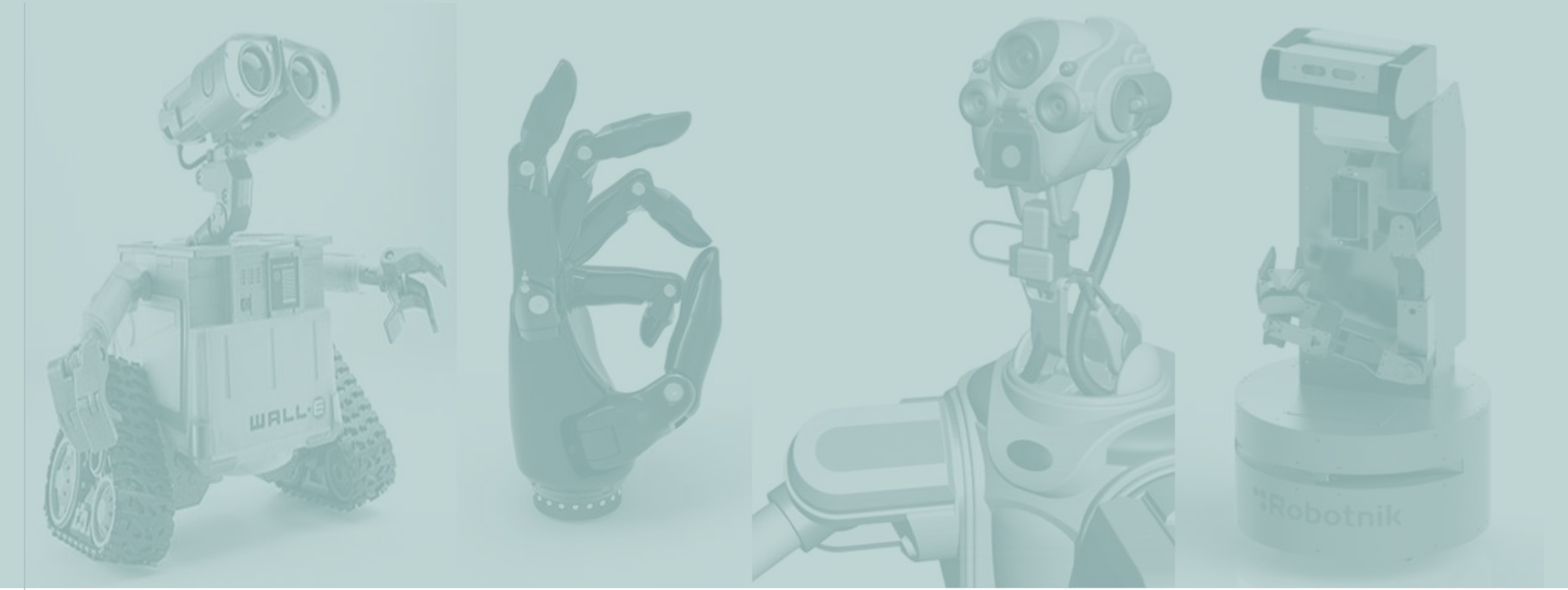


LABORATÓRIO DE SISTEMAS DINÂMICOS AULA 02: GRÁFICOS E PROGRAMAÇÃO

Larissa Driemeier
Marcilio Alves

NOSSA AGENDA

#	Data	Tópico
1	21/02	Introdução ao modelamento e uso do software
2	06/03	Introdução à gráficos e programação em MatLab
3	20/03	Resolução de Equações Diferenciais - Sistemas Lineares e Não Lineares
4	03/04	Transformada de Laplace e Funções de Transferência
5	24/04	Projeto
6	15/05	Diagrama de Blocos e Simulink
7	29/05	Análise de Sistemas de Primeira Ordem
8	19/06	Análise de Sistemas de Segunda Ordem

NOTA DE LABORATÓRIO

Atividades propostas em aula

1 ponto

Atividades em sala.

Parte 01 – 1,0

Parte 02 – 1,0

2 pontos

Projeto.

GRÁFICOS 2D

https://www.mathworks.com/examples/matlab/category/graphics

```
%% Plot 2D
x = -15:0.1:15;
y = sin(x)./x;
plot(x,y)
```

```
figure(1);
r = 50; L = 110; h = 40;
th = @(t) 80*(t+exp(-t)-1);
y = @(t) r*sin(th(t)) + sqrt(L^2-(r*cos(th(t))).^2) + h;
f = @(t) y(t) - 195;
fplot(y,[0 2])
figure(2)
fplot(f,[0 2])
```

Anonimous functions

```
x=-1:0.1:1;   % Cria vetor 'x': valores entre 1 e -1 espacados de 0.1
y=x.^2;       % Calcula y
z=x.^3;       % Calcula Z
figure(9);
plot(x,y,'r*',x,z,'b:')          % Traca os dois graficos - x vs y e x vs z
xlabel('Valor de x')             % Nomeia o eixo x
ylabel('y e z')                  % Nomeia o eixo y
title('Graficos sobrepostos')    % Atribui um titulo ao grafico
legend('y','z')                  % legenda
grid                             % Ativa as linhas de grade da janela
```

```
figure(10)
plot(x,y,'LineWidth',1) % Traca x vs y
hold on
plot(x,z,'LineWidth',2) % Traca x vs z no mesmo grafico
hold off
```

SUBPLOT

```
K = [1:100].^2;
Y = K.^(-0.4);
subplot(3,1,1);
plot(K, Y);
grid on
subplot(3,1,2);
semilogx(K, Y);
grid on
subplot(3,1,3);
loglog(K, Y);
grid on
```

subplot(m,n,p), or subplot(mnp), breaks the Figure window into an m-by-n matrix of small axes, selects the p-th axes for the current plot

COMO CUSTOMIZAR SEU GRÁFICO

Tabela 11, pág 31.

```
plot(K, Y,'rs-.');
```

Cores de linhas		Marcadores de ponto		Tipos de linhas	
Símb.	Cor	Símb.	Marcador	Símb.	Tipo
y	amarelo	.	ponto	—	sólida
m	magenta	o	círculo	:	pontilhada
c	azul-claro	x	x	—.	traço-ponto
r	vermelho	+	+	——	tracejada
g	verde	*	asterisco		
b	azul escuro	s	quadrado		
w	branco	d	losango		
k	preto	v	triângulo		
		^	triângulo		
		<	triângulo para esquerda		
		>	triângulo para direita		
		p	pentagrama		
		h	hexagrama		

Tabela 11: Códigos para cores, marcadores e tipos de linha em gráficos no MATLAB.

GRÁFICOS 3D

```
t = linspace(0,6*pi,100);
plot3(sin(t),cos(t),t,’rs’);
xlabel('seno(t)');
ylabel('cosseno(t)');
zlabel('z = t');
title('Grafico de helice');
grid on;
```

CRIAÇÃO DE GRADE

Aula anterior:
impõe **a quantidade de valores**

```
x = linspace(0,2,3) % Geracao de valores para 'x',
y = linspace(3,5,2) % Geracao de valores para 'y'
[X,Y] = meshgrid(x,y) % Criacao da matriz da malha 'xy'
Z=X.*Y
mesh(X,Y,Z)
```

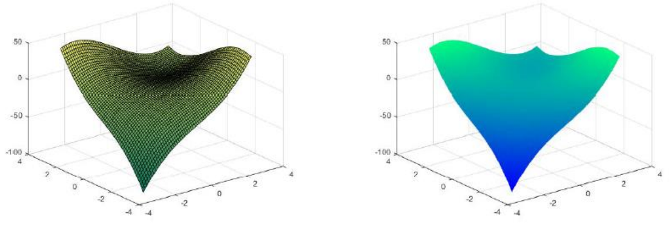
MESH VS SURF

```
x = -5:0.5:5;              % Definicao da malha de pontos no eixo 'x'
y = x;                     % Repeticao da malha do eixo x para o eixo 'y'
[X,Y] = meshgrid(x,y);     % Criacao da matriz da malha 'xy'
Z = X.^ 2 + Y.^ 2;         % Calculo da funcao z = f(x,y)
mesh(X,Y,Z)                % Tracado do grafico da funcao 'z'
```

```
[X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
surf(X,Y,Z)
colormap hsv % define o mapa de cores
colorbar % para colocar a barra de cores
```

Para eliminar as linhas de grade da superfície:

```
surf(X,Y,Z,'EdgeColor','none')
```



(a) *Summer.* (b) *Winter, sem linhas.*

Figura 18: Exemplo de uso do mapa de cores.

Tabela 12, pág 36.

Função	Mapa de cores
hsv	Escala com cores saturadas
hot	Preto-vermelho-amarelo-branco
gray	Escala linear de tons de cinza
bone	Escala de tons de cinza levemente azulados
copper	Escala linear de tons acobreados
pink	Tons pastéis de rosa
white	Mapa de cores totalmente branco
flag	Vermelho, branco, azul e preto alternados
jet	Uma variante do mapa hsv
prism	Mapa de cores denominado *prisma*
cool	Tons de ciano e magenta.
lines	Mapa de cores que usa as mesmas cores do comando `plot`
colorcube	Mapa de cores denominado *cubo colorido*
summer	Tons de amarelo e verde
autumn	Tons de vermelho e amarelo
winter	Tons de azul e verde
spring	Tons de magenta e amarelo

Tabela 12: Mapas de cores utilizados pelo MATLAB.

PROGRAMAÇÃO

HOME → New → Script

HOME → New → Function

Arquivos de script	Arquivos de funções
Não aceitam argumentos nem retornam valores ao *workspace*.	Aceitam argumentos e retornam valores ao *workspace*.
Trabalham com as variáveis definidas no *workspace*.	Trabalham com variáveis definidas localmente ao arquivo.
Principal aplicação: automatização de comandos que precisam ser executados em uma certa sequência.	Principal aplicação: adaptação da linguagem MATLAB a qualquer situação de programação necessária.

Tabela 14: Características das formas de programação MATLAB.

EDITOR → Run ou F5

SCRIPT – SALVE O ARQUIVO FREEFALL.M

```
% Script 1 - script file to compute the
%velocity of the free-falling bungee jumper for
%the case where the initial velocity is zero.
clear all %limpa toda a memoria
clc %limpa a janela de comando
g = 9.81; m = 68.1; t = 12; cd = 0.25;
% g = gravity (m/s^2)
% m = mass (kg)
% t = time (s)
% cd = second-order drag coefficient (kg/m)
v = sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t)
```

```
» lookfor script
freefall    Script 1 - script file to compute the
```

```
» help freefall
Script 1 - script file to compute the
velocity of the free-falling bungee jumper for
the case where the initial velocity is zero.
```

FUNCTION

VARIÁVEL DE SAÍDA

NOME DA SUA FUNCTION (salve para ver que o MatLab automaticamente dará esse nome)

VARIÁVEIS DE ENTRADA

```
function v = freefall(t, m, cd)
% freefall: bungee velocity with second-order drag
% v=freefall(t,m,cd) computes the free-fall velocity
% of an object with second-order drag
% input: t=time(s), m=mass(kg), cd=2nd drag coeff(kg/m)
% output: v = downward velocity (m/s)
g = 9.81; % acceleration of gravity
v = sqrt(g * m / cd)*tanh(sqrt(g * cd / m) * t);
```

texto usado para informar o usuário sobre a função (esta parte pode ser chamada digitando `help freefall` na janela de comandos)

Exemplo de execução da função `freefall`,

```
» freefall(12,68.1,0.25)
ans =
        50.6175
```

Tente explicitar a variável g,

```
» g
Undefined function or variable 'g'.
```

Cada função trabalha com variáveis locais, isoladas do espaço de memória do workspace.

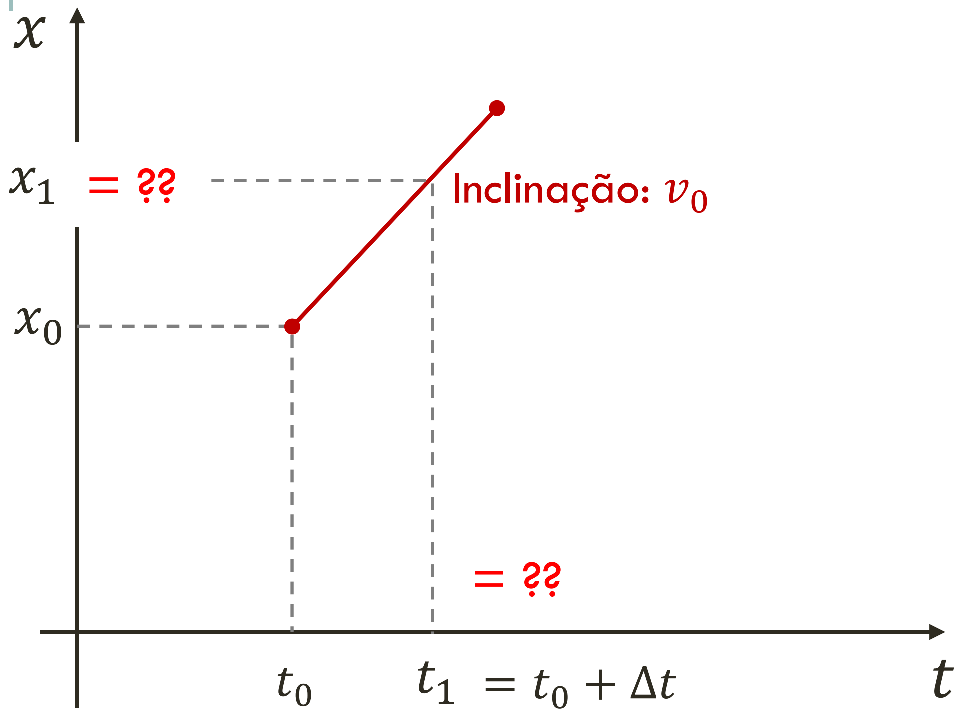
```
% Script 2 - script file to compute the
%velocity of the free-falling bungee jumper for
%the case where the initial velocity is zero.
clear all %limpa toda a memoria
clc %limpa a janela de comando
 g = 9.81; m = 68.1; t = 12; cd = 0.25;
% g = gravity (m/s^2)
% m = mass (kg)
% t = time (s)
% cd = second-order drag coefficient (kg/m)
v = freefall(t, m, cd)
```

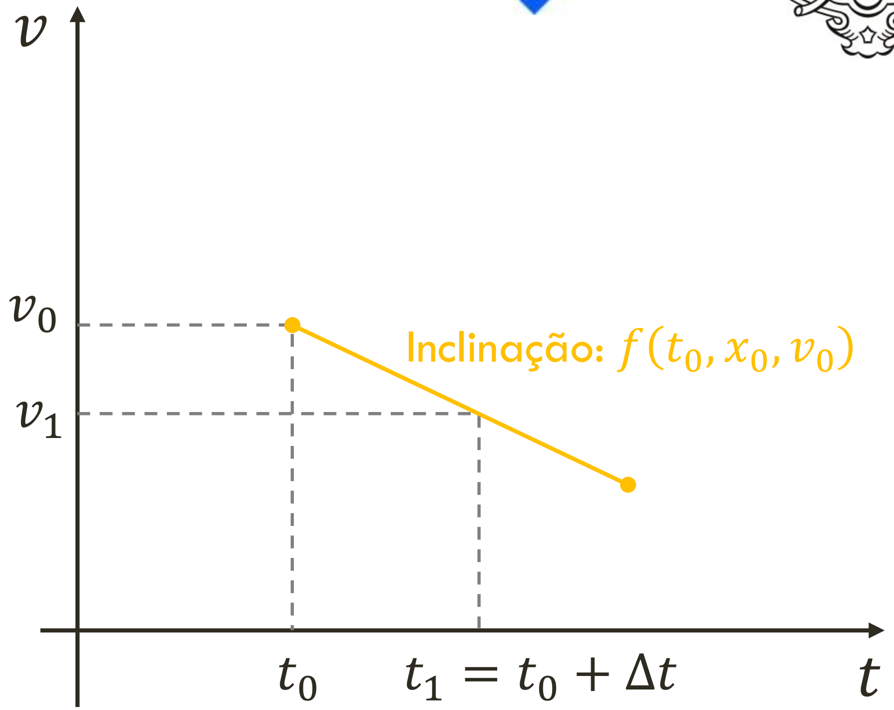
CRIE UM SCRIPT QUE PLOTE A VELOCIDADE DE QUEDA LIVRE AO LONGO DO TEMPO

```
clear all;clc;
m=66;cd=0.25;
```

```
plot(t,v)
```

SOLUÇÃO DE EQUAÇÕES DIFERENCIAIS: MÉTODO DE EULER





$$t_1 = t_0 + \Delta t$$
$$x_1 = x_0 + \Delta t v_0$$
$$v_1 = v_0 + \Delta t\, f(t_0, x_0, v_0)$$

EXEMPLO DE MÉTODO DE EULER

$$\dot{y}(t) = -6y(t)$$

Problema

$$y(t) = e^{-6t}$$

Solução Exata

Discretizando no domínio do tempo

$$\dot{y}(t) = \frac{dy(t)}{dt} = \frac{\Delta y(t)}{\Delta t}$$

$$\frac{\Delta y}{\Delta t} = \frac{y(t+\Delta t) - y(t)}{\Delta t} = -6y(t)$$

$$\frac{y(t+\Delta t) - y(t)}{\Delta t} = \frac{y_{n+1} - y_n}{\Delta t} = -6y_n$$

$$y_{n+1} = y_n + \Delta t(-6y_n)$$

SUA VEZ!!!

$$\dot{y}(t) = -6y(t)$$

Problema

$$y(t) = e^{-6t}$$

Solução Exata

$$y_{n+1} = y_n + \Delta t(-6y_n)$$

Implementar o método de Euler no problema anterior com intervalos de 0.25, 0.2, 0.1 e 0.01 além da solução exata, entre $t=0$ e $t=1$. Considerar $y(0)=1$.

```
close all; clear all; clc
figura_handler = figure;
a=0;                 % instante inicial
b=1;                 % instante final
y_init=1;            % condicao inicial
yp = @(i) -6*i;       % equacao diferencial
y(1)=y_init;

%% intervalo 0.1
dt=0.1;              % incremento de tempo
t=a:dt:b;            % Vetor tempo

for n=1:length(t)-1
```

Implementar você!!!

```
end
plot(t,y,'Linewidth',3)
hold on
clear t y;
y(1)=y_init;

%% intervalo 0.01
```

Continuar implementando para outros dt.

Implementar no fim para ajustar legenda e títulos de eixos, deixar fundo branco e SALVAR EM PNG!

```
%% Arrumando plot
grid on
legend({'dt=0.10', 'dt=0.01','Exata'},'FontSize',12)
xlabel('t','FontSize',20)
a=ylabel('y(t)','FontSize',20);
set(figura_handler,'Color',[1 1 1])
print(figura_handler,'-dpng')
```

SUA VEZ!!!

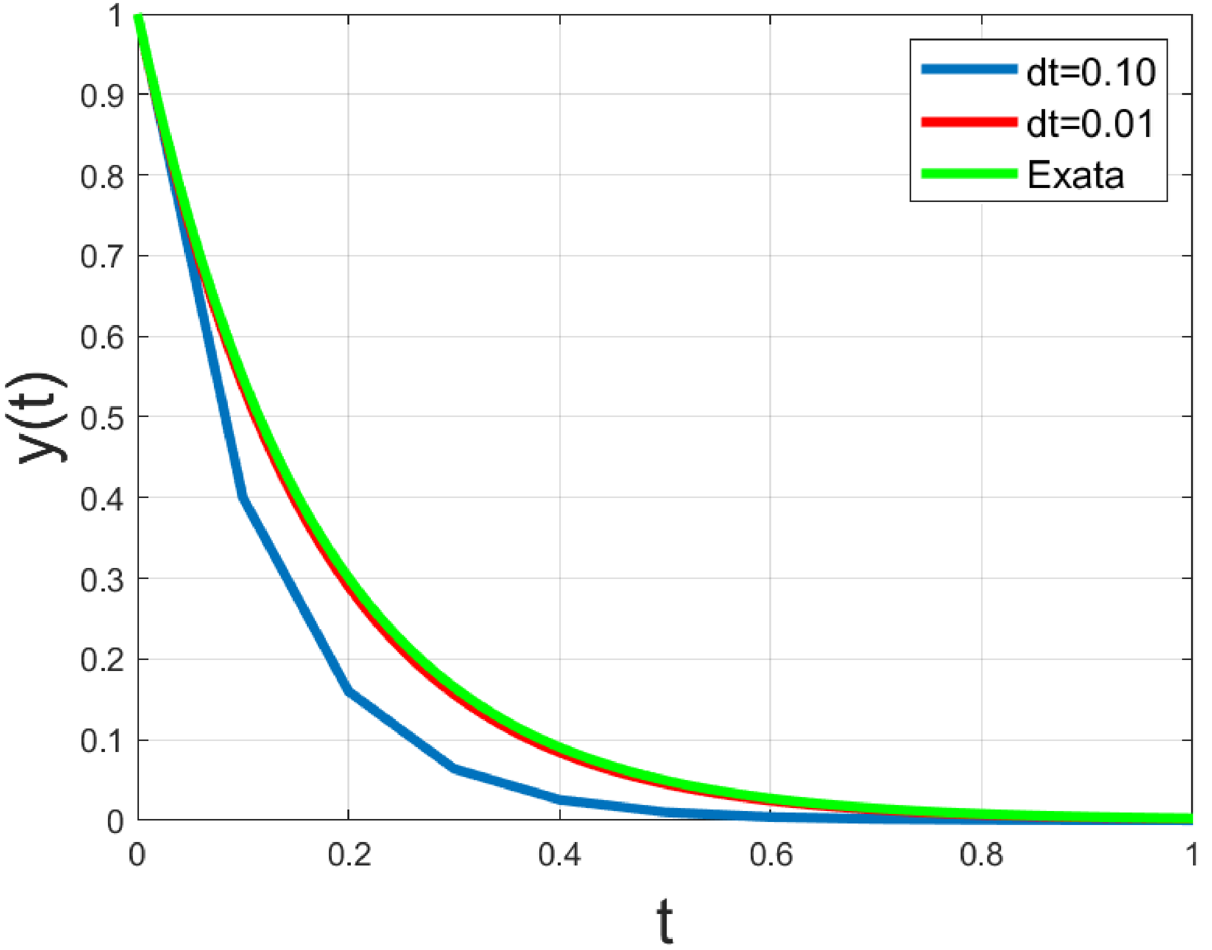
$$\dot{y}(t) = -6y(t)$$

Problema

$$y(t) = e^{-6t}$$

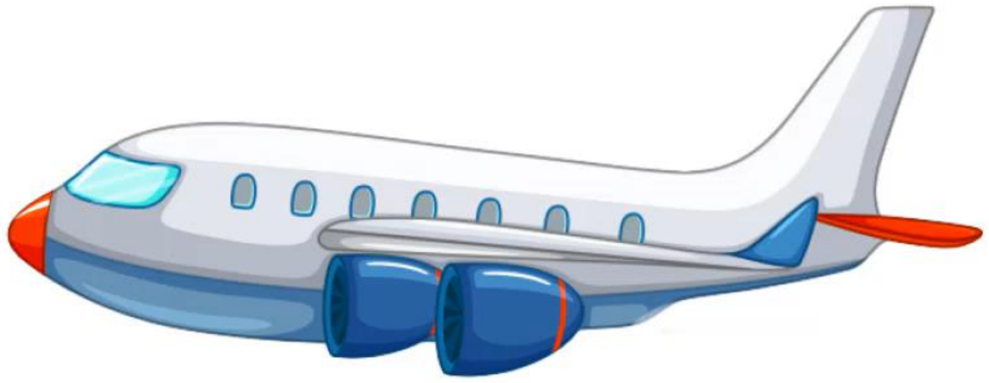
Solução Exata

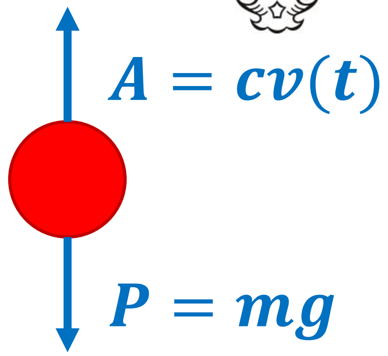
$$y_{n+1} = y_n + \Delta t(-6y_n)$$



EXEMPLO DE SOLUÇÃO DE SISTEMA DE 1ª ORDEM

Queda livre





$$mg - cv(t) = ma(t)$$

$$mg - cv(t) = m\dot{v}(t)$$

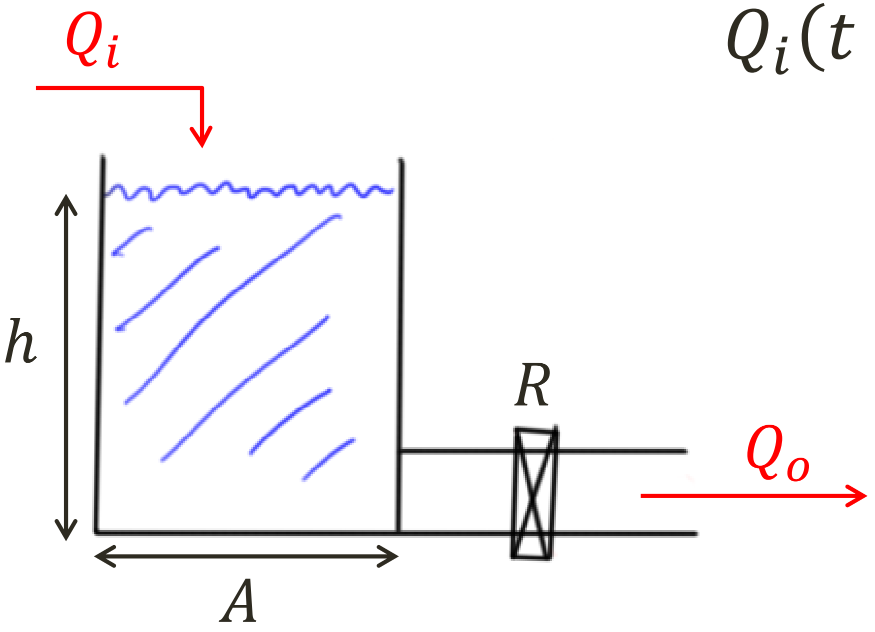
$$m\dot{v}(t) + cv(t) = mg$$

Resolver para 100s com para $m = 50, \quad c = 0.25, \quad v_0 = 0.$

EXERCÍCIOS PARA ENTREGAR

Tanque de Água

Resolver para 10s com para $Q_i(t < 5) = t$ e $Q_i(t \geq 5) = 5, h_0 = 10, R = 5 \cdot 10^3, A = 8$ e o fluido sendo a água.



$\dot{V}$

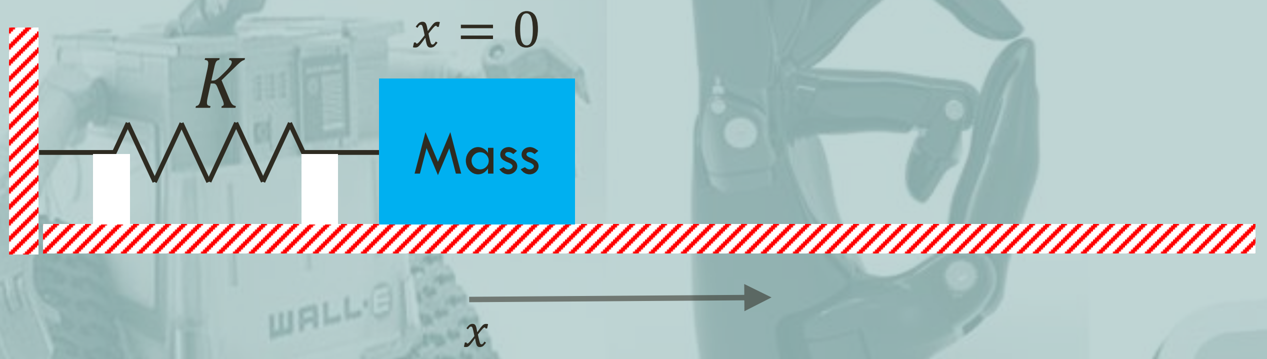
$$V = Ah$$
$$dV = dA * h + A * dh \ \rightarrow A = cte \ \rightarrow dA = 0$$
$$dV = \ Adh \ \rightarrow \dot{V} = A\dot{h}$$

$$\dot{V} = Q_i - Q_o = A\dot{h}$$

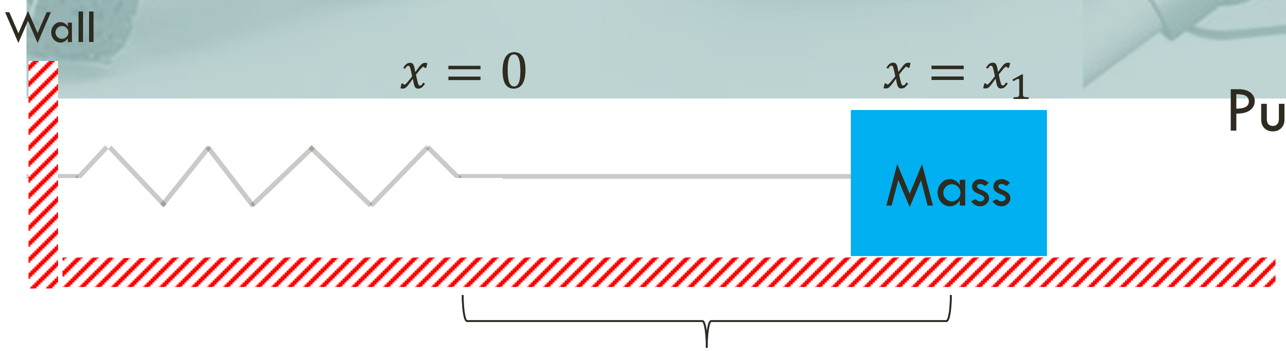
$$Q_o = \frac{Pressão}{Resistência} = \frac{\rho g h}{R}$$

Exemplo de Solução de Sistema de 2ª ordem

Massa Mola



A massa está livre sobre a superfície



$$\Delta x = x_1 - 0$$

Puxamos a massa e a soltamos em $x = x_1$

Condições iniciais @ $t = 0$ s $\begin{cases} x(0) = x_1 \\ \dot{x}(0) = 0 \end{cases}$

Exemplo de Solução de Sistema de 2ª ordem

Massa Mola

$$\sum F_x = m\ddot{x}(t) \rightarrow m\ddot{x}(t) = -kx(t) \rightarrow \boxed{EDO:\ \ddot{x}(t) + \frac{k}{m}x(t) = 0}$$

$$\ddot{x}(t) + \frac{k}{m}x(t) = 0 \xrightarrow{\text{Solução Geral}} x(t) = A\ \cos\left(\sqrt{\frac{k}{m}}\ t\right) + B\ \sin\left(\sqrt{\frac{k}{m}}\ t\right)$$

$$\text{Condições Iniciais} \begin{cases} x(0) = x_1 \\ \dot{x}(0) = 0 \end{cases} \xrightarrow{A = x_1, B = 0} \boxed{x(t) = x_1 \cos\left(\sqrt{\frac{k}{m}}\ t\right) \quad \textbf{Solução Exata!!!}}$$

Exemplo de Solução de Sistema de 2ª ordem

Massa Mola

Aplicando o método de Euler para resolver a EDO $\ddot{x}(t) = -\frac{k}{m}x(t)$

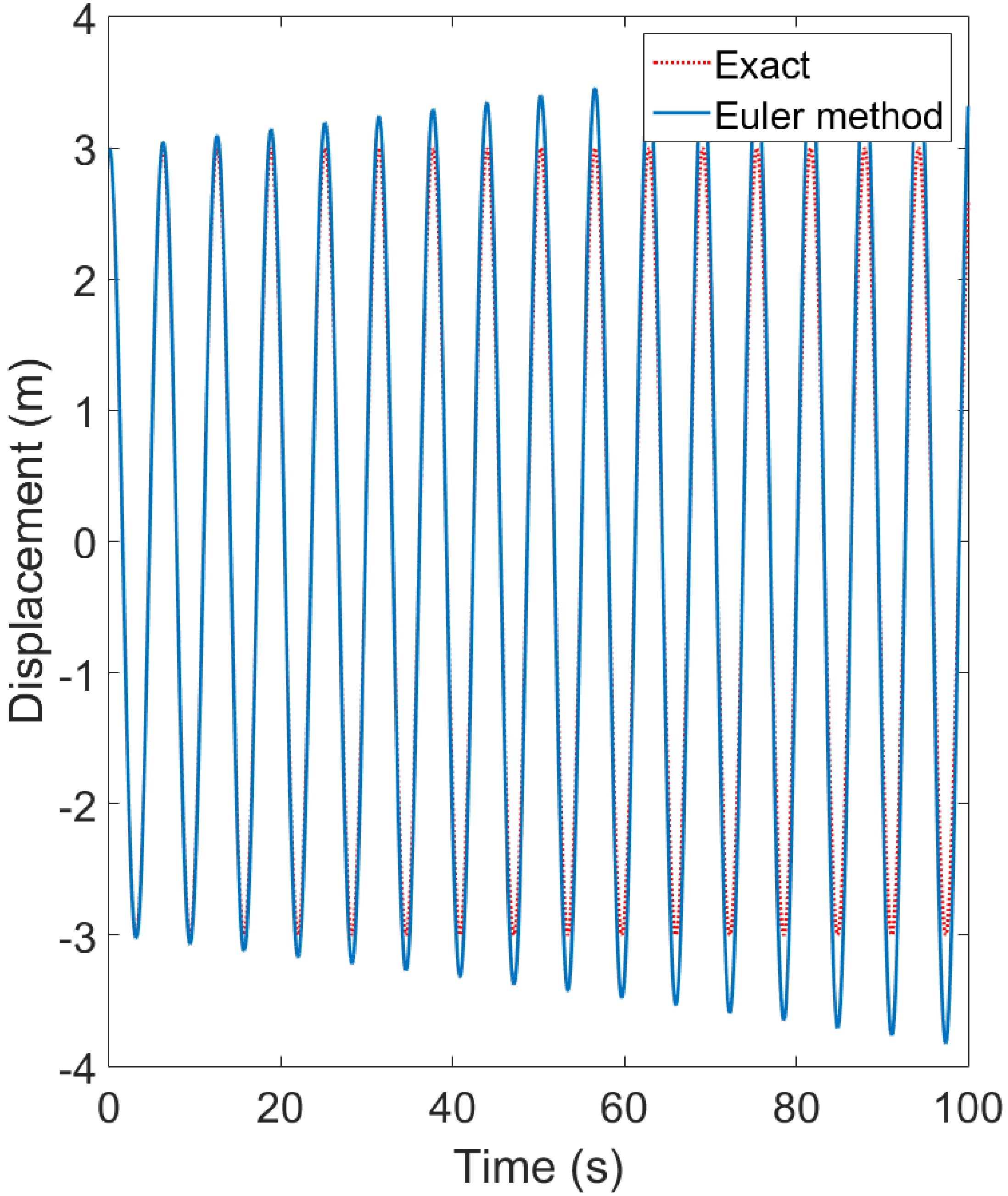
$$\ddot{x}(t) = \frac{\Delta\dot{x}\,(t)}{\Delta t} = \frac{\dot{x}(t+\Delta t) - \dot{x}(t)}{\Delta t} = \frac{\dot{x}_{n+1} - \dot{x}_n}{\Delta t} = -\frac{k}{m}x(t)$$

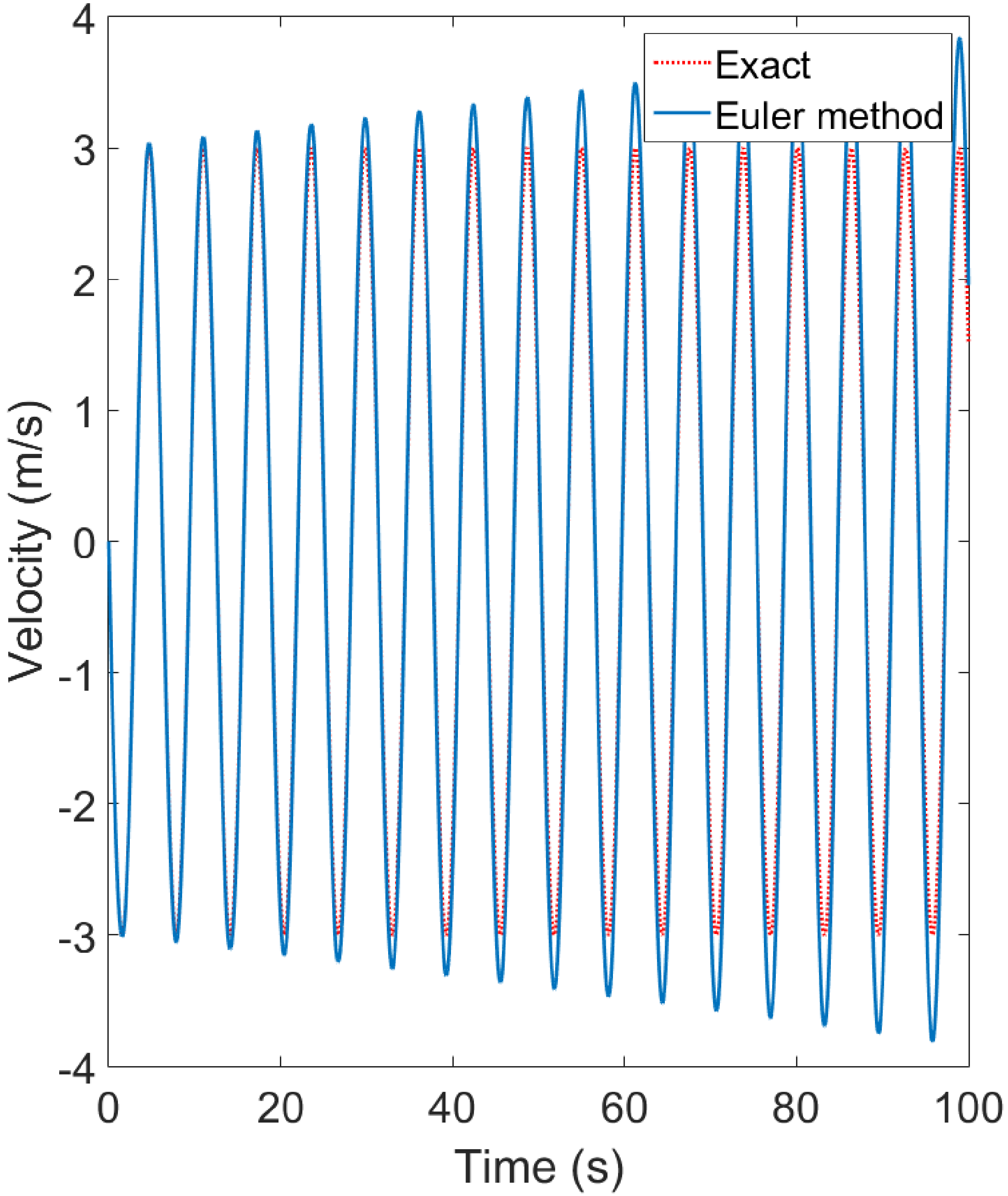
Chegando em:

$$\boxed{\begin{aligned} x_{n+1} &= x_n + \dot{x}_n\,\Delta t \\ \dot{x}_{n+1} &= \dot{x}_n + \left(-\frac{k}{m}\right) x_n\,\Delta t \end{aligned}}$$

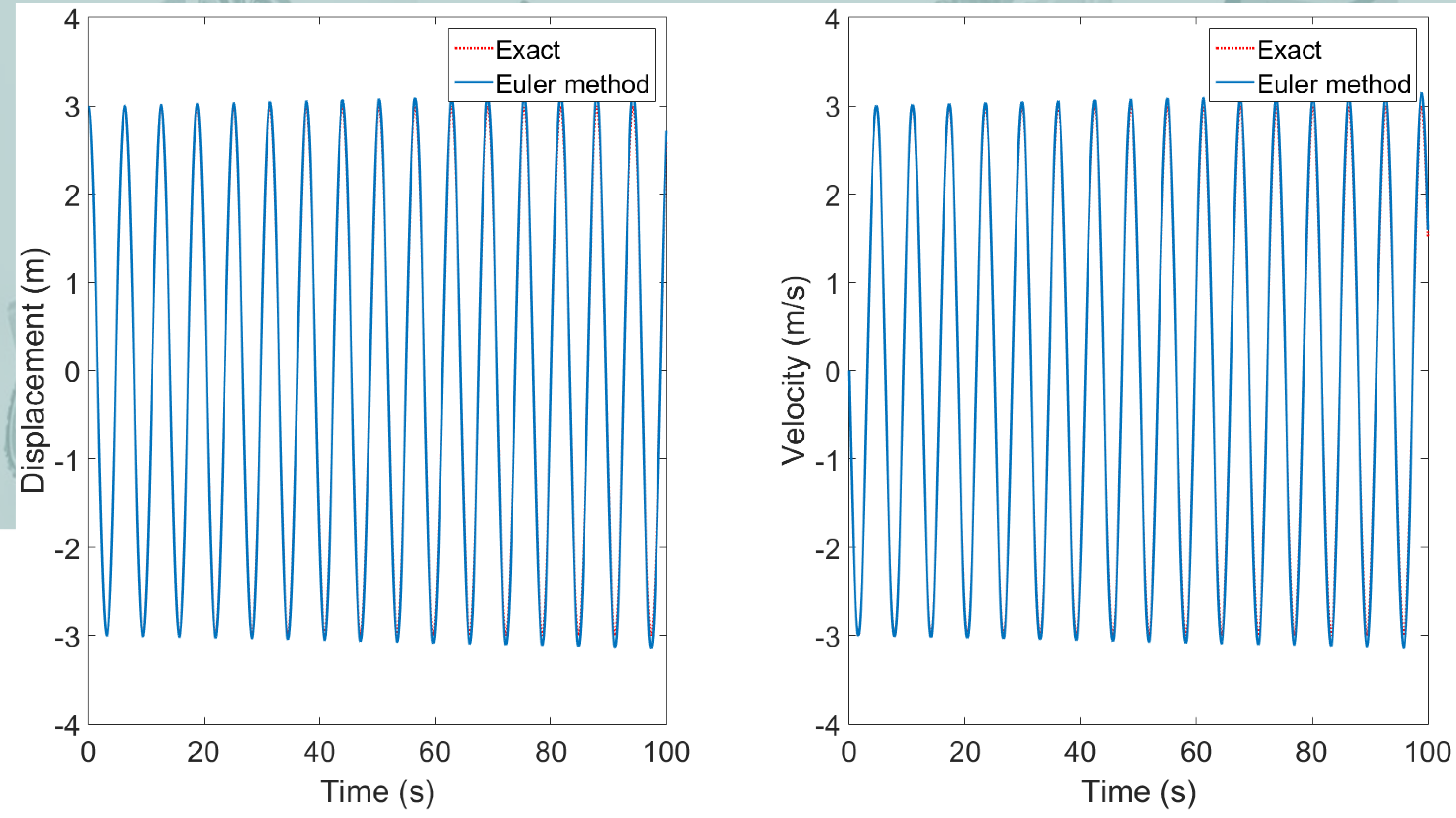
Resolver para 100s com para $k = 1$, $m = 1$, $v_0 = 0$ e $x_0 = 3$.

```
dt = 0.005; % Time interval
```





```
dt = 0.001; % Time interval
```



EXERCÍCIO PARA ENTREGA

Equações de movimento para o sistema ao lado

$T_p \rightarrow Energia\ cinética\ do\ pêndulo$ $\quad V_p \rightarrow Energia\ potêncial\ do\ pêndulo$

$T_b \rightarrow Energia\ cinética\ do\ bloco$ $\quad V_b \rightarrow Energia\ potêncial\ do\ bloco$

$$T = T_p + T_b = [Ml^2 + I + m(L+x)^2]\frac{\dot{\theta}^2}{2} + \frac{m\dot{x}^2}{2}$$

$$V = V_p + V_b = [Ml + m(L+x)]g \cdot \cos\theta + \frac{kx^2}{2}$$

$$Q_1 = 0$$
$$Q_2 = -c\dot{x}$$

Resolvendo:

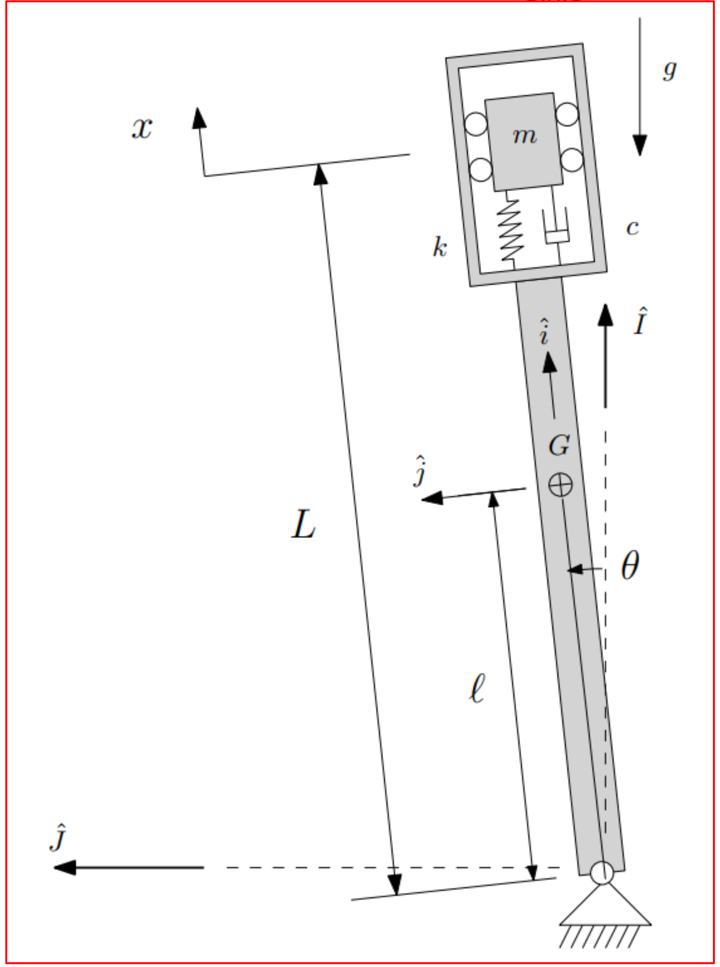
$$\frac{d}{dt}\frac{\partial(T-V)}{\partial\dot{\theta}} - \frac{\partial(T-V)}{\partial\theta} = 0$$

$$\frac{d}{dt}\frac{\partial(T-V)}{\partial\dot{x}} - \frac{\partial(T-V)}{\partial x} = -cx$$

Obtemos:

$$\ddot{\theta} = \frac{[Ml + m(L+x)]g \cdot \sin\theta - 2m(L+x)\dot{x}\dot{\theta}}{[Ml^2 + I + m(L+x)^2]}$$

$$\ddot{x} = \frac{[m(L+x)\dot{\theta}^2 - c\dot{x} - kx - mg \cdot \cos\theta]}{m}$$



1. Resolva estas equações usando o MÉTODO DE EULER com a condição inicial theta(0)=45° e x(0)=1

2. adote

L=1; l=L/2; m=1; k=1000; g=10; c=0.05;

I=10000;M=1;m=0.1;th(1)=pi/4;x(1)=0.1;a=0;b=10;xd(1)=0;thd(1)=0;

3. Plote os resultados e escreva um pequeno texto com seus comentários

4. Submeta um arquivo pdf com sua solução