Instituto Superior de Engenharia de Coimbra

LICENCIATURA EM ENGENHARIA INFORMÁTICA – PÓS LABORAL

ANÁLISE MATEMÁTICA II

Relatório Atividade 2

Métodos Numéricos SED

Rafael Filipe Martins Alves | 2014013189

Coimbra, 06 de Junho de 2020

Índice

1. Introdução

Este trabalho surge no âmbito da disciplina de Análise Matemática II, no qual nos foi proposto a realização de uma atividade prática sobre métodos numéricos para resolução e aproximação de Sistemas de Equações Diferenciais de ordem 1 através da ferramenta MATLAB.

Numa primeira fase vai ser abordada uma definição dos Sistemas de Equações Diferenciais (SED) e posteriormente vai ser feita a analise das fórmulas e dos algoritmos dos diferentes métodos de resolução de SED's.

Numa segunda fase vão ser apresentados exemplos de aplicação e os respetivos testes dos métodos analisados, bem como alguns exercícios resolvidos com a aplicação desenvolvida para testar as funções implementadas.

1.1. Enunciado da atividade proposta e interpretação do mesmo

Pretende-se com esta atividade adquirir conhecimentos sobre os métodos numéricos para resolução de SED, programando esses métodos e ao mesmo tempo desenvolvendo competências algorítmicas e de programação em MATLAB.

Assim sendo, iremos desenvolver um programa em MATLAB que aplique as técnicas lecionadas na unidade curricular de Análise Matemática II. A aplicação vai ser implementada através de uma GUI, tendo como objetivo implementar os seguintes métodos:

- Euler
- Euler melhorado
- Runge-Kutta de ordem 2 (RK2)

- Runge-Kutta de ordem 4 (RK4)

1.2. Definição de SED

Um sistema de equações diferenciais é um sistema constituído por duas ou mais equações envolvendo derivadas, neste caso de primeira ordem, de duas ou mais variáveis dependentes relativamente a uma só variável independente.

2. Métodos Numéricos para resolução de SED

2.1. Método de Euler

2.1.1. Fórmulas

$$\mathrm{u}(\mathrm{i}+1) = u_i + hf(t_i, u_i, v_i), i = 0,1,2 \dots, n-1$$
$$v(i+1) = v_i + hg(t_i, u_i, v_i), i = 0,1,2 \dots, n-1$$

2.1.2. Algoritmo/Função

```
INPUT: f,a,b,n,u0,v0
OUTPUT: u,v

h=(b-a)/n;
t=a:h:b;
u=zeros(1,n+1);
v=zeros(1,n+1);
u(1)=u0;
v(1)=v0;
for i=1:n
u(i+1)=u(i)+h*f(t(i),u(i),v(i));
v(i+1)=v(i)+h*g(t(i),u(i),v(i));
```

2.2. Método de Euler Melhorado ou Modificado

2.2.1. Fórmulas

$$u(i+1) = u_i + hf(t_i, u_i, v_i)$$
$$v(i+1) = v_i + hg(t_i, u_i, v_i)$$
$$u(i+1) = u_i + \frac{h}{2} * \left(f(t_i, u_i, v_i)\right) + (f(t(i+1), u(i+1), v(i+1)))$$
$$v(i+1) = u_i + \frac{h}{2} * \left(g(t_i, u_i, v_i)\right) + (g(t(i+1), u(i+1), v(i+1)))$$

2.2.2. Algoritmo/Função

```
INPUT: f,a,b,n,u0,v0
OUTPUT: u,v

h = (b-a)/n;
t = a:h:b;
u = zeros(1,n+1);
v = zeros(1,n+1);
u(1) = u0;
v(1) = v0;
for i=1:n
u(i+1)=u(i)+h*f(t(i),u(i),v(i));
v(i+1)=v(i)+h*g(t(i),u(i),v(i));
u(i+1)=u(i)+(h/2)*(f(t(i),u(i),v(i))+f(t(i+1),u(i+1),v(i+1)));
v(i+1)=v(i)+(h/2)*(g(t(i),u(i),v(i))+g(t(i+1),u(i+1),v(i+1)));
```

2.3. Método de RK2

2.3.1. Fórmulas

$$
\begin{aligned}
&k1u = hf(t_i, u_i, v_i)\\
&k1v = hg(t_i, u_i, v_i)\\
&k2u = hf(t_i + 1, u_i + k1u, v_i + k1u)\\
&k2v = hg(t_i + 1, u_i + k1v, v_i + k1v)\\
&u(i + 1) = u_i + (k1u + k2u)/2, \quad i = 0,1,2, \dots, n - 1\\
&u(i + 1) = v_i + (k1v + k2v)/2, \quad i = 0,1,2, \dots, n - 1
\end{aligned}
$$

2.3.2. Algoritmo/Função

```
INPUT: f,a,b,n,u0,v0
OUTPUT: u,v

h = (b-a)/n;
t = a:h:b;
u = zeros(1,n+1);
v = zeros(1,n+1);
u(1)  = u0;
v(1)  = v0;
for i=1:n
k1u=h*f(t(i),u(i),v(i));
k1v=h*g(t(i),u(i),v(i));
k2u=h*f(t(i+1),u(i)+k1u,v(i)+k1u);
k2v=h*g(t(i+1),u(i)+k1v,v(i)+k1v);
u(i+1)=u(i)+(k1u+k2u)/2;
v(i+1)=v(i)+(k1v+k2v)/2;
```

2.4. Método de RK4

2.4.1. Fórmulas

$$k1u = hf(t_i, u_i, v_i)$$
$$k1v = hg(t_i, u_i, v_i)$$
$$k2u = hf\left(t_i + \frac{h}{2}, u_i + (\frac{1}{2}k1u), v_i + (\frac{1}{2}k1u)\right)$$
$$k2v = hg\left(t_i + \frac{h}{2}, u_i + (\frac{1}{2}k1v), v_i + (\frac{1}{2}k1v)\right)$$
$$k3u = hf\left(t_i + \frac{h}{2}, u_i + (\frac{1}{2}k2u), v_i + (\frac{1}{2}k2u)\right)$$
$$k3v = hg\left(t_i + \frac{h}{2}, u_i + (\frac{1}{2}k2v), v_i + (\frac{1}{2}k2v)\right)$$
$$k4u = hf(t_i + h, u_i + k3u, v_i + k3u)$$
$$k4v = hg(t_i + h, u_i + k3v, v_i + k3v)$$

2.4.2. Algoritmo/Função

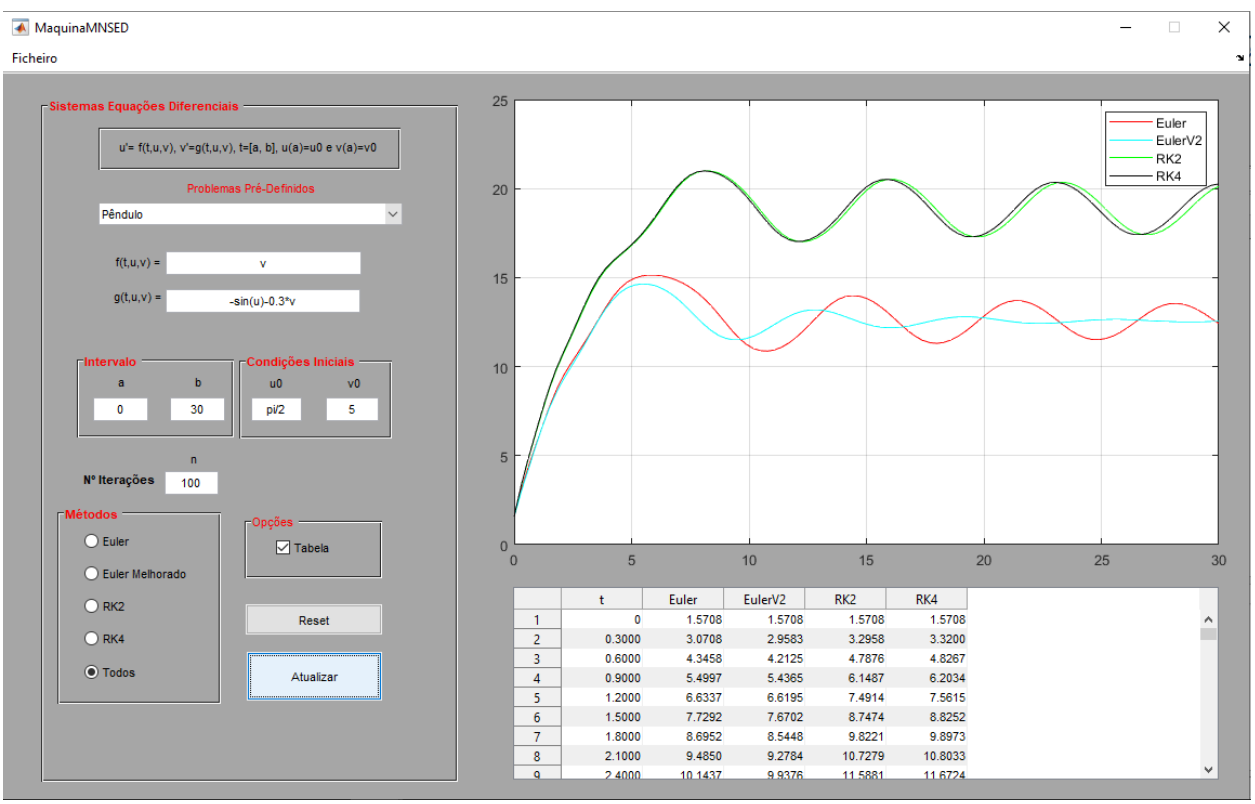
```
INPUT: f,a,b,n,u0,v0
OUTPUT: u,v

h = (b-a)/n;
t = a:h:b;
u = zeros(1,n+1);
v = zeros(1,n+1);
u(1) = u0;
v(1) = v0;
for i=1:n
k1u=h*f(t(i),u(i),v(i));
k1v=h*g(t(i),u(i),v(i));
k2u=h*f(t(i)+ h/2,u(i)+(1/2*k1u),v(i)+(1/2*k1u));
k2v=h*g(t(i)+ h/2,u(i)+(1/2*k1v),v(i)+(1/2*k1v));
k3u=h*f(t(i)+ h/2,u(i)+(1/2*k2u),v(i)+(1/2*k2u));
k3v=h*g(t(i)+ h/2,u(i)+(1/2*k2v),v(i)+(1/2*k2v));
k4u=h*f(t(i)+ h,u(i)+k3u,v(i)+k3u);
k4v=h*g(t(i)+ h,u(i)+k3v,v(i)+k3v);
u(i+1)=u(i)+(1/6)*(k1u+2*k2u+2*k3u+k4u);
v(i+1)=v(i)+(1/6)*(k1v+2*k2v+2*k3v+k4v);
```

3. Exemplos de aplicação e teste dos métodos

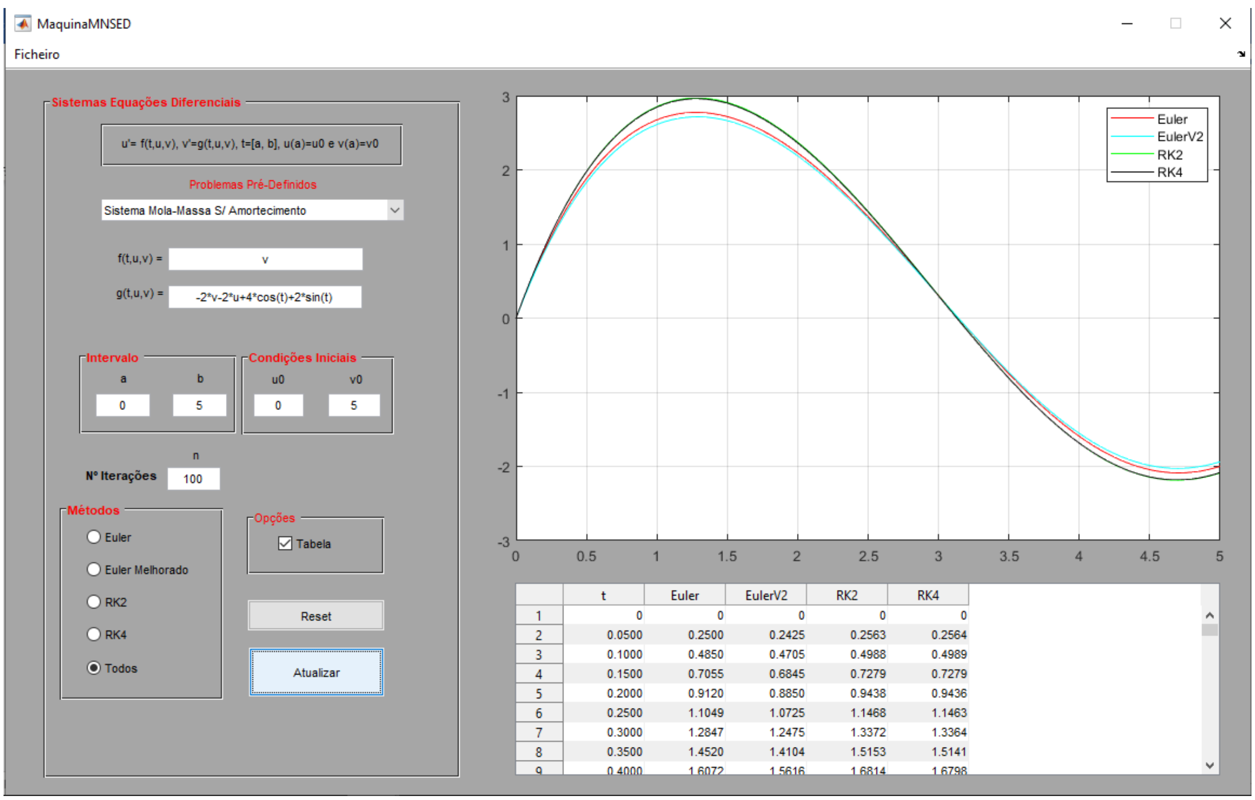
3.1. Problema do Pêndulo

```
f(t,u,v) = v
g(t,u,v) = -sin(u)-0.3*v
a = 0
b = 15
u0 = pi/2
v0 = 0
```



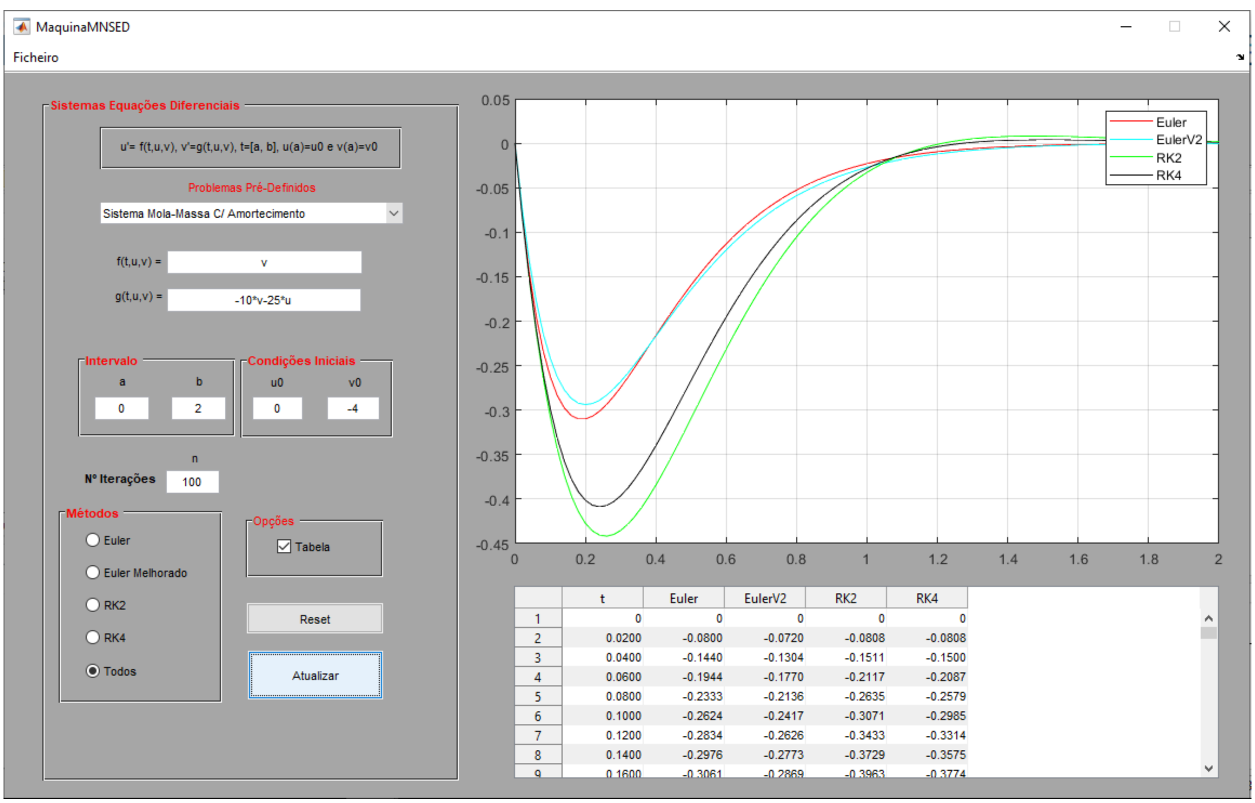
3.2. Problema Sistema Mola-Massa S/ Amortecimento

```
f(t,u,v) = v
g(t,u,v) = -2*v-2*u+4*cos(t)+2*sin(t)
a = 0
b = 15
u0 = 0
v0 = 3
```



3.3. Problema Sistema Mola-Massa C/ Amortecimento

```
f(t,u,v) = v
g(t,u,v) = -10*v-25*u
a = 0
b = 2
u0 = 0
v0 = -4
```



4. Conclusão

Concluindo a atividade apresento agora algumas considerações finais sobre o trabalho realizado.

Ao longo da sua execução verifiquei as verdadeiras vantagens da utilização de métodos numéricos como ferramenta fundamental e indispensável para a resolução de sistemas de equações diferenciais de uma forma rápida e eficaz, reforçando assim a ideia de que a criação de algoritmos que permitam resolver este e outros tipos de problema, são uma constante na vida de um/a Engenheiro/a.

Durante o desenvolvimento deste trabalho surgiram algumas dificuldades na execução em algumas tarefas, tanto a nível de programação e adaptação no ***MATLAB*** como na correta interpretação das questões (fórmulas) apresentadas.