Relatório - Métodos numéricos para resolução de Sistemas de ED

Departamento de Engenharia Informática e de Sistemas (DEIS)

Bruno Teixeira(2019100036), Rafael Ribeiro(2019131989), Gonçalo Correia(2019150122), Sofia Janeiro(2019132578)

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Em situações reais sobre quantidade da vida a taxa de variação depende mais do que uma variável. Por exemplo, a população de minhocas, embora possa ser representado por um número único, depende do tamanho das populações de predadores e a disponibilidade de alimento. Para representar e estudar esses problemas complicados, precisamos de usar sistemas de equações diferenciais. Este trabalho tem como objetivo conseguir perceber como é que podemos usar os métodos numéricos para a resolução de sistemas de equações diferenciais. Para isso vamos usar o Método de Euler, o Método de Euler Melhorado e os Métodos de Runge-Kutta de ordem 2 e de ordem 4. Com a implementação deste trabalho em MATLAB, vamos conseguir tambem resolver vários problemas conhecidos, como por exemplo o problema do pêndulo.

Capítulo 2

Desenvolvimento

Neste capítulo são apresentados os algoritmos em MATLAB referente a cada método que vamos usar.

2.1 SED

Um sistema de equações diferenciais é um sistema constituído por duas ou mais equações envolvendo derivadas (não necessariamente de primeira ordem) de duas ou mais variáveis dependentes relativamente a uma só variável independente.

$$\frac{dx_1}{dt} = F_1(x_1, \ldots, x_n, x_1', \ldots, x_n', t)$$
$$\frac{dx_2}{dt} = F_2(x_1, \ldots, x_n, x_1', \ldots, x_n', t)$$
$$\ldots$$
$$\frac{dx_n}{dt} = F_n(x_1, \ldots, x_n, x_1', \ldots, x_n', t)$$

Figura 2.1: Sistema de Equações Diferenciais

2.2 Métodos para o SED

2.2.1 Método de Euler

- O método de Euler, é um método numérico de primeira ordem para solucionar equações diferenciais ordinárias com um valor inicial dado. Neste caso iremos usar este método para solucionar um sistema de equações.

```
function [t,u,v]= NEulerSED(f,g,a,b,n,u0,v0)
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));
end

end
```

Figura 2.2: Método de Euler

2.2.2 Método de Euler Melhorado

- O método de Euler melhorado é semelhante ao método de Euler tradicional, a única diferença é que este método utiliza uma média das inclinações em cada ponto para cada iteração, dando assim uma precisão maior.

```
function [t, u, v] = NEuler_Melhorada(f,g,a,b,n,u0,v0)
    h = (b-a)/n;
    t = a:h:b;
    u = zeros(1,n+1);
    u(1) = u0;
    v = zeros(1,n+1);
    v(1) = v0;

    for i=1:n
        uFun = f(t(i),u(i),v(i));
        vFun = g(t(i),u(i),v(i));
        u(i+1)=u(i)+h*(uFun+f(t(i+1), u(i)+h*uFun, v(i)+h*uFun))/2;
        v(i+1)=v(i)+h*(vFun+g(t(i+1), u(i)+h*uFun, v(i)+h*uFun))/2;
    end

end
```

Figura 2.3: Método de Euler Melhorado

2.2.3 Método RK2

- É um método numérico com alguma precisão, isto deve-se em muito à sua fórmula que considera para cada iteração dois valores denominados normalmente por “k” onde o primeiro é a inclinação no início do intervalo, o segundo é a inclinação no final do intervalo.

```
function [t, u, v] = NRK2SED(f, g, a, b, n, u0, v0)
    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) + k1v);
        k2v = h * g(t(i+1), u(i) + k1u, v(i) + k1v);

        u(i+1) = u(i) + (1/2)*(k1u + k2u);
        v(i+1) = v(i) + (1/2)*(k1v + k2v);
    end
end
```

Figura 2.4: Método RK2

2.2.4 Método RK4

- É o método numérico com a maior precisão. A sua fórmula considera para cada iteração quatro valores denominados normalmente por "k" onde o primeiro é a inclinação no início do intervalo, o segundo é a inclinação no ponto médio do intervalo usando a primeira inclinação, o terceiro é novamente a inclinação no ponto médio do intervalo mas, desta vez, utilizando a segunda inclinação e, finalmente, o quarto é a inclinação no final do intervalo.

```
function [t, u, v] = NRK4SED(f, g, a, b, n, u0, v0)
    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)+(k1u/2), v(i)+(k1v/2));
        k2v = h * g(t(i)+(h/2), u(i)+(k1u/2), v(i)+(k1v/2));

        k3u = h * f(t(i)+(h/2), u(i)+(k2u/2), v(i)+(k2v/2));
        k3v = h * g(t(i)+(h/2), u(i)+(k2u/2), v(i)+(k2v/2));

        k4u = h * f(t(i)+h, u(i) + k3u, v(i) + k3v);
        k4v = h * g(t(i)+h, u(i) + k3u, 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);
    end
end
```

Figura 2.5: Método RK4

2.2.5 Solução Exata

- A solução exata, diz exatamente o resultado que um certo exercicio retorna, uma vez que é uma solução exata, esta é muitas vezes usada como modo de comparação com os outros métodos para tentar perceber qual deles é o mais eficaz e preciso.

```
function [t, exata] = SolExata(ODE, a, b, n, u0, v0)
    syms y(t);
    tempExata = dsolve(ODE, ['y(0)=',num2str(u0)], ...
        ['Dy(0)=',num2str(v0)], 't');
    if(~isempty(tempExata))
        ext=@(t) eval(vectorize(char(tempExata)));
        h=(b-a)/n;
        t=a:h:b;
        exata=ext(t);
    else
        exata=[];
end
```

Figura 2.6: Solução Exata

2.3 Problemas de aplicação

Note que para a resolução destes problemas irão ser mostradas todos os métodos criados no programa.

2.3.1 Problema do Pêndulo

- Usando o programa criado em MATLAB, conseguimos então resolver o problema to pêndulo de maneira simples.
- Uma vez que este problema é uma equação diferencial de ordem 2 homogénea **nao linear**, nao irá ser apresentado uma solução exata.
- No programa irá apresentar a função dada no enunciado, os parâmetros de entrada, os métodos que irão resolver o problema, um gráfico e uma tabela.
- Usando todos os métodos, para termos uma comparação entre eles, podemos ver a precisão de cada um para resolver este problema.

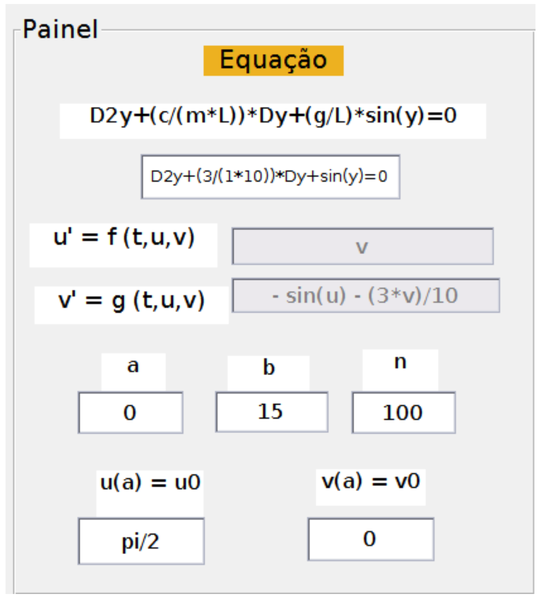

Figura 2.7: Parâmetros - Pêndulo

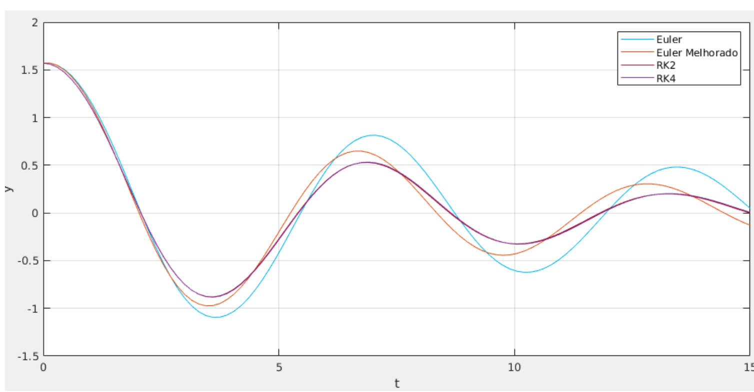

Figura 2.8: Gráfico - Pêndulo

	t	Euler	Euler Melhorado	RK2	RK4
1	0	1.5708	1.5708	1.5708	1.5708
2	0.1500	1.5708	1.5708	1.5595	1.5597
3	0.3000	1.5483	1.5466	1.5268	1.5271
4	0.4500	1.5043	1.4994	1.4735	1.4740
5	0.6000	1.4398	1.4303	1.4007	1.4013
6	0.7500	1.3558	1.3406	1.3093	1.3101
7	0.9000	1.2532	1.2314	1.2008	1.2018
8	1.0500	1.1332	1.1042	1.0764	1.0777
9	1.2000	0.9973	0.9610	0.9382	0.9397
10	1.3500	0.8471	0.8039	0.7883	0.7901
11	1.5000	0.6848	0.6357	0.6293	0.6315
12	1.6500	0.5129	0.4596	0.4644	0.4670
13	1.8000	0.3346	0.2794	0.2969	0.2998
14	1.9500	0.1532	0.0991	0.1303	0.1335
15	2.1000	-0.0275	-0.0773	-0.0317	-0.0284

Figura 2.9: Tabela - Pêndulo

2.3.2 Problema Mola-Massa Com Amortecimento

- Este problema é um exemplo clássico para sistemas mecânicos e são utilizados para estudos de oscilações.
- Sabemos que a mola tem um comportamento linear, o amortecedor tambem e as superficies em contacto não têm atrito.

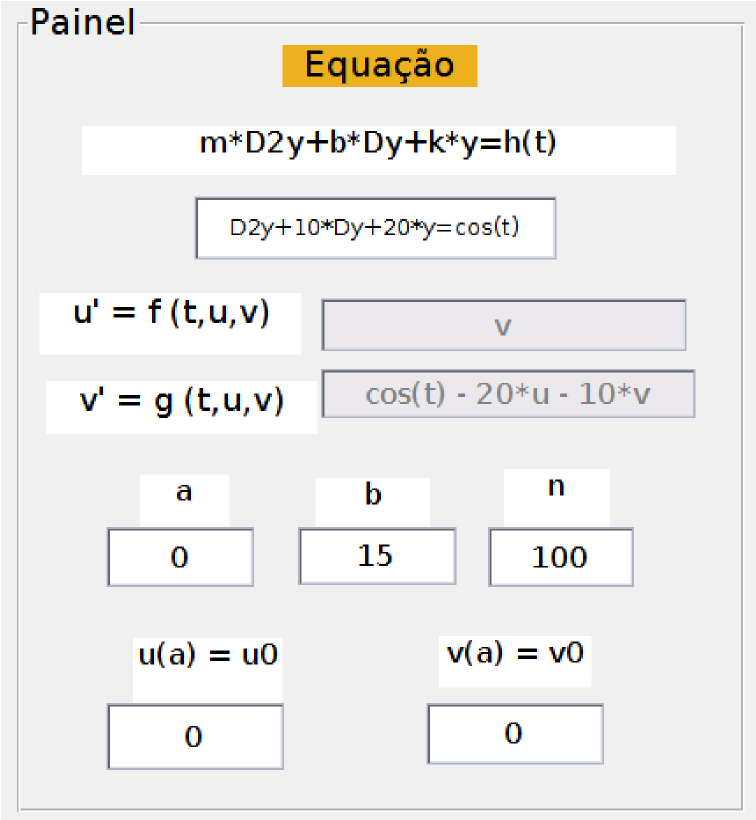

Figura 2.10: Parâmetros - Mola-Massa Com Amortecimento

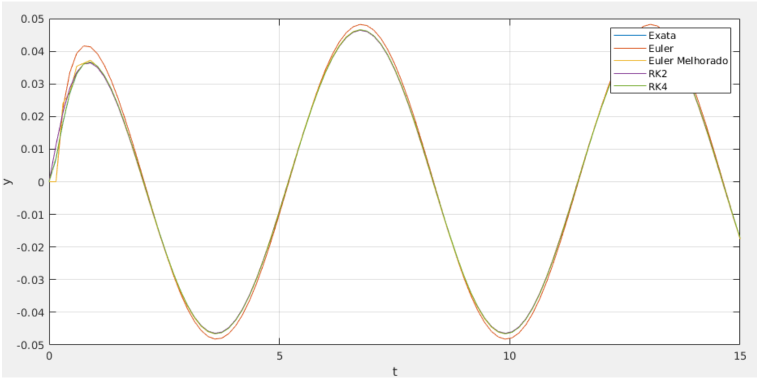

Figura 2.11: Gráfico - Mola-Massa Com Amortecimento

	t	Euler	Euler Melhorado	RK2	RK4	Exata	Erro Euler	Erro Euler Melhorado	Erro RK2	Erro RK4
1	0	0	0	0	0	6.9389e-...	6.9389e-...	6.9389e-18	6.9389e-...	6.9389e-...
2	0.1500	0	0	0.0113	0.0073	0.0070	0.0070	0.0070	0.0043	3.2227e-...
3	0.3000	0.0225	0.0241	0.0212	0.0182	0.0180	0.0045	0.0060	0.0032	2.2160e-...
4	0.4500	0.0335	0.0274	0.0287	0.0272	0.0271	0.0064	3.2845e-04	0.0016	1.1506e-...
5	0.6000	0.0394	0.0354	0.0336	0.0331	0.0330	0.0063	0.0024	5.5835e-...	5.4156e-...
6	0.7500	0.0416	0.0363	0.0360	0.0361	0.0361	0.0055	2.4935e-04	2.6926e-...	2.5095e-...
7	0.9000	0.0413	0.0373	0.0364	0.0367	0.0367	0.0046	5.4774e-04	2.9760e-...	1.2325e-...
8	1.0500	0.0392	0.0353	0.0350	0.0354	0.0354	0.0038	1.3994e-04	3.9443e-...	6.8138e-...
9	1.2000	0.0357	0.0324	0.0322	0.0326	0.0326	0.0031	1.1585e-04	4.0489e-...	4.2197e-...
10	1.3500	0.0310	0.0281	0.0281	0.0285	0.0285	0.0025	3.2986e-04	3.7696e-...	2.6559e-...

Figura 2.12: Tabela - Mola-Massa Com Amortecimento

2.3.3 Problema Mola-Massa Sem Amortecimento

- Neste caso, o processo será o mesmo mas com uma condição importante, nao existe amortecimento.

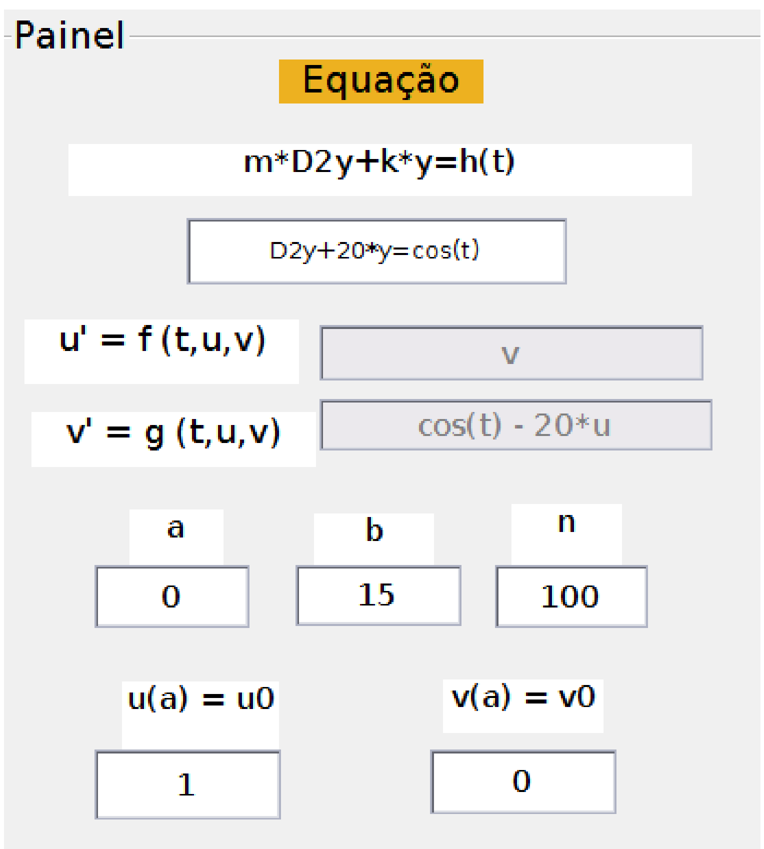

Figura 2.13: Parâmetros - Mola-Massa Sem Amortecimento

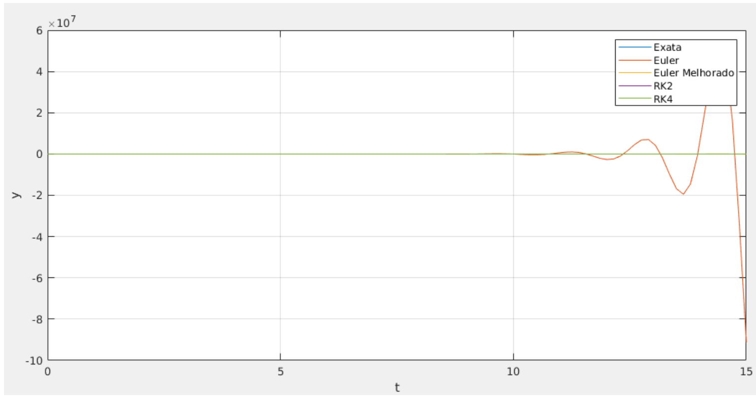

Figura 2.14: Gráfico - Mola-Massa Sem Amortecimento

	t	Euler	Euler Melhorado	RK2	RK4	Exata	Erro Euler	Erro Euler Melhorado	Erro RK2	Erro RK4
1	0	1	1	1	1	1	0	0	0	0
2	0.1500	1	1	0.7863	0.7942	0.7941	0.2059	0.2059	0.0079	1.1895e-...
3	0.3000	0.5725	0.5403	0.1928	0.2670	0.2655	0.3070	0.2748	0.0726	0.0015
4	0.4500	-0.2828	-0.2762	-0.5032	-0.3544	-0.3576	0.0748	0.0814	0.1456	0.0032
5	0.6000	-1.3741	-1.1479	-0.9598	-0.8025	-0.8062	0.5680	0.3418	0.1536	0.0036
6	0.7500	-2.3180	-1.6690	-0.9382	-0.8857	-0.8875	1.4305	0.7815	0.0507	0.0018
7	0.9000	-2.6250	-1.4987	-0.4273	-0.5705	-0.5685	2.0565	0.9303	0.1412	0.0020
8	1.0500	-1.8724	-0.5430	0.3392	0.0042	0.0104	1.8828	0.5534	0.3288	0.0062
9	1.2000	0.0755	0.9362	0.9875	0.5871	0.5955	0.5201	0.3407	0.3920	0.0085
10	1.3500	2.8771	2.3557	1.1837	0.9235	0.9304	1.9467	1.4253	0.2533	0.0069

Figura 2.15: Tabela - Mola-Massa Sem Amortecimento

2.3.4 Problema do Circuito Elétrico

- Podemos tambem determinar soluções de equações de diferencias de problemas como o problema do circuito elétrico, usando mais uma vez os métodos acima descritos..

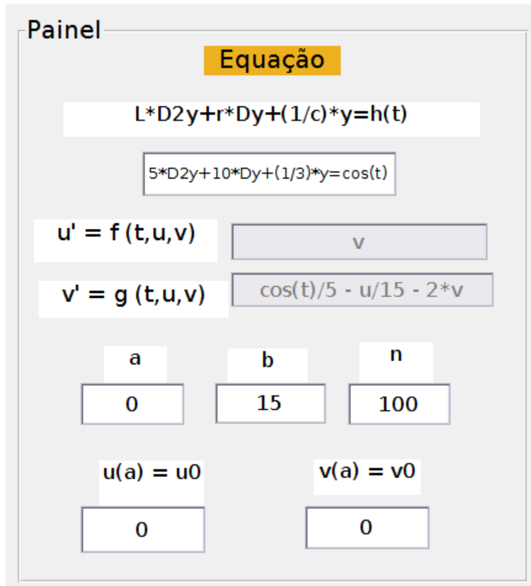

Figura 2.16: Parâmetros - Problema do Circuito Elétrico

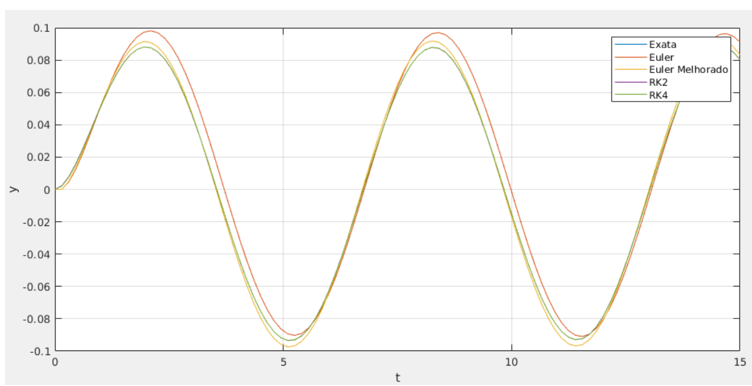

Figura 2.17: Gráfico - Problema do Circuito Elétrico

	t	Euler	Euler Melhorado	RK2	RK4	Exata	Erro Euler	Erro Euler Melhorado	Erro RK2	Erro RK4
1	0	0	0	0	0	-6.9389e...	6.9389e-...	6.9389e-18	6.9389e-...	6.9389e-...
2	0.1500	0	0	0.0023	0.0020	0.0020	0.0020	0.0020	2.1331e-...	1.0066e-...
3	0.3000	0.0045	0.0048	0.0077	0.0074	0.0074	0.0029	0.0026	3.2575e-...	1.5758e-...
4	0.4500	0.0121	0.0128	0.0154	0.0150	0.0150	0.0029	0.0023	3.7255e-...	1.8613e-...
5	0.6000	0.0217	0.0226	0.0245	0.0242	0.0242	0.0024	0.0015	3.7743e-...	1.9656e-...
6	0.7500	0.0325	0.0335	0.0344	0.0341	0.0341	0.0016	6.2528e-04	3.5625e-...	1.9554e-...
7	0.9000	0.0437	0.0445	0.0446	0.0442	0.0442	5.4271e-...	2.9298e-04	3.1949e-...	1.8740e-...
8	1.0500	0.0548	0.0552	0.0544	0.0541	0.0541	6.9594e-...	0.0011	2.7405e-...	1.7490e-...
9	1.2000	0.0653	0.0651	0.0635	0.0632	0.0632	0.0020	0.0019	2.2440e-...	1.5974e-...
10	1.3500	0.0748	0.0738	0.0715	0.0713	0.0713	0.0035	0.0025	1.7346e-...	1.4298e-...

Figura 2.18: Tabela - Problema do Circuito Elétrico

2.4 Conclusão

Com a elaboração deste trabalho, concluimos que os métodos numéricos funcionam tambem com sistemas de equações de equações diferenciais, no entanto para conseguirmos resolver esses sistemas, precisamos de adaptar os métodos uma vez que tanto o input como o output vão ser diferentes. Essa adaptação foi feita no MATLAB, no entanto no papel seria exatamente igual. Percebemos tambem que se tivermos uma equação diferencial **não linear**, não existe maneira de termos uma solução exata. Como não temos uma solução exata, tivemos de adaptar o nosso código para que se for introduzida uma equação diferencial não linear, não é mostrado o output da Solução Exata uma vez que o retorno dessa função é um array vazio.

DEIS Departamento de Engenharia Informática e de Sistemas