas

Regra dos Trapézios:

$$I_T(f) = \frac{h}{2}[f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)]$$
$$|E_T| \leq \frac{b-a}{12}h^2 M_2, \quad M_2 = \max_{x\in[a,b]} |f''(x)|$$

Regra de Simpson:

$$I_S(f) = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$
$$|E_S| \leq \frac{b-a}{180}h^4 M_4, \quad M_4 = \max_{x\in[a,b]} \left|f^{(4)}(x)\right|$$

4.2. Algoritmo/Função

Algoritmo:

```
RSimpson
Input: f,a,b,n
Output: t
h <- (b – a) / n
s <- 0
x <- a

Equanto i=1 ≤ n-1

        x <- x + h
        Se i/2 der resto 0
        s <- s + 2 * f(x)
        Se Não

                s <- s + 4 * f(x)

        Fim Se
Fim Enquanto
        s <- h * ( f(a) + f(b) + s ) /3
```

Função:

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

Algoritmo:

RTrapezios
Input: f,a,b,n
Output: t
h <- (b – a) / n
t <- 0
x <- a
Equanto i=1 ≤ n-1
    x <- x + h
    t <- t + 2 * f(x)

Fim Enquanto

t <- h * (f(a) + f(b) + t) / 2

Função:

```
function T=RTrapezios(f,a,b,n)
h=(b-a)/n;
t=0;
x=a;
for i=1:n-1
    x=x+h;
    t=t+2*f(x);
end
t=h*(f(a)+f(b)+t)/2;
```

5. Integração Simbólica no Matlab

5.1. Int

Para calcular integrais indefinidos pode-se usar o comando int:

```
>> int(f)                    % calcula a primitiva de f
ans = 3/4*x^4+4/3*x^3-5*x
```

Outro exemplo:

```
>> syms a b c x
>> h=a*x^2+b*x+c;
>> int(h,b)      % calcula a primitiva de h considerando que b é a variável
independente
ans = a*x^2*b+1/2*b^2*x+c*b
```

Ainda outro exemplo, agora com integrais definidos:

```
>> f=sym(sin(x))
f = sin(x)
>> int(f,0,pi)
ans = 2
```

6. Exemplos de aplicação e teste dos métodos

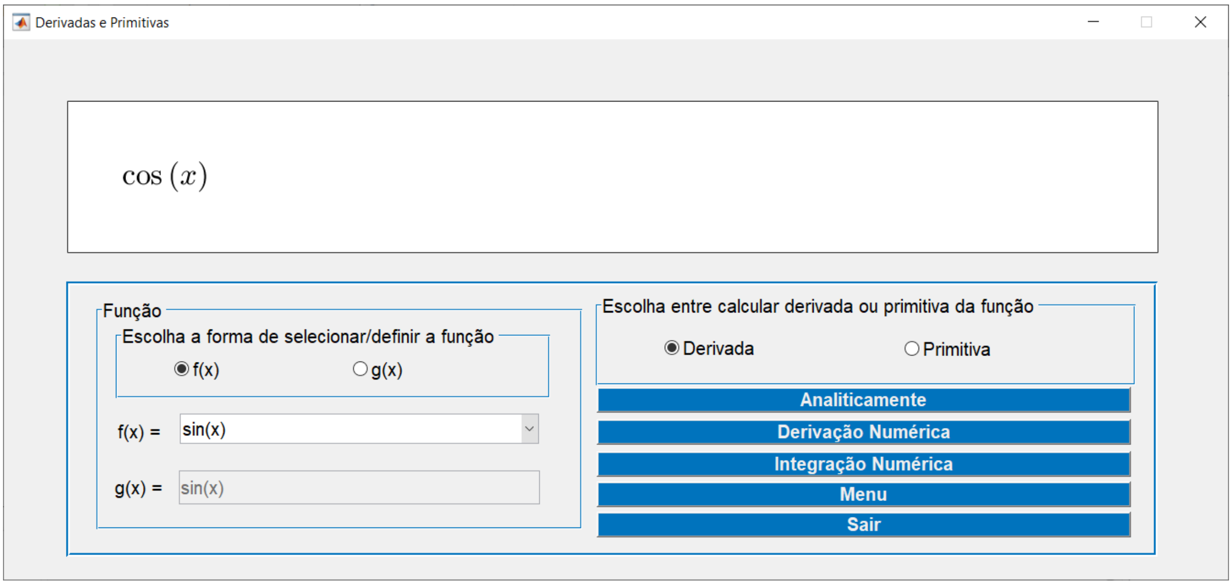


Figura 1: GUI - Derivadas e Primitivas

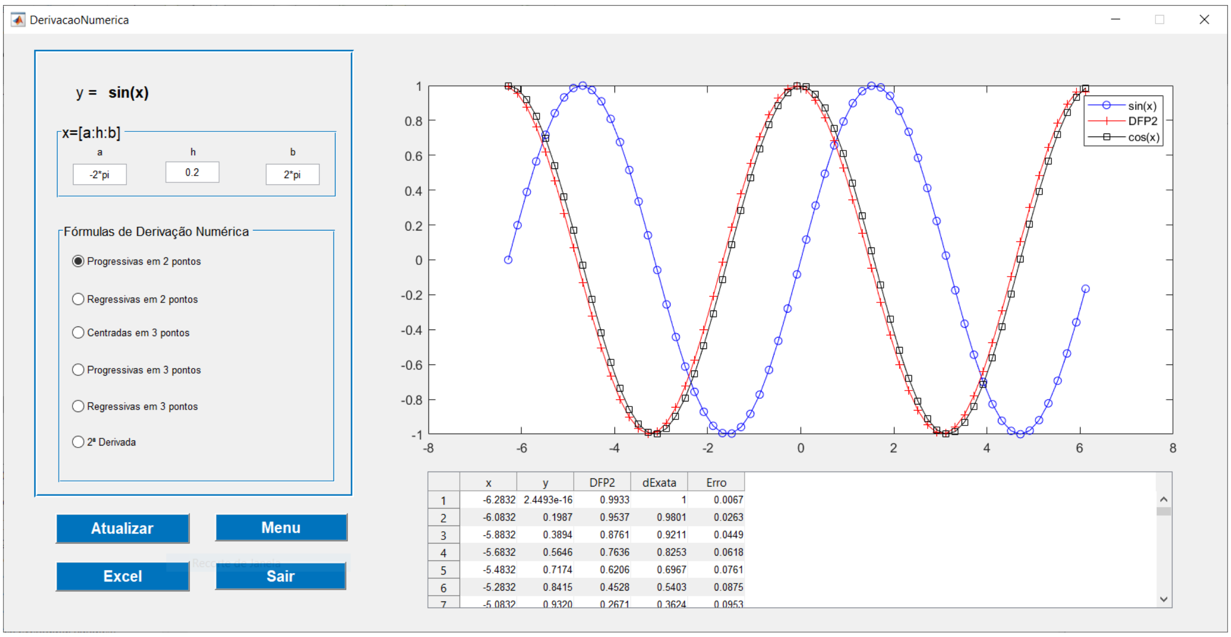
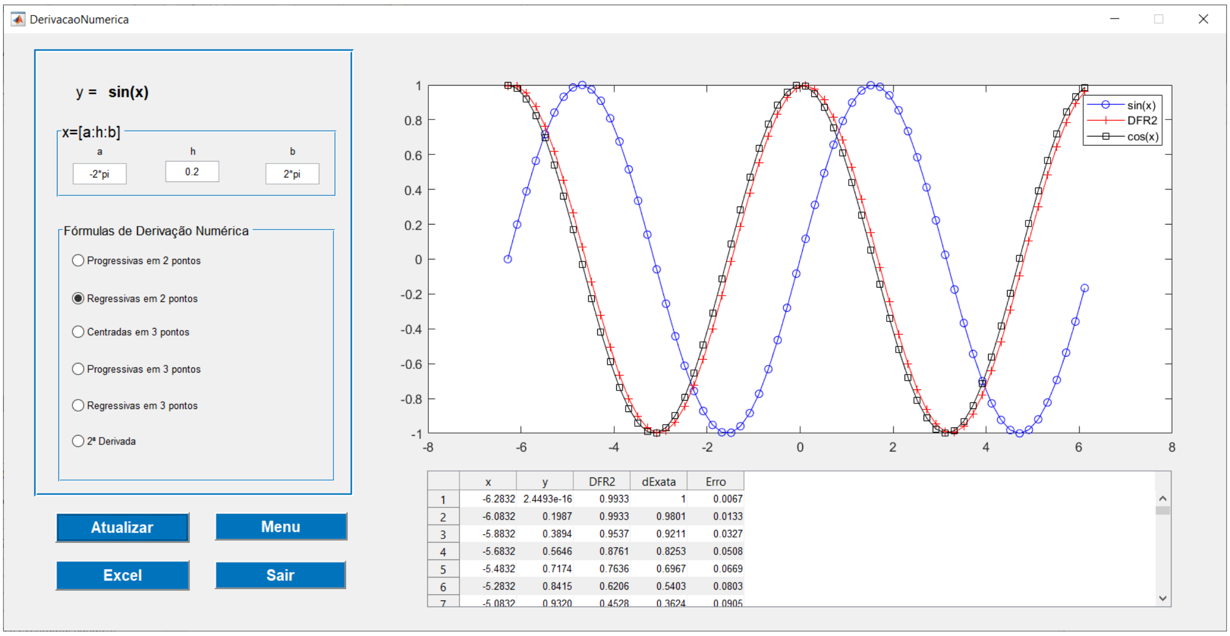
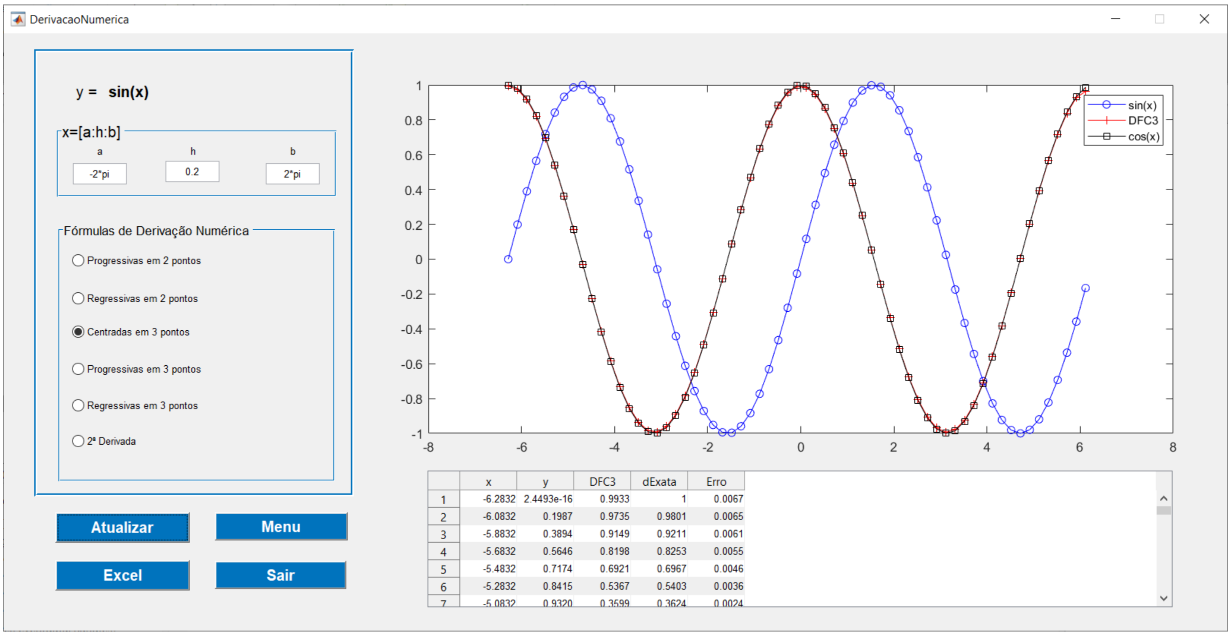


Figura 2: Função representada pelo Método das Diferenças Progressivas em 2 pontos



	x	y	DFR2	dExata	Erro
1	-6.2832	2.4493e-16	0.9933	1	0.0067
2	-6.0832	0.1987	0.9933	0.9801	0.0133
3	-5.8832	0.3894	0.9537	0.9211	0.0327
4	-5.6832	0.5646	0.8761	0.8253	0.0508
5	-5.4832	0.7174	0.7636	0.6967	0.0669
6	-5.2832	0.8415	0.6206	0.5403	0.0803
7	-5.0832	0.9320	0.4528	0.3624	0.0905

Figura 3: Função representada pelo Método das Diferenças Regressivas em 2 pontos



	x	y	DFC3	dExata	Erro
1	-6.2832	2.4493e-16	0.9933	1	0.0067
2	-6.0832	0.1987	0.9735	0.9801	0.0065
3	-5.8832	0.3894	0.9149	0.9211	0.0061
4	-5.6832	0.5646	0.8198	0.8253	0.0055
5	-5.4832	0.7174	0.6921	0.6967	0.0046
6	-5.2832	0.8415	0.5367	0.5403	0.0036
7	-5.0832	0.9320	0.3599	0.3624	0.0024

Figura 4: Função representada pelo Método das Diferenças Centradas

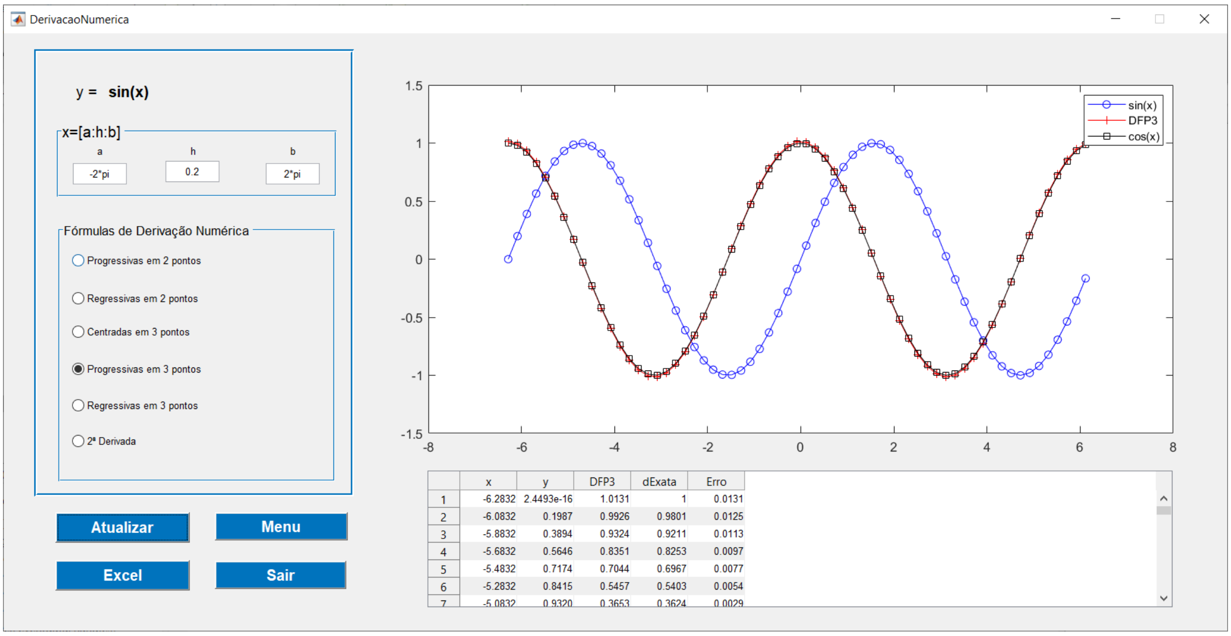


Figura 5: Função representada pelo Método das Diferenças Progressivas em 3 pontos

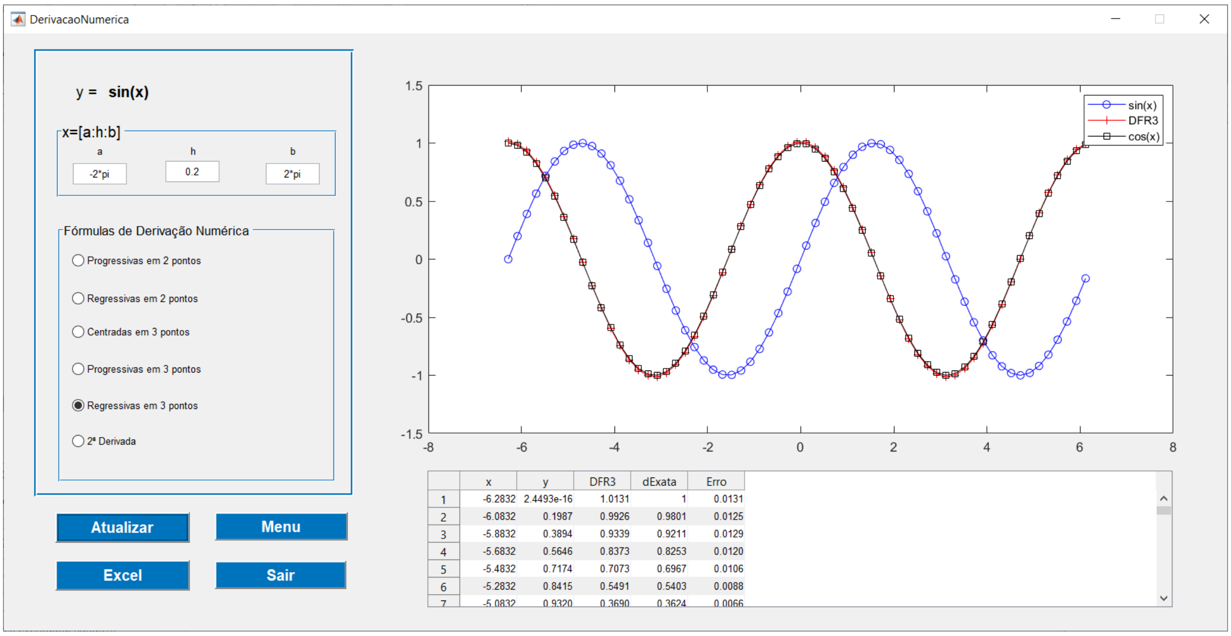


Figura 6: Função representada pelo Método das Diferenças Regressivas em 3 pontos

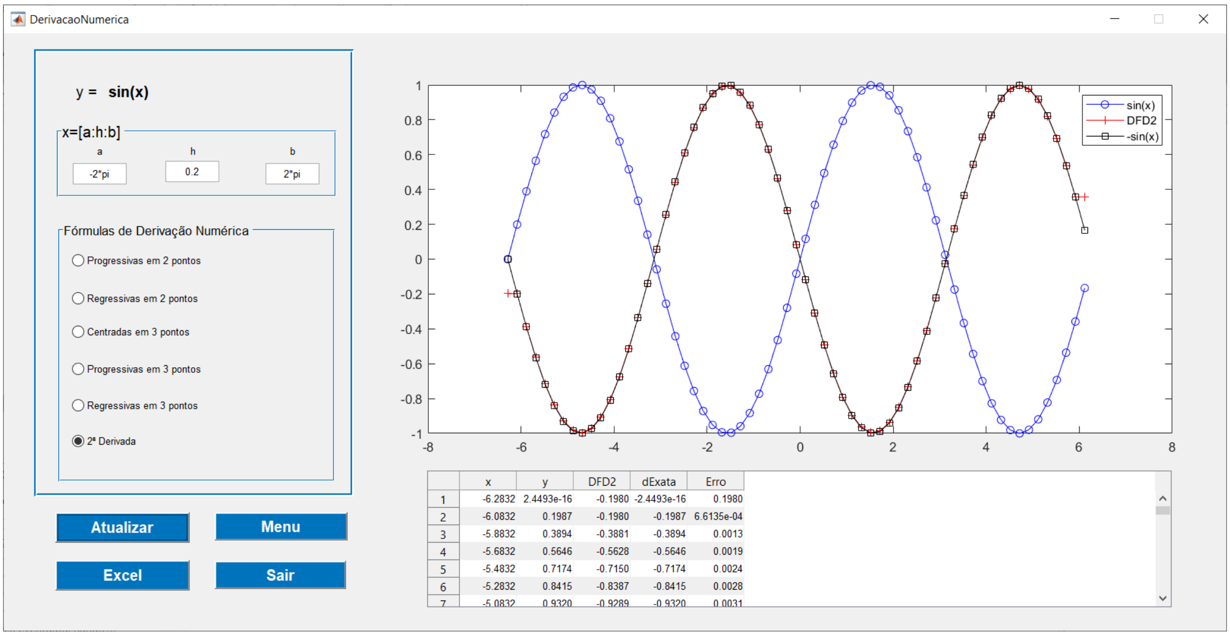


Figura 7: Função representada pelo Método da Segunda Derivada

	A	B	C	D	E	F
1	Derivação Numérica					
2	Fórmula das Diferenças Progressivas em 2 pontos					
3						
4	f(x)=	sin(x)				
5	a=	-6.2832	b=	6.2832	h=	0.2
6						
7	**x**	**y**	**dydx**	**dydxExata**	**Erro**	
8	-6,28319	2,45E-16	0,993347	1	0,006653346	
9	-6,08319	0,198669	0,953745	0,98006658	0,02632152	
10	-5,88319	0,389418	0,876121	0,92106099	0,044940339	
11	-5,68319	0,564642	0,763568	0,82533561	0,061767527	
12	-5,48319	0,717356	0,620574	0,69670671	0,07613224	
13	-5,28319	0,841471	0,452841	0,54030231	0,0874618	
14	-5,08319	0,932039	0,267053	0,36235775	0,095304534	
15	-4,88319	0,98545	0,070619	0,16996714	0,099347778	
16	-4,68319	0,999574	-0,12863	-0,0291995	0,099430339	
17	-4,48319	0,973848	-0,322751	-0,2272021	0,095548926	
18	-4,28319	0,909297	-0,504005	-0,4161468	0,087858278	
19	-4,08319	0,808496	-0,665166	-0,5885011	0,076664999	
20	-3,88319	0,675463	-0,799809	-0,7373937	0,062415328	

Figura 8:Dados de Derivação Numérica em Diferenças Progressivas em 2 pontos

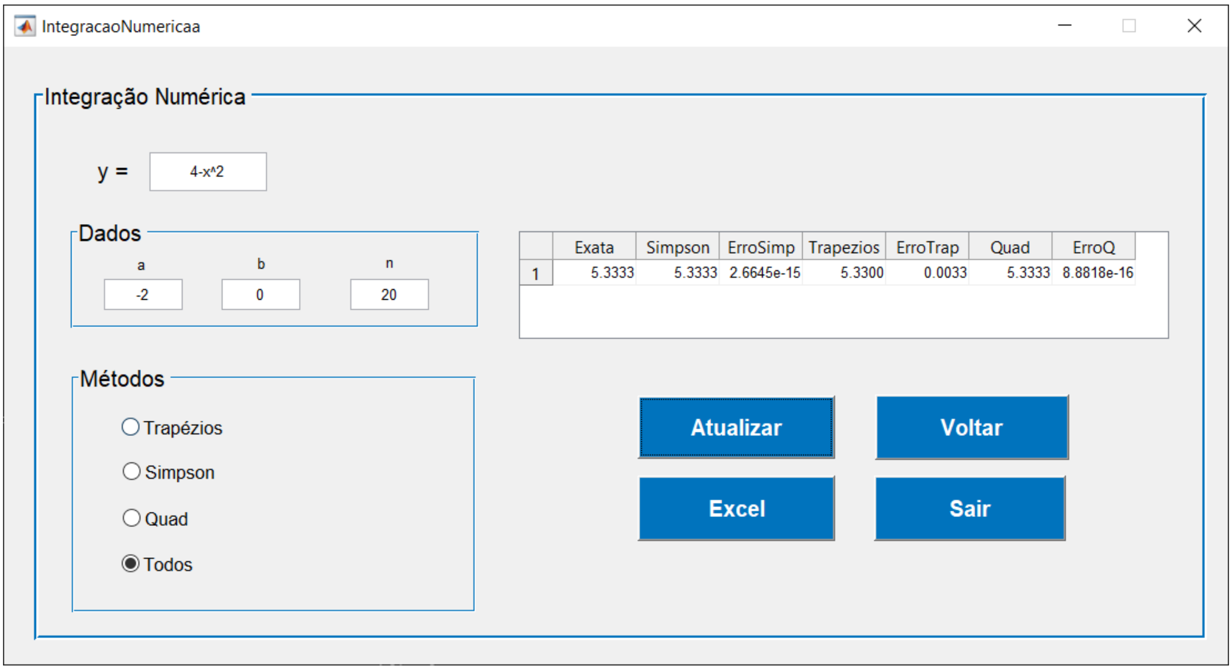


Figura 9: GUI - Integração Numérica

7. Conclusão

Com a realização desta atividade, ficámos a conhecer melhor os métodos numéricos para derivação e integração e também ganhámos mais experiência a trabalhar com o MATLAB.

Aprendemos também a funcionar com GUI's, sendo esta uma componente bastante prática que nos permite uma melhor perceção do conteúdo lecionado em aula.

Cumprindo todos os objetivos propostos pelo professor Arménio Correia, podemos concluir que todos os métodos nos dão uma boa aproximação dos valores pretendidos, de uma maneira mais simples e rápida.