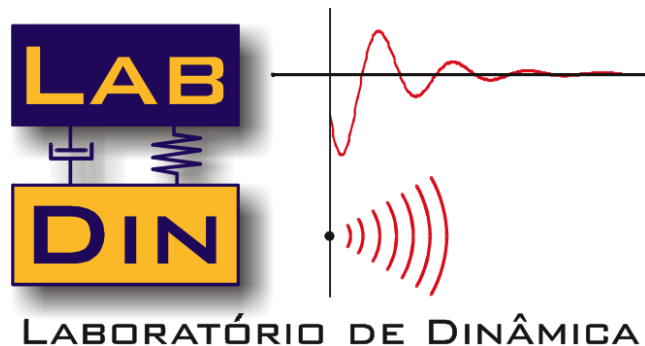


UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA



SEM 0530
PROBLEMAS DE ENGENHARIA
MECATRÔNICA II

Resolvendo EDOs no MATLAB

Objetivos

O objetivo desta aula é discutir a solução de EDOs e sistemas de EDOs usando o MATLAB

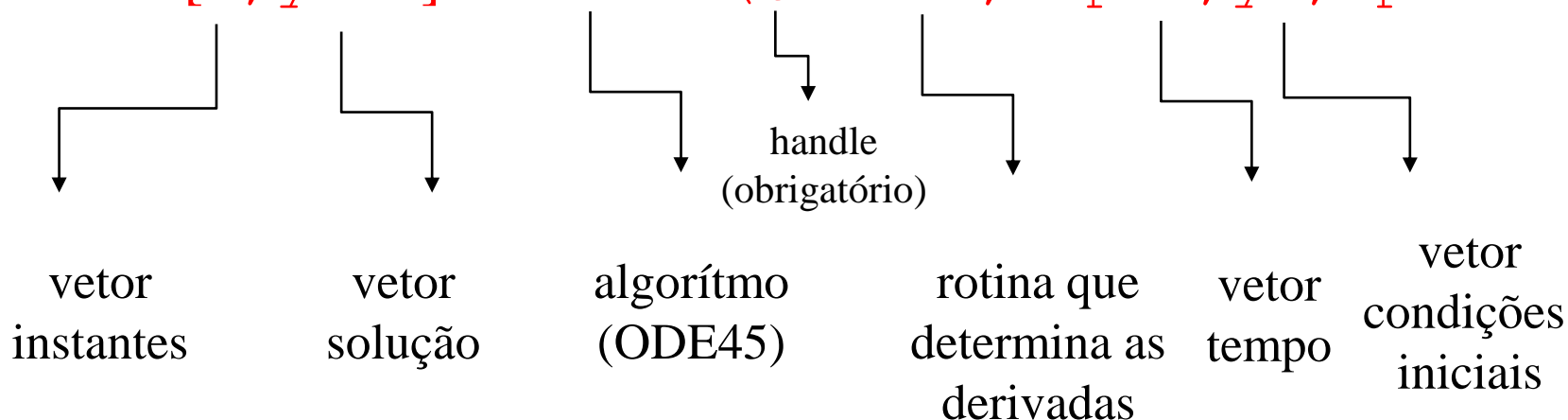
Bibliografia:

1 <https://www.mathworks.com>

EDOs no MATLAB

O MATLAB tem várias formas de resolvermos esta e outras EDOs. Vamos abordar uma das mais simples que realiza a solução numérica através do solver ODE45. Esta função emprega a seguinte sintaxe

```
>> [t, ysol] = solver(@deriv, tspan, y0, options)
```



Temos várias possibilidades. A mais simples compreende dois passos

1. Na janela de comandos defina o vetor de tempos e as condições iniciais
2. Crie uma **function** (**nome_arquivo.m**) separadamente que calcule as derivadas
3. Da janela de comandos chame a rotina do item 2 e resolva a EDO.

Criando uma Function no Matlab

Uma function ou sub-rotina é na verdade um programa destinado à execução de operações geralmente repetitivas e que é chamada tanto da janela de comandos ou de um outro programa no MATLAB.

Exemplo: Crie uma *function* que calcule $z = ax + by$ para vários valores de x e y

```
function z = plano(x,y)
    a = 2;
    b = 3;
    z = a*x + b*y
end
```

Agora da janela de comando (ou de outro programa) chame a rotina:

```
>> x = 2;
>> y = 3;
>> z = plano(x,y)

z =

    13
```

Cont. ...

Vamos a um exemplo. Queremos resolver numericamente

$$\frac{dy}{dt} - \alpha y + \gamma y^2 = 0 \quad y(0) = 10$$

Vamos reescrever a EDO da seguinte forma

$$\frac{dy}{dt} = \alpha y - \gamma y^2 \quad \begin{array}{l} \alpha = 2 \\ \gamma = 0.0001 \end{array}$$

Agora, vamos escrever uma rotina com o título deriv.m que vai calcular o lado direito da igualdade

```
function dydt = deriv(t,y)
    alpha=2;
    gamma=0.0001;
    dydt = alpha*y-gamma*y^2;
end
```

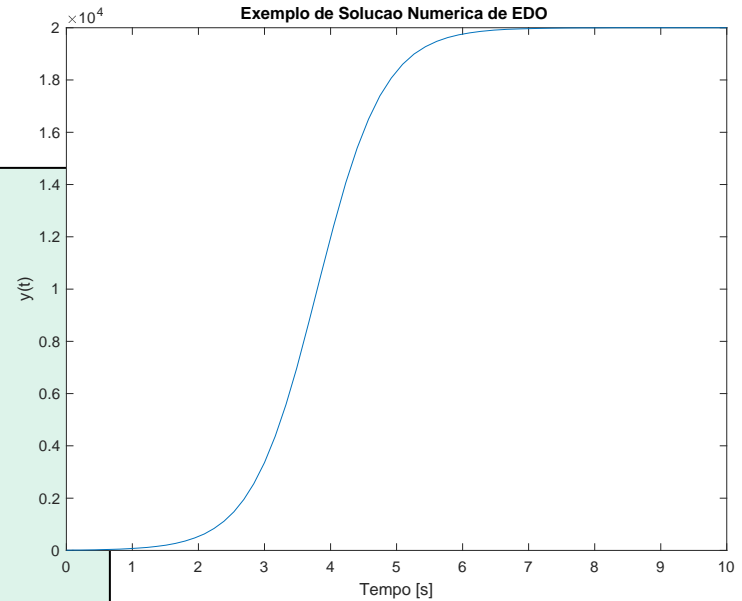
← salvar como

deriv.m

Cont. ...

Agora basta definir o vetor de tempo e as condições iniciais na janela de comando e chamar a rotina das derivadas

```
>> tspan = [0 10];  
>> y0 = 10 ;  
>> [t,y] = ode45(@deriv,tspan,y0);  
>> plot(t,y)  
>> grid on  
>> xlabel('Tempo [s]')  
>> ylabel('y(t)')  
>> title('Exemplo de Solucao Numerica de EDO')
```



Cont. ...

Vamos agora resolver um sistema de EDOs

$$\left\{ \begin{array}{l} \frac{dy_1}{dt} - y_2 = 0 \\ \frac{dy_2}{dt} - 1000(1 - y_1^2)y_2 + y_1 = 0 \end{array} \right. \quad y_1(0) = 0 \quad y_2(0) = 1$$

Inicialmente escrevemos

$$\left\{ \begin{array}{l} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = 1000(1 - y_1^2)y_2 - y_1 \end{array} \right.$$

Preciso de uma rotina
para calcular esta parte !

Cont. ...

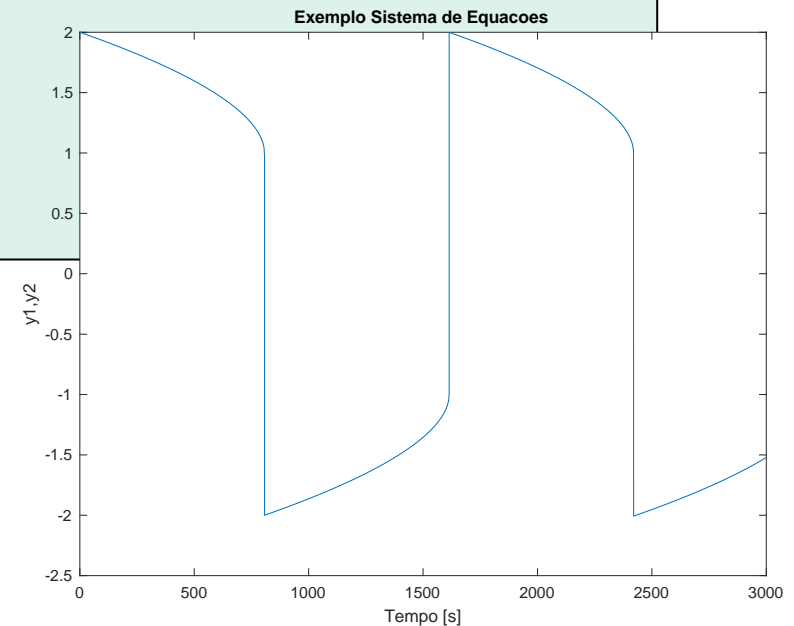
Então no editor do MATLAB abrimos um novo documento com o nome deriv1.m e escrevemos as linhas de comando para o cálculo das derivadas

salvar como deriv1.m

```
function dydt = deriv1(t,y)
    dydt = zeros(2,1); % cria um vetor coluna com zeros
    dydt(1) = y(2) ;
    dydt(2) = 1000*(1 - y(1)^2)*y(2) - y(1);
    % neste caso y(1) é y1 e y(2) é y2 e portanto dydt(1) será dy1/dt
    %e dydt(2) será dy2/dt
end
```

Agora ao invés de digitar os comandos diretamente na janela de comando vamos montar outro arquivo .m denominado sistema_ode.m e dele chamar a rotina deriv1.m


```
% Rotina matlab para solucao de um sistema de EDOs de
% primeira ordem
%
clear all
clc
t = [0 3000];
y1_0 = 2 ;
y2_0 = 0 ;
y0 = [y1_0 y2_0];
[t,y] = ode45(@deriv1,t,y0);
plot(t,y(:,1))
grid
xlabel('Tempo [s]')
ylabel('y1,y2')
title('Exemplo Sistema de EDOs')
```



FIIM

Bom Estudo !

