EC35F – Transmissão de Dados

Introdução ao Matlab

Prof. Eduardo Alves Hodgson
hodgson@utfpr.edu.br

2020-1

Os laboratórios da disciplina serão em Matlab.

É muito importante aprender as funções básicas de criação de vetores e gráficos para poder realizar os labs.

Sempre que aparecer o *prompt* >> copie e execute o comando no Command Window do Matlab para ver o resultado.

No final dos slides, fazer os exercícios 1 e 2 e apresentar o resultado ao professor.

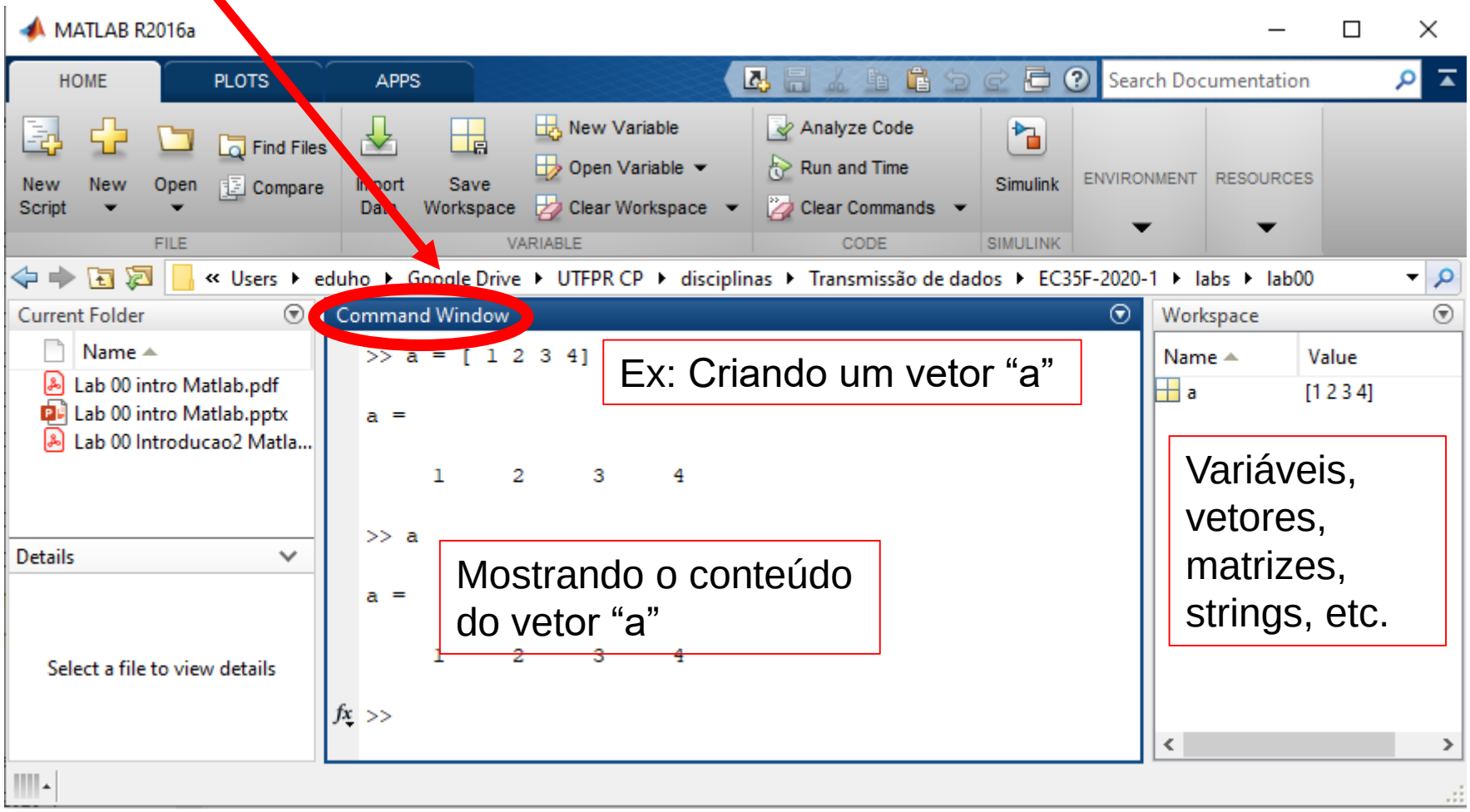
O que é Matlab?

Características

- MATLAB (MATrix LABoratory): software p/ cálculo numérico e científico de visualização de alta performance.
- Elementos Básicos: Matrizes que não requerem dimensionamento.
- Presença de TOOLBOXES: Grande coleção de funções.

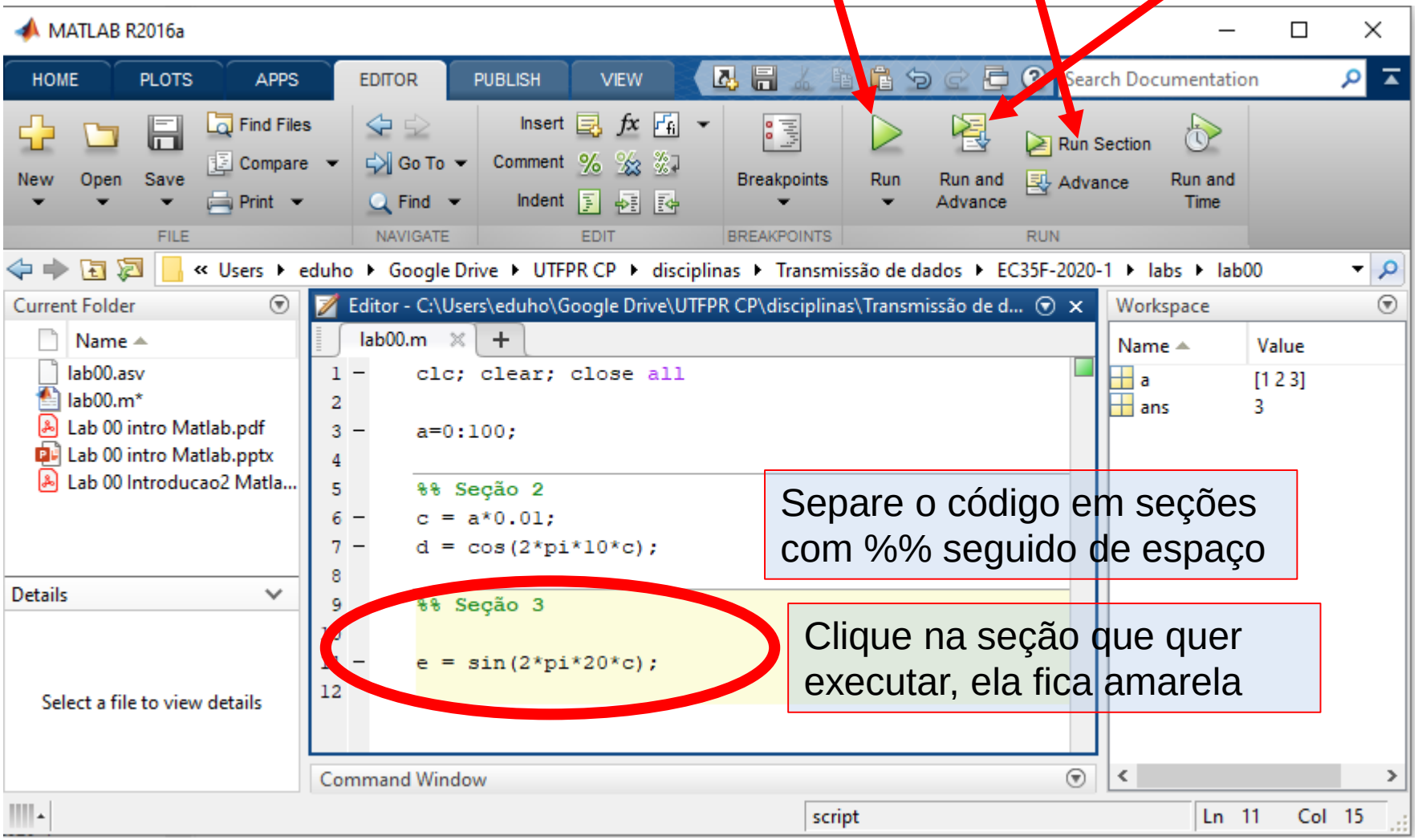
Command Window

Você pode executar comandos e funções do Matlab No *Command Window* e ver o resultado destes.



Command Window

Rode todo seu script aberto com o *Run*
Rode apenas uma seção de cada vez com *Run Section* ou *Run and Advance* (rode a seção atual, e pule para a seguinte)



- Escalares: matriz 1x1

>> a=1

- Vetores: Linha e Coluna

>> a = [1 2 4]

- Matrizes: bi e multidimensionais
- Palavras Reservadas: ans, pi, i, j, inf, version, flops, NaN, computer
- Expressões: Lógicas e Aritméticas
- Gráficos: 2D e 3D
- Controle de Fluxo: while, for, if, etc.

- **Tipos de Dados:**
  - Inteiro: 5; 1000; -564; ...
  - Real: 3; 1.23; .2343; -2.5365e-2, ...
  - Complexo: 2; 1+2i; -47i; -1-3i; ...
    - Distinção entre os tipos numéricos é realizada em tempo de execução.
  - Tipo *String*: 'matlab'; 'figura'; '2ab'; ...
- **Nomes Para Identificadores:**
  - *String* com até 32 caracteres.
  - 1º caractere deve ser uma letra.
  - São sensíveis a maiúsculo e minúsculo
  - Aceita '_' no meio da variável.

Variável = expressão; ➔{não mostra o valor da variável no 'Command Window'}

Variável = expressão {mostra o valor da variável}

Exemplos:

```
>> x = sin(5);
>> x = sin(5)
>> y = 1/3
>> w = 1.602e-20
>> r = .0001
>> soma = 3+2i
>> c = 'ABC'
```

Construção de Vetores:

- Vetor linha:

```
>> x = [5.2  6.3  1.2]
```

- Vetor coluna:

```
>> y = [5.2  6.3  1.2] '
```

➔ transposto

- Referenciação a elementos:

```
>> y(1)
ans =5.2
>> y(end)
ans =1.2
```

→ 'end' contém posição do último elemento de y

- **O primeiro elemento do vetor tem sempre índice 1, e não 0, como em linguagem C.**

- Funções Para Geração de Vetores:

>> x = 1:5 {Vetor linha de 1 até 5 com incremento de 1}

>> x = 0 : 0.05 : 2 {Vetor linha de 0 até 2 com incremento de 0.05}

>> y = ones(1,10) {Vetor linha de 10 elementos de valor 1}

>> y(1:10) = 1 {Vetor linha de 10 elementos de valor 1}

>> w = zeros(1,8) {Vetor linha com 8 elementos nulos}

>> w(1:10) =0 {Vetor linha com 8 elementos nulos}

>> z = rand(1,15) {Vetor linha de 15 elementos aleatórios uniformes de 0 a 1}

Construção de Matrizes

1a. Forma >> x = [5.2 6.3 1.2
6.4 8.3 7.1
9.2 1.2 3.1]

2a. Forma >> y = [5.2 6.3 1.2 ; 6.4 8.3 7.1 ; 9.2 1.2 3.1]

Referenciação a elementos: **`x(linha,coluna)`**

→ x(1,2); x(3,2); y(2,1); y(3,3); ...

O primeiro elemento da matriz tem índice (1,1).

Funções Para Geração de Matrizes:

x = rand(10) {Matriz 10x10 com valores aleatórios uniforme}

y = ones(10,5) {Matriz 10x5 com elementos de valor 1}

z = randn(15) {Matriz 15x15 com elementos aleatórios N(0,1)}

w = zeros(5,8) {Matriz 5x8 com 8 elementos nulos}

OPERADOR	OPERAÇÃO
+	Adição
-	Subtração
*	Multiplicação Matricial
.*	Multiplicação Escalar
/	Divisão Matricial
./	Divisão Escalar
^	Potência
.^	Potência escalar
´	Transposta
()	Precedência

Multiplicação de dois vetores a e b, elemento por elemento

```
>> a = [1 2 3 4]
>> b = [5 5 5 5]
```

- **Forma errada:**

```
>> c = a*b
Error using  *
Inner matrix dimensions must agree.
```

- **Forma correta:**

```
>> c= a.*b
c =
   5   10   15   20
```

.* ⇒ realiza a multiplicação escalar, elemento por elemento.

Multiplicação de dois vetores a e b

```
>> a = [1 2 3 4]
>> b = [5 5 5 5]
```

MATLAB entende que * sem o ponto (.*) é multiplicação matricial, ou seja, linha vezes coluna. Por isso o Matlab retorna o erro de dimensão de matrizes.

“`Inner matrix dimensions must agree`”.

Ou seja, “o número de colunas de a tem que ser igual (must agree) com número de linhas de b”.

Se usarmos um vetor coluna b’ (transposto de b), não temos mais o erro:

```
>> c = a * b'
c =
   50
```

exp	Exponencial	**det**	Determinante
log	Logaritmo natural	**abs**	Valor absoluto
log10	Log de base 10	**sqrt**	Raiz quadrada
max	Máximo valor	**real**	Parte real de complexo
min	Mínimo valor	**imag**	Parte imag de complexo
mean	Média aritmética	**round**	Arredondar para próximo inteiro

Operador	Significado
<	Menor que
<=	Menor ou igual que
==	Igual
~=	Não igual
>	Maior que
>=	Maior ou igual que

Operadores lógicos:

& → Para conjunção (AND)

| → Para disjunção (OR)

~ → Para a negação

```
for c = vi : vf
            <bloco de comandos>
end
```

onde: *c* é a variável de controle

vi é o valor inicial de *c*

vf é o valor final de *c*

Exemplo:

```
>> for i=1:10
v(i) = 3*i;
end
```

Cria um vetor transposto com 10 elementos:

[3 6 9 12 15 18 21 24 27 30]

Veja também “help while”

Primeira Forma:

```
if  <condição verdadeira>
  <bloco de comandos>
end
```

Segunda Forma:

```
if  <condição1 verdadeira>
      <bloco1 de comandos>
  elseif <condição2 verdadeira>
      <bloco2 de comandos>
  else
      <bloco3 de comandos>
end
```

Controle de fluxo

if, elseif, else

Exemplo:

Primeira Forma:

```
if a>3
   a=a+1;
end
```

Segunda Forma:

```
if I == J
  A(I,J) = 2;
elseif I == 1
  A(I,J) = 1;
else
  A(I,J) = 0;
end
```

Criação de scripts

- ***Exemplo:*** *fazer um programa que armazene num vetor os n primeiros termos da sequência de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13...*

% Salve o código abaixo em um arquivo "fibonacci.m"

```
n = input('Entre com o valor de n > 2: ');
f(1) = 0;
f(2) = 1;
for i = 3:n
f(i) = f(i-1) + f(i-2);
end
```

- `>> fibonacci` *{Exemplo de chamada}*

Funções Gráficas com Matlab

Funções gráficas

- **plot(.)** - cria um gráfico com escalas lineares sobre ambos eixos.

- **semilogx(.)** - cria um gráfico com escala linear no eixo Y e logarítmicas no eixo X.

- **semilogy(.)** - cria um gráfico com escala logarítmica no eixo Y e linear no eixo X.

Funções gráficas

Alteração de marcadores e traços

Parâmetros da função plot (ver mesma tabela com “help plot”). Utilizar no formato string como abaixo:

```
>> plot(x,y,'m+:') %plot com cor magenta, símbolo de + e pontilhado
```

Símbolo	Cor
y	amarelo
m	magenta
c	ciano
r	vermelho
g	verde
b	azul
w	branco
k	preto

Símbolo	Estilo de Marcador
.	ponto
o	círculo
x	x
+	sinal positivo
*	estrela
s	quadrado
d	losango
^	triângulo p/cima

Símbolo	Estilo de Linha
-	sólida
:	pontilhada
-.	traço e ponto
--	tracejada

Funções gráficas

Configuração das funções básicas

- **plot(y)** - plota os valores de y no eixo das ordenadas sendo que o eixo x é incrementado de uma unidade para cada valor de y.
- **plot(x,y)** - plota o valor de x no eixo das abscissas e y no eixo das ordenadas.
- **plot(x,y,'r')** - plota o gráfico em cor vermelha referente aos valores de x e y.
- **plot(x,y,'b:d')** - plota o gráfico em cor azul, estilo de linha pontilhada e estilo de marcador em losango.
- **plot(x1,y1,'b:d',x2,y2,'r-o',x3,y3,'k--*')** – plota três curvas na mesma figura. É o mesmo que:

```
plot(x1,y1,'b:d')
hold on
plot(x2,y2,'r-o')
plot(x3,y3,'k--*')
```

Funções gráficas

- **linspace(ínício, fim, N)**
  - ✓ Cria um vetor uniforme de tamanho 'N' começando por 'início' até 'fim'. Muito útil para eixo x (tempo) em plots.

>> t= linspace(0, 1, 5) → vetor com 5 valores de a 0 a 1:

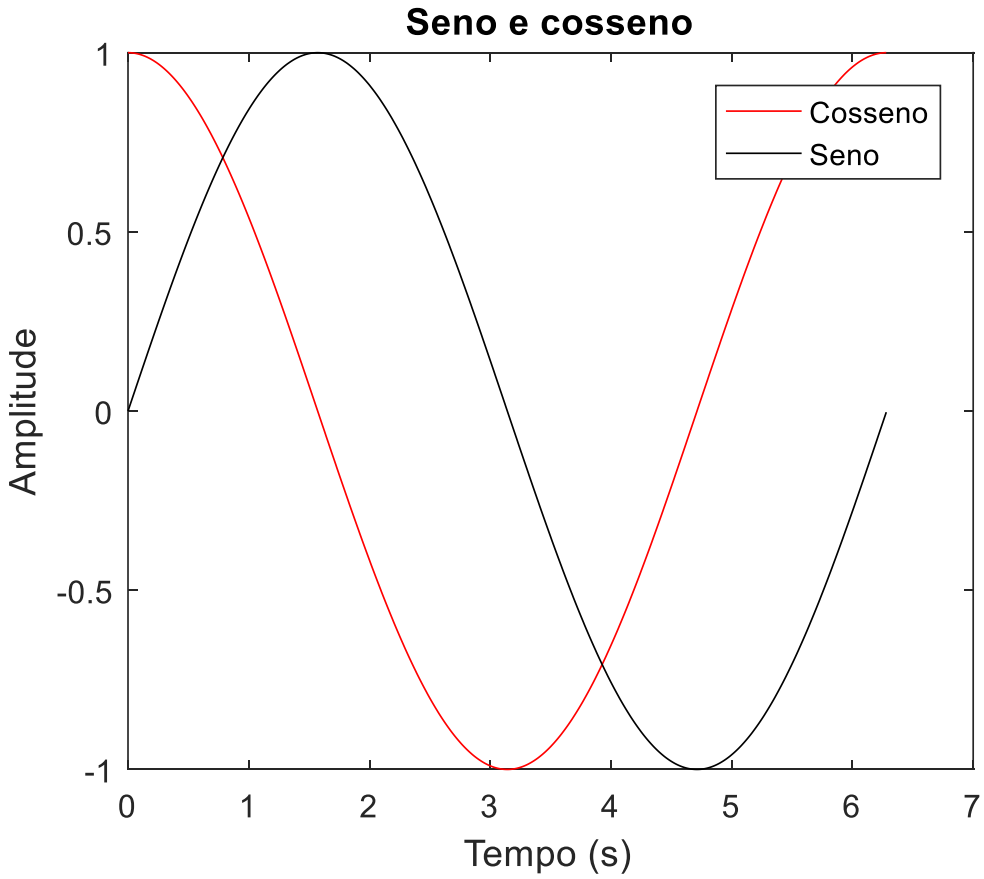
t = 0 0.2500 0.5000 0.7500 0.1000

- Manipulação de textos em gráficos:
  - **title('título')** - insere o título no gráfico.
  - **xlabel('label1')** - insere legenda no eixo x.
  - **ylabel('label2')** - insere legenda no eixo y.
  - **legend('leg1','leg2',...)** - define rótulos para os plots do gráfico.

Exemplo:

```
x=0:0.01:2*pi;
y1=cos(x);
y2=sin(x);

figure
plot(x,y1,'r-');
hold on
plot(x,y2,'k-');
title('Seno e cosseno')
ylabel('Amplitude')
xlabel('Tempo (s)')
legend('Cosseno','Seno')
```



Funções gráficas

Controle de eixos

- **axis([xmin xmax ymin ymax])** – define os valores mínimos e máximos dos eixos com base nos valores fornecidos pelo vetor linha.

- **xlim([xmin xman])** e **ylim([ymin ymax])** – fazem o mesmo que o comando anterior.

- **axis square** - faz com que o gráfico atual tenha a forma de um quadrado em lugar do retângulo habitual.

- **grid on** - mostra as linhas de grade da figura. **Sempre utilizar este comando nos plots.**

Funções gráficas

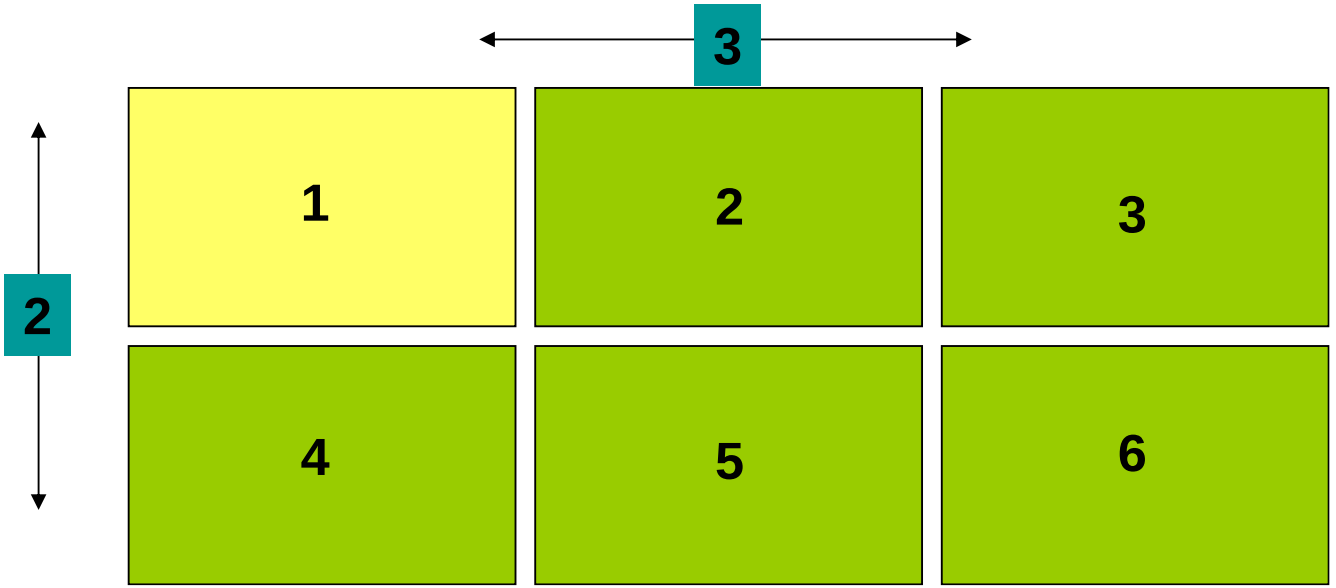
Funções auxiliares para controle de eixos

- **figure(n)** - utilizada para criar figuras numeradas.
- **figure** – o comando cria apenas uma nova figura, iniciando pela figura 1. Novo comando cria figura 2, e assim vai. **Utilizar sempre antes de plotar uma figura! Caso contrário, um novo plot apaga a figura anterior.**
- **close all** - fecha todas as janelas de gráfico.
- **hold on** - mantém os plots da mesma figura fixos para os próximos plots.

Funções gráficas

Trabalhando com sub gráficos

- O comando subplot(m,n,k) subdivide a janela de figuras atual em uma matriz com m por n regiões nas quais se pode traçar gráficos, ativando a região de ordem k.
- Exemplo: subplot(2,3,1) ativa o 1º gráfico em amarelo, dividindo em 2 linhas e 3 colunas. Em seguida, você fazer o plot(.) desejado.
- Com subplot(2,3,6) podemos plotar no último "quadrinho" inferior à direita.



Funções gráficas

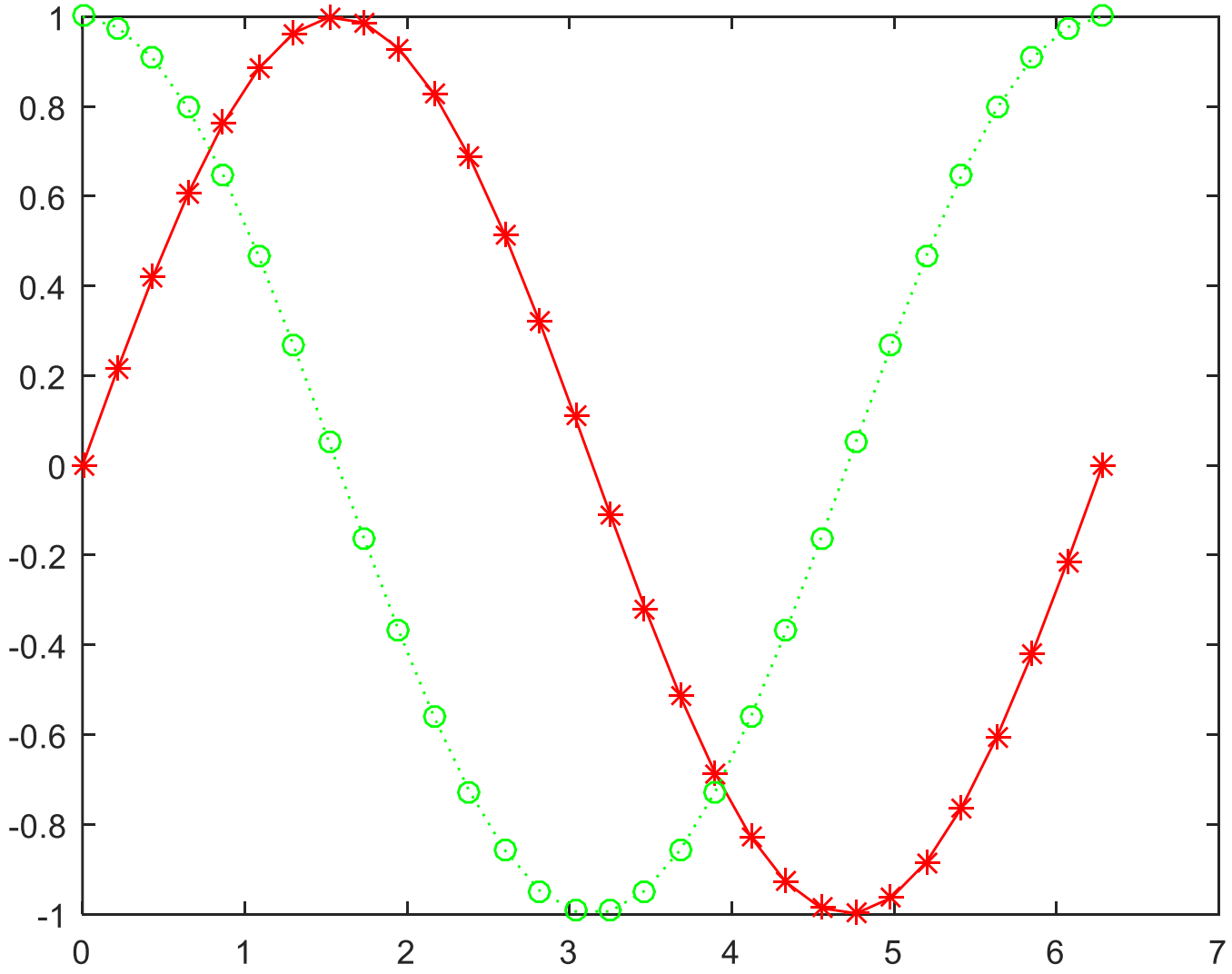
Exercício 1

Para as funções sin(x) e cos(x) faça as seguintes ações:

- Defina os valores de x para ambas entre 0 e 2pi com 30 pontos. (ver slide 24)
- Traçar no mesmo gráfico ambas as funções, tendo as seguintes características (ver slide 23):
- Função sen(x) na cor vermelha, estilo de linha cheia (sólido) e marcador em estrela.
- Função cos(x) na cor verde, estilo de linha pontilhado e marcador em círculo.

Funções gráficas

Exercício 1



Funções gráficas

Exercício 2

- Dada as seguintes funções:
  - 1) f(x) = sen(x);
  - 2) f(x) = cos(x);
  - 3) f(x) = 2*sen(x)*cos(x)
  - 4) f(x) = sen(x)/exp(x)

- Trace cada uma delas em sub-gráficos, com valores de x entre 0 e 5pi/2, tendo ainda este intervalo 500 pontos e eixos dimensionados automaticamente.

- Coloque rótulos nos eixos das ordenadas (y).

Funções gráficas

Exercício

Troque as interrogações '?' pelo código correto e termine as linhas que faltam. Crie um novo script e cole o código abaixo:

```
>> x= linspace( ? )
>> subplot(2,2,1)
>> plot(x,sin(x))
>> ylabel( ? )
>> subplot(2,2,2)
>> plot( ? )
>> ylabel( ?  )
>> subplot(2,2,3)
```

...

Funções gráficas

Exercício 1