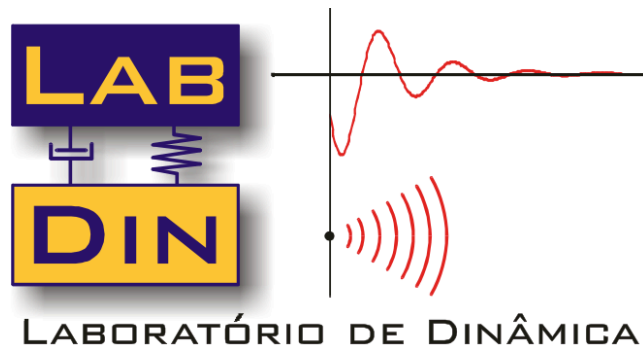


UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA



SEM 0530
PROBLEMAS DE ENGENHARIA MECATRÔNICA II

MATLAB
Controle de Fluxo de Variáveis
Introdutório de Gráficos

Objetivos

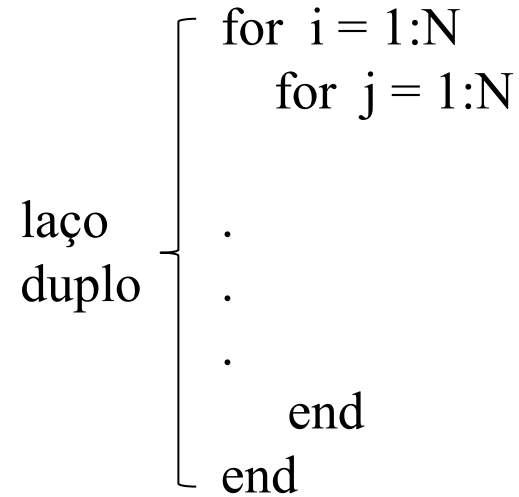
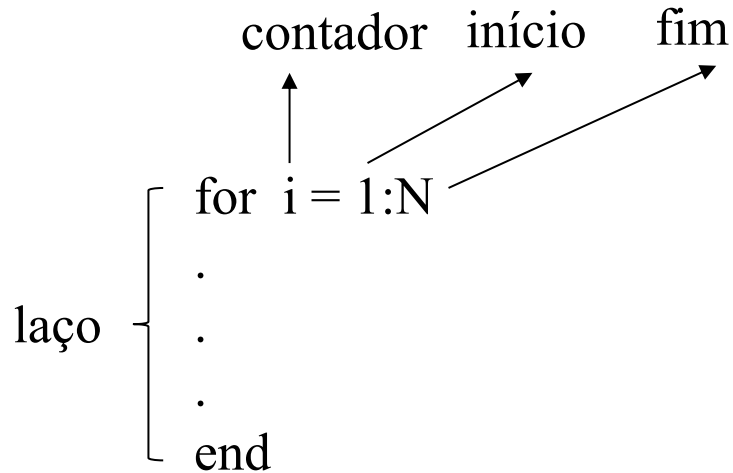
- Apresentar formas de controle de fluxo para variáveis
- Resolver expressões numéricas
- Introduzir ferramentas para construção de gráficos

Bibliografia:

- 1 <https://www.mathworks.com>
- 2 Notas de aula

Controle de Fluxo de Operações Aritméticas

- Loops de **FOR** : Destinado à repetição sequencial de operações. Estrutura



```
>> for i = 1:10
x(i) = i;
end
>> x

x =

     1     2     3     4     5     6     7     8     9    10

>>
```

```
>> for i = 1:2
for j = 1:2
a(i,j) = -1;
end
end
>> a

a =

    -1    -1
    -1    -1
```

Cont. ...

- Loops de **WHILE** : Repetição enquanto condição se mantém

laço { while *expressão*
·
·
·
end

```
>> a_contador = 0 ;
while a_contador < 3
a_contador = a_contador + 1;
disp('O contador é igual a:'),disp(a_contador);
end
O contador é igual a:
1

O contador é igual a:
2

O contador é igual a:
3

>>
```

```
>> A = zeros(5,1);
>> A(2) = 1;
>> m = 1

m =
1

>> while A(m) == 0
m = m + 1;
end
>> m

m =
2
```

Diferença entre FOR e WHILE:
O comando FOR executa um número fixo de vezes enquanto o WHILE executa tantas vezes até a condição não ser mais válida !

Cont. ...

- Condição **if-else-end** : Executar diferentes sequências

laço

```
if expressão 1
.
. % executa se expressão 1 for verdade
.
elseif expressão 2
.
.% executa se expressão 1 for falsa e 2 verdade
.
else
expressão 3
% executa se expressões 1 e 2 forem falsas
end
```

laço

```
if expressão 1
.
.
.
else
expressão 2
end
```

```
>> if 1<2
disp('primeira condição')
elseif 1<3
disp('Segunda Condição')
end
primeira condição
```

Cont. ...

- Condição **switch-case-otherwise** : muito parecido com if-elseif-else

laço {

```
switch condição
case 1
expressão 2
case 2
expressão 2
.
.
.
otherwise
expressão final
end
```

```
>> num = 3;
>> switch num
case 1
disp('Janeiro')
case 2
disp('Fevereiro')
case 3
disp('Março')
case 4
disp('Abril')
otherwise
disp('Não encontrei !')
end
Março
>>
```

Construindo Gráficos

Inicialmente gráficos bi-dimensionais (x,y)

| Função | Descrição |
|--------------------|-----------------------------------|
| figure | Cria uma nova janela para gráfico |
| plot(x,y) | Constrói o gráfico de y versus x |
| figure(n) | Seleciona a janela do gráfico n |
| title('sentença') | Inserir um título |
| legend('sentença') | Inserir legendas no gráfico |
| xlabel('sentença') | Inserir título eixo abscissa |
| ylabel('sentença') | Inserir título eixo ordenada |
| grid on/off | Inserir (remover) linhas de grid |
| hold on/off | Mantém um gráfico na janela |
| subplot(m,n,p) | Cria múltiplos gráficos |

Vejamos alguns exemplos a seguir

Cont. ...

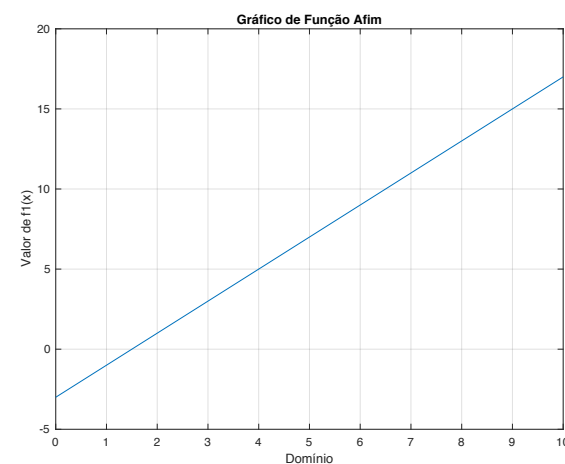
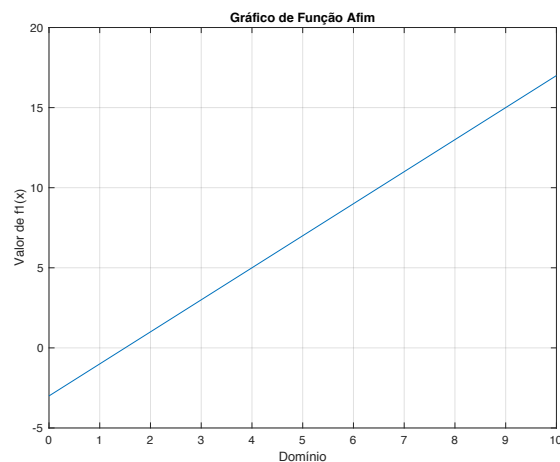
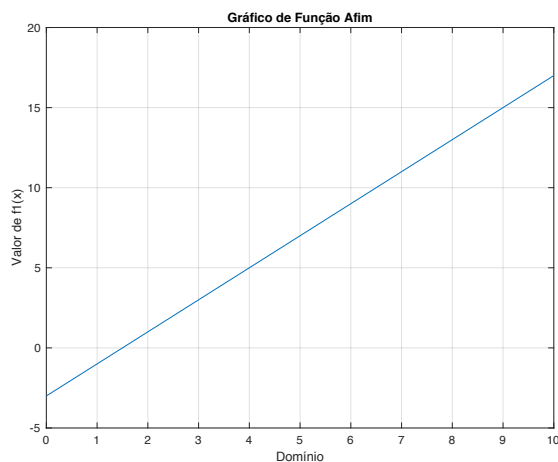
Exemplo 1: Construir o gráfico das seguintes funções

a) $f_1(x) = 2x - 3$

```
x = 0:10;  
>> f1 = 2*x - 3 ;  
>> plot(x,f1)
```

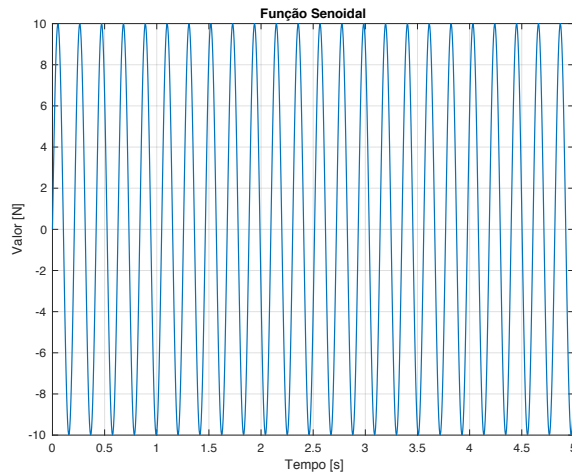
```
>> x = 0:0.001:10;  
>> f1 = 2*x-3;  
>> plot(x,f1)
```

```
x = linspace(0,10,1000)';  
>> f1 = 2*x - 3 ;  
>> plot(x,f1)
```

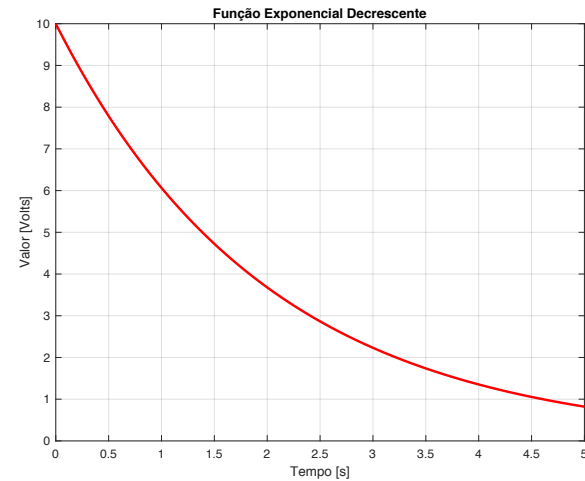


Cont. ...

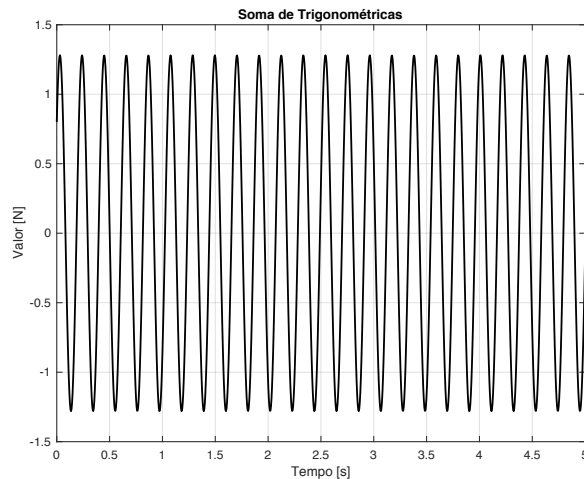
$$b) f(t) = f_0 \text{sen}\omega t$$



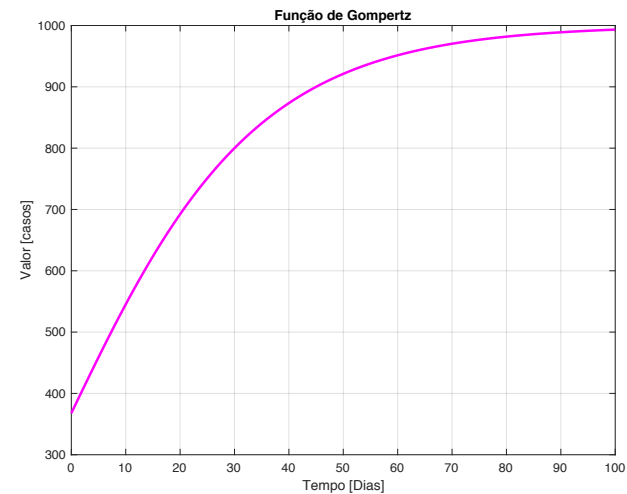
$$c) f(t) = f_0 e^{-\tau t}$$



$$d) f(t) = a_1 \text{sen}\omega t + a_2 \text{cos}\omega t$$



$$e) f(t) = N e^{-e^{-k(t-T)}}$$



FINIM

Bom Estudo !

