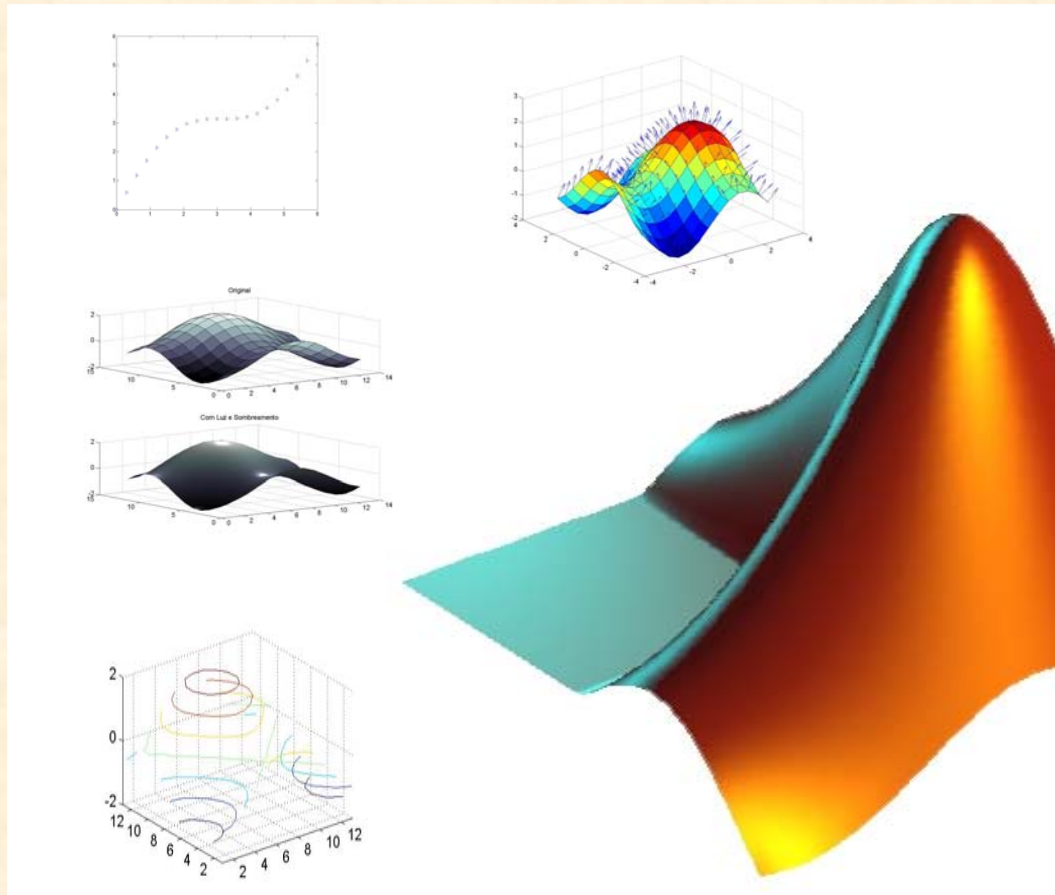


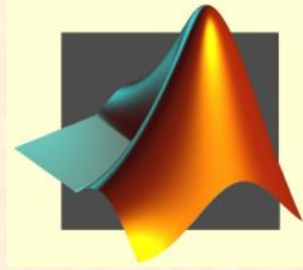
CURSO MATLAB



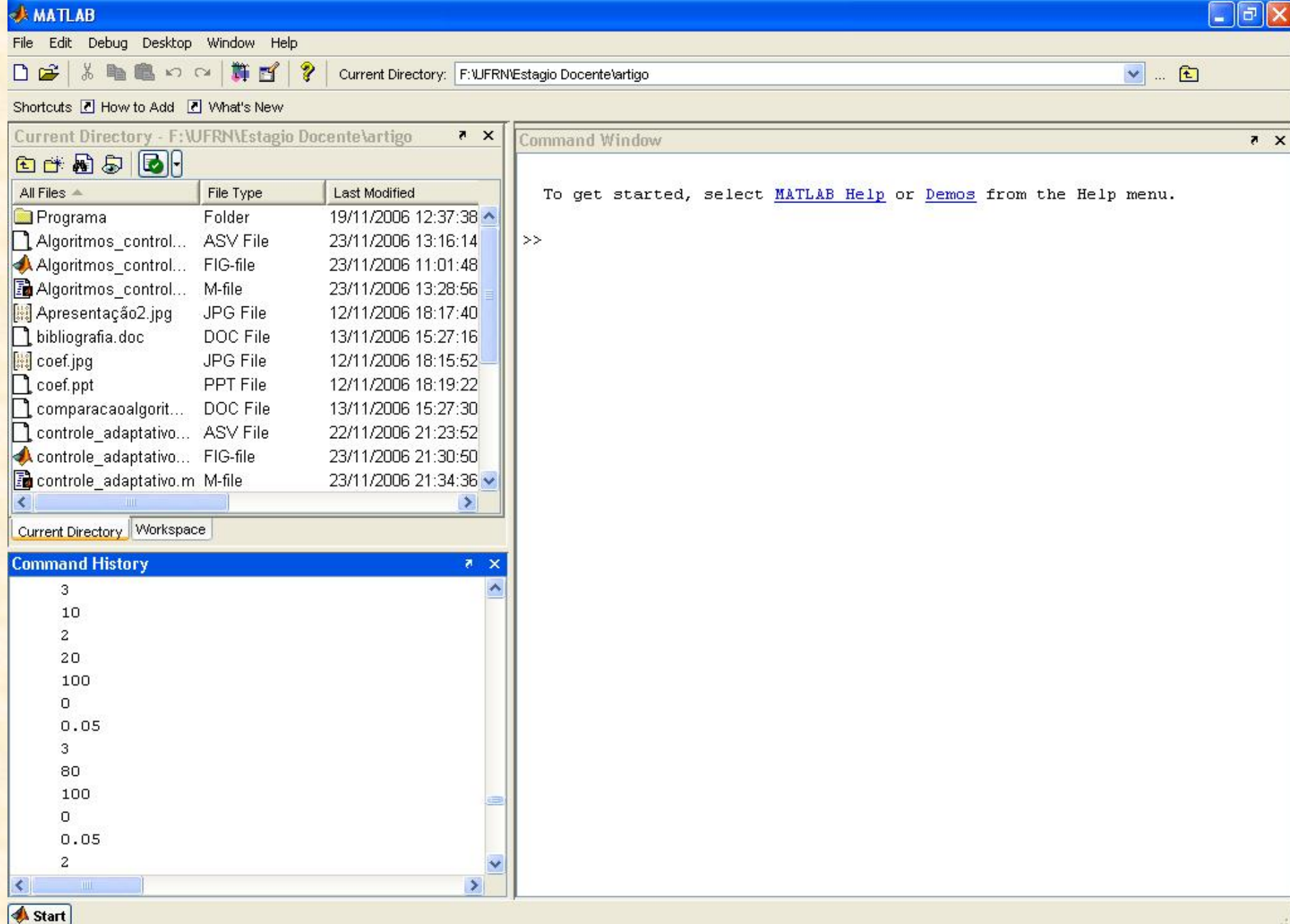
Instrutores: Marcelo Brandão e Marcelo Guerra
CT-DEE

brandao@dca.ufrn.br

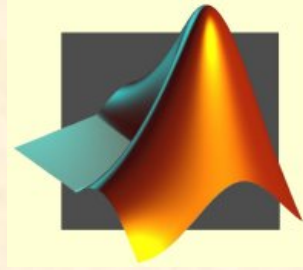
1-Visão Geral



- Janela de Comandos (*command window*)
- Área de Trabalho (*workspace*)
- Pasta de Arquivos (*current directory*)
- Histórico de Comandos (*command history*)

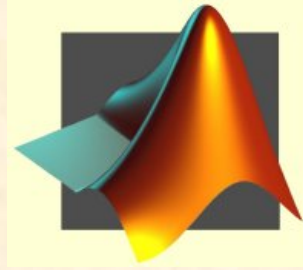


1-Visão Geral



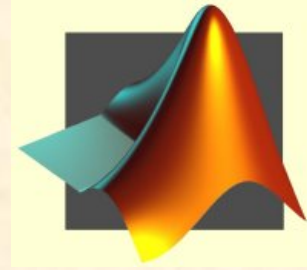
- Menus
- Configuração de Diretórios:
 - File/Set Path...
- Help:
 - help
 - help general
 - help ops

1-Visão Geral



- Bibliografia:
 - Matlab 7, Fundamentos, Élia Yathie Matsumoto, Ed. Érica, 2004.
 - www.mathworks.com

2-Criação de Matrizes



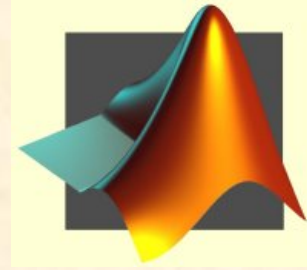
- Escalares
 - '.' indica casa decimal;
 - 'e' para notação científica;
 - 'i' e 'j' para notação de números complexos.

Ex: $a = 4 + j * 5$

- Constantes numéricas predefinidas:

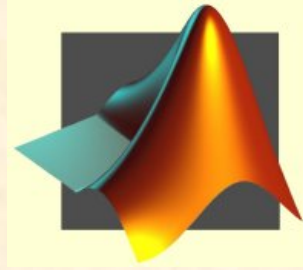
Constante	
pi	3.1415926...
eps	2.220446049250313 e-016
realmin	2.22507385850720 e-308 (menor número real)
realmax	1.797693134862316 e+308
Inf	infinito
NaN	<i>Not-a-number</i> (exemplo: 0/0)

2-Criação de Matrizes



- Operadores

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
^	Potência
()	Precedência



2-Criação de Matrizes

- Vetores e matrizes:

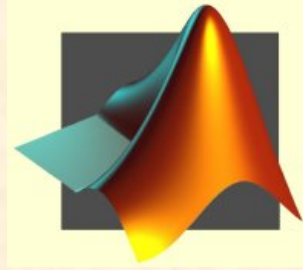
- delimitadores:

- Coluna: "espaço" ou ",",
 - Linha: ";"

- Exemplos:

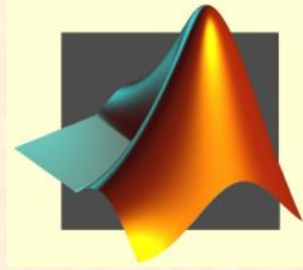
- $a=3$ ou $a=[3]$ (1x1)
 - $b=[4\ 5\ 6\ 7]$ ou $b=[4,5,6,7]$ (1x4)
 - $c=[4\ 5\ 6\ 7;8\ 9\ 10\ 11]$ (2x4)
 - $d=[4\ 5\ 6\ 7]'$ ou $d=[4;5;6;7]$ (4x1)

2-Criação de Matrizes



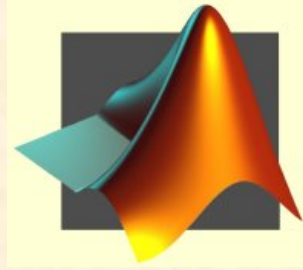
- Criando um vetor:
 - `x = 0:0.1:1`
 - `x = 1:10` (operador ":")
 - `x = linspace(0,1,11);` (operador ";")
- `help linspace`
- Concatenação de matrizes
 - `c = [a b]` ou `c = [a;b]`

2-Criação de Matrizes



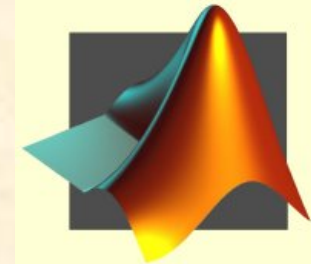
- Denominação das Variáveis:
 - alfanuméricas iniciadas por letra
 - Ex: x1, alfa
 - máximo de 63 caracteres. (Windows)
 - *case-sensitive*: $x1 \neq X1$
 - 'ans' é a variável reservada para o Matlab armazenar o resultado da última operação.

2-Criação de Matrizes



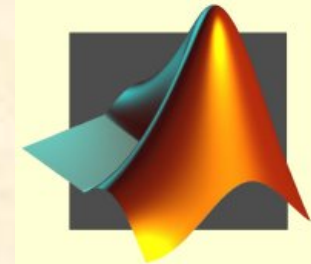
- Matrizes elementares:
 - zeros
 - ones
 - eye
 - rand Ex: $n = \text{round}(5 * \text{rand}(1,6))$

2-Criação de Matrizes

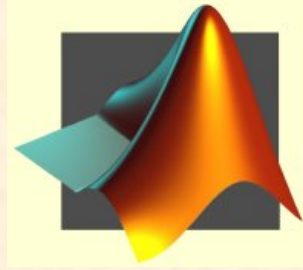


- Outras funções úteis:
 - round → aprox. p/ inteiro mais próximo
 - floor → aprox para baixo
 - ceil → aprox para cima
 - rem → resto da divisão
 - sign → retorna o sinal

3-Formatos Numéricos

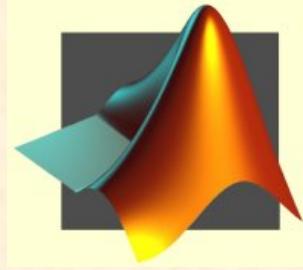


- help format
 - format short – precisão simples
 - format long – precisão dupla
 - format short e – precisão simples E
 - format long e – precisão dupla E+
 - format hex – hexadecimal
 - format bank – duas casas decimais
 - format rat – quociente



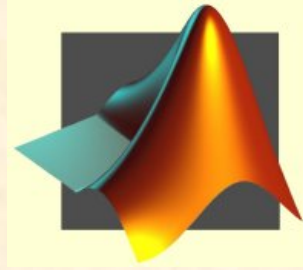
4-Manipulação de Matrizes

- Indexação: (i,j) → (i- linha , j- coluna)
 - $a=[1:10]$
 - $a_2=a(2)$
 - $a_7=a(7)$
 - $b=a(1:5)$
 - $c=a(1:2:9)$
 - $d=3*\text{rand}(3,4)$
 - $e=d(1,2:\text{end})$
 - $f=d(2,:)$
 - $g=d(:,3:4)$



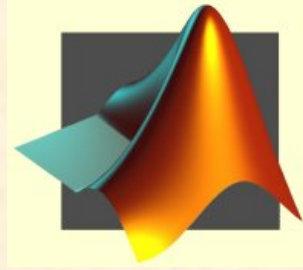
4-Manipulação de Matrizes

- Indexação: (p)
 - Além do padrão de indexação (i,j), o Matlab aceita também, indexação sequênciada.
 - A contagem da posição é feita, seguindo primeiro por linha e depois coluna.
 - Ex: $d(3,2) \approx d(6)$



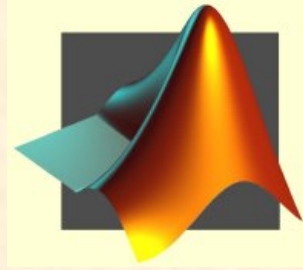
4-Manipulação de Matrizes

- Operações com matrizes:
 - `*` multiplicação matricial
 - `.*` multiplicação escalar
 - `/` divisão matricial ($a/b \rightarrow a*b^{-1}$)
 - `./` divisão escalar
 - `\` divisão à esquerda ($a\b \rightarrow a^{-1}*b$)



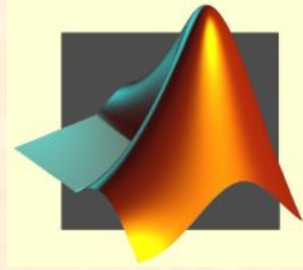
4-Manipulação de Matrizes

- Operações com matrizes:
 - \wedge potência
 - \wedge potência escalar
 - $'$ transposta
 - $\det(a)$ - determinante
 - $\text{inv}(a)$ - inversa
 - $\text{eig}(a)$ - autovalores
 - $\text{poly}(a)$ - polinômio característico



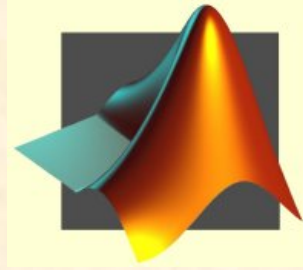
4-Manipulação de Matrizes

- Eliminando linhas ou colunas de uma matriz:
 - $a(1,:)=[]$ - elimina a primeira linha
 - $a(:,3)=[]$ - elimina a terceira coluna



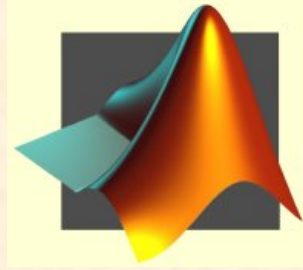
4-Manipulação de Matrizes

- Modificando elementos de uma matriz:
 - Faz-se referencia ao elemento a ser modificado e o seu novo valor:
 - Ex: `A=rand(4);`
`A(2,2)=3;`
`A(1,2:4)=0;`



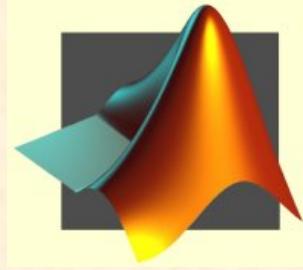
4-Manipulação de Matrizes

- Funções de Informação:
 - `size (A)`
 - `length (A)`
 - `find(exp)` – retorna os índices dos elementos que satisfazem a condição *exp*



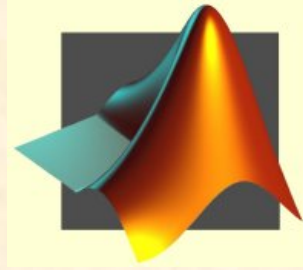
4-Manipulação de Matrizes

- Funções de Manipulação:
 - diag (diagonal)
 - tril e triu (matrizes triangulares)
 - fliplr e flipud (inverte a ordem)
 - sort (organiza em ordem crescente)



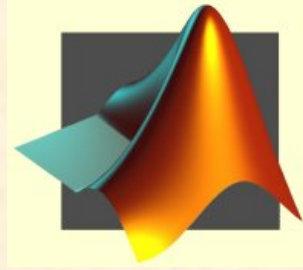
4-Manipulação de Matrizes

- Solução de sistemas lineares ($A*X=B$):
 - $X=\text{inv}(A)*B$
 - Ex: $4x_1+5x_2=9$ $A=[4 \ 5; 7 \ 1]$ e $B=[9;10]$
 $7x_1+x_2=10$ $x_1=1.3226$ e $x_2=0.7419$
- Para sistemas sobre-determinados:
Pseudo-inversa (matriz não quadrada):
 - $X=\text{pinv}(A)*B$ ($\text{pinv}(A)=\text{inv}(A^T*A)$)



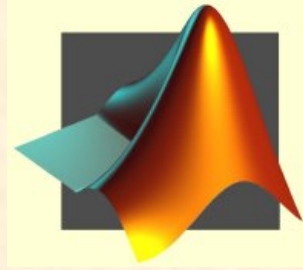
4-Manipulação de Matrizes

- Funções para números complexos
 - `abs(z)` - módulo
 - `angle(z)` - ângulo
 - `real(z)` - parte real
 - `imag(z)` - parte imaginária
 - `conj(z)` - retorna o complexo conjugado de z
 - `plot(z,'o')` - traça um gráfico com a parte real no eixo **x** e a parte imaginária em **y**.



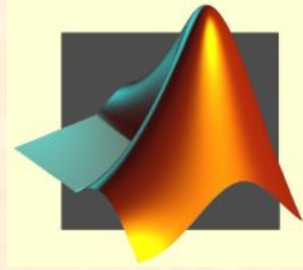
4-Manipulação de Matrizes

- Mudança de Base Numérica:
 - bin2dec – binária para decimal
 - dec2bin – decimal para binária
 - base2dec – base especificada para decimal
 - dec2base – decimal para base especificada



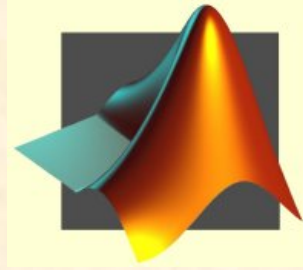
4-Manipulação de Matrizes

- Outras Funções Úteis:
 - `primes(x)` – Retorna todos os primos menores que x
 - `gcd(a,b)` – Máximo divisor comum (MDC).
 - `lcm(a,b)` – Mínimo múltiplo comum (MMC).
 - `factorial(n)` – Retorna o fatorial de n



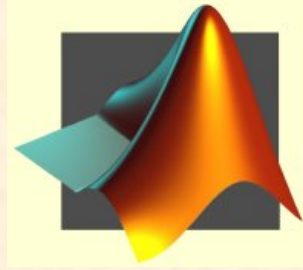
5-Polinômios

- Vetores com os coeficientes:
 - $x^5 + 4x^3 + 5x^2 - x - 10 = 0$
 - $p = [1 \ 0 \ 4 \ 5 \ -1 \ -10]$
- Achar as raízes (zeros): $r = \text{roots}(p)$
- Encontrar o polinômio a partir das raízes:
 $p = \text{poly}(r)$



5-Polinômios

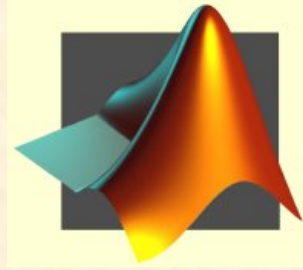
- `polyval(p,x)` e `polyvalm(p,X)`
 - calculam o valor do polinômio .
- `[R,P,K]=residue(B,A)`
 - encontra os resíduos, os pólos e os termos diretos da expansão em frações parciais $B(s)/A(s)$



5-Polinômios

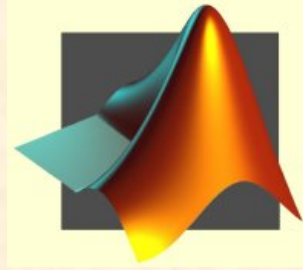
- `polyfit` – ajusta polinômio aos dados.
 - Ex: `x=1:100;`
`y=x.^2+3;`
`p=polyfit(x,y,n);` onde `n` é a ordem do polinômio desejado
- `conv` – multiplicação de polinômios
 - Ex: `c=conv(a,b)`
- `deconv` – divisão de polinômios
 - Ex: `a=deconv(c,b)` e `b=deconv(c,a)`

5-Polinômios



- polyder – derivada de um polinômio
- polyint – integral analítica de um polinômio

6-Gráficos

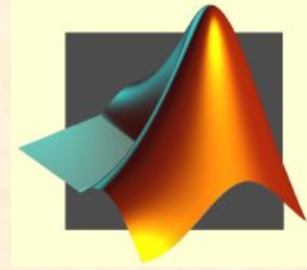


- Plot

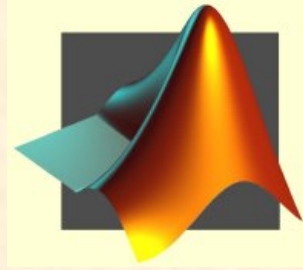
- help plot
- xlabel / ylabel
- grid
- title
- Ex: `plot(X,Y,'S');`

(S= cores e tipos de linha)

6-Gráficos



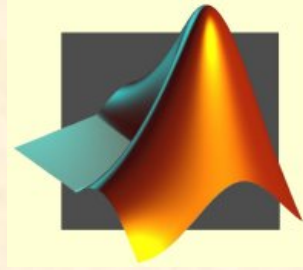
Cores		Tipos de Linha	
b	blue	.	point
g	green	-	solid
r	red	--	dashed
c	cyan	*	star
m	magenta	o	circle
y	yellow	x	x-mark
k	black	s	square
			...



6-Gráficos

- `loglog` - escala logarítmica
- `semilogx` - escala logarítmica em x
- `semilogy` - escala logarítmica em y
- `polar` - Ex: `polar(t,2*t)`
- `figure` - abre uma nova janela gráfica

6-Gráficos



- subplot

- Ex1: >> x=0:.1:10;

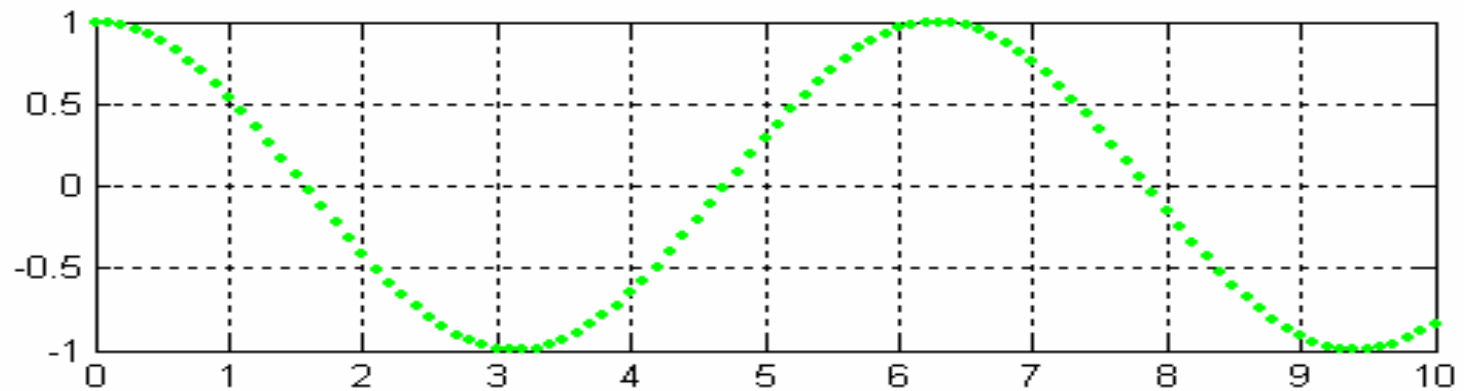
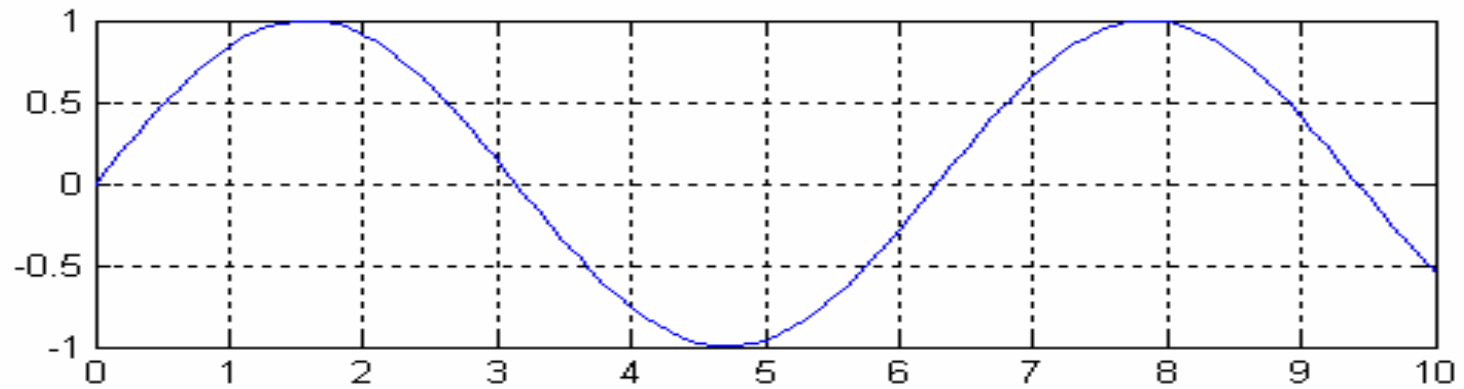
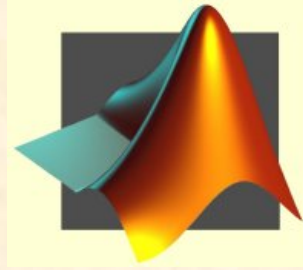
- >> y=sin(x);

- >> y2=sin(x+pi/2);

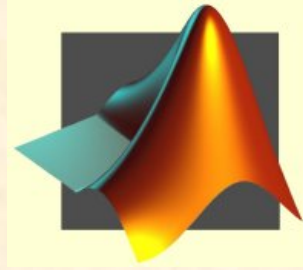
- >> subplot(2,1,1);plot(x,y);grid

- >> subplot(2,1,2);plot(x,y,'g. ');grid

6-Gráficos

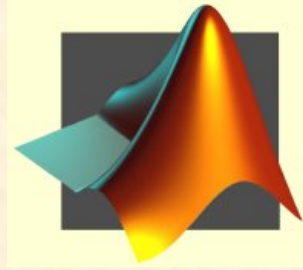


6-Gráficos

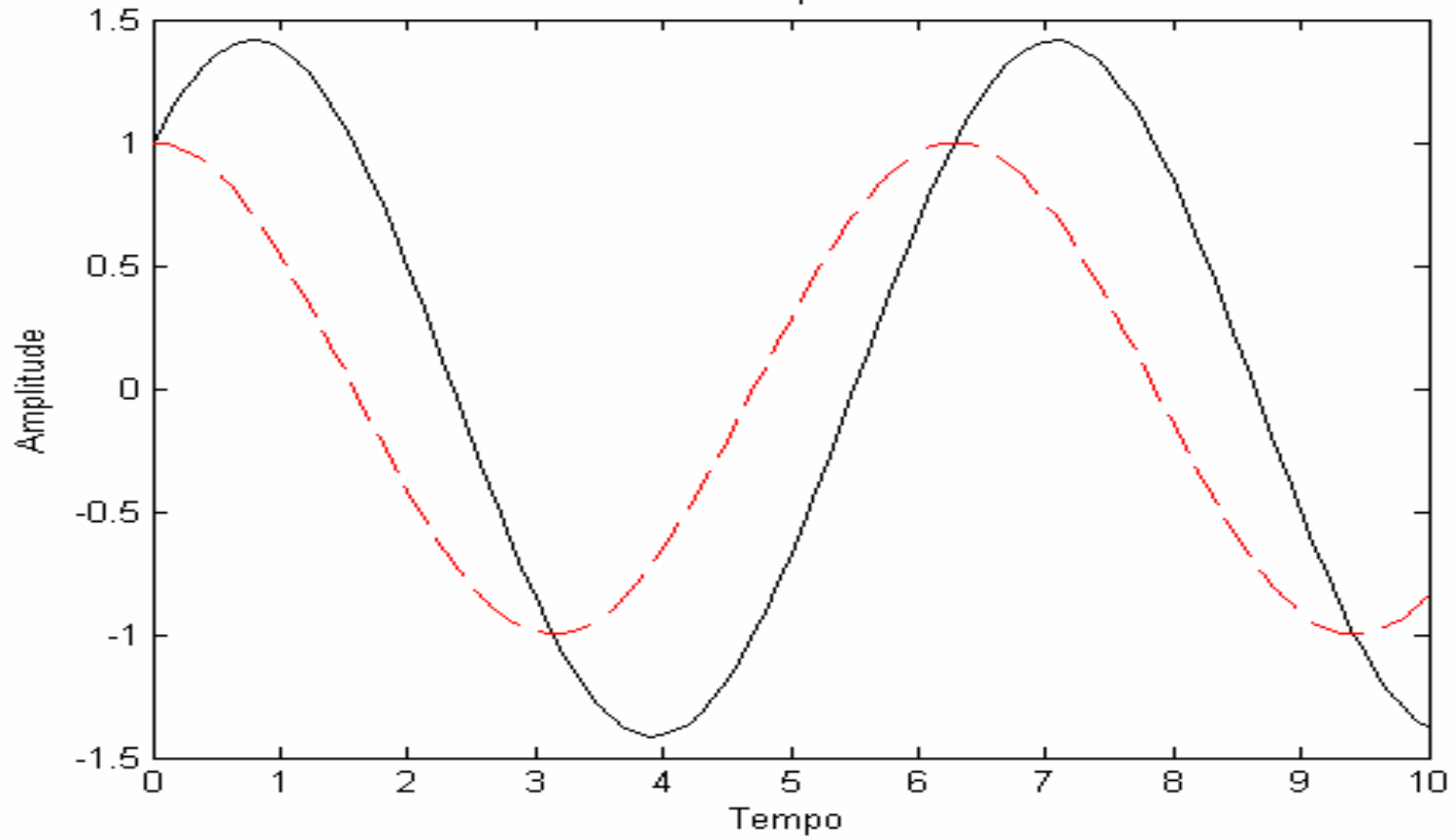


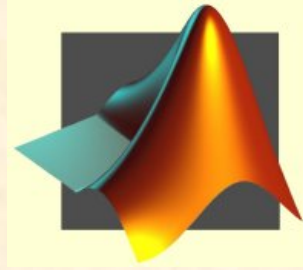
- Ex2:
 >> y3=y+y2;
 >> y4=y-y2;
 >> plot(x,y3,'k',x,y2,'r--');
 >> xlabel('Amplitude')
 >> ylabel('Amplitude')
 >> title('Exemplo 2')

6-Gráficos



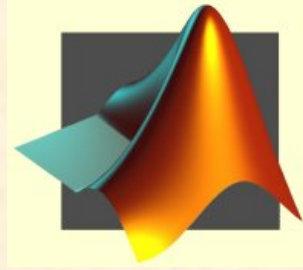
Exemplo 2





6-Gráficos

- Edição na janela gráfica:
 - zoom
 - `ginput(n)` → n =número de pontos
 - Texto
 - Propriedades dos eixos
 - Legendas

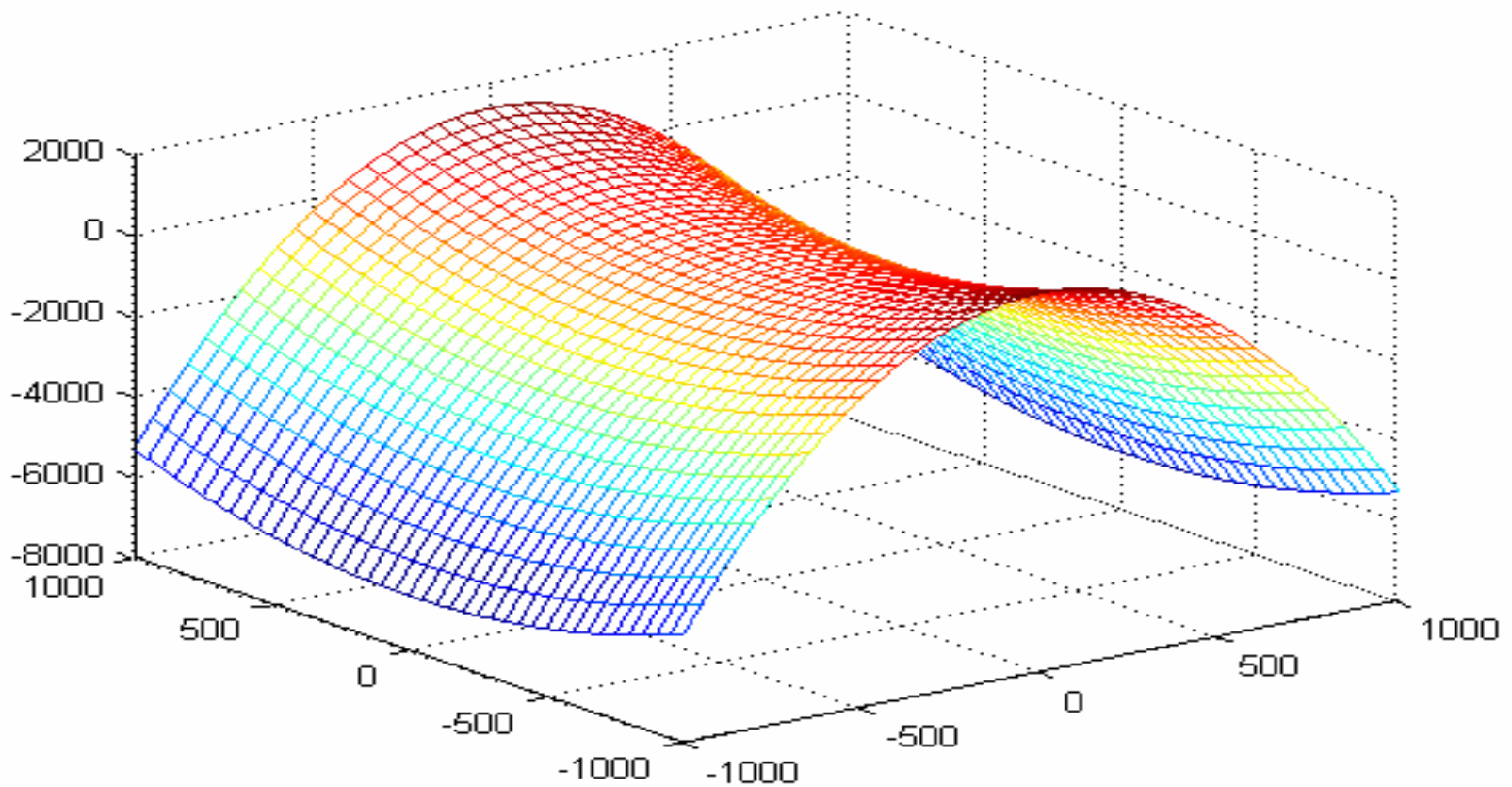
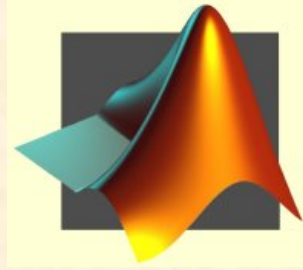


6-Gráficos

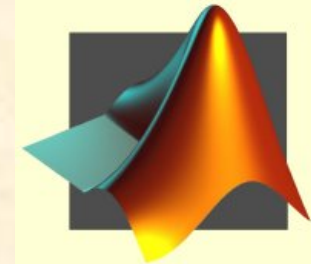
- Ex3 – Gráfico 3D:

```
>> a=-1000:50:1000;  
>> b=a;  
>> [A,B]=meshgrid(a,b);  
>> Z=(-A.^2+B.^2/5)/150;  
>> mesh(A,B,Z)
```

6-Gráficos



7- Derivadas e Integrais



- Derivada Numérica:

- diff – Exemplo:

```
x=linspace(0,5,101);
```

```
y=cos(2*x);
```

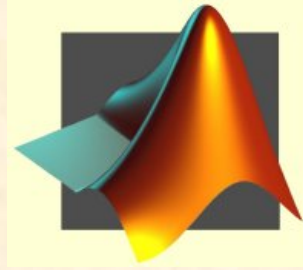
```
der=diff(y)./diff(x);
```

```
plot(x(1:100),der,'--k',x,y)
```

```
grid
```

OBS: por ser um processo numérico iterativo, “perde-se” um ponto do vetor.

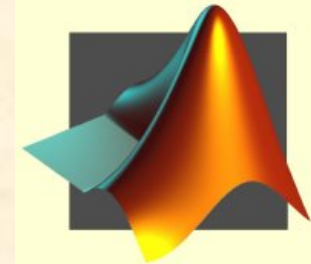
7- Derivadas e Integrais



- Integral Numérica
 - cumsum- calcula uma aproximação da integral através de uma soma acumulada dos elementos. Exemplo:

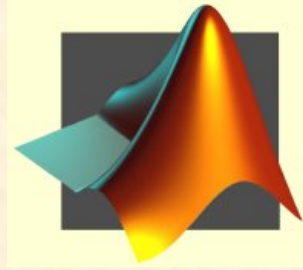
```
inty=cumsum(der).*diff(x);  
plot(x(1:100),der,'--k',x,y,x(1:100),inty,'r')  
grid
```

7- Derivadas e Integrais



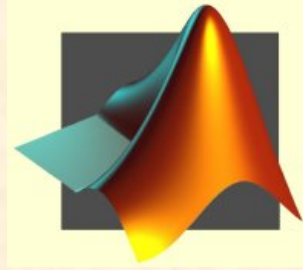
- Integral definida:
 - quad ou quadl
 - dblquad – Integral dupla
 - triplequad – Integral Tripla
- Ex: $Q = \text{quad}(\text{FUN}, A, B)$
a função FUN deve ser definida como:
 $\text{FUN} = \text{inline}('1./(x.^3-2*x-5)');$
A e B são os intervalos de integração.

7- Derivadas e Integrais



- Integral definida:
 - Para o cálculo de integrais duplas e triplas, deve-se prestar atenção nos intervalos de integração:
 - `help dblquad`
 - `help triplequad`

7- Derivadas e Integrais



- Derivadas e Integrais Analíticas

- Variáveis Simbólicas

- `syms x` – define x como variável simbólica

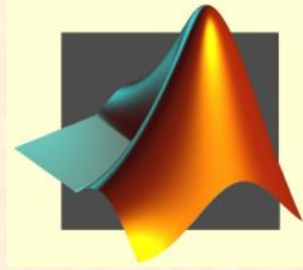
- Ex: `syms t`

- `dt=diff(cos(t))`

- `dt2=diff(t*cos(t))`

- `dt3=diff(exp(-t)*cos(t))`

7- Derivadas e Integrais



– Ex (cont):

```
int(cos(t))
```

```
int(cos(t),0,pi/2)
```

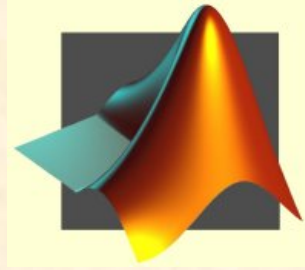
```
int(dt3)
```

```
syms a
```

```
int(1/(t^2+1))
```

```
int(1/(t^2+1),-a,a)
```

8- Outras funções de variáveis simbólicas:



- `simple`

Ex: `syms t`

`z=cos(t)^2+sin(t)^2;`

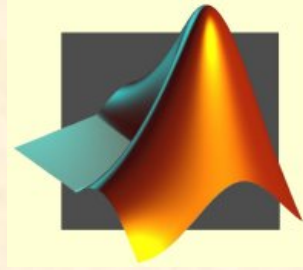
`y=simple(z)`

- `solve` – resolver: $10/t^2 = 4-t$

Ex: `f=solve(10/t^2 - 4+t)`

`eval(f)`

8- Outras funções de variáveis simbólicas:



- Resolver equações simultâneas :

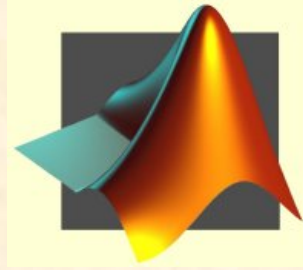
Ex: `syms x y`

`eq1=2*x^2-x+2*y^2-8*y;`

`eq2=x^2+2*x+y^2-6*y+1;`

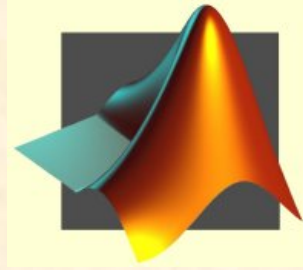
`[X Y]=solve(eq1,eq2)`

9-Programação no Matlab



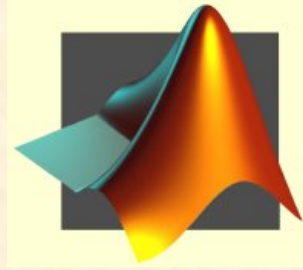
- Arquivos *.m (m-files)
- File / New / m-file
- >> edit

9-Programação no Matlab



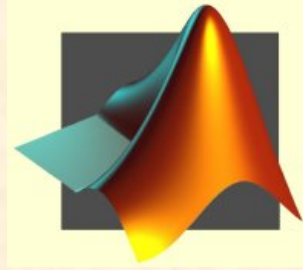
- Arquivos *.m (m-files)
 - Script : seqüência de comandos para automatizar processos repetitivos.
 - Função: aceita parâmetros de entrada e retorna parâmetros de saída.

9-Programação no Matlab



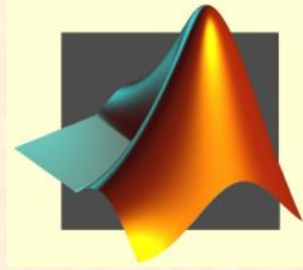
- Observações sobre funções:
 - Utiliza o seu próprio espaço de memória para armazenar variáveis.
 - É possível a implementação de estruturas lógicas.
 - Aceita parâmetros de entrada e retorna parâmetros de saída.

9-Programação no Matlab



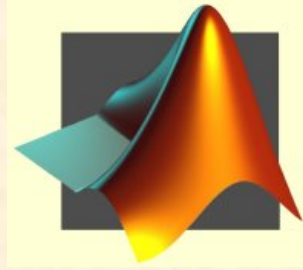
- % - transforma em comentário
- Cabeçalho
- `help nome_da_função` – exibe o cabeçalho da função

9-Programação no Matlab



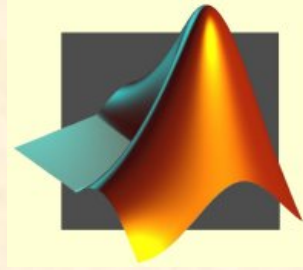
- O código das funções do Matlab pode ser visualizado:
 - `edit nome_da_função`
 - Ex: `edit poly`
`edit roots`
`edit quad`

9-Programação no Matlab



- IF, ELSE IF, ELSE
- SWITCH
- FOR
- WHILE

9-Programação no Matlab



- Ex: IF / ELSE IF / ELSE

```
if I<3
```

```
N=pi;
```

```
elseif I==3
```

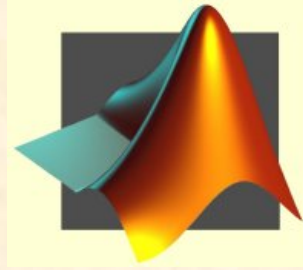
```
N=2*pi;
```

```
else
```

```
N=0;
```

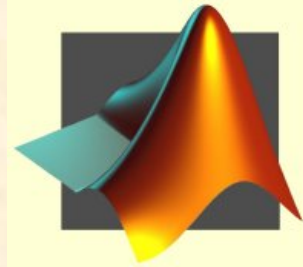
```
end
```

9-Programação no Matlab



- Ex SWITCH:
function tipo=teste(valor)
switch valor
case 3
tipo= 'Numero 3'
case 5
tipo= 'Número 5'
otherwise
tipo='valor nao tabelado'
end

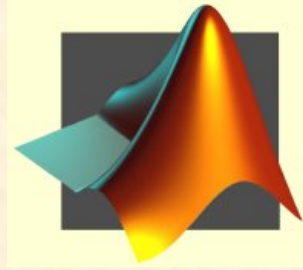
9-Programação no Matlab



- Ex FOR:

```
soma=0;  
for n=1:30  
soma=soma+n;  
end
```

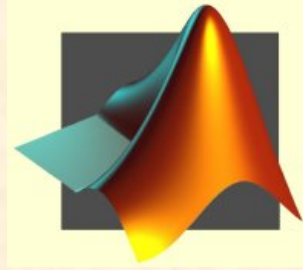
9-Programação no Matlab



- Ex WHILE:

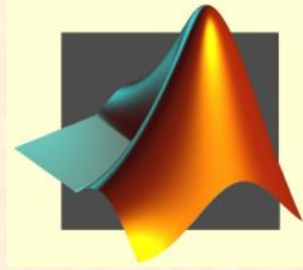
```
while I<m  
    <executa expressão>  
end
```


9-Programação no Matlab



- Códigos eficiente para Matlab
 - Evitar criação de variáveis desnecessárias.
 - Vetorizar os algoritmos, evitando, quando possível o uso de estruturas de repetição.

9-Programação no Matlab



- Ex- script para automatizar o exemplo 6.3:

```
% script para automatizar o exemplo 6.3
```

```
a=-1000:50:1000;
```

```
b=a;
```

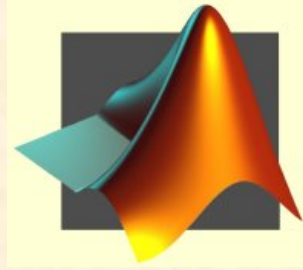
```
[A,B]=meshgrid(a,b);
```

```
Z=(-A.^2+B.^2/5)/150;
```

```
mesh(A,B,Z)
```

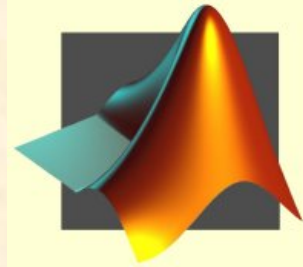
```
%salvar como script1.m
```

9-Programação no Matlab



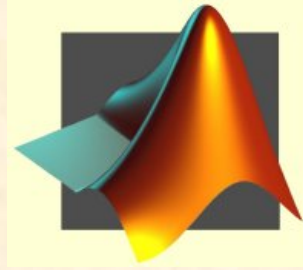
- Ex Função para converter graus em radianos (deg2rad / rad2deg):
- ```
function y=g2r(grau)
y=pi*grau/180;
```

# 9-Programação no Matlab

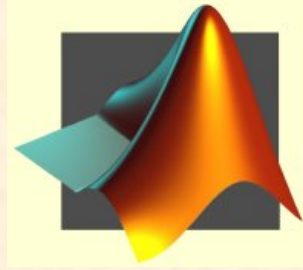


- Ex: Função para traçar o gráfico do seno e cosseno de  $x$ , num intervalo especificado:

# 9-Programação no Matlab



- Ex: Função para calcular o somatório de 2 senóides.
  - Parâmetros de Entrada:  $a_1$ ,  $a_2$ ,  $\omega_1$ ,  $\omega_2$ ,  $\Phi_1$  e  $\Phi_2$ .
  - Parâmetros de Saída:  $Y$ , `plot(x,y)`.
  - Garantir que seja exibidos no gráfico pelo menos 2 períodos da função de menor frequência.



```
%Exemplo 9.2
%Somatorio de 2 senoides
% Y=som(a1,a2,w1,w2,f1,f2)
% onde:
% a-amplitude
% w-frequencia em rad/s
% f-fase em radianos
```

```
function Y=som(a1,a2,w1,w2,f1,f2)
```

```
%cria a base de tempo e garante
%que sejam exibidos 2 periodos da
%onda de menor frequencia
```

```
if w1<w2
 T1=2*pi/w1;
 t=linspace(0,2*T1,200);
else
 T2=2*pi/w2;
 t=linspace(0,2*T2,200);
end
...
```

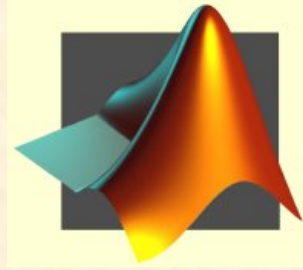
```
...
```

```
Y1=a1*sin(w1*t+f1);
Y2=a2*sin(w2*t+f2);
Y=Y1+Y2;
```

```
%plota o grafico de Y, Y1 e Y2
```

```
plot(t,Y,t,Y1,'k--',t,Y2,'r--')
grid
xlabel('Tempo')
ylabel('Amplitude')
title('Exemplo 2 - Somatorio de 2
 senoides')
```

# 10- Importação e Exportação de Arquivos



- Arquivos de Texto:

```
>> type teste.txt
```

```
1,2,3,4
```

```
5,6,7,8
```

```
9,10,11,12
```

```
>> test=dlmread('teste.txt','')
```

```
test =
```

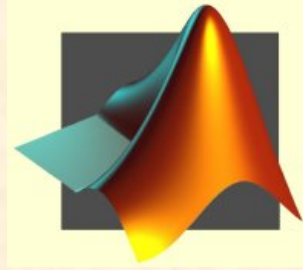
```
1 2 3 4
```

```
5 6 7 8
```

```
9 10 11 12
```



# 10- Importação e Exportação de Arquivos



- Arquivos de Texto:

```
>> test2=test+7
```

```
test2 =
```

```
8 9 10 11
```

```
12 13 14 15
```

```
16 17 18 19
```

```
>> dlmwrite('teste2.txt',test2,',')
```

```
>> type teste2.txt
```

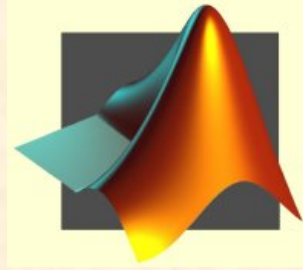
```
8,9,10,11
```

```
12,13,14,15
```

```
16,17,18,19
```

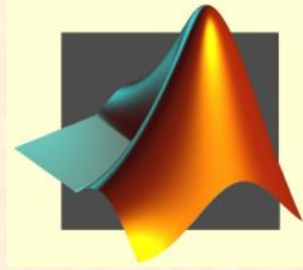


# 10- Importação e Exportação de Arquivos



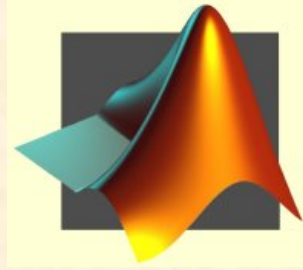
- Arquivos de Texto:
  - load / save
  - Carregar arquivo a partir do *current directory*.

# 10- Importação e Exportação de Arquivos



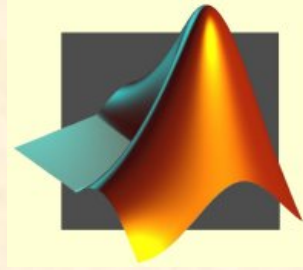
- Arquivos de Som:
  - `[ring fs]=wavread('ring.wav');`
  - `wavplay(ring,fs);`
  - `x=1:length(ring);`
  - `y=sin(10*x);`
  - `ring2=ring+y';`
  - `wavplay(ring2,44100)`
  - `wavwrite(ring2,44100,'ring2.wav');`

# 9- Importação e Exportação de Arquivos



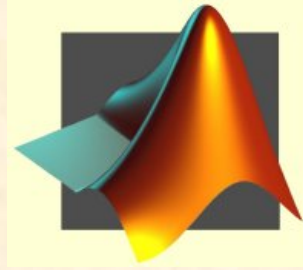
- Arquivos de Imagem:
  - help images : (toolbox proc. Imagem)
  - I=imread('imagem.jpg');
  - J=imnoise(I,'gaussian');
  - truesize
  - J=imrotate(J,60);
  - imwrite(J,'teste.jpeg')

# 10- Importação e Exportação de Arquivos



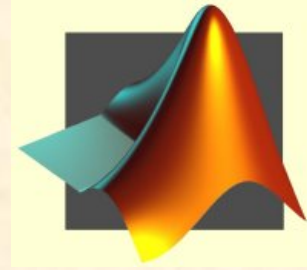
- Dados em Geral:
  - `X=importdata('arquivo');`
  - Verifica a extensão do arquivo especificado e utiliza a função de importação mais conveniente.

# 11- Funções Estatísticas de um Sinal

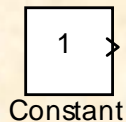


- `max(x)`
- `min(x)`
- `mean(x)`
- `std(x)`

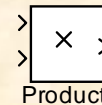
# 12- Simulink



- Algoritmo em Diagramas de Blocos



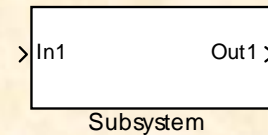
Sinal constante



Produto



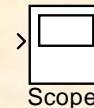
Ganho



Subsistema

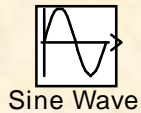
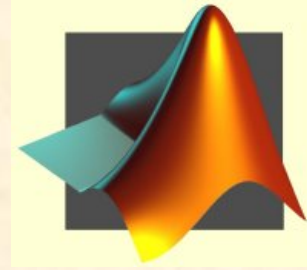


Somador

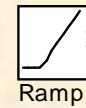


Osciloscópio

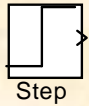
# 12- Simulink



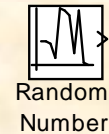
Sinal senoidal



Sinal rampa



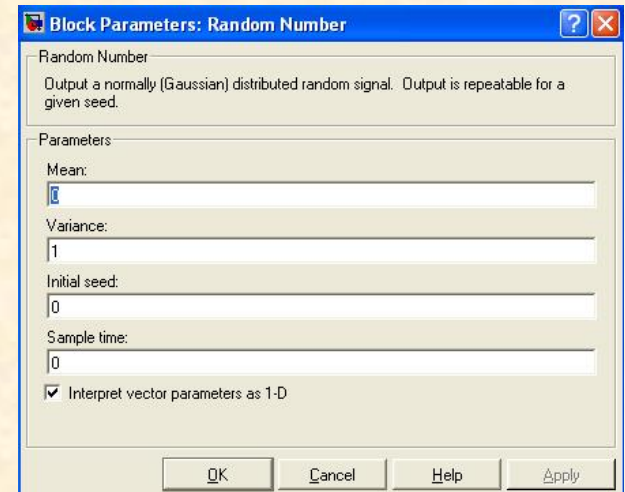
Sinal degrau

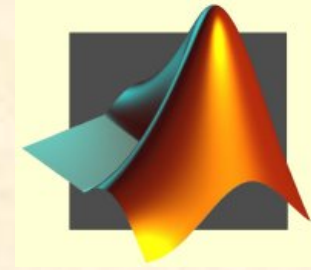


Número aleatório



Tempo

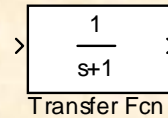




# 12- Simulink



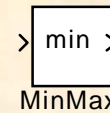
Multiplexador



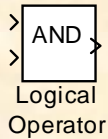
Função de Transferência



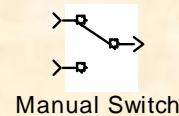
Demultiplexador



Mínimo e máximo



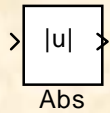
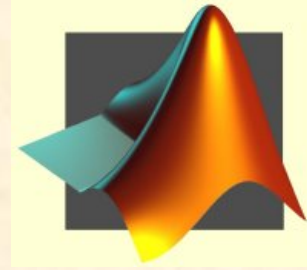
Operador Lógico



Switch manual



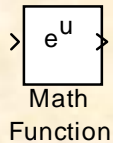
# 12- Simulink



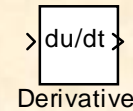
Valor Absoluto



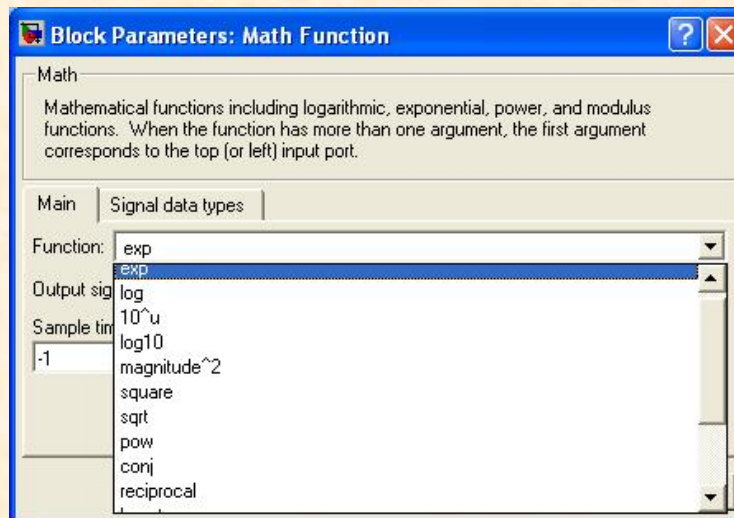
Função Sinal



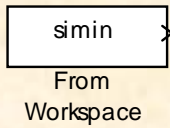
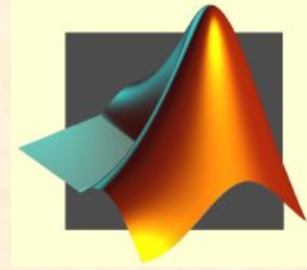
Função Matemática



Derivada



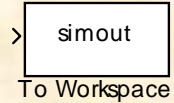
# 12- Simulink



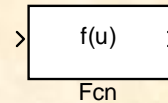
Exportar da área e  
trabalho



Saturador



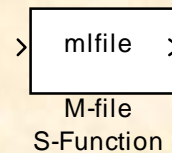
Importar para a área  
de trabalho



Função

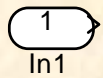
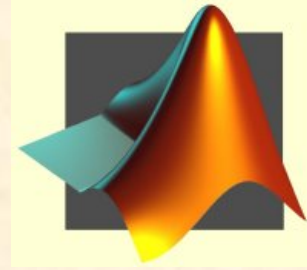


Terra

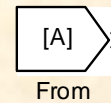


Função escrita num  
arquivo m-file

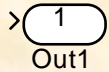
# 12- Simulink



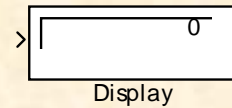
Entrada



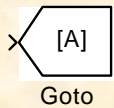
Vem de



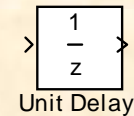
Saída



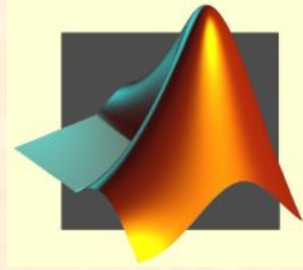
Display



Ir para

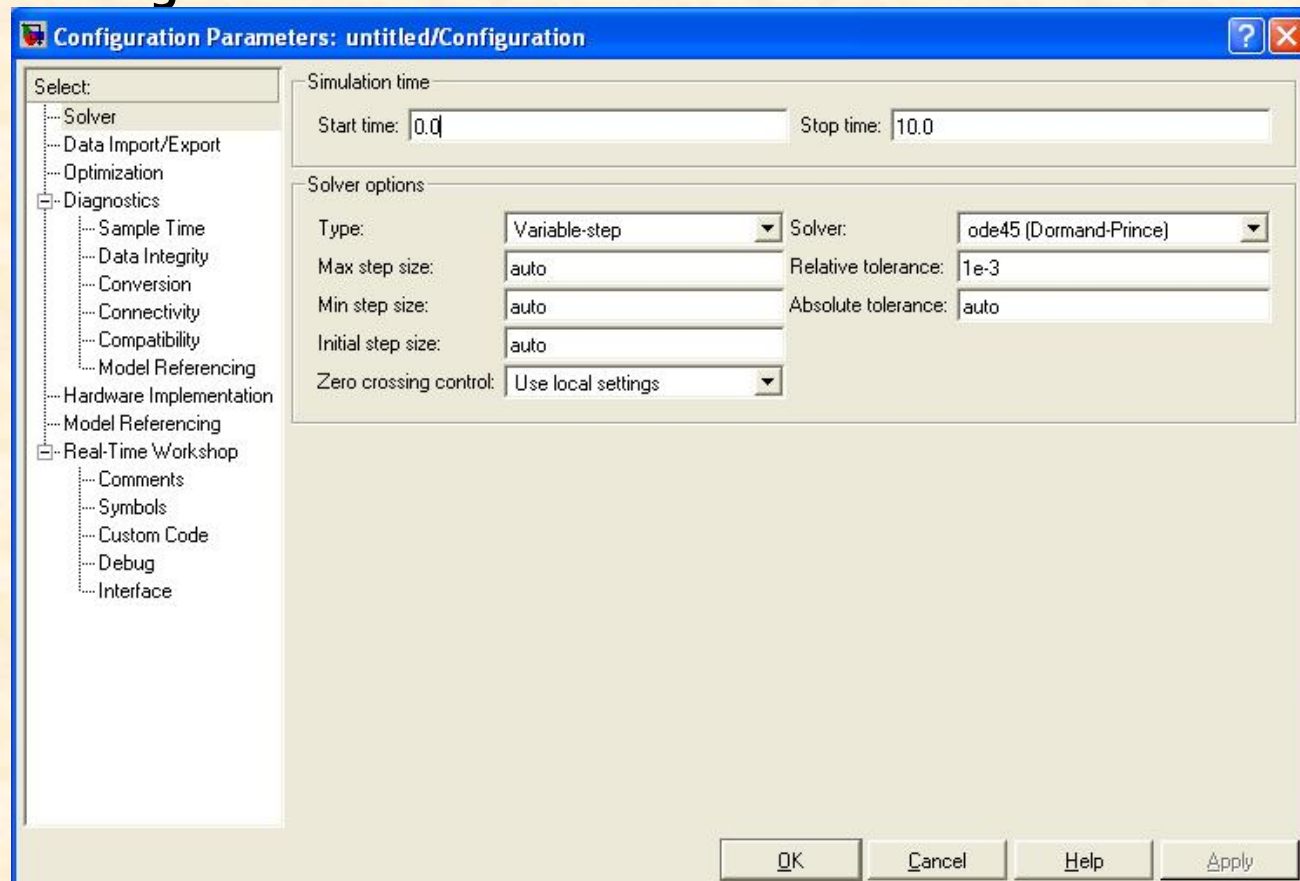


Atraso Unitário

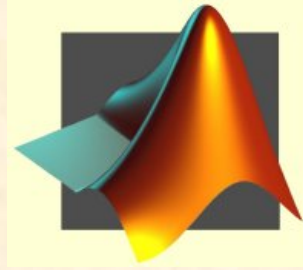


# 12- Simulink

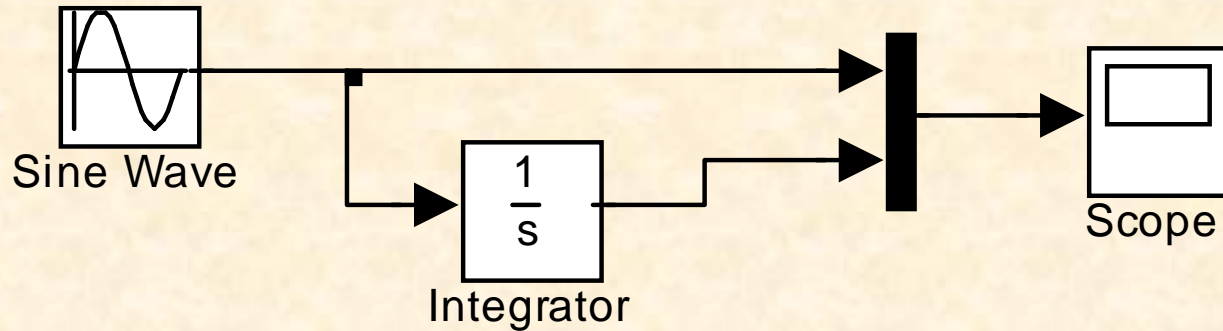
- Configurando os parâmetros da Simulação



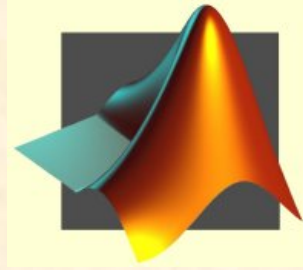
# 12- Simulink



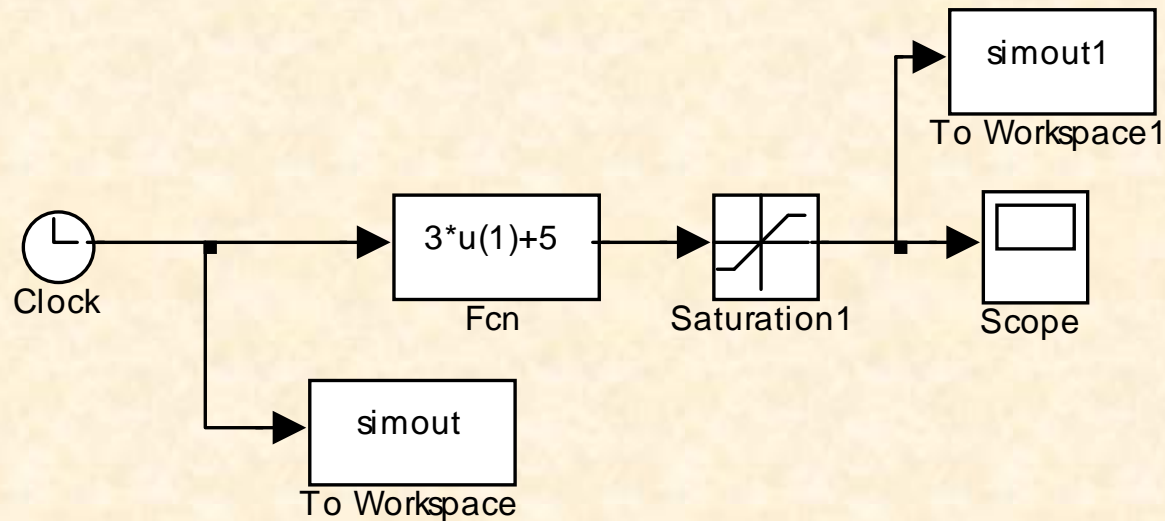
- Exemplos



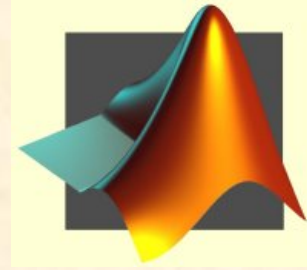
# 12- Simulink



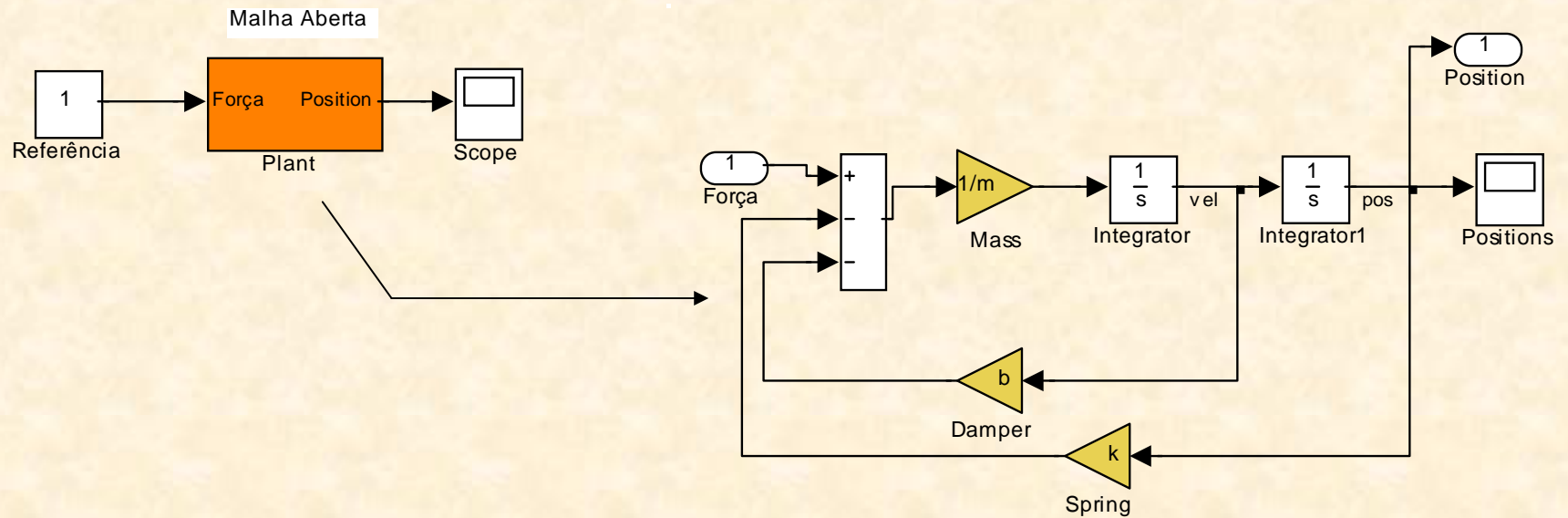
- Exemplos

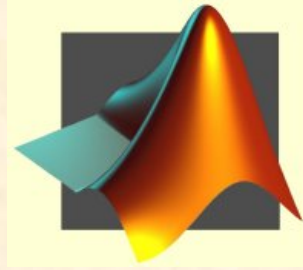


# 12- Simulink



- Exemplos

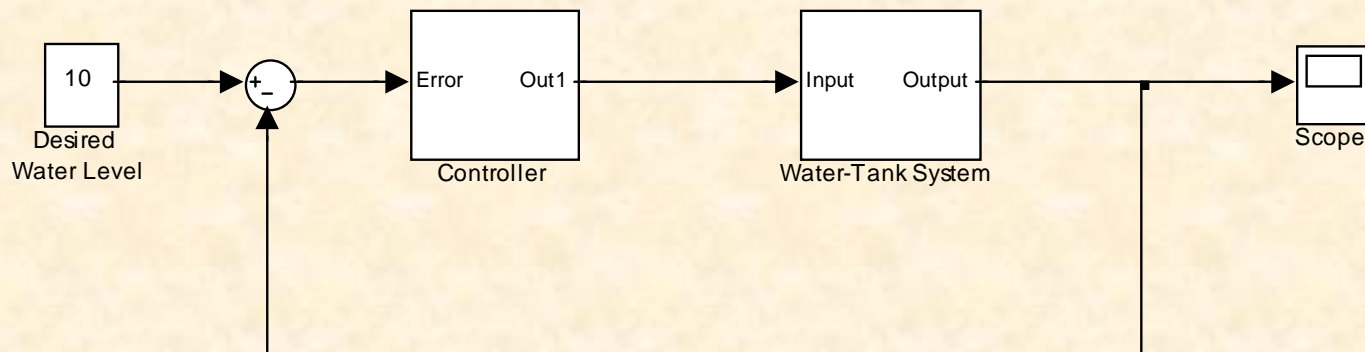




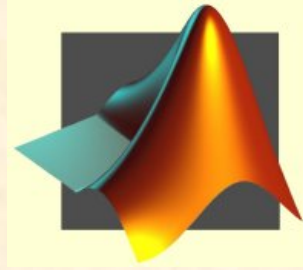
# 12- Simulink

- Demos

Start – Demos – Simulink – Simulink  
Control Design – Water Tank



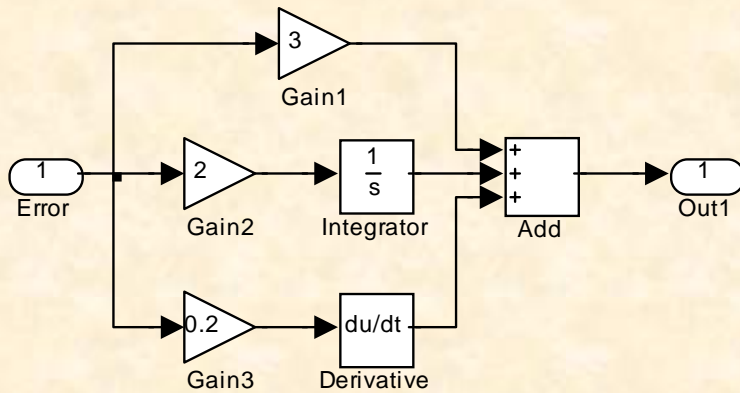




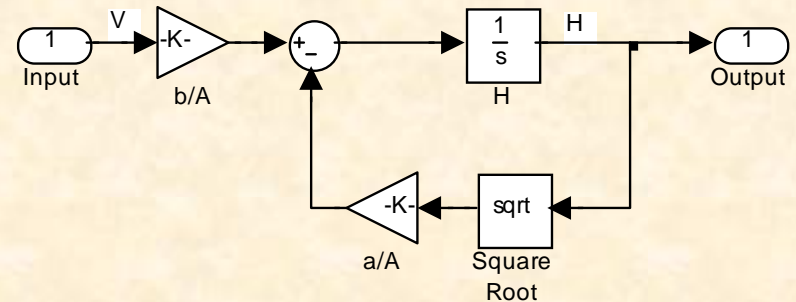
# 12- Simulink

- Demos

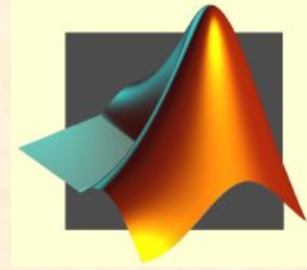
Controller



Water Tank System



# 12- Simulink



- Exercícios

## Sistema Massa – Mola

- Equações básicas da Física:

Aceleração:  $a = \frac{dv}{dt} = \frac{d^2x}{dt^2}$

Velocidade:  $v = \frac{dx}{dt} = \int a dt$

Deslocamento:  $x = \int v dt$

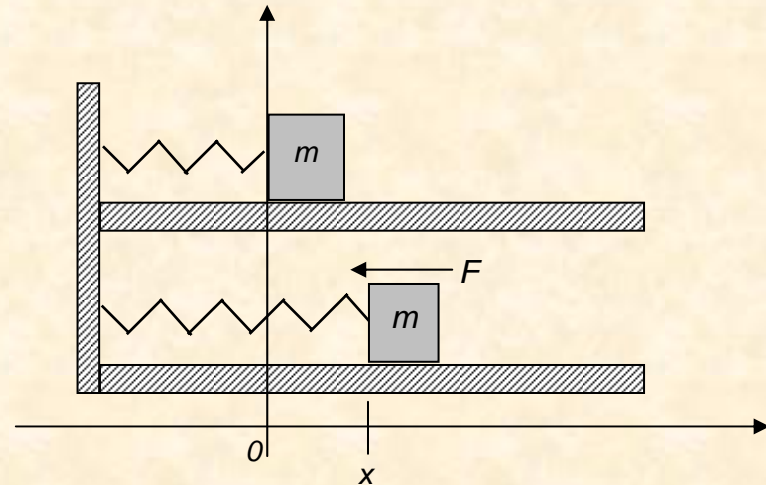
Força:  $F = -kx = ma$

$$kx = m \frac{d^2x}{dt^2}$$

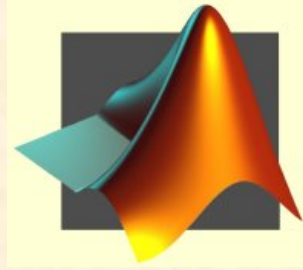
$$a = \frac{d^2x}{dt^2} = -\frac{k}{m}x$$

$$E_p = \frac{1}{2}kx^2$$

$$E_c = \frac{1}{2}mv^2$$



# 12- Simulink

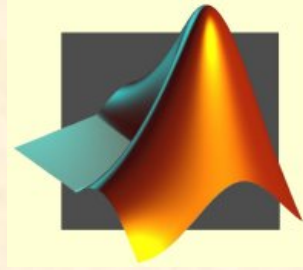


- Exercícios

Plotar:

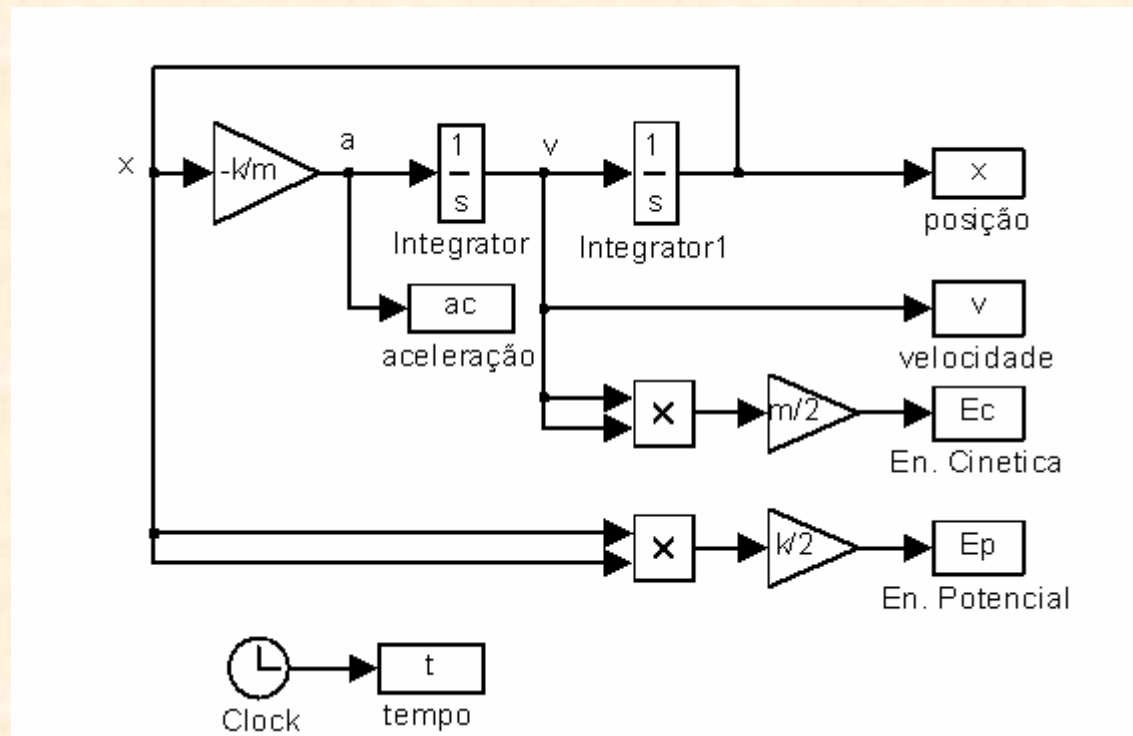
- ✓ Energia Cinética e Potencial *versus* Deslocamento
- ✓ Energia Cinética e Potencial *versus* Velocidade
- ✓ Velocidade *versus* Deslocamento

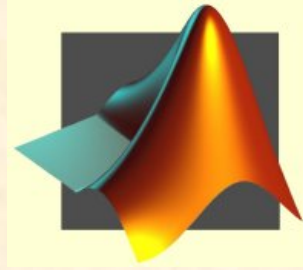
# 12- Simulink



- Exercícios

Diagrama:





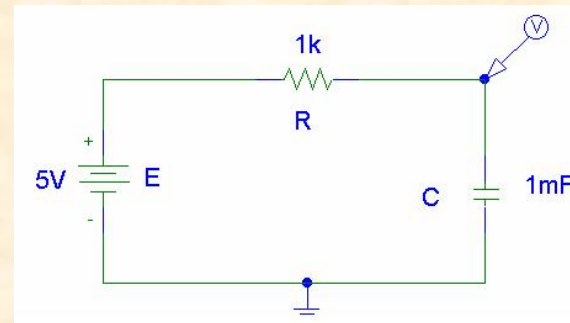
# 12- Simulink

- Exercícios

## Circuito RC – Série

- Equações básicas da Física:

Lei de Kirchhoff:  $e - v_R - v = 0$

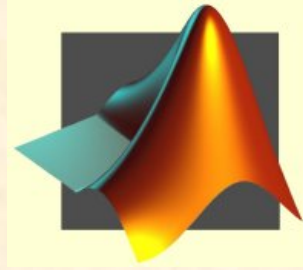


Equações:  $v_R = Ri(t)$   $i(t) = C \frac{dv}{dt}$   $\frac{dv}{dt} + v = 5$

$$e - RC \frac{dv}{dt} - v = 0 \quad \frac{dv}{dt} + \frac{1}{RC}v = \frac{1}{RC}e \quad \text{onde: } v(0) = 10$$

$$v = \frac{1}{RC} \int (e - v) dt$$

# 12- Simulink

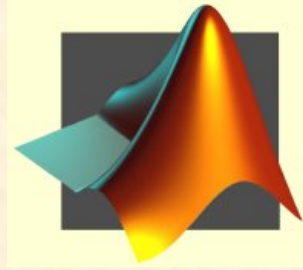


- Exercícios

Plotar:

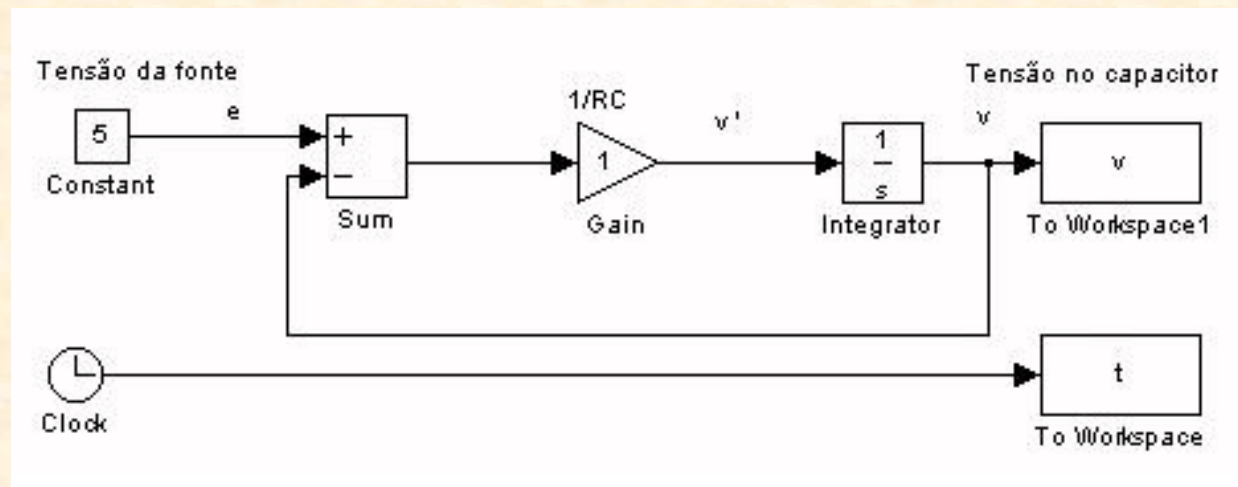
✓  $V(t)$  versus  $t$

# 12- Simulink

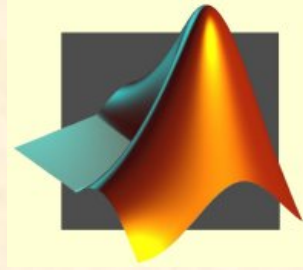


- Exercícios

Diagrama:

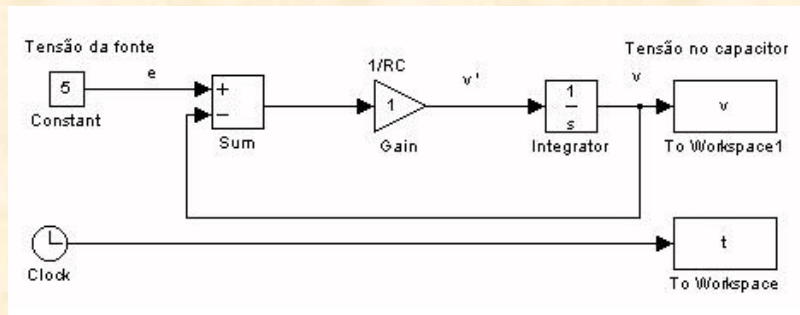


# 12- Simulink



- Exercícios

## Simulação:



$$\dot{x} = 5 - x$$

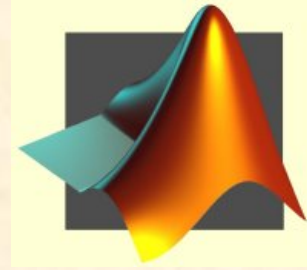
*Metodo de Euler*

$$dx = 5 - x$$

$$x = x + dx * h$$



# 12- Simulink



- Exercícios

Guide:

