

Atividade 02 (2018/2019) » MNEDOPVI

Unidade Curricular: Análise Matemática II

Docentes: Arménio Correia & Nuno Baeta

Grupo de trabalho: Alexandre Costa Reis - 21280926

Celso André Ferreira Jordão - 21130067

Fabio Capobianchi de Souza - 21280924

Atividade 02 (2018/2019) » MNEDOPVI	1
Introdução	3
O que é o MATLAB?	3
Interpretação do problema e definição de PVI	3
Métodos Numéricos para a resolução de um PVI	4
Método de Euler	4
Fórmulas	4
Algoritmo & Função	4
Método de Euler Melhorado	5
Fórmulas	5
Algoritmo & Função	5
Runge Kutta de Ordem 2	5
Fórmulas	5
Algoritmo & Função	6
Runge Kutta de Ordem 3	6
Fórmulas	6
Algoritmo & Função	6
Runge Kutta de Ordem 4	7
Fórmulas	7
Algoritmo & Função	7
ODE45	8
Fórmulas	8
Algoritmo & Função	8
Exemplos de aplicação e teste dos métodos	9
Exercício 4 do um teste A de 2015/2016	9
Alínea (c)	9
Alínea (d)	9
Exemplos de output - GUI com gráfico e tabela	10
Extração de CSV	12
Conclusão	13

Introdução

Esta actividade prática, foi sugerida, pelos professores da UC de AM2, e consiste na realização de uma interface/aplicação que resolva problemas de valor inicial (PVI), recorrendo a métodos distintos (Euler, Runge-Kutta de 2ª e 4ª Ordem e, numa fase mais avançada, a implementação de outros 2 métodos, Euler Modificado e ODE45).

Esta actividade tem como objectivo principal, promover a familiarização com estes métodos numéricos assim como o aprofundamento de conhecimentos a nível de programação, em Matlab.

O que é o MATLAB?

MATLAB (MATrix LABoratory) é um software interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos num ambiente fácil de usar, onde problemas e soluções são expressos somente como eles são escritos matematicamente, ao contrário da programação tradicional.

O MATLAB é um sistema interativo, cujo elemento básico de informação é uma matriz que não requer dimensões padrão.

Este software permite programar numa linguagem bastante semelhante a linguagem C, permitindo ainda a elaboração de janelas de interface, através da opção **guide**.

Interpretação do problema e definição de PVI

Uma equação diferencial que satisfaça certas condições adicionais é denominada **Problema de Valor Inicial (PVI)**.

Exemplo:

$$\begin{aligned} y' + 2y &= \arctan(x) \\ y(0) &= \pi \end{aligned}$$

Se forem conhecidas as condições adicionais, podemos obter soluções particulares para a equação diferencial caso contrário, são conhecidas condições adicionais poderemos obter a solução geral.

Métodos Numéricos para a resolução de um PVI

Neste capítulo, vamos proceder à explicação da algoritmia e desenvolvimento das funções em causa, isto é:

- Euler
- Euler Melhorado
- Runge Kutta de ordem 2
- Runge Kutta de ordem 3 (Extra)
- Runge Kutta de ordem 4
- ODE 45

Método de Euler

Este é o mais simples e mais antigo dos métodos numéricos utilizados resolução de um PVI. Foi criado no séc. XVIII pelo matemático Leonhard Euler (1707 – 1783).

Fórmulas

$$y_{m+1} = y_m + hf(t_m, y_m)$$

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

y(1)=y0

t(1)=a

h=(b-a)/n

Para i de 1 até n fazer

 y(i+1)=y(i)+h*f(t(i),y(i))

 t(i+1)=t(i)+h

Fim

```
function y = N_Euler(f, a, b, n, y0)
```

```
h = (b-a)/n;
```

```
t(1) = a;
```

```
y(1) = y0;
```

```
for i = 1:n
```

```
    y(i+1) = y(i) + h*f(t(i), y(i));
```

```
    t(i+1) = t(i) + h;
```

```
end
```

Método de Euler Melhorado

Fórmulas

```

$$y(i+1) = y(i) + h * f(t(i), y(i));$$

$$y(i+1) = y(i) + h/2 * (f(t(i), y(i)) + f(t(i+1), y(i+1)));$$

```

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

y=zeros(1,n+1);

y(1)=y0;

t=a:h:b

h=(b-a)/n

Para i de 1 até n fazer

 y(i+1)=y(i)+h*f(t(i),y(i));

 y(i+1)=y(i)+h/2*(f(t(i), y(i))+f(t(i+1), y(i+1))));

Fim

```
function y=N_Euler_v2(f,a,b,n,y0)
h=(b-a)/n;
t=a:h:b;
y=zeros(1,n+1);
y(1)=y0;
for i=1:n
    y(i+1)=y(i)+h*f(t(i),y(i));
    y(i+1)=y(i)+h/2*(f(t(i), y(i))+f(t(i+1), y(i+1))));
end
```

Runge Kutta de Ordem 2

Fórmulas

```

$$k1 = h * f(t(i), y(i));$$

$$k2 = h * f(t(i+1), y(i) + k1);$$

$$y(i+1) = y(i) + (k1+k2)/2;$$

```

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

h = (b-a)/n;
t(1) = a;
y(1) = y0;

Para i de 1 até n fazer

t(i+1) = t(i) + h;
k1 = h*f(t(i), y(i));
k2 = h*f(t(i+1), y(i) + k1);
y(i+1) = y(i) + (k1+k2)/2;

Fim

function y = N_RK2(f, a, b, n, y0)

h = (b-a)/n;
t(1) = a;
y(1) = y0;

for i = 1:n

t(i+1) = t(i) + h;
k1 = h*f(t(i), y(i));
k2 = h*f(t(i+1), y(i) + k1);
y(i+1) = y(i) + (k1+k2)/2;

end

Runge Kutta de Ordem 3

Fórmulas

```
k1=h*f(t(i),y(i));
k2=h*f(t(i)+ h/2, y(i) + k1/2);
k3=h*f(t(i)+h,y(i)-k1+2*k2);
y(i+1) = y(i)+(1/6)*(k1+4*k2+k3);
```

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

h = (b-a)/n;
t(1) = a;
y(1) = y0;

Para i de 1 até n fazer

t(i+1) = t(i) + h;
k1 = h*f(t(i),y(i));
k2=h*f(t(i)+ h/2, y(i) + k1/2);
k3=h*f(t(i)+h,y(i)-k1+2*k2); y(i+1) = y(i)+(1/6)*(k1+4*k2+k3);

Fim

function y = N_RK3(f, a, b, n, y0)

h = (b-a)/n;
t(1) = a;
y(1) = y0;

for i = 1:n

t(i+1) = t(i) + h;
k1 =h*f(t(i),y(i));
k2=h*f(t(i)+ h/2, y(i) + k1/2);
k3=h*f(t(i)+h,y(i)-k1+2*k2);
y(i+1) = y(i)+(1/6)*(k1+4*k2+k3);

end

Runge Kutta de Ordem 4

Fórmulas

```
 $k_1 = h * f(t(i), y(i));$   
 $k_2 = h * f(t(i) + (h/2), y(i) + (k_1/2));$   
 $k_3 = h * f(t(i) + (h/2), y(i) + (k_2/2));$   
 $k_4 = h * f(t(i+1), y(i)+k_3);$   
 $y(i+1) = y(i) + (k_1+(2*k_2)+(2*k_3)+k_4)/6;$ 
```

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

h = (b-a)/n;

t(1) = a;

y(1) = y0;

Para i de 1 até n fazer

 t(i+1) = t(i) + h;

 k1 = h*f(t(i), y(i));

 k2 = h*f(t(i) + (h/2), y(i) + (k1/2));

 k3 = h*f(t(i) + (h/2), y(i) + (k2/2));

 k4 = h*f(t(i+1), y(i)+k3);

 y(i+1) = y(i) + (k1+(2*k2)+(2*k3)+k4)/6;

Fim

```
function y = N_RK4(f, a, b, n, y0)
```

```
h = (b-a)/n;
```

```
t(1) = a;
```

```
y(1) = y0;
```

```
for i = 1:n
```

```
    t(i+1) = t(i) + h;
```

```
    k1 = h*f(t(i), y(i));
```

```
    k2 = h*f(t(i) + (h/2), y(i) + (k1/2));
```

```
    k3 = h*f(t(i) + (h/2), y(i) + (k2/2));
```

```
    k4 = h*f(t(i+1), y(i)+k3);
```

```
    y(i+1) = y(i) + (k1+(2*k2)+(2*k3)+k4)/6;
```

```
end
```

ODE45

Fórmulas

$[t,y] = \text{ode45}(f,tsp,y0);$

Algoritmo & Função

Ler entradas: f, a, b, n, y0

Saídas: y

h=(b-a)/n;
tsp = a:(b-a)/n:b;
y(1) = y0;

Para i de 1 até n fazer

 [t,y] = ode45(f,tsp,y0);

Fim

function [t,y] = ODE45 (f,a,b,n,y0)

h=(b-a)/n;
tsp = a:(b-a)/n:b;
y(1) = y0;
[t,y] = ode45(f,tsp,y0);

Exemplos de aplicação e teste dos métodos

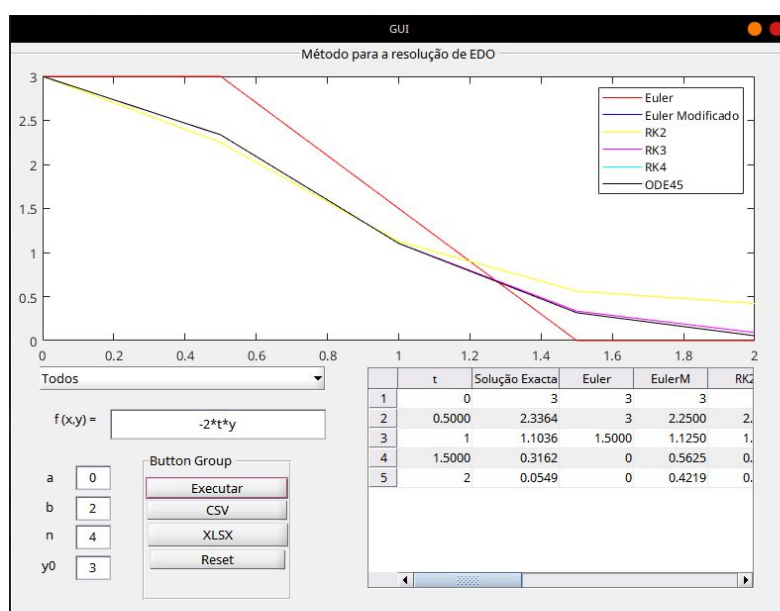
Exercício 4 do um teste A de 2015/2016

Alínea (c)

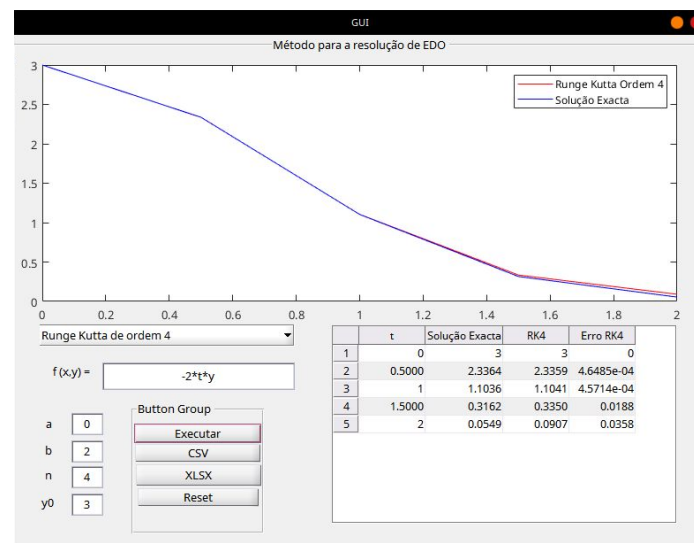
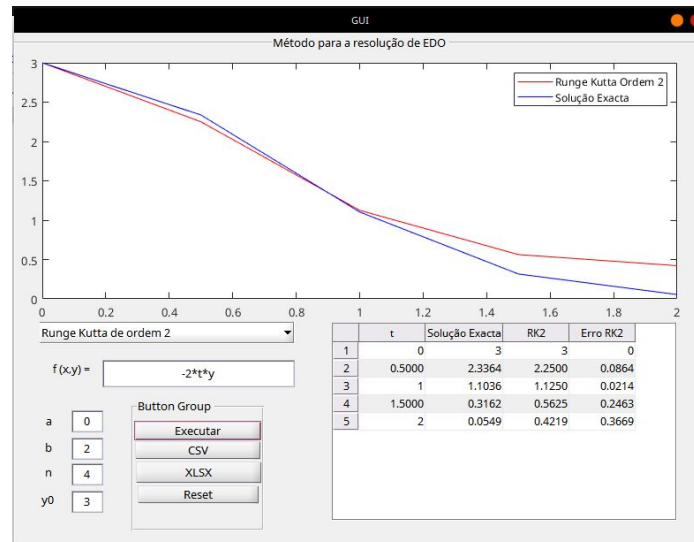
i	t _i	Exata	Euler	RK2	RK4	Erro Euler	Erro RK4	Erro RK2
0	0	3	3	3	3	0	0	0
1	0.5	2.3364	3	2.2500	2.3359	0.6636	0.0864	0.0005
2	1	1.1036	1.5000	1.1250	1.1041	0.3964	0.0214	4.5714e-04
3	1.5	0.3162	0	0.5625	0.3350	0.3162	0.2463	0.0188
4	2	0.0549	0	0.4219	0.0907	0.0549	0.3669	0.0358

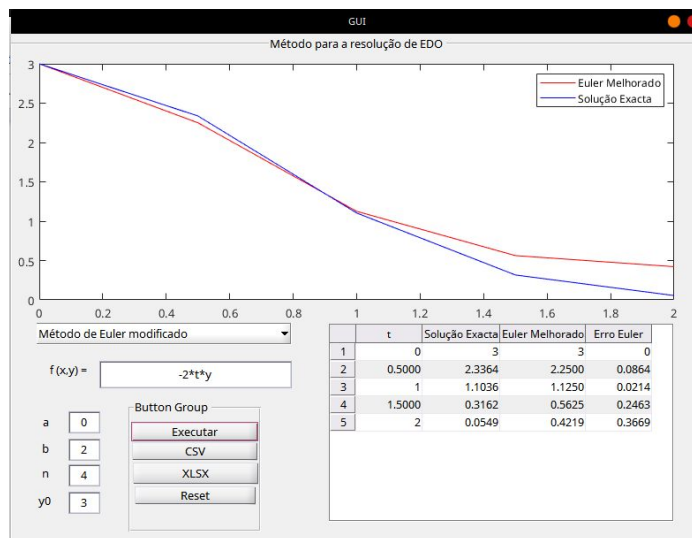
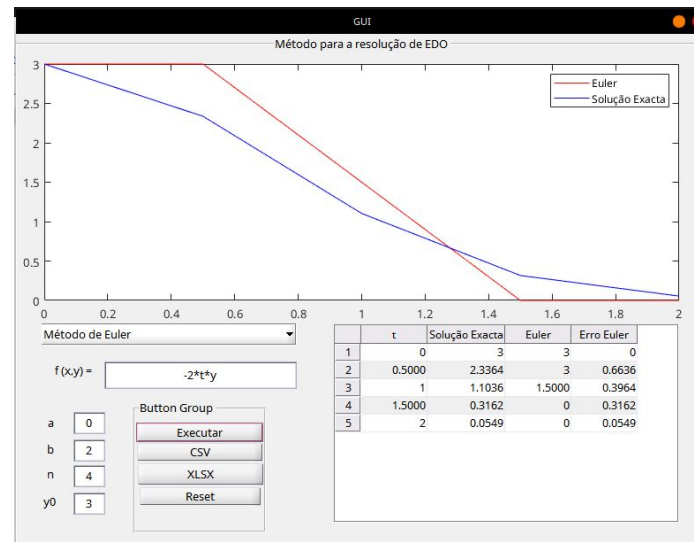
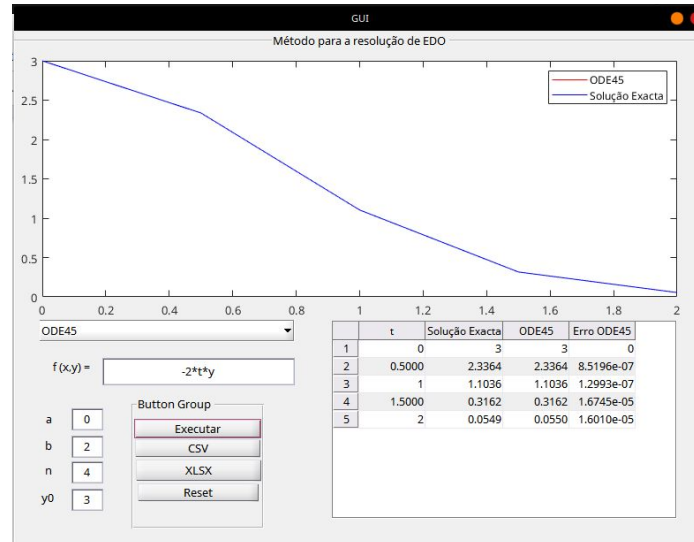
Alínea (d)

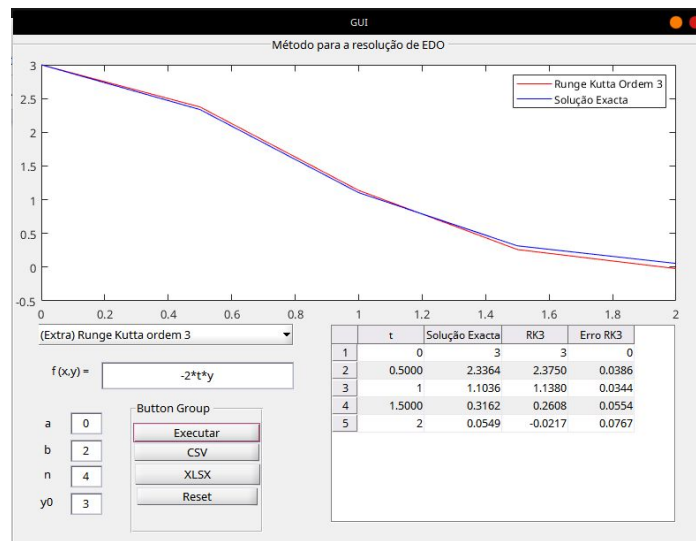
A figura 9 é a imagem que melhor representa a solução gráfica do problema PVI, pois o Y0 (ponto inicial) é definido pelo enunciado como 0 e o intervalo é de 0 a 2, o que não se verifica na figura 8.



Exemplos de output - GUI com gráfico e tabela







Extração de CSV

```

1 function csv = extractCSV(headers, handles)
2
3 fileDir=uigetdir('','Selecione a pasta para guardar o teu CSV');
4 csvDetail=inputdlg({'Filename:'},...
5 'CSV workbook details',1,['Test.csv']) ;
6 fileName = [fileDir, '/', csvDetail{1}];
7
8 commaHeader = [headers; repmat({','},1,numel(headers))];
9 commaHeader = commaHeader(:)';
10 textHeader = cell2mat(commaHeader);
11
12
13
14 fp = fopen(fileName, 'w');
15 fprintf(fp, '%s\n', textHeader);
16 fclose(fp);
17 dlmwrite(fileName, handles, '-append');
18

```

Esta função pede ao utilizador que especifique onde quer guardar o ficheiro csv e qual o nome do mesmo.

Após isto, a função faz o parse dos headers e de seguida insere-os no ficheiros através da função fprintf. Para inserir a informação no csv, usamos o dlmwrite, com o argumento append, que ao invés de limpar tudo, adiciona ao que foi escrito anteriormente, ou seja, os headers.

Conclusão

Neste relatório, foram abordados foram abordadas as funções faladas nas lecionadas nas aulas, tratando-os minuciosamente desde as fórmulas utilizadas, algoritmos e alguns exercícios de aplicação para testar o funcionamento da GUI.

Os objectivos da actividade prática foram alcançados, pois a aplicação, possivelmente, não tem erros assinaláveis, o que foi bom para a nossa evolução na programação em MATLAB.

Todas as funções foram desenvolvidas dentro dos padrões definidos pelos professores Arménio Correia e Nuno Baeta, sendo assim útil a nossa percepção de métodos numéricos e de programação.