Análise Matemática II

Atividade 03 – Máquina CDI

Ricardo Almeida de Aguiar Tavares - 2021144652 – LEI

João Choupina Ferreira da Mota - 2020151878 – LEI

Coimbra, 26 de junho de 2022

Índice

1. Introdução

Trabalho realizado no âmbito da unidade curricular de Análise Matemática II que consiste na implementação em MatLab de métodos de derivação e integração numérica.

O principal objetivo será implementar funções através do desenvolvimento de uma GUI, nomeadamente:

- Diferenças finitas em 2 e 3 pontos (Progressivas, Regressivas, Centradas)
- Segunda Derivada
- Regra dos Trapézios e Simpson

2. Métodos Numéricos para resolução de Sistemas de ED

2.1. Fórmulas de Diferenças Finitas em dois pontos

2.1.1. Progressivas

Fórmula:

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

Algoritmo:

```
function [x, y, dydx] = NDerivacaoP2(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;
end
```

2.1.2. Regressivas

Fórmula:

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

Algoritmo:

```
function [x, y, dydx] = NDerivacaoR2(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 = n:-1:2
    dydx(i) = (y(i) - y(i - 1)) / h;
end

dydx(1) = (y(2) - y(1)) / h;

end
```

2.2. Fórmulas de Diferenças Finitas em três pontos

2.2.1. Progressivas

Fórmula:

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

Algoritmo:

```
function [x, y, dydx] = NDerivacaoP3(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) = (-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
```

2.2.2. Regressivas

Fórmula:

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

Função:

```
function [x, y, dydx] = NDerivacaoR3(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 = n: -1: 3
    dydx(i) = (y(i - 2) -4 * y(i - 1) + 3 * y(i)) / (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
```

2.2.3. Centradas

Fórmula:

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

Função:

```
function [x, y, dydx] = NDerivacaoC3(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 = 2: (n - 1)
    dydx(i) = (y(i + 1) - y(i - 1)) / (2 * h);
end

dydx(1) = (y(3) - y(1)) / (2 * h);
dydx(n) = (y(n) - y(n - 2)) / (2 * h);
end
```

2.3. 2ª Derivada

Fórmula:

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

Função:

```
function [x, y, dydx] = NDerivacaoD2(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 = 2: (n - 1)
    dydx(i) = (y(i + 1) - 2 * y(i) + y(i - 1)) / h^2;
end

dydx(1) = (y(3) - 2 * y(2) + y(1)) / h^2;
dydx(n) = (y(n) - 2 * y(n - 1) + y(n - 2)) / h^2;

end
```

3. Métodos numéricos para integração

3.1. Regra dos Trapézios

Fórmula:

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

Função:

```
function T = Trapezios(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) );
```

3.2. Regra de Simpson

Fórmula:

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

Função:

```
function out_S = Simpson(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) );
```

4. Conclusão

Com este trabalho podemos concluir que os métodos numéricos possuem diversas aplicações para derivadas / integrais que facilitam a resolução de problemas com uma maior abrangência de cada uma das áreas da ciência, onde normalmente seria necessário calcular integrais analiticamente (que por vezes se verificava impossível).

Como já verificado em outros trabalhos, quanto maior for ***n*** (número de sub-intervalos), menor é o erro dos respetivos métodos e fórmulas. Contudo, a introdução de ***h*** (dimensão de cada sub-intervalo) verifica-se o contrário.

5.Bibliografia

.Fórum .: Matlab

Matlab .: Atividade03Trabalho - MáquinaCDI » Dúvidas e Questões

(https://moodle.isec.pt/moodle/mod/forum/discuss.php?d=33833)