Instituto Superior
de Engenharia
Politécnico de Coimbra

MáquinaCDI

Autores:

Pedro Miguel Martins Jácome - 2022137038

Ricardo Rodrigues Duarte - 2022137878

Guilherme de Pinho Domingos - 2022136668

INSTITUTO POLITÉCNICO DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, 05/2024

Índice

1. Introdução 4
2. Implementação de funções de Diferenças Finitas 5
  2.1. Fórmulas de diferenças finitas em 2 pontos 6
  2.2. Fórmulas de diferenças finitas em 3 pontos 7
3. Regras dos trapézios e Simpson 9
  3.1. Regra dos Trapézios 9
  3.2. Regra de Simpson 10
  3.3. Código MatLab 11
4. Avaliar se uma função real de duas variáveis reais é harmónica 12
5. MáquinaCDI 13
6. Conclusão 14
7. Bibliografia 15
8. Autoavaliação e heteroavaliação 16

Figura 1- Fórmula das diferenças regressivas em 2 pontos 6
Figura 2 - Fórmula das diferenças progressivas em 2 pontos 6
Figura 3 - Fórmula das diferenças progressivas em 3 pontos 7
Figura 4 - Fórmula das diferenças regressivas em 3 pontos 7
Figura 5 - Fórmula das diferenças centrada em 3 pontos 8
Figura 6 - Algoritmo regra dos Trapézios 9
Figura 7 - Algoritmo regra de Simpson 10
Figura 8 - Regra dos Trapézios Matlab 11
Figura 9 - Regra de Simpson MatLab 11
Figura 10 - Avaliar se é harmónica MatLab 12
Figura 11- Ilustração MaquinaCDI 13

1. Introdução

No ambito da disciplina de Análise de Matemática II foi nos solicitado a realização de um trabalho com o principal objetivo o cálculo diferencial e integram de funções reais de uma variável real e de duas variáveis reais.

Este relatório está organizado em quatro partes principais, cada uma abordando um aspecto fundamental da análise numérica e cálculo. Os pontos que abordamos são Implementação de Diferenças Finitas onde em *MATLAB* implementaremos funções que calculam as derivadas utilizando métodos de diferenças finitas, implementação das Regras dos Trapézios e Simpson, também teremos avaliação de Funções Harmônicas e por fim baseando na máquina apresentada nas aulas práticas, que serve como uma ferramenta para o cálculo diferencial e integral, esta parte foca em expandir e completar essa máquina. O objetivo é aprimorar as suas capacidades para realizar cálculos diferenciais e integrais de funções reais de uma variável real e de duas variáveis reais. A implementação dessas funcionalidades permitirá uma análise mais abrangente e eficiente das funções.

2. Implementação de funções de Diferenças Finitas

Para compreendermos a derivação numérica, é fundamental entender o conceito de derivada. A derivada de uma função f em um ponto pode ser visualizada graficamente com a inclinação da reta tangente à curva da função naquele ponto. É aqui que entra a derivação numérica. A derivação numérica é uma técnica utilizada para estimar a derivada de uma função quando temos apenas um conjunto de pontos discretos pertencentes a essa função. Isso é particularmente útil para funções que não são deriváveis em todo o seu domínio ou cuja derivação é complexa. À medida que o intervalo h entre os pontos diminui, o valor da derivada numérica se aproxima do valor real da derivada. Contudo, existe sempre um erro de arredondamento a esse método, mesmo que h seja muito pequeno. Uma forma eficaz de minimizar esse erro é utilizar múltiplos pontos para a estimativa.

Assim sendo o objetivo é, portanto, partir de um conjunto de pontos que definem um intervalo $[a, b]$ e determinar uma função f que os represente, ou seja, realizar a interpolação desse conjunto de pontos. Com a função f interpolada, podemos então calcular sua derivada e aplicá-la a qualquer ponto dentro do intervalo .

2.1. Fórmulas de diferenças finitas em 2 pontos

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

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

```
% Fórmula das diferenças regressivas em 2 pontos
% f'(xk) = f(xk)-f(xk-1) / h

% Onde:
%   f'(xk) - aproximação do valor da derivada no ponto de abcissa xk;
%   f(xk) - valor da função no ponto de abcissa atual;
%   f(xk-1) - valor da função na abcissa anterior;
%   h - valor de cada sub-intervalo.

% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

function [x, y, dydx] = NDerivacao2PR(f, a, b, h, y)
x = a: h: b;
n = length(x);
if nargin == 5
    y = f(x);
end
dydx = zeros(1, n);
for k = n:-1:2
    dydx(k) = (y(k)-y(k-1)) / h;
end
dydx(1) = (y(2)-y(1)) / h;
end
```

Figura 1- Fórmula das diferenças regressivas em 2 pontos

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

Figura 2 - Fórmula das diferenças progressivas em 2 pontos

2.2. Fórmulas de diferenças finitas em 3 pontos

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

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

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

```
% Fórmula das diferenças progressivas em 3 pontos
% f'(xk) = -3f(xk)+4f(xk+1)-f(xk+2) / 2*h

% Onde:
%   f'(xk) - aproximação do valor da derivada no ponto de abcissa xk;
%   f(xk) - valor da função no ponto de abcissa atual;
%   f(xk+1) - valor da função na próxima abcissa;
%   f(xk+2) - valor da função 2 abcissas à frente;
%   h - valor de cada sub-intervalo.

% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

function [x, y, dydx] = NDerivacao3PP(f,a,b,h,y)
x = a: h: b;
n = length(x);
if nargin == 5
    y = f(x);
end
dydx = zeros(1, n);
for k = 1: (n-2)
    dydx(k) = (-3*y(k)+4*y(k+1)-y(k+2)) / (2*h);
end
dydx(n-1) = (-3*y(n-2)+4*y(n-1)-y(n)) / (2 * h);
dydx(n) = (-3*y(n-2)+4*y(n-1)-y(n)) / (2 * h);
end
```

Figura 3 - Fórmula das diferenças progressivas em 3 pontos

```
% Fórmula das diferenças regressivas em 3 pontos
% f'(xk) = -3f(xk)+4f(xk+1)-f(xk+2) / 2*h

% Onde:
%   f'(xk) - aproximação do valor da derivada no ponto de abcissa xk;
%   f(xk) - valor da função no ponto de abcissa atual;
%   f(xk-2) - valor da função 2 abcissas atrás;
%   f(xk-1) - valor da função na abcissa anterior;
%   h - valor de cada sub-intervalo.

% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

function [x, y, dydx] = NDerivacao3PR(f, a, b, h, y)
x = a: h: b;
n = length(x);
if nargin == 5
    y = f(x);
end
dydx = zeros(1, n);
for k = n:-1:3
    dydx(k) = (y(k-2)-4*y(k-1)+3*y(k)) / (2*h);
end
dydx(2) = (y(1)-4*y(2)+3*y(3)) / (2*h);
dydx(1) = (y(1)-4*y(2)+3*y(3)) / (2*h);
end
```

Figura 4 - Fórmula das diferenças regressivas em 3 pontos

```
% Fórmula das diferenças centradas em 3 pontos
% f'(xk) = f(xk+1)-f(xk-1)/ 2*h

% Onde:
%   f'(xk) - aproximação do valor da derivada no ponto de abcissa xk;
%   f(xk+1) - valor da função na próxima abcissa;
%   f(xk-1) - valor da função na abcissa anterior;
%   h - valor de cada subintervalo.

% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

function [x, y, dydx] = NDerivacao3PC(f, a, b, h, y)
x = a: h: b;
n = length(x);
if nargin == 5
    y = f(x);
end
dydx = zeros(1, n);
for k = 2:(n-1)
    dydx(k) = (y(k+1)-y(k-1)) / (2*h);
end
dydx(1) = (y(3)-y(1)) / (2*h);
dydx(n) = (y(n)-y(n-2)) / (2*h);
end
```

Figura 5 - Fórmula das diferenças centrada em 3 pontos

3. Regras dos trapézios e Simpson

A integração numérica é uma técnica essencial na análise numérica para calcular integrais definidas quando a função não tem uma derivada simples ou quando os dados são discretos. Dois dos métodos mais comuns são a Regra dos Trapézios e a Regra de Simpson. Vamos explorar ambos os métodos e sua implementação em MATLAB.

3.1. Regra dos Trapézios

A Regra dos Trapézios calcula a integral de uma função num certo intervalo com base à interpolação polinomial, obtendo assim uma aproximação razoável através dos polinómios. Quantos mais intervalos em [a,b], maior será a sua precisão.

Integração numérica → $\int_a^b f(x)dx \approx$

$$I_T(f) = \frac{h}{2}[f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)]$$

$$|E_T| \leq \frac{b-a}{12}h^2 M_2, \qquad M_2 = max_{x\,\in[a,b]}|f''(x)|$$

```
Trapezios rule Algorithm
Input parameters: f, a, b e n
Output parameter: T

h := (b-a)/n;
x := a;
s := 0;
For i by 1 to n-1 do
   x := x + h;
   s := s + f(x);
End for
T := h/2(f(a)+2s+f(b))
```

Figura 6 - Algoritmo regra dos Trapézios

3.2. Regra de Simpson

Como a Regra dos Trapézios, calcula uma aproximação da integral de uma função f, mas desta vez, calcula-o através de um polinómio interpolador de 2ª grau.

Integração numérica → $\int_a^b f(x)dx \approx$

$$I_S(f) = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 4f(x_{n-1}) + f(x_n)]$$

$$|E_S| \leq \frac{b-a}{180} h^4 M_4, \qquad M_4 = max_{x \in [a,b]} |f^4(x)|$$

Simpson's rule Algorithm

```
Input parameters: f, a, b e n
Output parameter: out_S

h := (b-a)/n;
x := a;
s := 0;
For i by 1 to n-1 do
   x := x + h;
   If i is even
   Then s := s + 2f(x);
   Else s := s + 4f(x);
End for
out_S :=h/3(f(a)+s+f(b))
```

Figura 7 - Algoritmo regra de Simpson

3.3. Código MatLab

```
%Regra dos Trapézios
% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

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

Figura 8 - Regra dos Trapézios Matlab

```
%Regra de Simpson
% 24/05/2024  Ricardo Duarte
% 24/05/2024  Pedro Jácome
% 24/05/2024  Guilherme Domingos

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
out_S = h/3*(f(a)+s+f(b));
```

Figura 9 - Regra de Simpson MatLab

4. Avaliar se uma função real de duas variáveis reais é harmónica

Para verificar se uma função é harmônica, calculamos suas segundas derivadas parciais em relação às variáveis independentes e somamos essas derivadas. Se a soma for zero em todos os pontos do domínio, a função é harmônica. Este método chama-se equação de Laplace.

$$\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \cdots + \frac{\partial^2 f}{\partial x_n^2} = 0$$

```
function YN = isHarmonica(~,f)
    syms z(x,y)
    if (diff(f,x,2)+diff(diff(f,y),y)==0)
        YN = 1;
    else
        YN = 0;
    end
end
```

Figura 10 - Avaliar se é harmónica MatLab

5. MáquinaCDI

Com base na máquina representada na figura seguinte, que foi objeto de trabalho nas aulas práticas, ampliamos as suas potencialidades para o cálculo diferencial e integral de funções reais de uma variável real e de duas variáveis reais e implementamos as regras dos trapézios e Simpson. A figura abaixo apresenta a representação visual da MaquinaCDI.

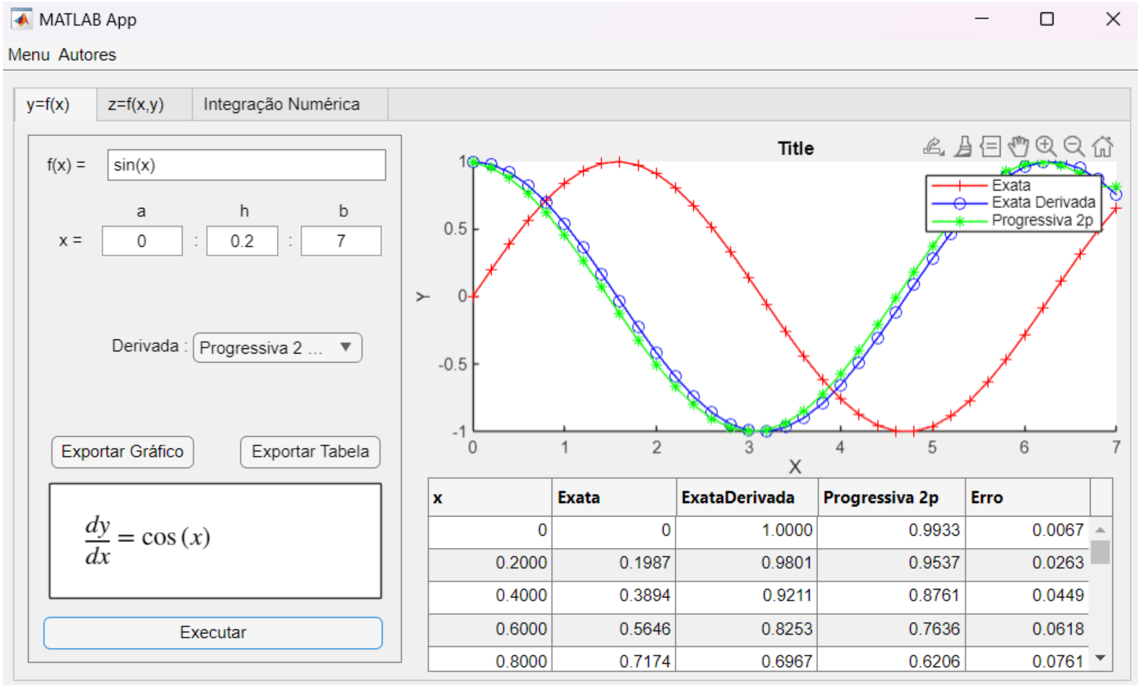

Figura 11- Ilustração MaquinaCDI

6. Conclusão

Em resumo, os métodos numéricos de derivação e integração são ferramentas valiosas para calcular derivadas e integrais de funções de forma aproximada. Embora esses métodos possam fornecer resultados precisos com um número adequado de subintervalos, é importante lembrar que são apenas aproximações e podem conter erros. Portanto, antes de utilizar essas técnicas, é fundamental avaliar sua precisão. No caso específico dos Métodos de Derivação usando as Fórmulas das Diferenças Finitas, observamos que o aumento no número de pontos de diferença finita pode aumentar a precisão da estimativa da derivada.

Concluimos que esta atividade nos ajudou a compreender que os métodos numéricos têm várias aplicações, quer para integrais e para derivadas o que ajuda na resolução de problemas de resolução mais complexa.

7. Bibliografia

Moodle Disciplina - https://moodle.isec.pt/moodle/course/view.php?id=20386

8. Autoavaliação e heteroavaliação

Como parte desta atividade, realizamos uma análise crítica do nosso desempenho e aprendizado, e com isto consideramos que o nosso trabalho deverá ser autoavaliado em 4.