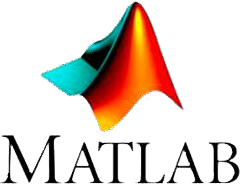


Análise Matemática II | Engenharia Informática Trabalho Prático Nº3

Máquina para Derivação e Integração



Diogo Silva a2020138438

Hugo Ferreira a2020128305

Rúben Mendes a2020138473

Índice

1. [Introdução 4](#_bookmark0)
2. [Métodos Numéricos para Derivação 5](#_bookmark1)
   1. [Formulas 5](#_bookmark2)
   2. [Algoritmo/Função 5](#_bookmark3)
3. [Derivação Simbólica no Matlab 12](#_bookmark4)
   1. [Diff 12](#_bookmark5)
4. [Métodos Numéricos para integração 13](#_bookmark6)
   1. [Formulas 13](#_bookmark7)
   2. [Algoritmo/Função 13](#_bookmark8)
5. [Verificar se a função é harmonica 16](#_bookmark9)
6. [Exemplos de aplicação e teste dos métodos 17](#_bookmark11)
7. [Conclusão 22](#_bookmark13)

Índice de Imagens

Figura 1: GUI - Derivadas 17

Figura 2: Função representada pelo Método das Diferenças Progressivas em 2 pontos 17

Figura 3: Função representada pelo Método das Diferenças Regressivas em 2 pontos 18

Figura 4: Função representada pelo Método das Diferenças Centradas

............................................................................................ 18

Figura 5: Função representada pelo Método das Diferenças Progressivas em 3 pontos 19

Figura 6: Função representada pelo Método das Diferenças Regressivas em 3 pontos 19

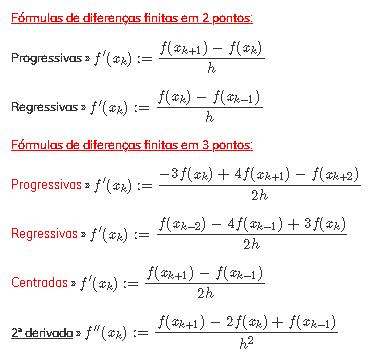
Figura 7: Função representada pelo Método da Segunda Derivada 20

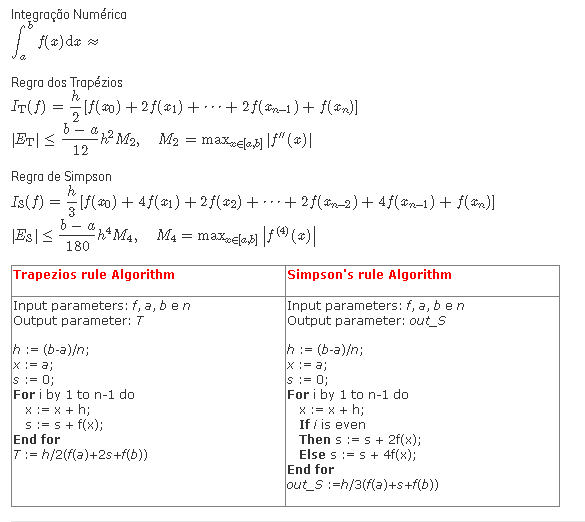
Figura 8: Verificar se a função é harmónica 20

# 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, com esta atividade desenvolvemos também a derivação numérica e simbólica em Matlab bem como a implementação da regra dos trapezios e de simpsons.

O professor Arménio Correia propôs a seguinte atividade:





# Métodos Numéricos para Derivação

* 1. Formulas

**Fórmulas de diferenças finitas em 2 pontos:**

Progressivas: Regressivas:



**Fórmulas de diferenças finitas em 3 pontos:**

Progressivas: Regressivas: Centradas:



2ª Derivada:



* 1. 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 = 5 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]=DFProgressivas\_2(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)=(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 = 5 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]=DFRegressivas\_2(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

*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 = 5 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]=DFProgressivas\_3(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);

*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 = 5 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]=DFRegressivas\_3(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;

*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 = 5 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]=DFCentradas\_3(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;

*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 = 5 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==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);

1. Derivação Simbólica no Matlab

3.1. Diff

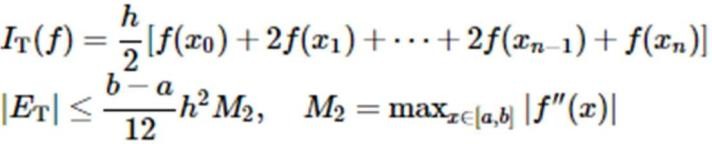
Funciona também com a função diff mas neste caso calcula as diferenças entre elementos adjacentes de X ao longo de um array cujo tamanho não é igual a 1. Pode ser representada das seguintes maneiras:

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

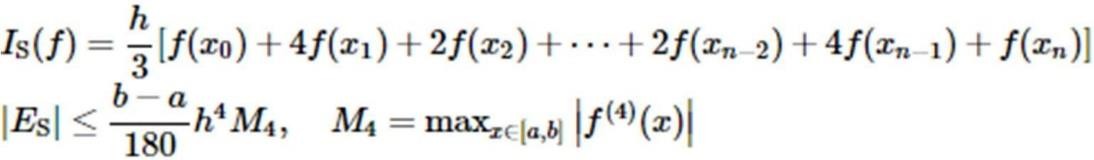
# Métodos Numéricos para integração

* 1. Formulas

**Regra dos Trapézios:**



**Regra de Simpson:**



* 1. 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 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;

# Verificar se a função é harmónica

*Função:*

function verifica = harmonica(f)

syms x y;

if(diff(f,x,2)+diff(f,y,2)==0)

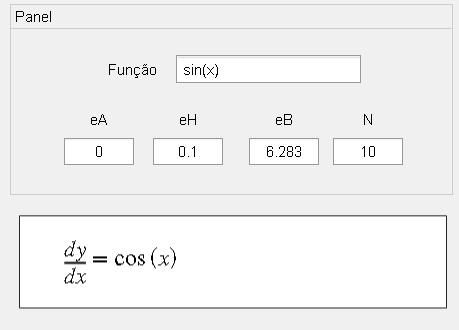
verifica = 1;

else

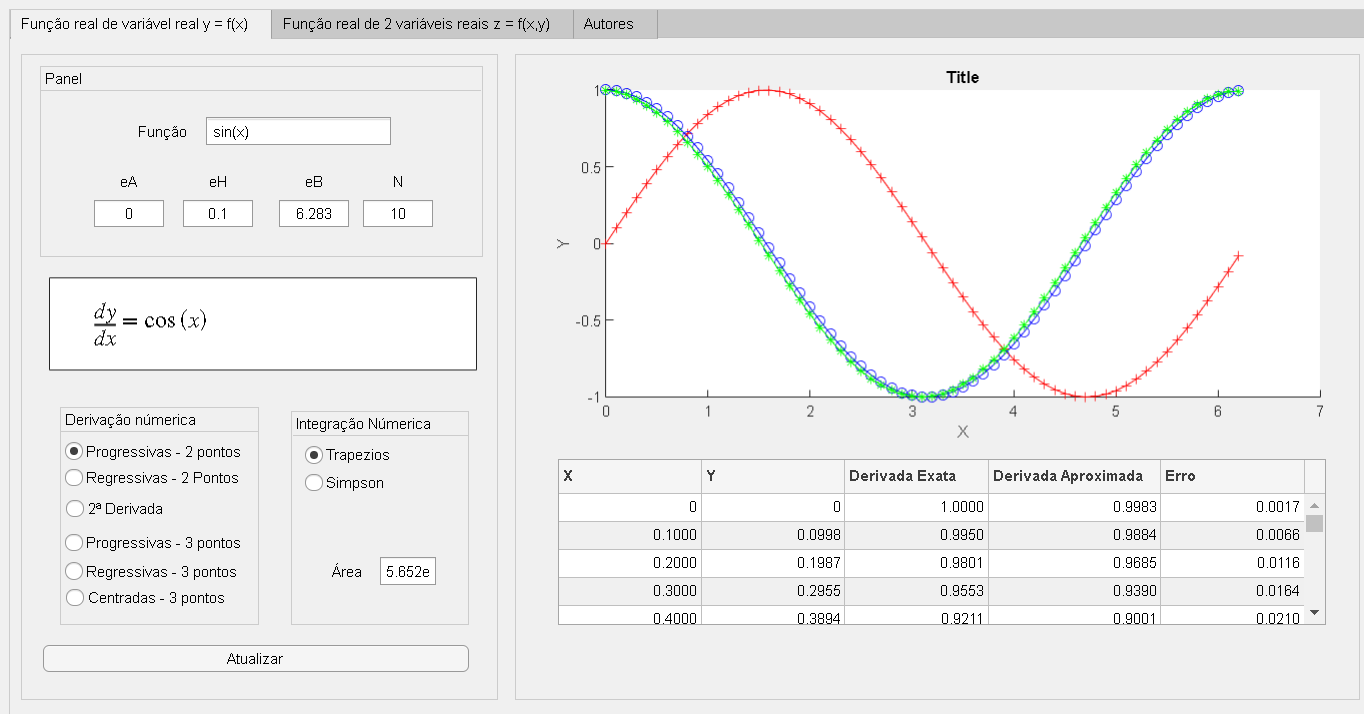
verifica = 0;

end

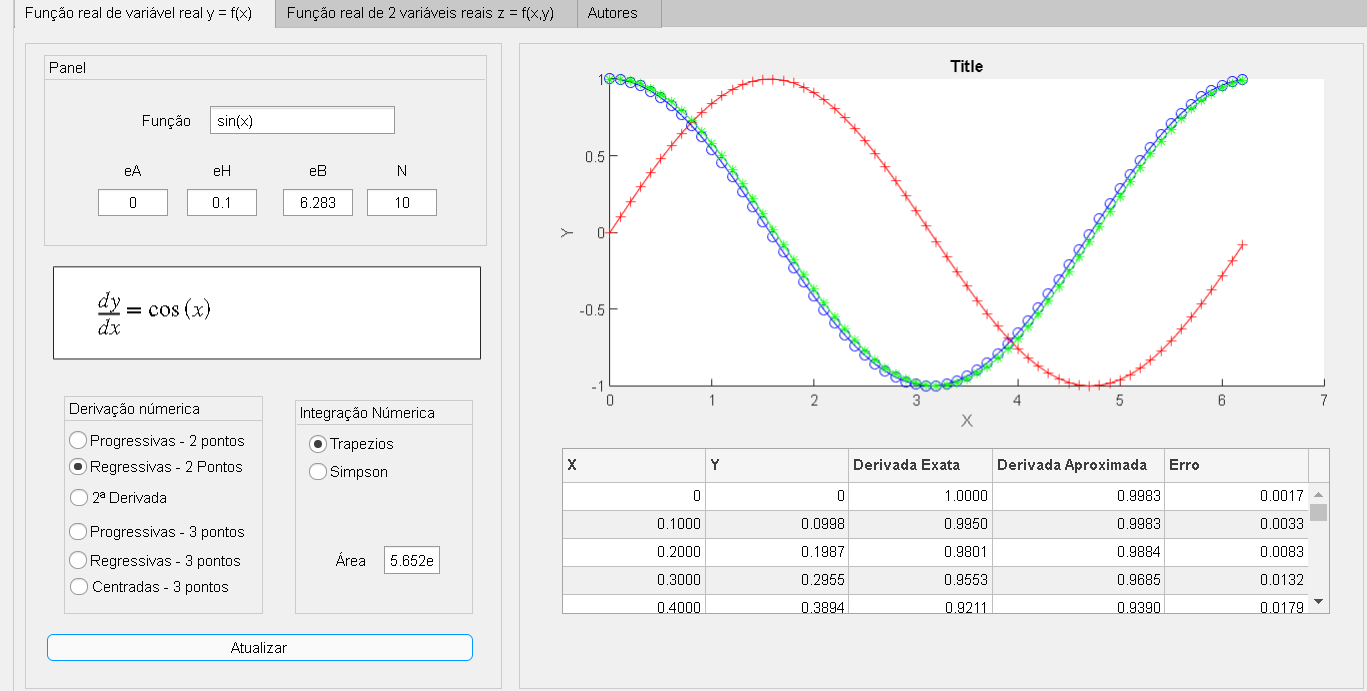
# Exemplos de aplicação e teste dos métodos



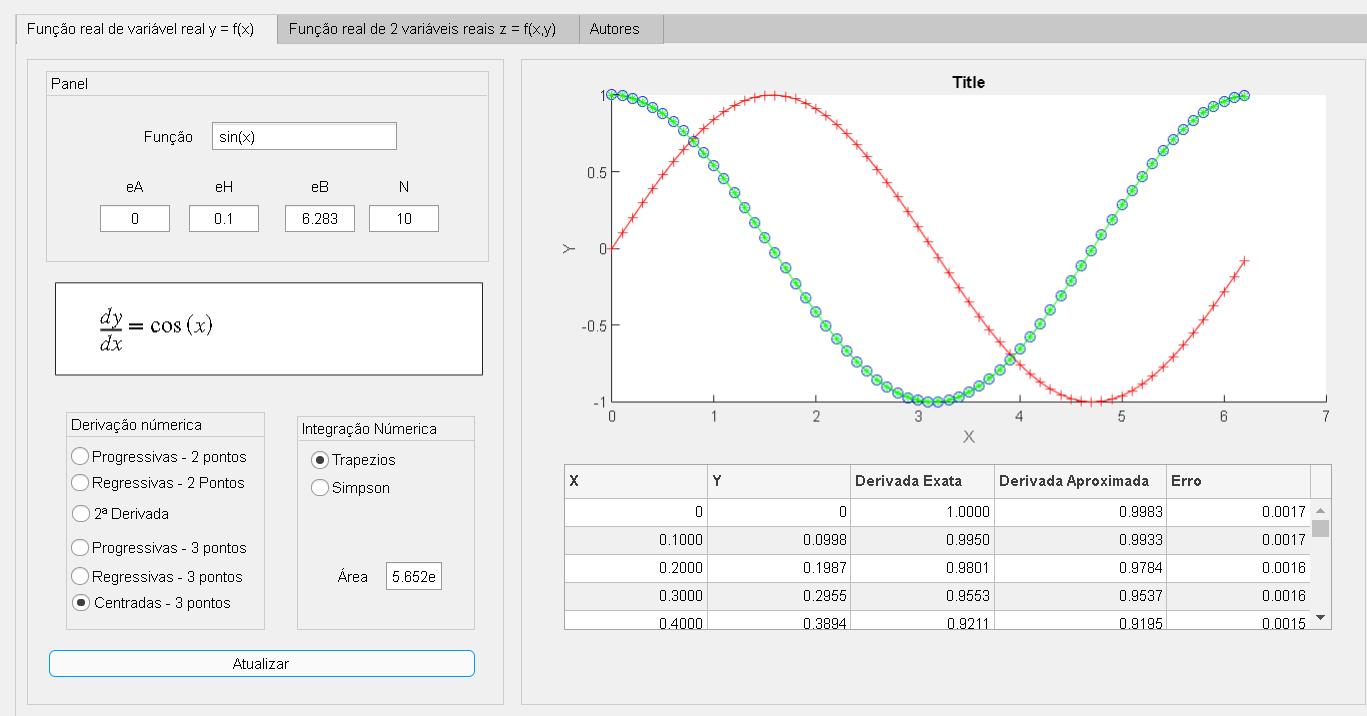
*Figura 1: GUI - Derivadas*



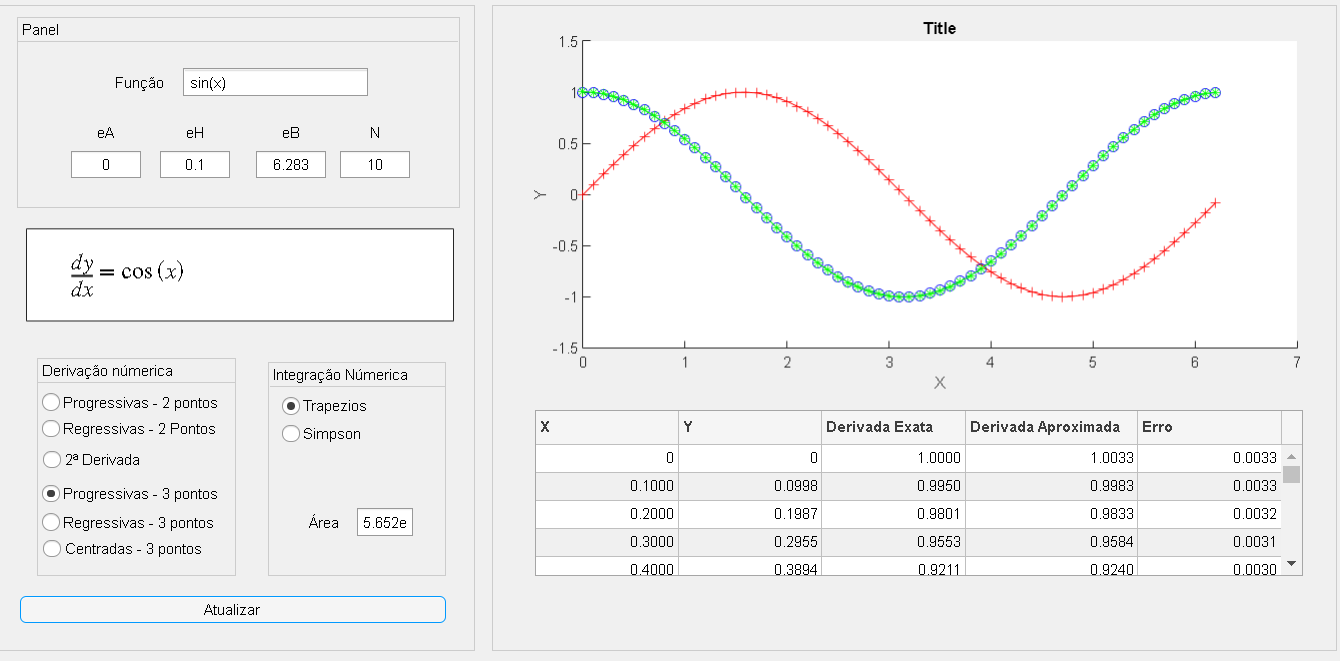
*Figura 2: Função representada pelo Método das Diferenças Progressivas em 2 pontos*



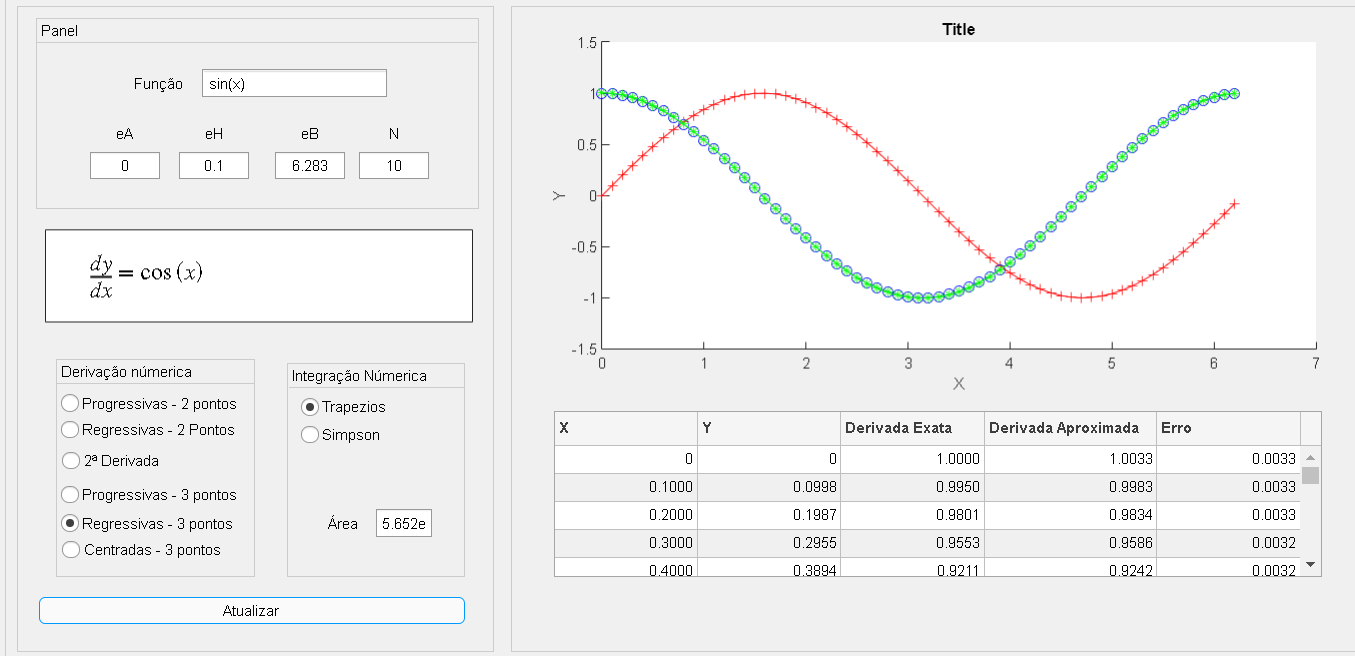
*Figura 3: Função representada pelo Método das Diferenças Regressivas em 2 pontos*



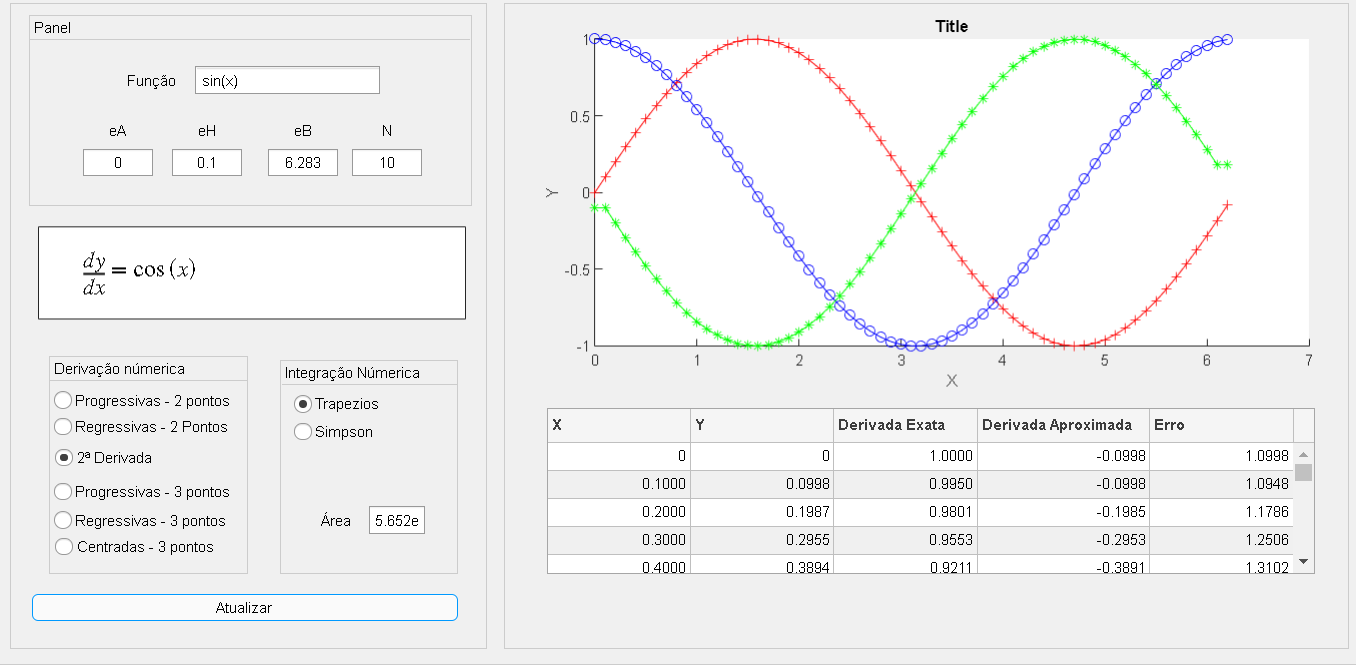
*Figura 4: Função representada pelo Método das Diferenças Centradas*



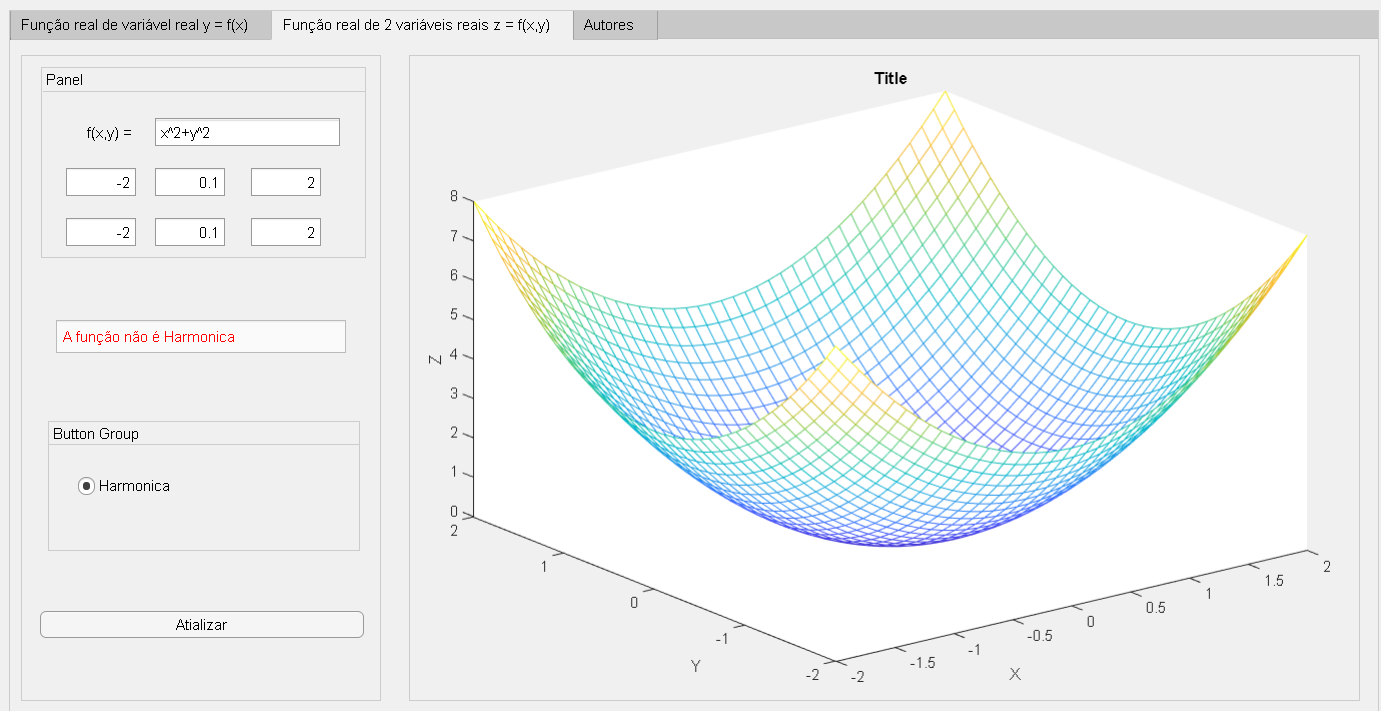
*Figura 5: Função representada pelo Método das Diferenças Progressivas em 3 pontos*



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



*Figura 7: Função representada pelo Método da Segunda Derivada*



*Figura 8:Verificar se a função é harmónica e construir o gráfico*

# 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 APPS, 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.