Relatório de Métodos Numéricos para PVI

Análise Matemática II - Atividade 1

Dinis Meireles de Sousa Falcão, 2020130403
David Seco Rodrigues, 2019130152

Índice

1 INTRODUÇÃO

O presente relatório corresponde à **«Atividade01Trabalho»**, cujo os objetivos são a aquisição de conhecimentos sobre métodos numéricos para resolução de equações diferenciais ordinárias e/ou problemas de valor inicial, e a sua implementação em MatLab.

Os computadores são uma ferramenta extremamente útil no estudo de equações diferenciais, uma vez que através deles é possível executar algoritmos que constroem aproximações numéricas para as soluções destas equações. Apresentamos os métodos numéricos de **Euler**, **Euler Melhorado** e a classe de **Métodos de Runge-Kutta (ordem 2 e 4)**.

1.1 Enunciado da atividade proposta e interpretação do mesmo

"Com base nos ficheiros programados nas aulas T, TP e P de AM2 e ainda nos ficheiros existentes nas sub-pastas «**ficheiros de suporte à Atividade01Trabalho**» que deve adaptar para a versão mais recente do MatLab, por exemplo, utilizar a função dsolve com parâmetros de entrada que sejam expressões simbólicas e não strings, implementar os seguintes métodos:

- **Euler** (versão beta com "alocação" de memória);
- **Euler Melhorado** ou modificado;
- **Runge-Kutta de ordem 2** (RK2);
- **Runge-Kutta de ordem 4** (RK4);
- Função que utilize o **ODE45** do MatLab;
- Pesquisa de outro método [**Runge-Kutta de ordem 3** (RK3)].

1ª Parte da atividade: [até 19 de abril]
Com base nos ficheiros da pasta de suporte a esta atividade e/ou outros implementados nas aulas, implementar e acrescentar os métodos anteriores (com "alocação" de memória), ajustar e reconstruir a interface de texto com o utilizador para uma LiveScript, sem esquecer a validação dos parâmetros de entrada.

2ª Parte da atividade: [até 30 de abril]
Implementar uma App de de interface com utilizador. »Na sub-pasta **at01_MN_PVI_GUI**, encontram um esquiço de uma GUI (Graphical User Interface) com radio buttons e outros objetos como um modelo e ideia para construção de uma App de interface com o utilizador.

3ª Parte da atividade: [até 30 de abril]
Elaboração de um relatório."

1.2 Definição de PVI

"Chama-se problema de valores iniciais (PVI) ou **Problema de Cauchy** a todo o problema que consiste numa equação diferencial satisfazendo a condições dadas num único ponto do intervalo em que a equação é considerada. Estas condições denominam-se condições iniciais.

Para uma equação de ordem n, um problema de valores iniciais consiste então em resolver, num intervalo I⊆IR, a equação

$\frac{d^n y}{dx^n} = f(x,y,y',...,y^{(n-1)})$

sujeita às condições

$y(x_0) = y_0, y'(x_0) = y_1 \ldots, y^{(n-1)}(x_0) = y_{n-1},$

em que x0é um ponto pertencente ao intervalo I, e y_0, y_1, y_{n-1}, $\ldots$,

são constantes reais."

in Capítulo 1 - Equações Diferenciais Ordinárias, Alves, Maria; Rosa, Paulo

2. MÉTODOS NUMÉRICOS PARA RESOLUÇÃO DE PVI

2.1. Método de Euler

2.1.1. Fórmula

$y_{i+1} = \boldsymbol{yi} + \boldsymbol{hf}(\boldsymbol{ti}, \boldsymbol{yi})$, $\boldsymbol{i} = \boldsymbol{0}$, $\boldsymbol{1}$,..., $\boldsymbol{n-1}$

2.1.2. Algoritmo/Função

Inputs:

I. f - Função do segundo membro de uma Equação Diferencial

II. [a,b] - Extremos do intervalo de uma variável independente t

III. n - Número de iterações do método

IV. y0 - Condição inicial, com t = a implicando y=y0

Outputs:

I. y - Vetor das aproximações discretas da solução exata

II. y(i+1) = y(i)+h*f(t(i),y(i)) , i =0,1,...,n-1

Função:

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

2.2 Método de Euler Melhorado ou Modificado

2.2.1 Fórmulas

$$\boldsymbol{y_{i+1}} = \boldsymbol{yi} + \boldsymbol{h}[\tfrac{1}{2}(\boldsymbol{f}(\boldsymbol{ti},\boldsymbol{yi}) + \boldsymbol{f}(\boldsymbol{ti} + \boldsymbol{hyi} + \boldsymbol{hf}(\boldsymbol{ti},\boldsymbol{yi})))]$$
$$\boldsymbol{i} = \boldsymbol{0}, \boldsymbol{1}, \ldots, \boldsymbol{n-1}$$

2.1.2 Algoritmo/Função

Inputs:

I. f - Função do segundo membro de uma Equação Diferencial

II. [a,b] - Extremos do intervalo de uma variável independente t

III. n - Número de iterações do método

IV. y0 - Condição inicial, com t = a implicando y=y0

Outputs:

I. y - Vetor das aproximações discretas da solução exata

II. y(i+1) = y(i)+h*f(t(i),y(i)) , i =0,1,...,n-1

Função:

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

2.3 Método de RK2

2.3.1 Fórmulas

Método de Runge-Kutta de ordem 2 (RK2)

$\boldsymbol{k1}=\boldsymbol{hf}(\boldsymbol{ti},\boldsymbol{yi})$
$\boldsymbol{k2}=\boldsymbol{hf}(\boldsymbol{ti}+\mathbf{1},\boldsymbol{yi}+\boldsymbol{k1})$
$y_{i+1}=\boldsymbol{yi}+\frac{1}{2}(\boldsymbol{k1},\boldsymbol{k2}),\ \boldsymbol{i}=\mathbf{0},\mathbf{1},\dots,\boldsymbol{n}-\mathbf{1}$

2.3.2 Algoritmo/Função

Inputs:

I. f - Função do 2.º membro da Equação Diferencial

II. [a, b] - Extremos do intervalo da variável independente t

III. n - Número de subintervalos ou iterações do método

IV. t=a - y=y0 - condição inicial

Outputs:

I. y - Vetor das aproximações discretas da solução exata

II. y(i+1) = y(i)+h*f(t(i), y(i)), i =0, 1, ..., n-1

Função:

```
function y=RK2(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
    k1=h*f(y(i),t(i));
    k2=h*f(y(i)+k1,t(i+1));
    y(i+1)=y(i)+(k1+k2)/2;
end
```

2.4 Método de RK4

2.4.1 Fórmulas

Método de Runge-Kutta de ordem 4 (RK4)

$\boldsymbol{k1} = \boldsymbol{hf}(\boldsymbol{ti}, \boldsymbol{yi});$
$\boldsymbol{k2} = \boldsymbol{hf}(\boldsymbol{ti}+\boldsymbol{h2}, \boldsymbol{yi}+\boldsymbol{12k1})$
$\boldsymbol{k3} = \boldsymbol{hf}(\boldsymbol{ti}+\boldsymbol{h2}, \boldsymbol{yi}+\boldsymbol{12k2})$ $\boldsymbol{k4}=\boldsymbol{hf}(\boldsymbol{ti}+\boldsymbol{1}, \boldsymbol{yi}+\boldsymbol{k3})$
$y_{i+1}= \boldsymbol{yi}+\frac{1}{6}(\boldsymbol{k1}, \boldsymbol{2k2}+\boldsymbol{2k3}+\boldsymbol{k4}), \boldsymbol{i}=\boldsymbol{0}, \boldsymbol{1}, \dots, \boldsymbol{n}-\boldsymbol{1}$

2.4.2 Algoritmo/Função

Inputs:

I. f - Função do 2.º membro da Equação Diferencial

II. [a, b] - Extremos do intervalo da variável independente t

III. n - Número de subintervalos ou iterações do método

IV. t=a - y=y0 - condição inicial

Outputs:

I. y - Vetor das aproximações discretas da solução exata

II. y(i+1) = y(i)+h*f(t(i), y(i)), i =0, 1, ..., n-1

Função:

```
function y = 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) + (1/2) * K1);
    K3 = h * f(t(i) + (h/2),y(i) + (1/2) * K2);
    K4 = h * f(t(i) + h,y(i) + K3);
    y(i+1) = y(i) + ((1/6) * (K1 + (2*K2) + (2*K3) + K4));
end
```

2.5 Função ODE45 do Matlab

2.5.1 Fórmulas

$[t, y] = ode45(f, t, y0)$

2.5.2 Algoritmo/Função

Inputs:

I. f - Função do 2.º membro da Equação Diferencial
II. [a, b] - Extremos do intervalo da variável independente t
III. n - Número de subintervalos ou iterações do método
IV. t=a - y=y0 - condição inicial

Outputs:

I. t - discretização do domínio
II. y - Vetor das aproximações discretas da solução exata

Função:

```
function y = ODE45(f,a,b,n,y0)
h = (b-a)/n;
t = a:h:b;
y=zeros(1,n+1);
[t,y] = ode45(f,t,y0);
end
```

2.6 Método de RK3

2.6.1 Fórmulas

$k1 = hf(ti,yi)$
$k2 = hf(ti+12h,yi+12hk1)$
$k3 = hf(ti+h,yi-hk1+2k2)$
$y_{i+1} = yi+\frac{1}{6}h(k1+4k2+k3)$

2.6.2 Algoritmo/Função

Inputs:

I. f - Função do 2.º membro da Equação Diferencial

II. [a, b] - Extremos do intervalo da variável independente t

III. n - Número de subintervalos ou iterações do método

IV. t=a - y=y0 - condição inicial

Outputs:

I. y - Vetor das aproximações discretas da solução exata

II. y(i+1) = y(i)+h*f(t(i), y(i)), i =0, 1, ..., n-1

Função:

```
function y=RK3(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
    k1 = h*f(t(i),y(i));
    k2 = h*f(t(i)+h*0.5,y(i)+h*k1*0.5);
    k3 = h*f(t(i)+h, y(i)-h*k1 +2*k2*h);
    y(i+1) =y(i)+h*(1/6)*(k1 +4*k2+k3);
end
```

3. EXEMPLOS DE APLICAÇÕES E TESTES DOS MÉTODOS

3.1 Exercício 3 do Teste Farol

3.1.1 PVI - Equação diferencial de 1.ª Ordem e Condições Iniciais

Equação Diferencial de 1ª Ordem:

$$y' = -2ty$$

Intervalo:

t ∈ [0,1.5]
a = 0
b = 1.5

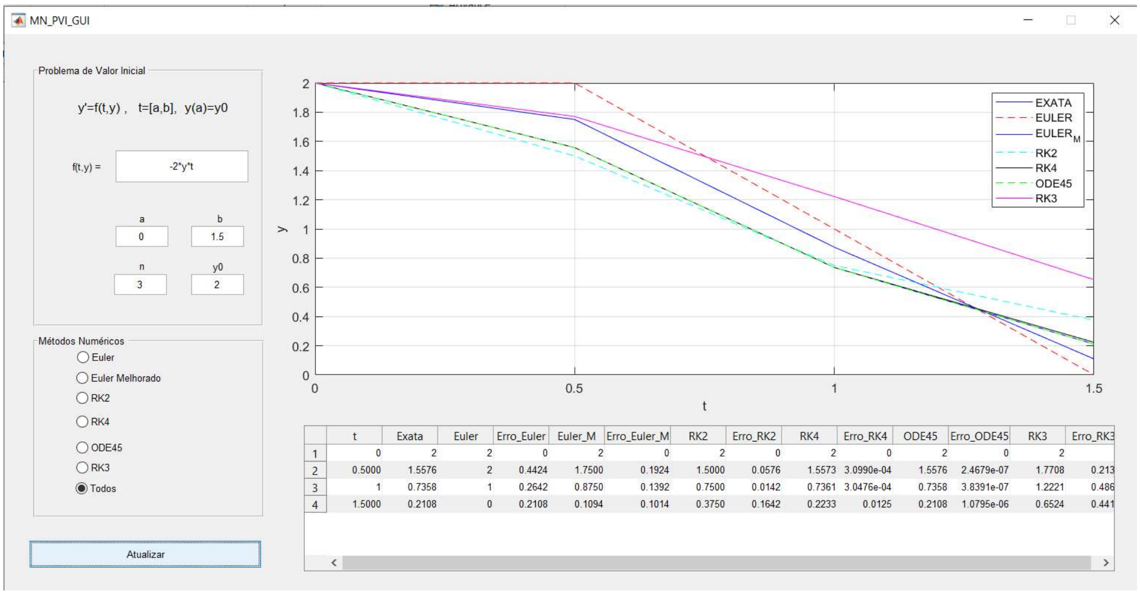
Condições Iniciais:

y0 = 2
n = 3

Solução Exata e Particular:

$$y(t) = e^{(-t)^2}$$

3.1.2 Exemplos de output - GUI com gráficos e tabela



	t	Exata	Euler	Erro_Euler	Euler_M	Erro_Euler_M	RK2	Erro_RK2	RK4	Erro_RK4	ODE45	Erro_ODE45	RK3	Erro_RK3
1	0	2	2	0	2	0	2	0	2	0	2	0	2	
2	0.5000	1.5576	2	0.4424	1.7500	0.1924	1.5000	0.0576	1.5573	3.0990e-04	1.5576	2.4679e-07	1.7708	0.213
3	1	0.7358	1	0.2642	0.8750	0.1392	0.7500	0.0142	0.7361	3.0476e-04	0.7358	3.8391e-07	1.2221	0.486
4	1.5000	0.2108	0	0.2108	0.1094	0.1014	0.3750	0.1642	0.2233	0.0125	0.2108	1.0795e-06	0.6524	0.441

3.2 Problema de aplicação

3.2.1 Modelação matemática do problema

Como: k=0.125, m=5 e g=32ft/s2, temos que:

$$m\frac{dv}{dt} = mg - kv^2 \Leftrightarrow mv' = mg - kv^2$$

3.2.2 Resolução através da aplicação criada

Resolução através da aplicação criada – Exercício 1

a) Use the Runge-Kutta method with h=1 to find an approximation to the velocity of the falling max at t=5s.

I. y' = (5*32-0.125*y^2)/5;
II. y0 = v(0) = 0;
III. t ∈ [0,5];
IV. a = 0;
V. b = 5;
VI. h = 1;
VII. n = (b-a)/h = (5-0) /1 = 5;

Resolução com o Método RK2:

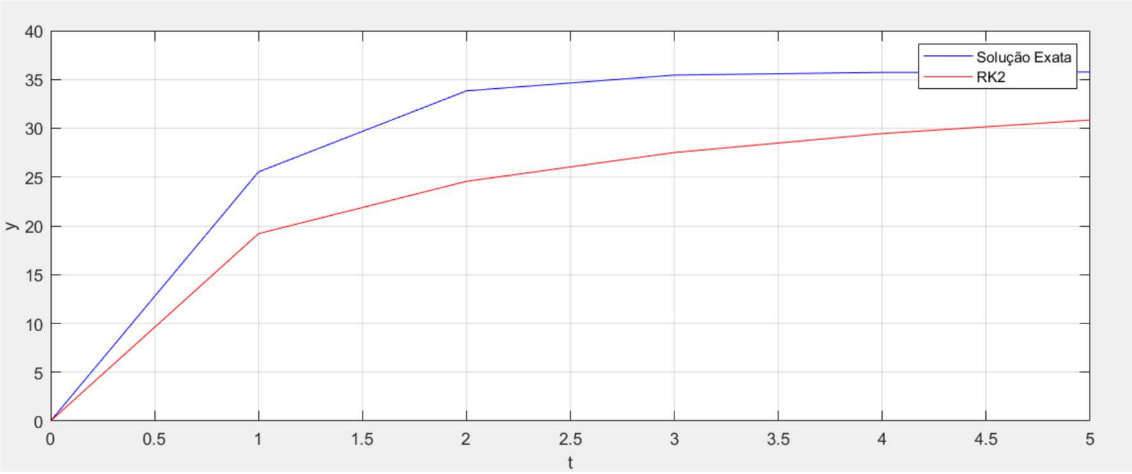
	t	Solução Exata	RK2	Erro RK2
1	0	0	0	0
2	1	25.5296	19.2000	6.3296
3	2	33.8322	24.5588	9.2734
4	3	35.4445	27.5118	7.9327
5	4	35.7213	29.4569	6.2644
6	5	35.7678	30.8457	4.9221

Resolução com o Método RK4:

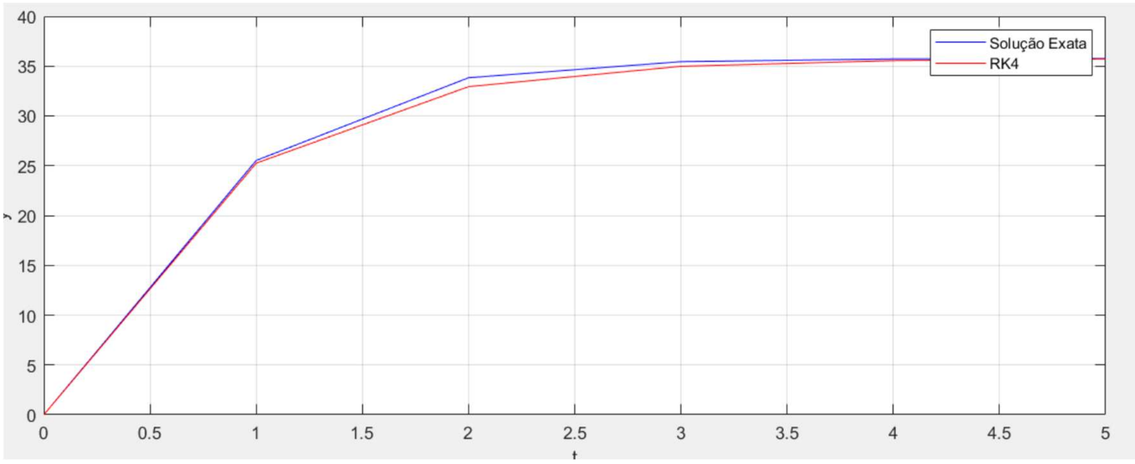
	t	Solução Exata	RK4	Erro RK4
1	0	0	0	0
2	1	25.5296	25.2570	0.2726
3	2	33.8322	32.9390	0.8932
4	3	35.4445	34.9772	0.4673
5	4	35.7213	35.5503	0.1709
6	5	35.7678	35.7128	0.0550

b) Use a numerical solver to graph the solution of the initial-value problem.

Resolução com o Método RK2:



Resolução com o Método RK4:



a) Use the Runge-Kutta method with h=0.5 to complete the following table:

I. y' = y*(2.218-0.0432*y);
II. y0 = 0.24;
III. t ∈ [1,5];
IV. a = 1;
V. b = 5;
VI. h = 0.5;
VII. n = (b-a)/h = (5-1) /0.5 = 8.

Com Runge-Kutta de Ordem 2:

T (Days)	1	2	3	4	5
A (Observed)	2.78	13.53	36.30	47.50	49.40
A (Approximated)	0.2400	1.7401	10.9768	35.4325	47.4155

Com Runge-Kutta de Ordem 4:

T (days)	1	2	3	4	5
A (Observed)	2.78	13.53	36.30	47.50	49.40
A (Approximated)	0.2400	2.1025	14.3923	40.1006	49.7660

4. CONCLUSÃO

Com a realização deste trabalho, concluímos que o Método mais preciso de todos os abordados é o ODE45, com valores extremamente próximos à solução exata. Em contrapartida, o método numérico menos preciso é o método de Euler, que apresenta valores com erros muito elevados, excetuando-se alguns casos.