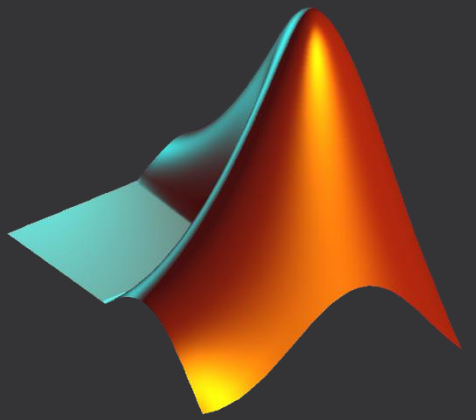


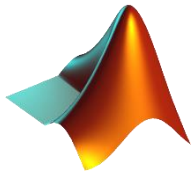
Minicurso de MATLAB Básico

PET – Programa de Educação Tutorial

Engenharia Elétrica

Universidade Federal do Espírito Santo

pet.eletrica.ufes@gmail.com



AULA 1

- O que é Matlab
- Plataforma
- Variáveis e Funções Elementares
- Vetores
- Matrizes

PET
ENGENHARIA ELÉTRICA

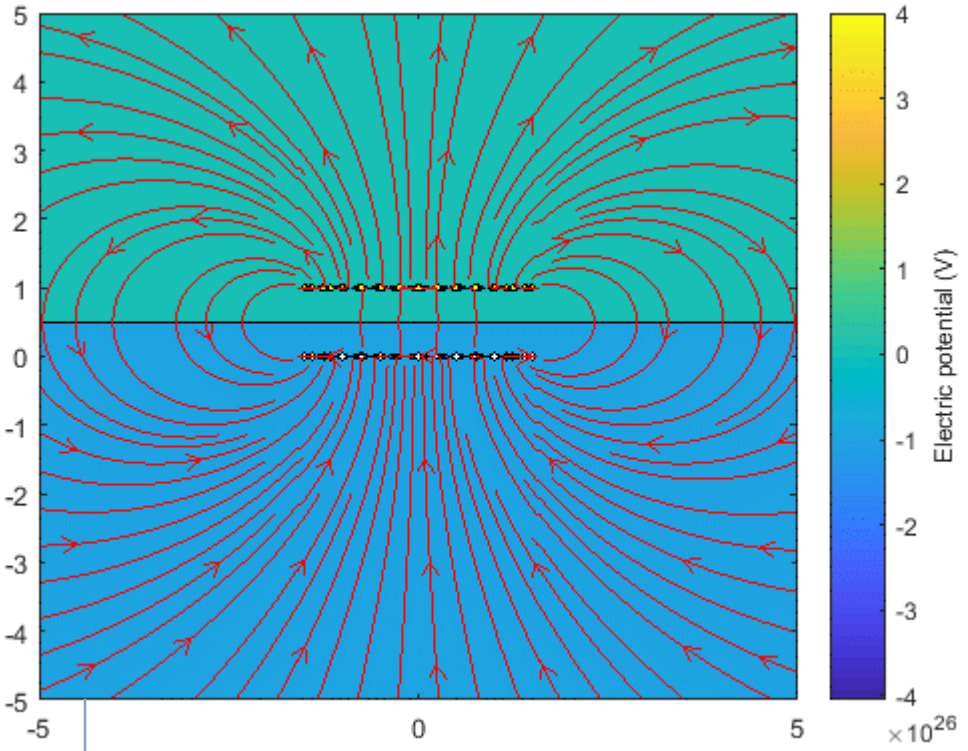
MATLAB

- *Matrix Laboratory*

É um *software*, também conhecido como linguagem de programação, desenvolvido no final da década de 80, que visa principalmente o tratamento de cálculos de matrizes e vetores

- Linguagem interpretada

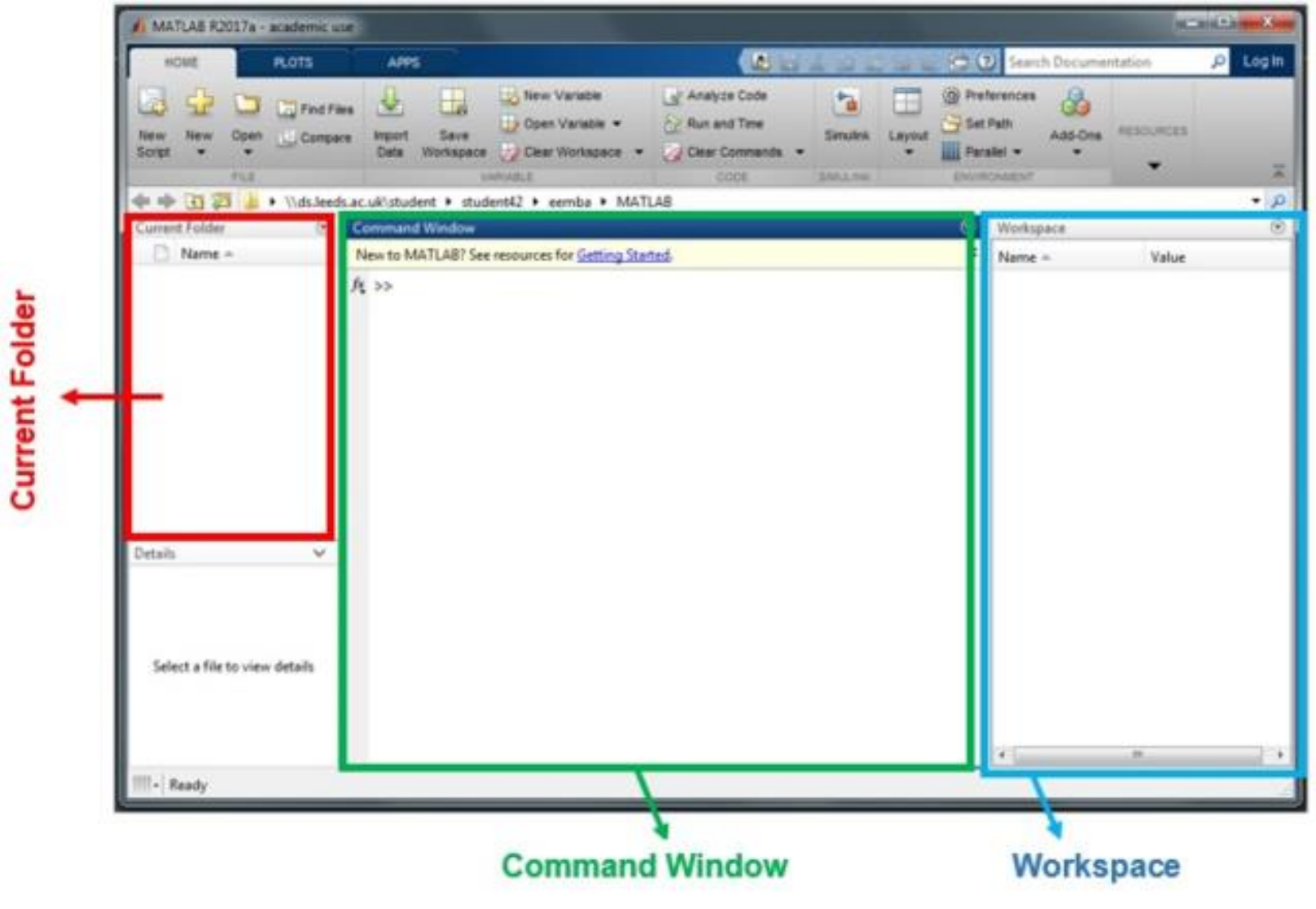
MathWorks



Linhas de campo elétrico dentro de um capacitor de placas paralelas

PET
ENGENHARIA ELÉTRICA

Área de trabalho



Comandos de suporte

- >> help

  Dica: Caso queira usar uma função, mas não sabe se ela foi implementada no MATLAB → Traduza para o inglês e pesquise na documentação do software (doc)

- >> doc função

- >> demo

- Comentar o código: %

PET
ENGENHARIA ELÉTRICA

Variáveis

Tipo	Descrição
Números escalares	Insere números reais.
Pré-definidas	Usadas para cálculos com constantes ou números complexos (por exemplo i, j, pi, ans).
Simbólicas (syms)	Representação de variáveis que não tem necessariamente um valor numérico. Muito usado para fórmulas.
Char	Insere letras.
Struct Array	Vetores cujos os elementos podem ser chamados por instruções.
Cell Array	Vetores com dados não estruturados

PET
ENGENHARIA ELÉTRICA

Variáveis

- Atenção: Algumas variáveis são guardadas. Exemplos:

Variável	Significado
ans	Variável padrão usada para resultados.
pi	Número irracional π
inf	Infinito

- Comandos úteis:
  - >> who / whos
  - >> clc / clear all
  - >> save / load

PET ENGENHARIA ELÉTRICA

Operações Aritméticas

- Expressões Aritméticas básicas

Operação	Forma Algébrica	Matlab
Adição	a + b	a + b
Subtração	a - b	a – b
Multiplicação	a x b	a*b
Divisão à direita	a ÷ b	a/b
Divisão à esquerda	b ÷ a	a\b
Exponencial	a^b	a^b

PET
ENGENHARIA ELÉTRICA

Funções elementares

- Funções úteis para números complexos

Comando	Utilidade
real(x)	retorna a parte real do número complexo x
imag(x)	retorna a parte imaginária do número complexo x
abs(x)	retorna o módulo do número complexo x
angle(x)	retorna a fase, em radianos, no número complexo x

PET
ENGENHARIA ELÉTRICA

Funções elementares

Comando	Utilidade
sin(x) ou sind(x)	seno de x (em radianos ou em graus)
cos(x) ou cosd(x)	cosseno de x (em radianos ou em graus)
tan(x) ou tand(x)	tangente de x (em radianos ou em graus)
asin(x)	arco cujo seno é x
acos(x)	arco cujo cosseno é x
atan(x)	arco cujo tangente é x
round(x)	Arredondamento para o inteiro mais próximo de x

Comando	Utilidade
exp(x)	exponencial e elevado a x
gcd(x,y)	máximo divisor comum de x e y
lcm(x,y)	mínimo múltiplo comum de x e y
log(x)	logaritmo de x na base e (ln)
log10(x)	logaritmo de x na base 10
rem(x,y)	resto da divisão de x por y
sqrt(x)	raiz quadrada de x

PET
ENGENHARIA ELÉTRICA

Funções elementares

Comando	Utilidade
diff(f)	calcula a derivada de f
compose(f,g)	determina a composta f(g(x))
expand(expr)	expande uma expressão expr
finverse(expr)	determina a inversa funcional da expressão expr.
disp('texto')	exibe o texto informado entre aspas simples
sprintf('texto')	associa o texto fornecido entre aspas simples a uma variável e plota seu valor em seguida
simplify(expr)	simplifica a expressão expr
solve(expr)	acha a(s) solução(es) da equação expr = 0
subs(expr,x,a)	substitui na expressão expr a variável x por a
syms x y z a b	define as variáveis simbólicas x, y, z, a e b
root(p, x)	encontra as raízes do polinômio p de x (Obs: Retorna valores simbólicos)
vpa(r)	substitui um valor simbólico por seu equivalente numérico

PET
ENGENHARIA ELÉTRICA

Exercícios de uso

1. Calcule o seno e o cosseno de 45° e a partir das variáveis ache a tangente
2. Defina a expressão $y = x^3 + 2x^2 + 4$ e calcule $\frac{dy}{dx}$
3. Calcule o módulo e a fase de a = 3 + j10
4. Encontre as raízes da expressão $y = x^4 + 5x^2 + 10$
5. Escreva ‘Olá mundo!’

PET
ENGENHARIA ELÉTRICA

Resolução

1) Calcule o seno e o cosseno de 45° e a partir das variáveis ache a tangente

- >> a = sind(45);
- >> b= cond(45);
- >> tang = a/b %tang(x) = sen(x)/cos(x)

- OU :
- >> A= sin(pi/4);
- >> B= cos(pi/4);
- >> TANG= A/B

Resolução

2) Defina a expressão y = x³ + 2x² + 4 e calcule $\frac{dy}{dx}$

- >> syms x y;
- >> y = x^3 + 2*x^2 + 4;
- >> diff(y)fplot(y)

3) Calcule o módulo e a fase de a = 3 + j10

- >> modulo = abs(3 + j*10);
- >> angulo = angle(10/3);
- >> sprintf('O módulo é igual a: %f\nO ângulo é de: %f', modulo, angulo)

Resolução

4) Encontre as raízes da expressão y= $x^4 + 5x^2 + 10$

- >> %simbólica para a forma extensa numérica.
- >> syms x y;y = x^4 + 5*x^2 + 4;
- >> r = root(y);
- >> vpa(r)

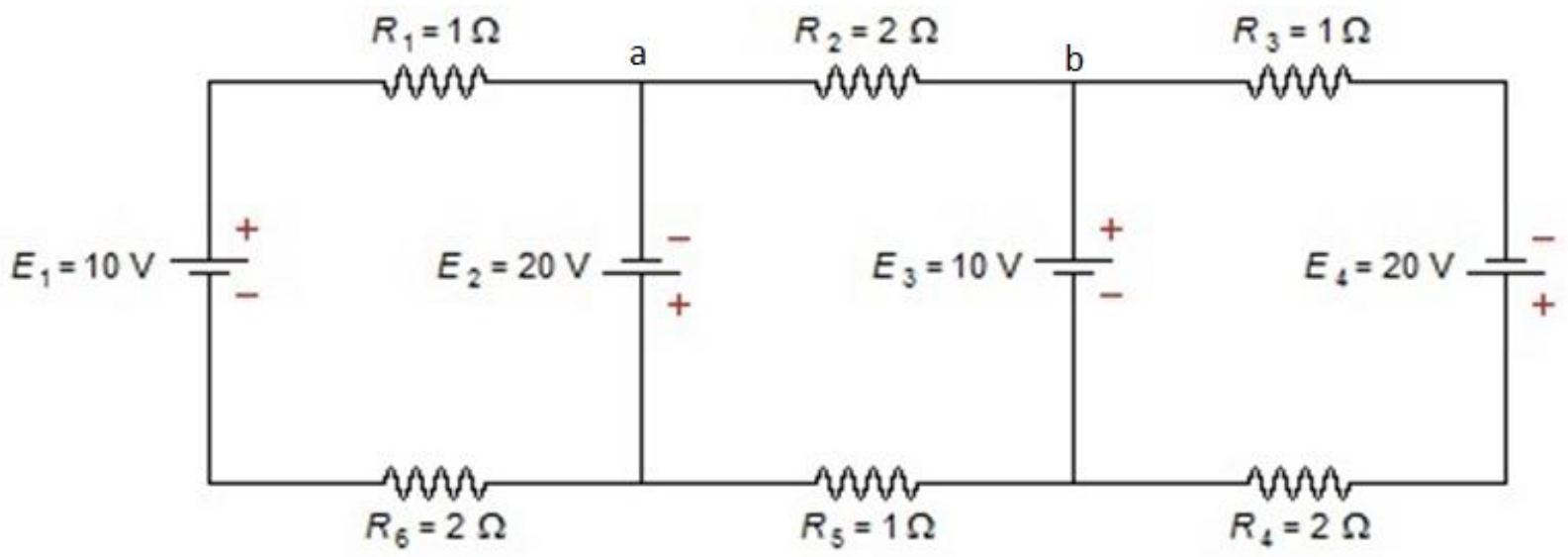
Observação: O comando vpa "traduz" a formatação do comando de sua forma

5) Escreva ‘Olá mundo!’

- >> %Apresenta a string numa formatação prórpia do matlab, só é capaz de printar um vetor por vez. (Uma única entrada).
- >> disp('Hello world’)
- >> %Apresenta a string numa formatação parecida com C
- >> sprintf('Hello world')

Exercícios

- Resolva o circuito abaixo com a Lei de Nó no Matlab.



PET
ENGENHARIA ELÉTRICA

Vetores

- Declarando um vetor

```
>> x1 = [0  2  4  6  8]
>> x1 =
     0     2     4     6     8
```

```
>> x2 = [ pi  2^3  (3+sqrt(81))*8]
    3.1416    8.0000   96.0000
```

- Gerando um vetor

```
>> y = 1 : 5
>> y =
     1     2     3     4     5
```

PET
ENGENHARIA ELÉTRICA

Vetores

- Incrementos (início : incremento : fim)

```
>> z = 0 : pi/4 : pi

>> z =  0.0000  0.7854  1.5708  2.3562  3.1416
```

- Usando o linspace (início, fim, quantidade)

```
>> z = linspace (0,1,6)

>> z = 0  0.2000  0.4000  0.6000  0.8000  1.0000
```

- Usando o logspace

```
>> z = logspace (0,2,3)

>> z = 10^0  10^1  10^2
```

PET ENGENHARIA ELÉTRICA

Vetores

- Observação:

Sendo Y :

>> Y = [0 pi/2 10 -50]

>> Y =

0 1.5708 10 -50

>> lenght(Y)

>> ans =

4

>> c = max(Y)

>> c=

10

>> d = min(Y)

>> d=

-50

PET
ENGENHARIA ELÉTRICA

Vetores - Operações

- Sendo **x** = $[x_1\ x_2\ \ldots\ x_n]$, **y** = $[y_1\ y_2\ \ldots\ y_n]$ e **c** um escalar:

Operação	Expressão	Resultado
Adição escalar	x+c	$[x_1$+c x_2+c ... x_n+c]
Adição vetorial	x+y	$[x_1+y_1\ \ x_2+y_2\ \ \ldots\ \ x_n+y_n]$
Multiplicação escalar	x*c	$[x_1$*c x_2*c ... x_n*c]
Multiplicação vetorial	x.*y	$[x_1*y_1\ \ x_2*y_2\ \ \ldots\ \ x_n*y_n]$
Divisão	x./y	$[x_1/y_1\ \ x_2/y_2\ \ \ldots\ \ x_n/y_n]$
Potenciação	x.^c	$[x_1$^c x_2^c ... x_n^c]
	c.^x	[c^x_1 c^x_2 ... c^x_n]
	x.^y	$[x_1$^y_1 x_2^y_2 ... x_n^y_n]

PET
ENGENHARIA ELÉTRICA

Vetores - Polinômios

- Pode-se representar um polinômio como um vetor linha, contendo os coeficientes do polinômio em ordem decrescente. Exemplo:

>> p = [2 -4 -3 8 -1] teremos: p= 2*x^4 – 4*x^3 – 3*x^2 + 8*x - 1

Função	Descrição
root(p)	raízes de polinômio
poly(p)	cálculo do polinômio característico e remontagem de um polinômio a partir das raízes
conv(p,r)	produto de polinômios
deconv(p,r)	divisão de polinômios
polyder(p)	derivação de polinômios

PET ENGENHARIA ELÉTRICA

Exercícios

1) Crie um vetor coluna contendo os seguintes elementos: ;

sen235°; 6,1; ln292; 0,00552; ln229 e 133

2) Crie um vetor com 9 elementos igualmente espaçados no qual o primeiro elemento é 81 e o último elemento é 12.

3) Use um único comando para criar um vetor linha de zeros (atribua esse vetor a uma variável com o nome a) com 9 elementos e faça com que o último elemento seja 7,5. Não digite o vetor explicitamente.

PET
ENGENHARIA ELÉTRICA

Resolução

1) Crie um vetor coluna contendo os seguintes elementos: ;

sen235°; 6,1; ln292; 0,00552; ln229 e 133

- >> Vetor = [sind(235); 6; 1; log(292); 0.00552; log(229); 133]

2) Crie um vetor com 9 elementos igualmente espaçados no qual o primeiro elemento é 81 e o último elemento é 12.

- >> Vetor = linspace(81,12,9)

Resolução

3) Use um único comando para criar um vetor linha (atribua esse vetor a uma variável com o nome a) com 9 elementos de modo que o último elemento é 7,5 e todos os outros elementos são 0's. Não digite o vetor explicitamente.

- >> a = zeros(1, 9);
- >> a(1,9) = 7.5;
- >> disp(a);

Matrizes

- Entrando com uma matriz

```
>> A=[ 0  9  2;  2  4  6;  5   5  7]
```

```
>> A =

     0     9     2

     2     4     6

     5     5     7
```

```
>> x = [-1.3 sqrt(2) ((1+2+3)*4/5)^2]
```

```
>>  x =

   -1.3000    1.4142   23.0400
```

PET
ENGENHARIA ELÉTRICA

Matrizes

Rand(nº de Linhas, nº de Colunas)

- >> z=rand(3,2)

```
>> z =

    0.1320    0.5752
    0.9421    0.0598
    0.9561    0.2348
```

- >> z=rand(3)

```
>> z =

    0.4709    0.1948    0.2277
    0.2305    0.2259    0.4357
    0.8443    0.1707    0.3111
```

PET
ENGENHARIA ELÉTRICA

Matrizes

- >> z = randperm(15,4)

```
>> z =

     3    13     4     9
```

Vetor de números inteiros aleatórios entre 1 e o número do primeiro argumento. O segundo argumento corresponde a quantidade de números.

- >> z = randn(3,2)

```
>> z =

    0.0700   -1.2226   -1.3429
   -0.6086    0.3165   -1.0322
```

Matriz de elementos aleatórios com distribuição normal com media zero e desvio padrao 1.

PET
ENGENHARIA ELÉTRICA

Matrizes

- >> z = randi(15,3)

>> z =

10	14	7
12	12	7
2	8	5

Matriz de elementos aleatórios uniformemente distribuídos entre 1 e o primeiro argumento.

- >> magic(3)

>> ans =

8	1	6
3	5	7
4	9	2

PET
ENGENHARIA ELÉTRICA

Matrizes - Operações

- Transposta:

```
>> A = [1 2 3; 4 5 6; 7 8 9]

>> A'

ans =

     1     4     7
     2     5     8
     3     6     9
```

- Adição:

```
>> A = [1 2 3; 4 5 6; 7 8 9]      B = [9 8 7; 6 5 4; 3 2 1]

>> A + B

ans =

    10    10    10
    10    10    10
    10    10    10
```

PET
ENGENHARIA ELÉTRICA

Matrizes - Operações

- Subtração:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
   B = [9 8 7; 6 5 4; 3 2 1]

>> A - B

ans =

    -8    -6    -4
    -2     0     2
     4     6     8
```

- Multiplicação:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
   B = [9 8 7; 6 5 4; 3 2 1]

>> A*B

ans =

    30    24    18
    84    69    54
   138   114    90
```

PET ENGENHARIA ELÉTRICA

Matrizes - Operações

- Divisão a direita:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
   B = [5 2 10; 9 3 11; 8 7 15]

        >> A/B

        ans =

-0.0662    -0,1985    0.3897

-0.9485    0.1544     0.9191

-1.8309    0.5074     1.4485
```

- Divisão a esquerda:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
   B = [5 2 10; 9 3 11; 8 7 15]

        >> A\B

        ans =

0.5294     0.5294     0.5294

1.3088     1.1838     1.0588

-0.4265    -0.3015    -0.1765
```

PET
ENGENHARIA ELÉTRICA

Matrizes - Operações

- Inversão:

```
>> A = [7 13 11; 5 2 7; 21 9 10]
>> inv(A)
ans =
-0.0433   -0.0312   0.0695
-0.0977   -0.1621   0.0060
0.0030    0.2115   -0.0514
```

- Determinante:

```
>> A = [7 13 11; 5 2 7; 21 9 10]
>> det(A)
ans =
   993.0000
```

PET
ENGENHARIA ELÉTRICA

Matrizes - Operações

- Potenciação:

```
>> A = [1 2 3; 4 5 6; 7 8 9]                                    B =
[9 8 7; 6 5 4; 3 2 1]

>> A.^B

ans =

     1   256  2187
  4096  3125  1296
   343    64     9
```

PET
ENGENHARIA ELÉTRICA

Matrizes - Referência de elementos

- Seja a matriz A:

```
>> A = [1 2 3; 4 5 6; 7 8 9]

>> B=A(:,3)

B =

     3

     6

     9
```

```
>> E=A(2:3,:)
 E =

     4     5     6

     7     8     9
```

```
>>  C=A(2,:)

C =

     4     5     6
```

PET
ENGENHARIA ELÉTRICA

Matrizes - Funções

Função	Descrição
sum	Soma dos elementos de uma mesma coluna
prod	Produto dos elementos de uma mesma coluna
mean	Média aritmética dos elementos de uma mesma coluna
std	Desvio padrão dos elementos de uma mesma coluna
max	Maior elemento de uma mesma coluna
min	Menor elemento de uma mesma coluna
sort	Ordena os elementos de uma mesma coluna em ordem crescente
size	Retorna as dimensões da matriz
rank	Determina o posto
rref	Retorna a forma escalonada reduzida (por linhas)

PET
ENGENHARIA ELÉTRICA

Matrizes Elementares

- >> A = ones(3,4)

```
>> A =

     1     1     1     1
     1     1     1     1
     1     1     1     1
```

- >> A = zeros(2,3)

```
>> A =

     0     0     0
     0     0     0
```

Matrizes Elementares

- >> A = eye(3,4)

```
>> A =

     1     0     0     0
     0     1     0     0
     0     0     1     0
```

Matrizes - Funções

- Para o cálculo dos autovetores e autovalores:

```
>> A = [ 2 5 7; 3 4 6; 9 2 1]
>> eig(A)
        ans =
            13.0120
            -5.6151
            -0.3969
```

- Uso dos dois pontos (:) : para significar “todos” ou “até”

```
>> A = [ 2 5 7; 3 4 6; 9 2 1]
>> A(:,3)
 ans =
      7
      6
      1
```

```
>> A(1:2,1:3)
 ans =
      2     5     7
      3     4     6
```

PET
ENGENHARIA ELÉTRICA

Exercícios de Matrizes

1) Dada uma matriz M (5 x 7), aleatória, mostrar:

- o maior elemento de cada coluna da matriz;
- o maior elemento de cada linha da matriz;
- a média dos elementos de cada coluna;
- o produto de todos os elementos diferentes de zero;
- quantos elementos são negativos;
- posição ocupada (linha-coluna) por um elemento cujo valor será lido pelo programa.

PET
ENGENHARIA ELÉTRICA

Resolução

1) Dada uma matriz M (5 x 7), preenchê-la por leitura e mostrar:

```
>> M = rand(5,7);                    %criação da matriz aleatória
```

- o maior elemento de cada coluna da matriz;

```
>> max(M)
```

- o maior elemento de cada linha da matriz;

```
>> max(max(M))
```

- a média dos elementos de cada coluna;

```
>>  mean(M)
```

- o produto de todos os elementos diferentes de zero;
- >> for i:=1 to MAXLIN do
- for j:=1 to MAXCOL do
- if M[i,j]<>0 then produto:=produto*M[i,j];
- writeln('O produto dos elementos nao nulos eh ',produto);
- quantos elementos são negativos;
- >> for i:=1 to MAXLIN do
- for j:=1 to MAXCOL do
- if M[i,j]<0 then neg:=neg+1;
- writeln('O numero de elementos negativos eh ',neg);
-

AULA 2

- Scripts M-file
- Funções M-file
- Operadores lógicos e relacionais
- Comandos de fluxo

Editor - C:\Users\Matheus Lessa Boy\Desktop\S'matlab\ExemploParaSerPrintado.m

ExemploParaSerPrintado.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
1   %Esse script plota duas circunferências para os raios dados e coordenadas
2   %dos centros
3
4   %%
5   %Primeiro passo: adquirir os raios e coordenadas
6   raio1 = input('Por favor entre com o valor do raio da primeira circunferência: ');
7   raio2 = input('Por favor entre com o valor do raio da segunda circunferência: ');
8   cord1 = input('Por favor entre com as coordenadas para o centro da primeira circunferência: ');
9   cord2 = input('Por favor entre com as coordenadas para o centro da segunda circunferência: ');
10
11  %%
12  %Segundo passo: gerar as coordenadas dos pontos das dadas circunferências em coordenadas
13  %polares
14  theta = 0:0.001:2*pi;
15  x1 = raio1*cos(theta);
16  y1 = raio1*sin(theta);
17  cordc1 = [x1 ; y1];
18  x2 = raio2*cos(theta);
19  y2 = raio2*sin(theta);
20  cordc2 = [x2 ; y2];
21
22  %%
23  %Terceiro passo: deslocar os pontos do segundo passo para garantir as
24  %coordenadas do centro
25
26  cordc1(1,:) = cordc1(1,:) + cord1(1,1);
```

PET ENGENHARIA ELÉTRICA

Arquivos de texto que contêm comandos MATLAB

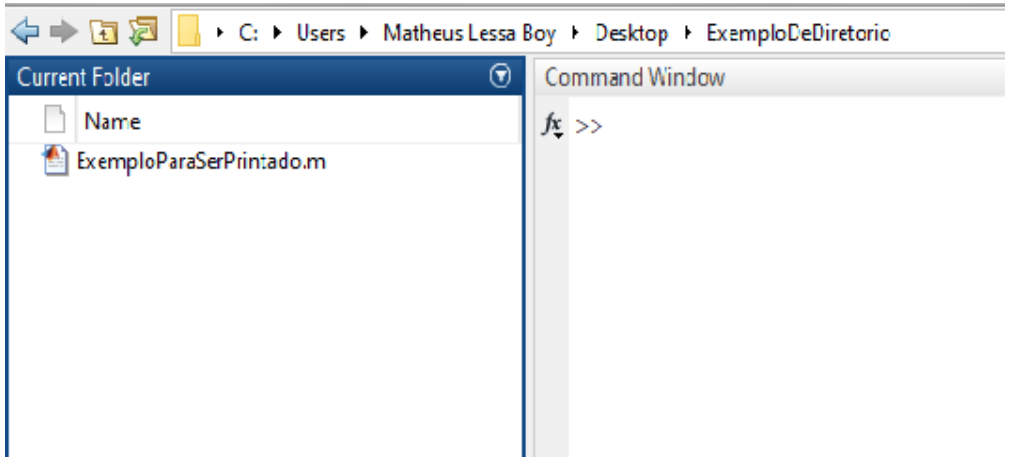
Scripts M-file

- Um “roteiro”, sequência de comados, para uma aplicação.
- Útil para testes em sequência ou rotinas extensas e repetitivas.
- Exemplos:
  - Processamento de imagens para contagem de objetos;
  - Plotagens gráficas para estudo com diferentes entradas;
  - Etc.

PET ENGENHARIA ELÉTRICA

Scripts M-file

- Certificar de que está no diretório que deseja trabalhar:



- Para carregar e executar o script selecionado é preciso estar no diretório onde ele está salvo.

PET
ENGENHARIA ELÉTRICA

Scripts M-file

- Passos para criação de um script:
  - Criar o arquivo.m:

Editor - C:\Users\Matheus Lessa Boy\Desktop\ExemploDeDiretorio\ExemploParaSerPrintado.m

ExemploParaSerPrintado.m

```
%Esse script plota duas circunferências para os raios dados e coordenadas
%dos centros

%%
%Primeiro passo: adquirir os raios e coordenadas
raio1 = input('Por favor entre com o valor do raio da primeira circunferência: ');
raio2 = input('Por favor entre com o valor do raio da segunda circunferência: ');
cord1 = input('Por favor entre com as coordenadas para o centro da primeira circunferência: ');
cord2 = input('Por favor entre com as coordenadas para o centro da segunda circunferência: ');

%%
%Segundo passo: gerar as coordenadas dos pontos das dadas circunferências em coordenadas
%polares
theta = 0:0.001:2*pi;
x1 = raio1*cos(theta);
y1 = raio1*sin(theta);
cordc1 = [x1 ; y1];
x2 = raio2*cos(theta);
y2 = raio2*sin(theta);
cordc2 = [x2 ; y2];

%%
```

PET ENGENHARIA ELÉTRICA

Funções para o M-file

Função	Descrição
disp(‘texto’)	Exibi textos e variáveis sem escrever o nome da variável.
Echo	Controla o eco dos comandos, presentes no m-file, que vão sendo executados (echo on e echo off).
input(‘texto’)	Entrada de valor para variável.
publish(‘script.m’, ’formato’)	Exporta o arquivo .m em MATLAB para o formato de arquivo desejado.
keyboard	Interrompe a execução de um m-file dando liberdade ao utilizador para executar outros comandos. Retoma-se a execução do m-file através de dbcont.
fprintf(‘texto’)	Exibe textos e variáveis com opção de formatação da saída. Também usado para gravar dados em arquivos.
pause(n)	Há uma pausa de n segundos na execução.
waitforbuttonpress	Existe uma pausa na execução do m-file até que se carregue numa tecla do mouse ou do teclado.

PET ENGENHARIA ELÉTRICA

Funções para o M-file

Função	Descrição
disp(‘texto’)	disp(‘Qual o seu CPF?’);
Echo	echo on e echo off
input(‘texto’)	X = input('Por favor entre com o valor de X = ');
publish(‘script.m’, ’formato’)	Manualmente ou pela janela de comando.
keyboard	keyboard
fprintf(‘texto’)	fprintf(‘Ola pessoal esse é o meu numero %d’, meuNumero);
pause(n)	pause(2)
waitforbuttonpress	Existe uma pausa na execução do m-file até que se carregue numa tecla do mouse ou do teclado.

PET
ENGENHARIA ELÉTRICA

Scripts M-file

- Passos para criação de um script:
  - Executar o script .m:

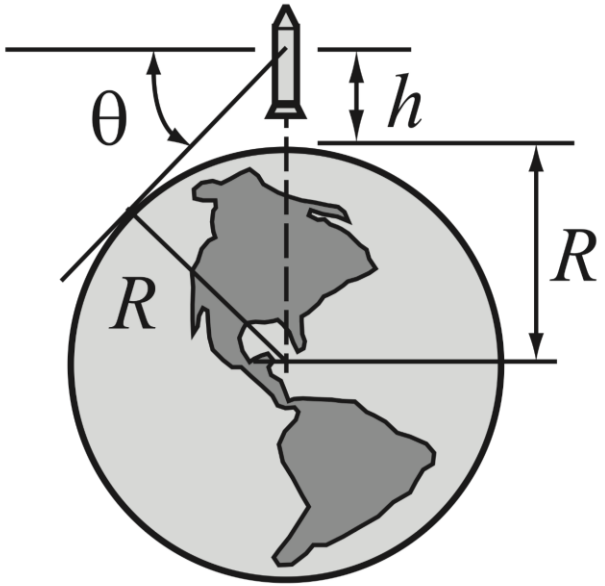
```
%%
%Primeiro passo: adquirir os raios e coordenadas
raio1 = input('Por favor entre com o valor do raio da primeira circunfe
raio2 = input('Por favor entre com o valor do raio da segunda circunfe:
cord1 = input('Por favor entre com as coordenadas para o centro da pri
cord2 = input('Por favor entre com as coordenadas para o centro da segu

%%
%Segundo passo: gerar as coordenadas dos pontos das dadas circunferênc:
%polares
theta = 0:0.001:2*pi;
x1 = raio1*cos(theta);
y1 = raio1*sin(theta);

ot muito simples serve para mostrar alqumas propri
```

PET Engenharia Elétrica

Script M-file

- Exemplo 1:

Um foguete partindo do solo mede o ângulo θ com o horizonte em diferentes alturas h. Escreva um script MATLAB que calcule o raio R da terra (considerando-a uma esfera perfeita) em cada ponto dado e encontre a média desses valores.



Através de uma análise trigonométrica concluímos que

$$R = \frac{h\cos\theta}{(1-\cos\theta)}$$

Logo, colocando em função de R temos:

$$R\cos\theta - R = h\cos\theta$$

h (km)	4	8	12	16	20	24
θ (graus)	2.0	2.9	3.5	4.1	4.5	5.0

PET
ENGENHARIA ELÉTRICA

Script M-file

- Exemplo 1:

```
 7 -     theta = [ 2.0 2.9 3.5 4.1 4.5 5.0 ];
18
19      %%
20      %Plotando os resultados
21
22 -    tabel = [ R ; h ; theta ];
23 -    fprintf('Valor médio de R: %4.2f  km \n\t\t\tValores de R\t\t\tValores de h\t\t\tValores de theta\n',mediaR);
24 -    fprintf('\t\t\t %4.2f km\t\t\t\t   %2.0f km\t\t\t\t    %1.1f graus\n',tabel);
25
```

```
17 -    mediaR = mean(R);
18
```

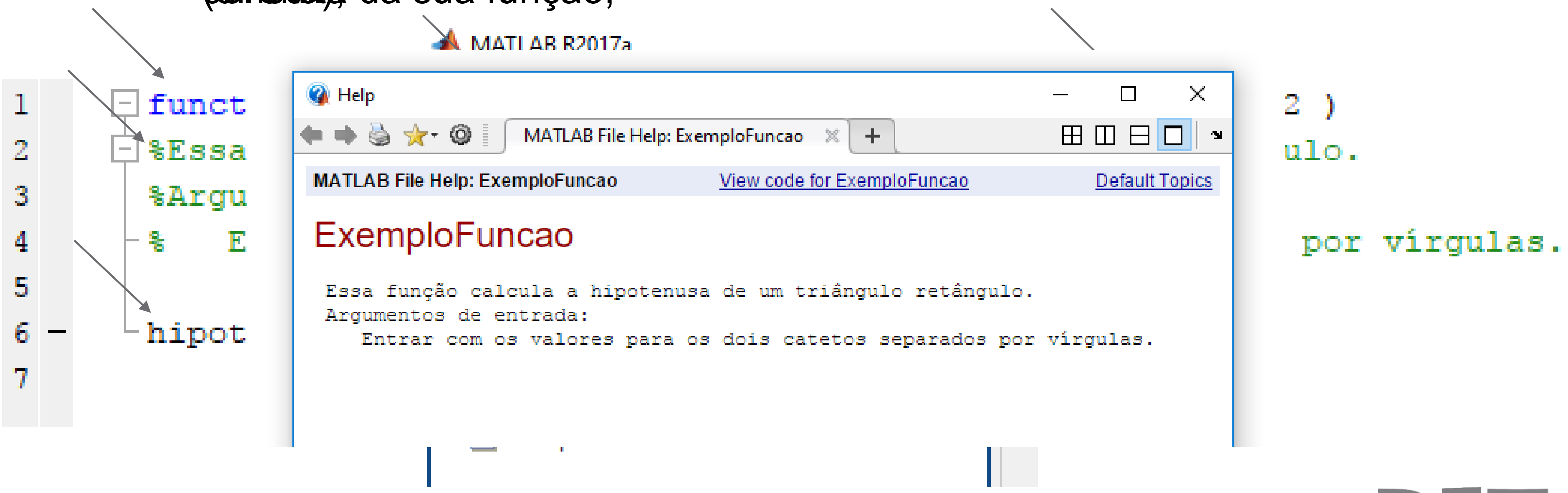
PET
ENGENHARIA ELÉTRICA

Funções M-file

- Tipos de arquivos .m que possuem variáveis locais (seus valores não ficam gravados no workspace);
- Usados para a definição de funções de usuários permitindo a personalização e especificação para o tratamento de problemas;
- Podem ser chamadas em scripts ou no próprio command Windows.

Funções M-file

- Passos para a criação de uma função:
  - [illegible] da sua função;



PET
ENGENHARIA ELÉTRICA

Funções M-file

- Exemplo 2:

Editor - ex2.m

ex2.m

```
1   function [Tensoes,Potencias] = ex2(R1,R2,R3,Vs)
2   %Essa função resolve o exemplo 2 da aula 2.
3   %Entradas: Resistores e tensão.
4
5   %Cálculo Corrente
6   I=Vs/(R1+R2+R3);
7
```

PET
ENGENHARIA ELÉTRICA

Funções M-file

- Exemplo 2:

```
%Tensão de cada resistor
TR1= I*R1;
TR2= I*R2;
TR3= I*R3;

%Potência de cada resistor
PR1= TR1*I;
PR2= TR2*I;
PR3= TR3*I;

%Valores de Saida
Tensoes = [TR1 TR2 TR3];
Potencias = [PR1 PR2 PR3];
```

PET ENGENHARIA ELÉTRICA

Comandos de fluxo

- Muitas vezes os dados precisam ser tratados com distintas condições ou em loops que seriam inviáveis, se não impossíveis, de serem feitos a mão;

- Exemplo:

    - Delimitação de dados quanto a um determinado valor;

PET
ENGENHARIA ELÉTRICA

Operadores lógicos e relacionais

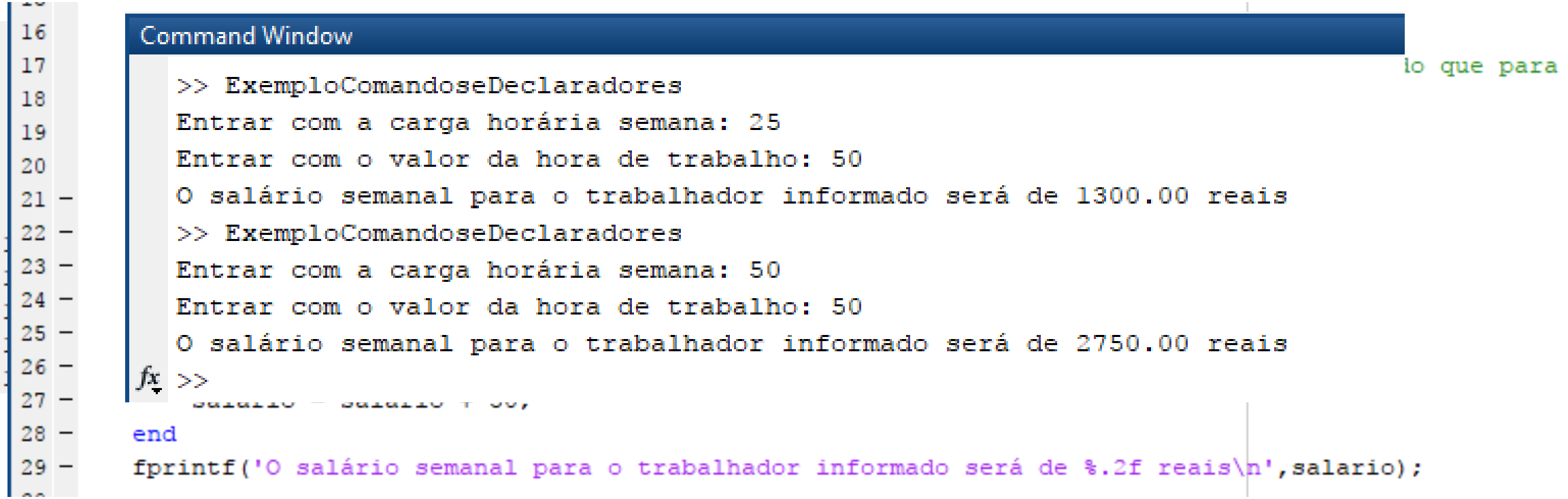
Operador	Descrição
<	Menor que
<=	Menor ou igual que
>	Maior
>=	Maior ou igual que
==	Igual a
~=	Diferente de

Operador	Descrição
&	E
|	Ou
~	Não

PET
ENGENHARIA ELÉTRICA

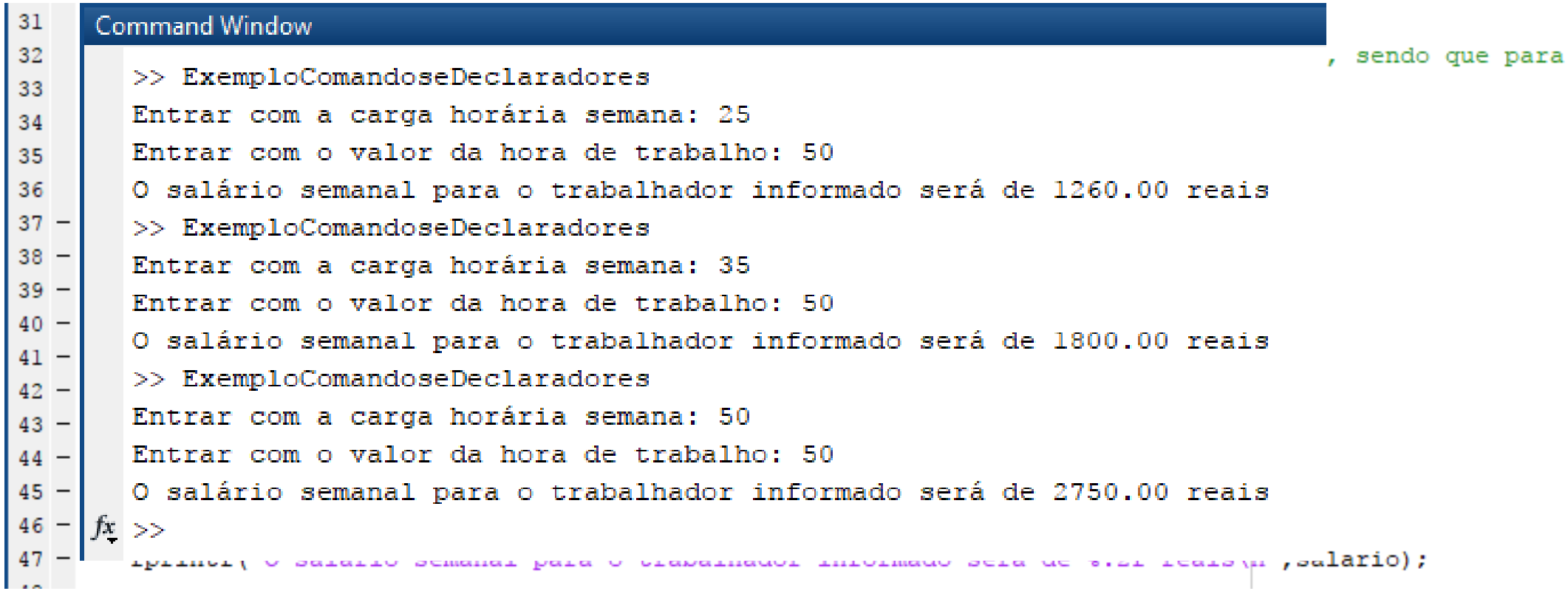
Declaradores Condicionais

- Laço *if-else*:

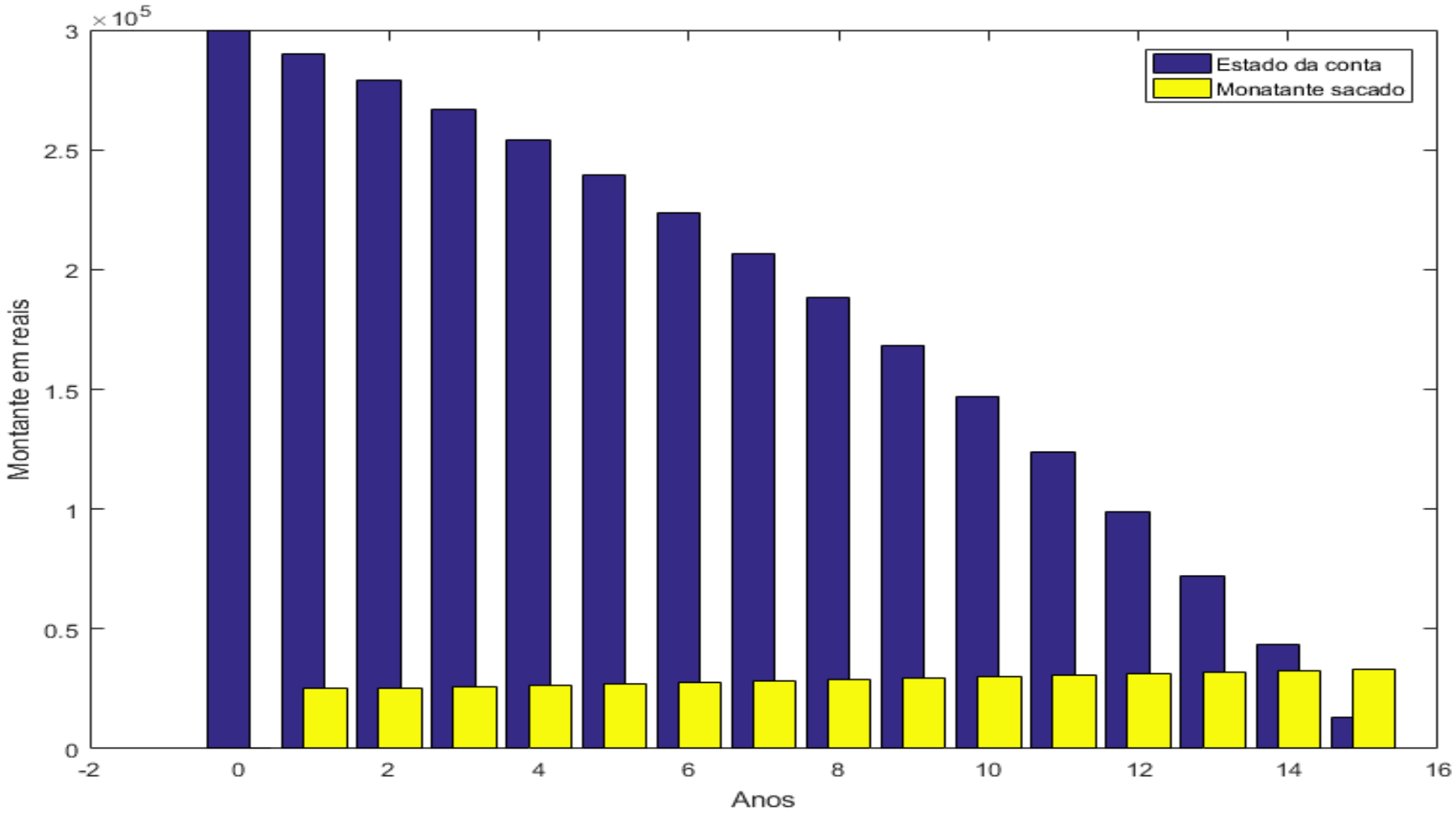


Declaradores Condicionais

- Laço *if-elseif-else* :

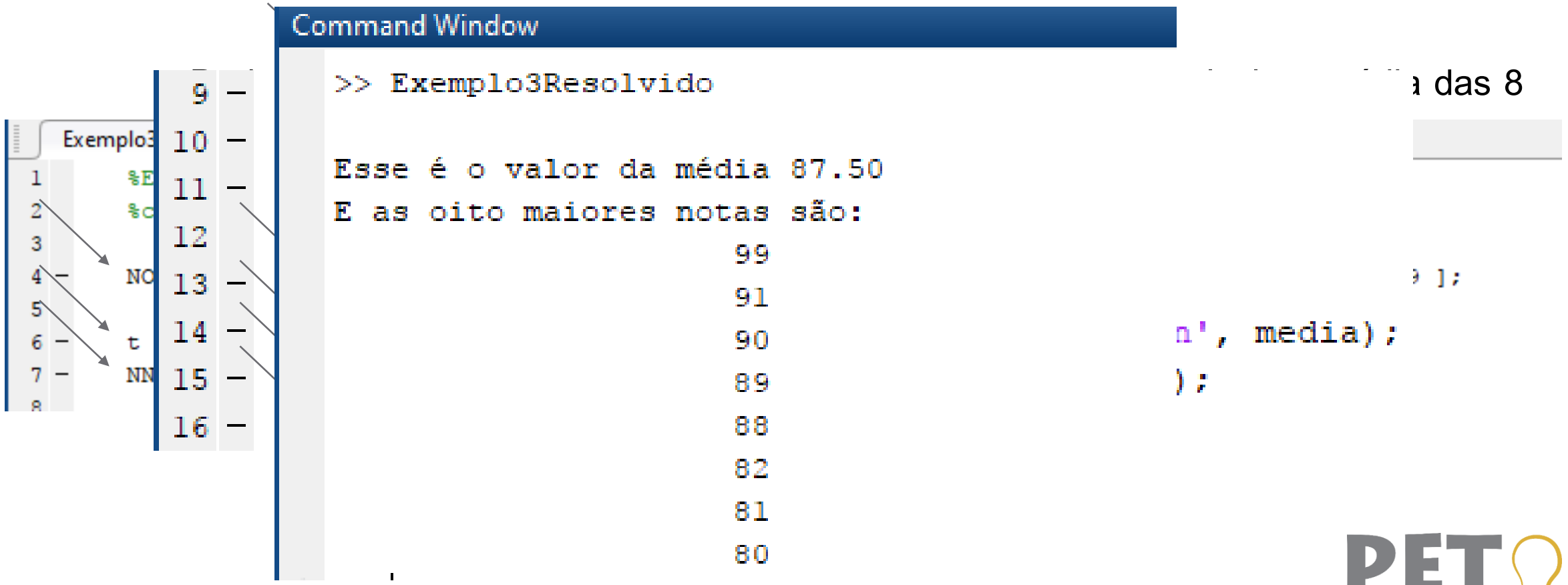


Loops


×10^5
3
2.5
2
1.5
1
0.5
0
Montante em reais
-2
0
2
4
6
8
10
12
14
16
Anos
Estado da conta
Monatante sacado

Loops

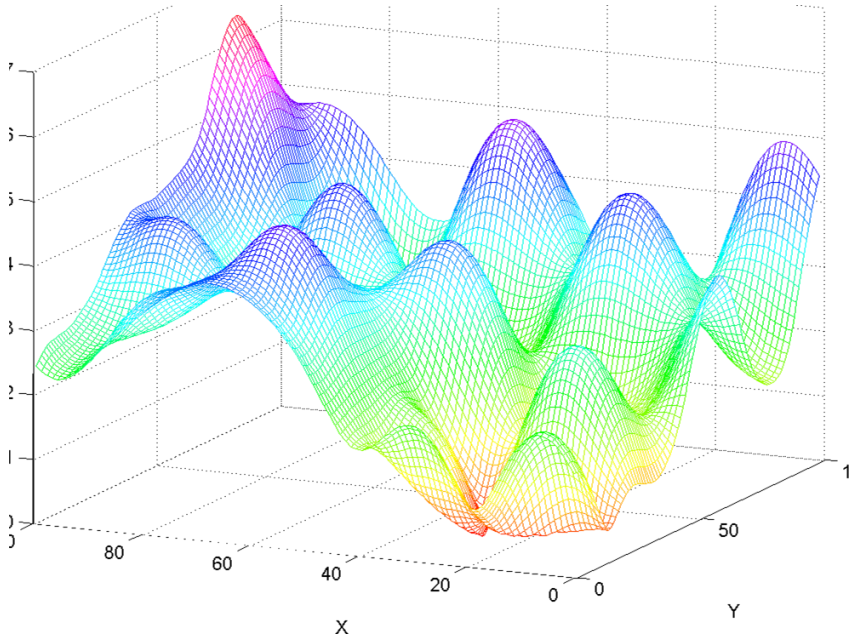
- Exemplo 3:



PET
ENGENHARIA ELÉTRICA

AULA 3

- Gráficos 2D
- Gráficos 3D
- Comandos importantes



PET
ENGENHARIA ELÉTRICA

Gráficos 2D

- Plot

Plot é uma função para se “plotar” gráficos contínuo 2D.

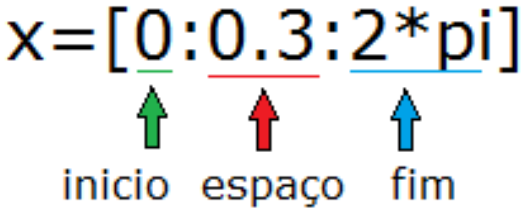
plot(abscissa , ordenada)

plot(x , y)

1° - Definir os intervalos do eixo das abscissas e o espaço entre um ponto e outro.

T=[inicio:espaço:fim]

Ex:



Gráficos 2D

- Plot

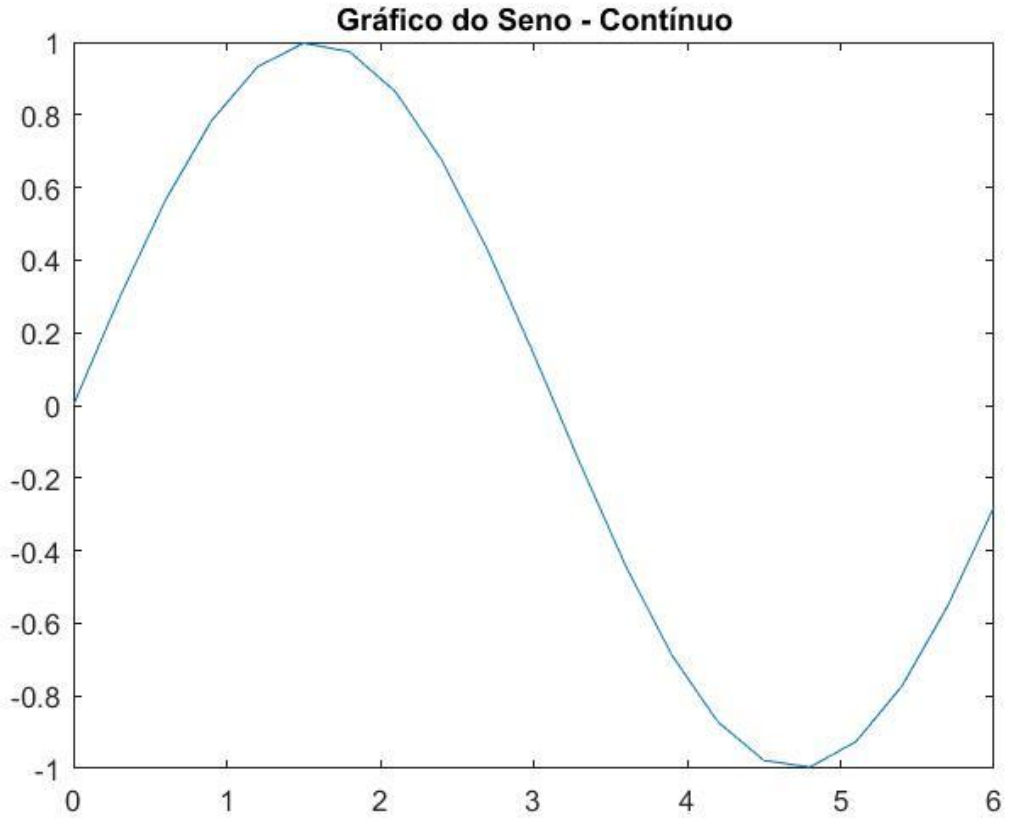
2° - Definir a função

3º - Utilizar o comando plot

plot(abscissa , ordenada)

Ex:

```
x = 0:.3:2*pi;
y = sin(x);
plot(x, y)
title('Gráfico do Seno - Contínuo');
```



Gráficos 2D

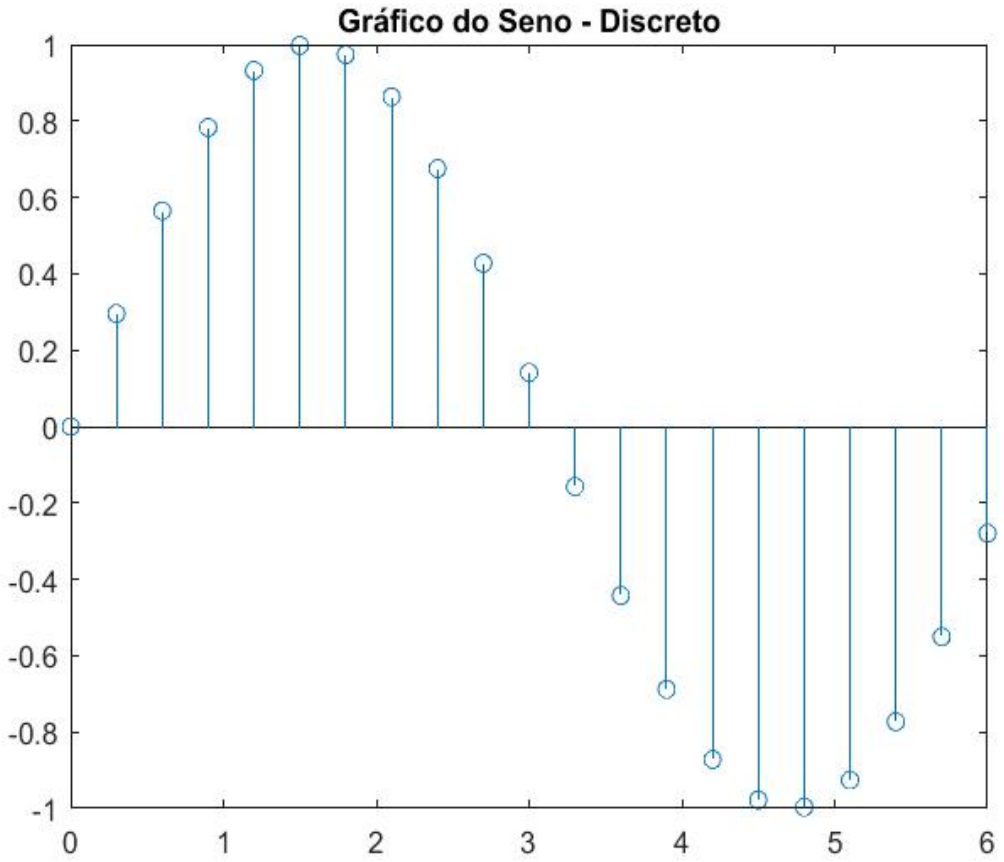
- Stem

Stem é uma função para se “plotar” gráficos discreto 2D.

stem(abscissa , ordenada)

stem(x , y)

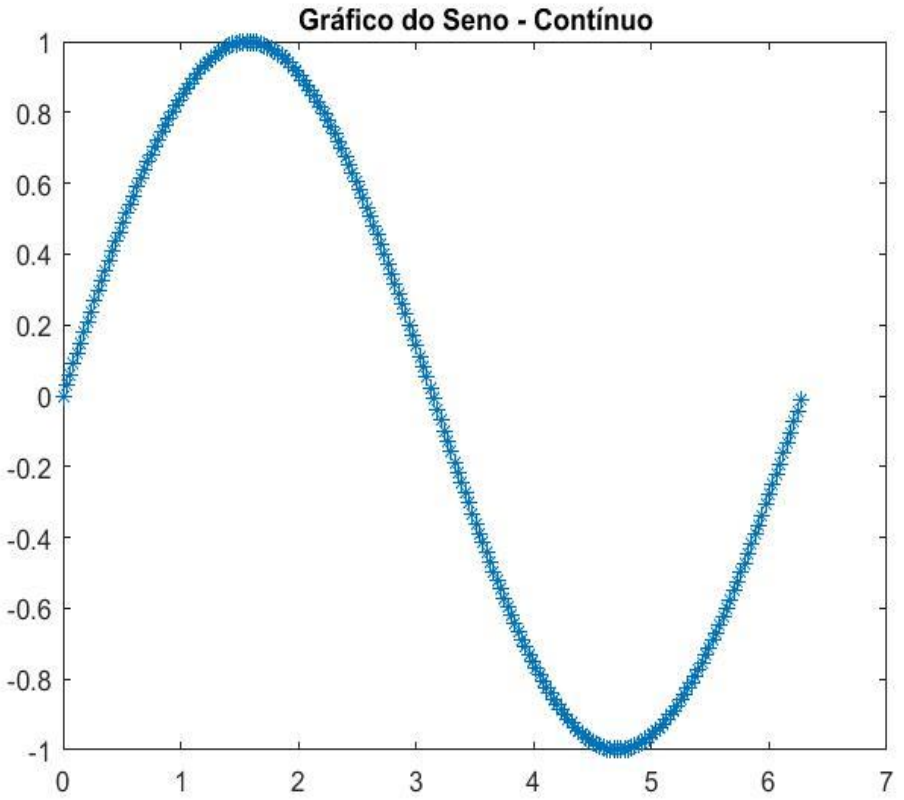
```
x = 0:.3:2*pi;
y = sin(x);
stem(x, y)
title('Gráfico do Seno - Discreto');
```



Gráficos 2D

- Detalhes de plotagem:

Título
title(‘Seu título’)
red yellow magenta green blue
Eixos
xlabel(‘Nome do eixo X’)
ylabel(‘Nome do eixo Y’)
plot(x,y,’:’)
Grade
grid(‘on’) ou grid(‘off’)
plot(x,y,’O’)
Hold
hold(‘on’) ou hold(‘off’)



Outros detalhes de plotagem

Símbolo	Cor
b	Azul
g	Verde
r	Vermelho
c	Ciano
m	Magenta
y	Amarelo
k	Preto
w	Branco

Símbolo	Tipo de linha
-	Contínua
:	Pontilhada
-.	Traço e ponto
--	Tracejada

Outros detalhes de plotagem

Símbolo	Marca
.	Ponto
o	Círculo
x	Xis
s	Quadrado
d	Losango
v	Triângulo pra baixo
^	Triângulo pra cima
p	Pentagrama
h	Hexagrama
<	Triângulo pra esquerda
>	Triângulo pra direita

Outros detalhes de plotagem

Comando	Utilidade
grid	Coloca linhas de grade no gráfico
title	Permite acrescentar um título ao gráfico
xlabel	Permite acrescentar um título no eixo das abscissas
ylabel	Permite acrescentar um título no eixo das ordenadas
legend(y1, y2,..)	Relaciona um tipo de linha a uma curva
hold (on/off)	Não apaga o gráfico atual antes de fazer o seguinte
axis ([Xmim Xmax Ymim Ymax])	Permite setar o tamanho do gráfico

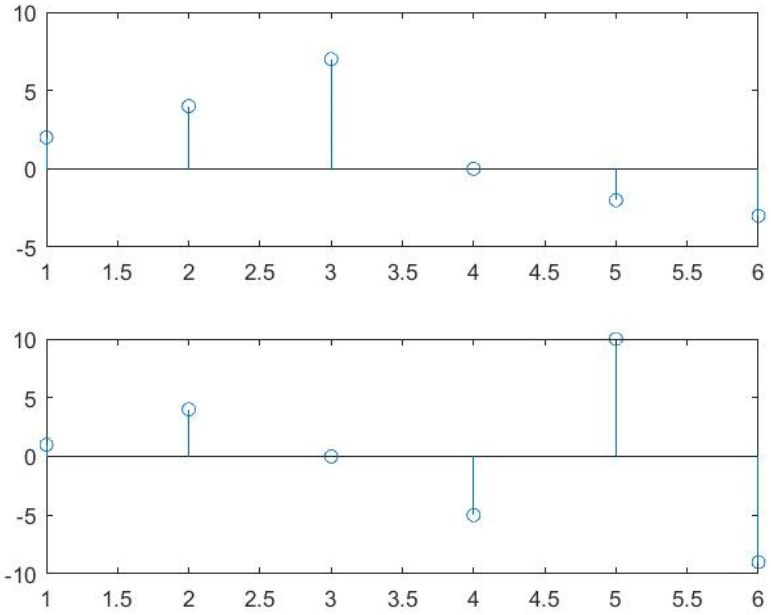
Outros detalhes de plotagem

- Subplot

Plota mais de um gráfico dentro de uma mesma figure, mas posicionado em eixos diferentes.

subplot(m, n, p), com m, e n sendo a quantidade de linhas e colunas em que a figure será dividida e p a posição da figura nessa matriz.

```
%Exemplo do uso de um subplot
clc, clear variables, close all;
A = [2, 4, 7, 0, -3, -5];
B = [3, 4, 0, -5, 10, -8];
set(gcf, 'Color', 'White');
subplot(2, 1, 1);
stem(A);
subplot(2, 1, 2);
stem(B);
```



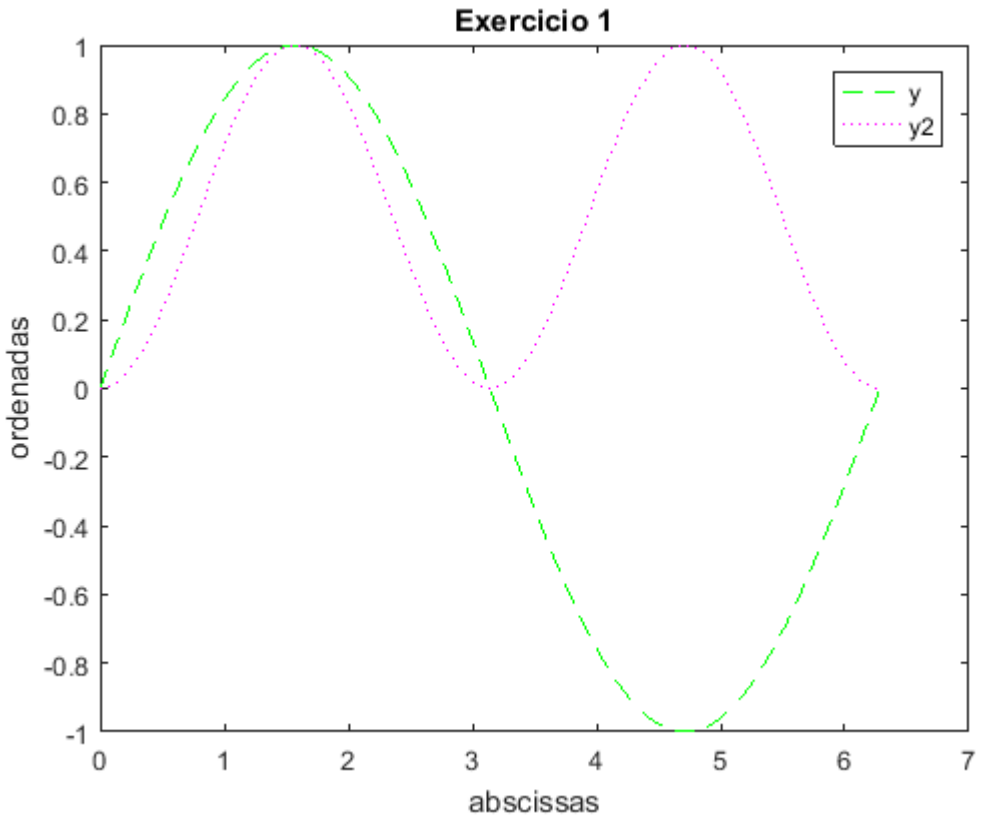
Exercício

1. Plote o gráfico do sen(x) e do $sen^2(x)$. Os gráficos devem ser plotados num mesmo plano sendo que o gráfico do sen(x) deverá ser verde e tracejado, enquanto o gráfico do $sen^2(x)$ deverá ser rosa e pontilhado.
2. Plote o gráfico discreto da função $\text{sen}(\frac{\pi}{2}k)$ com $-5 \leq k \leq 5$

- PS: Acrescente título, dê nome aos eixos e legenda ao gráfico.

Resposta – Exercício 1

```
%%Exercicio 1
x=linspace(0,2*pi,1000);
y=sin(x);
y2=sin(x).^2;
plot(x,y,'g--',x,y2,'m:');
legend('y','y2');
title('Exercicio 1');
xlabel('abscissas');
ylabel('ordenadas');
fig=figure(1);
set(fig,'color','w');
```



Resposta – Exercício 2

```
% --- Seno discreto ---- %
clear all;
k=-5:5;
y=sin((pi/2)*k);
stem(k,y,'g');
fig=figure(1);
set(fig,'color','w');
title('Seno discretizado');
legend('y');
xlabel('valores de k');
ylabel('valores de y');
```

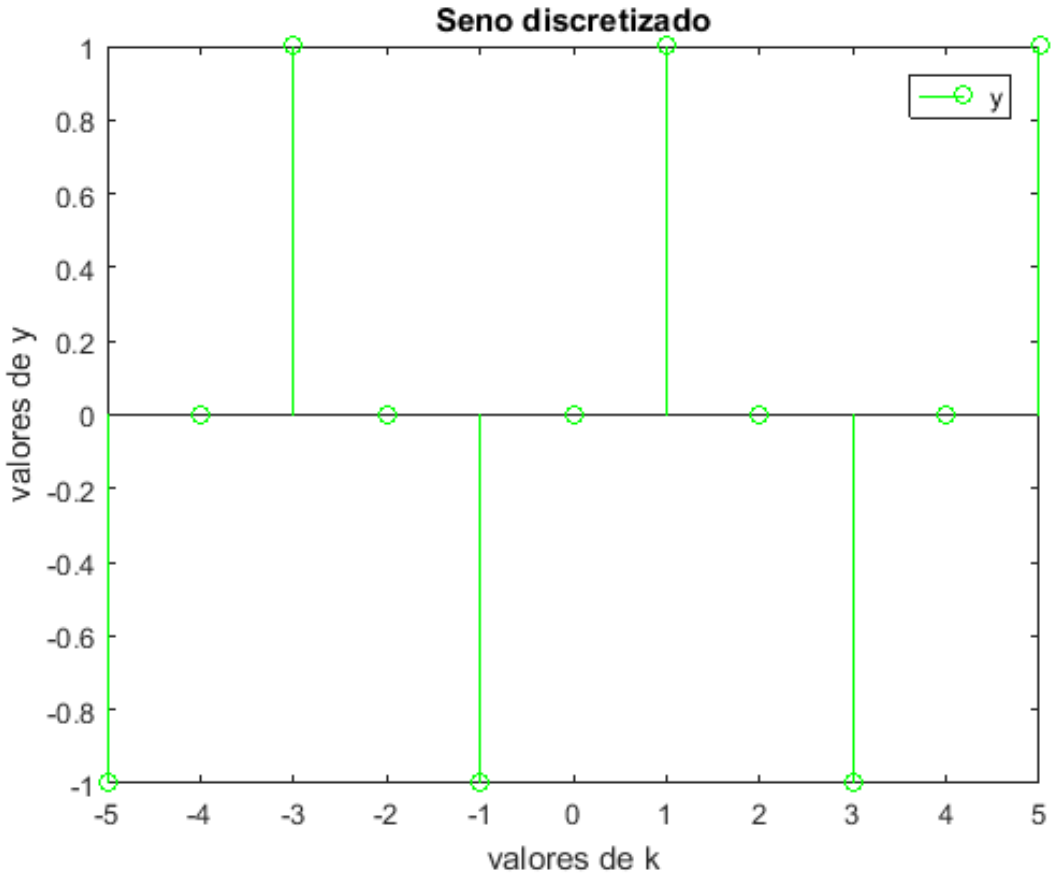


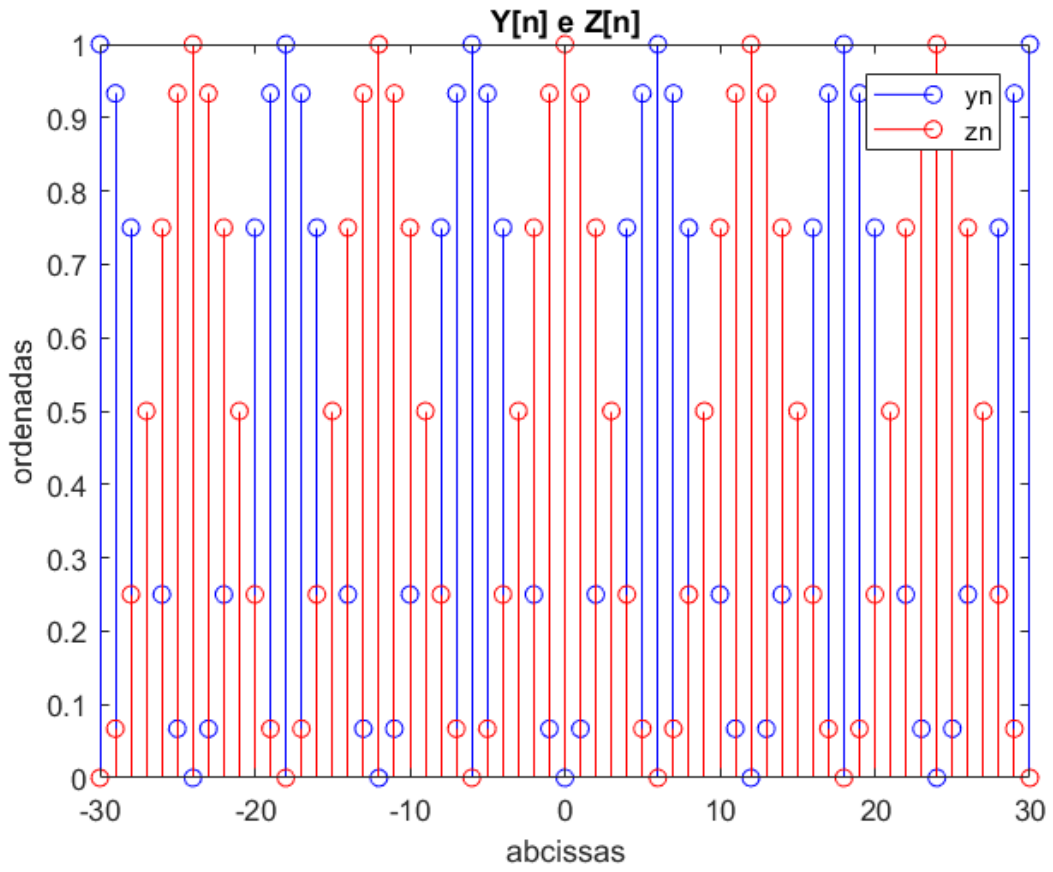
Gráfico 2D

1. Faça um gráfico de $y[n] = \sin^2\left(\frac{\pi}{12}n\right)$ e $z[n] = \cos^2\left(\frac{\pi}{12}n\right)$ para $-30 \leq n \leq 30$ na mesma figura. O gráfico de y[n] deverá ser azul e o de z[n] em vermelho.

Obs: Uma função (f) discreta é representada por f[x], tente publicar o arquivo m-file usando o comando publish.

Resposta

```
%exercício de matlab
clear variables  clc;
close all;
n = -30:30;
yn = sin(pi/12 .* n).^2;
zn = cos(pi/12 .* n).^2;
stem(n, yn, 'blue');
hold on
stem(n, zn, 'red');
xlabel('abcissas');
ylabel('ordenadas');
title('Y[n] e Z[n]');
legend('yn', 'zn');
set(gcf, 'color', 'white');
```



Gráficos 2D

- Pizza

Propriedades

1ª - Substituir os espaços (vazios) pelos nomes (Label's)

2º - Criar um vetor de strings, ou seja, um array de STRING'S.

pie(conjunto , label)

pie(conjunto , propriedades)

pie(conjunto , explode)

Observamos que label, neste caso, é um vetor de STRING'S

Ex: `explode = [0, 0, 1, 1];`

Ex:
```
labels = {'A', 'B', 'C', 'D'};
pie(A, labels)
```

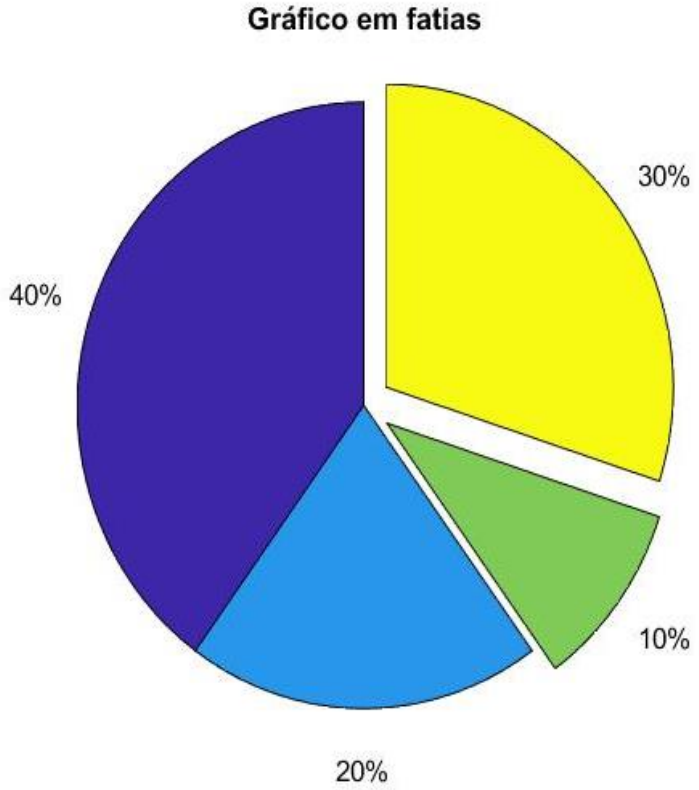


Gráfico 2D – outros tipos

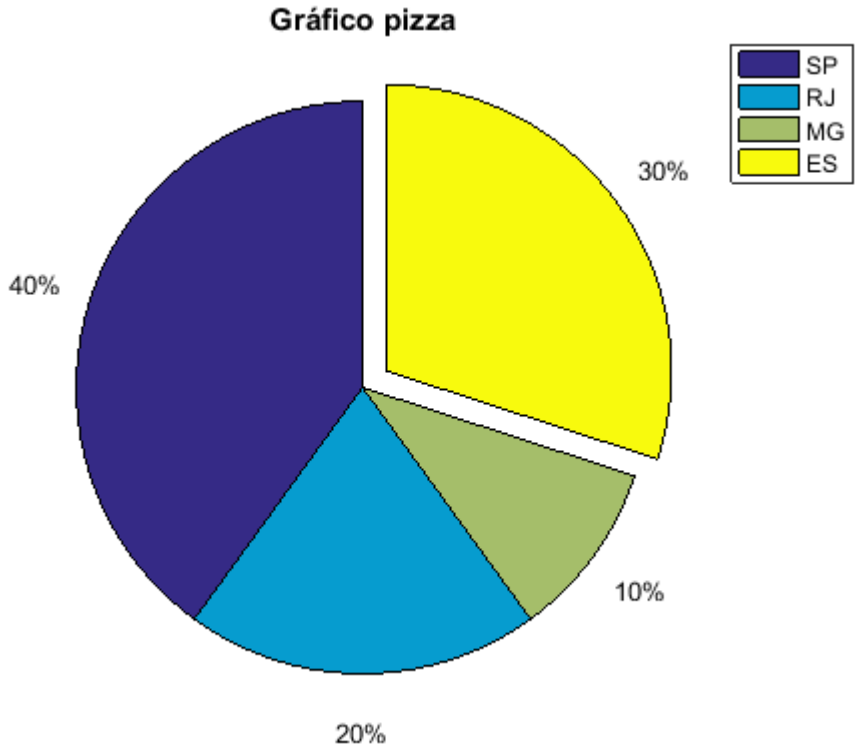
1. Faça um gráfico pizza com os dados dos seguintes estados:

SP : 4	RJ : 2
ES : 3	MG : 1

Destaque no gráfico o estado do Espírito Santo.

Resposta

```
%exercício de matlab
clear variables;
percentuais = [.4 .2 .1 .3];
estados = {'SP', 'RJ', 'MG', 'ES'};
pizza = subplot(1, 1, 1);
destacar = [0 0 0 1];
pie(percentuais, destacar);
legend(estados, 'Location', 'northeast');
title(pizza, 'Gráfico pizza');
set(gcf, 'Color', 'white');
```



Gráficos 2D

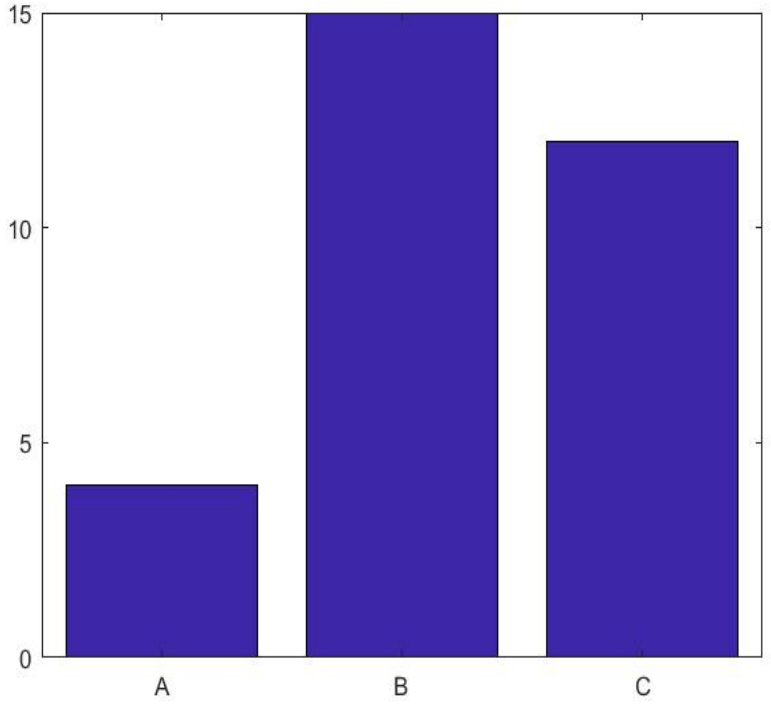
- Barras
  - Propriedades

1° - Definir os intervalos nos eixos X e Y

2º - Chamar o comando bar(X, Y)

Ex:

```
% --- Gráfico em Barras
clc, clear all, close all;
y = [4 15 12];
legenda = categorical({'A', 'B', 'C'});
%Obs: O comando categorical converte
%um tipo de variável qualquer em
%vetores de dados
bar(legenda, y);
```



Gráficos 3D - Curvas no Espaço

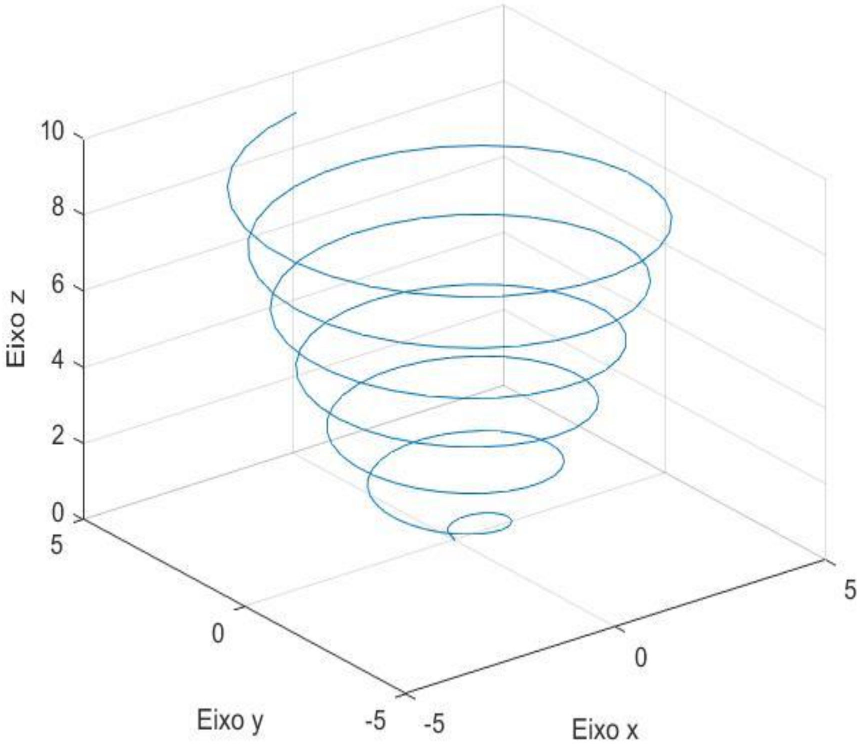
- Plot3

Ex:

1° - ... as
entr... as
pod... s.

2° - ...
plot...

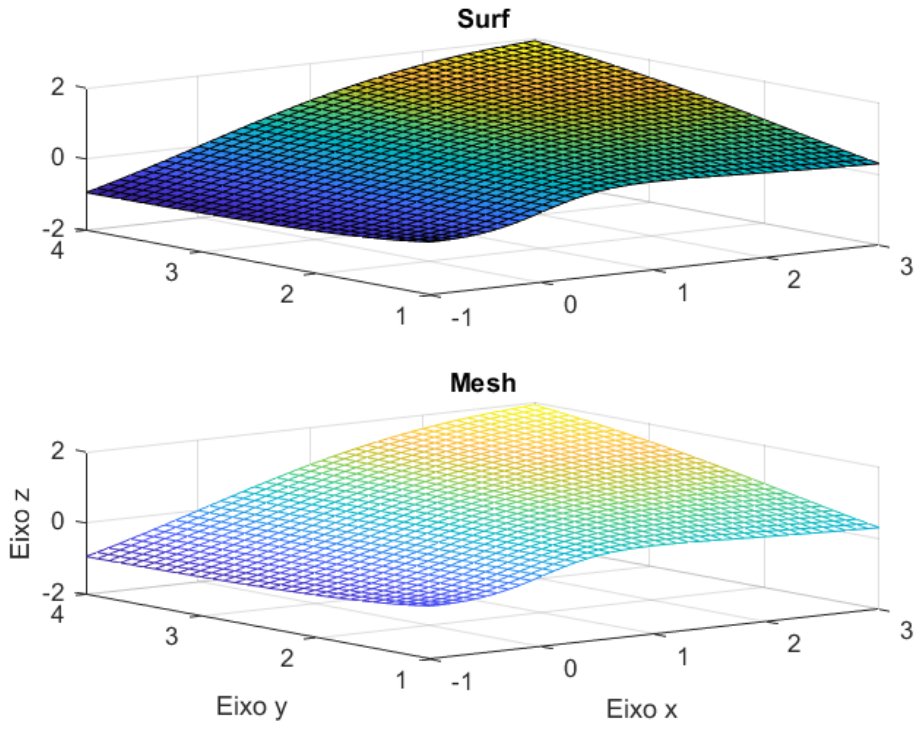
```
%Exemplo plot3
clear variables;
t = 0: .1 : 6*pi;
x = sqrt(t) .* sin(2*t);
y = sqrt(t) .* cos(2*t);
z = 0.5*t;
plot3(x, y, z);
```



Gráficos 3D - Malhas e Superfícies

São duas formas de representar graficamente funções de variável dependente no espaço. A utilização desses comandos é realizada por etapas.

1. Cria-se um grid no plano cartesiano (meshgrid).
2. Calcula-se o valor da função de variável dependente para todos os valores desse grid.
3. Plota-se a função usando mesh ou surf.

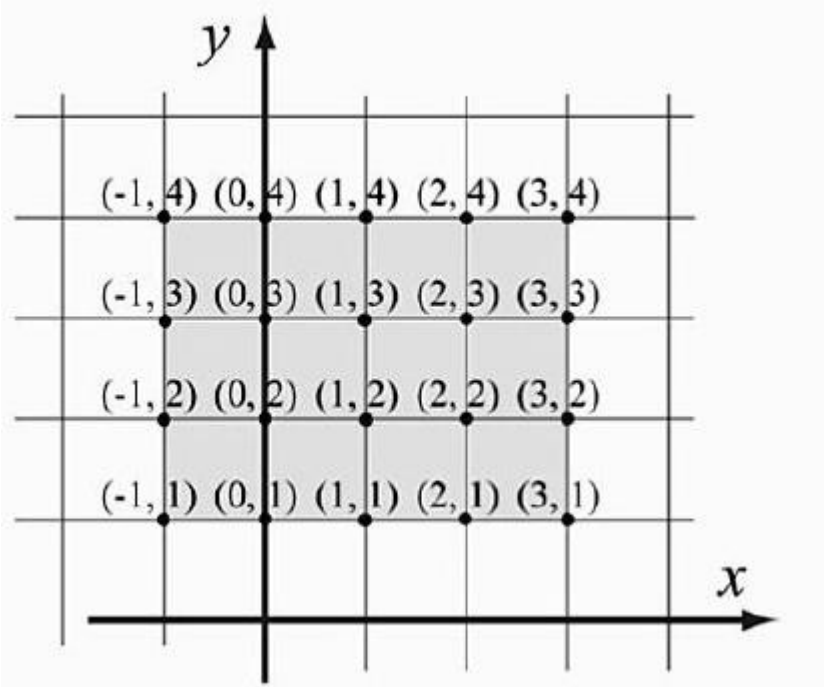


Gráficos 3D - Criando uma Grid

- Meshgrid

Retorna as matrizes de coordenadas em X e Y relativa aos vetores aplicados.

```
%Exemplo do uso de um meshgrid
x = -1:3;
y =  1:4;
[X,Y] = meshgrid(x,y);
```



Gráficos 3D - Calculando f(x,y)

Para determinar o valor da função em cada ponto das matrizes de coordenadas X e Y, basta utilizar-se da lei de formação da função dependente em sua representação matricial, por exemplo :

```
%Exemplo grids e surf
clear variables;
x = -1:3;
y = 1:4;
[X, Y] = meshgrid(x, y);
Z = X.*Y.^2./(X.^2 + Y.^2);
```

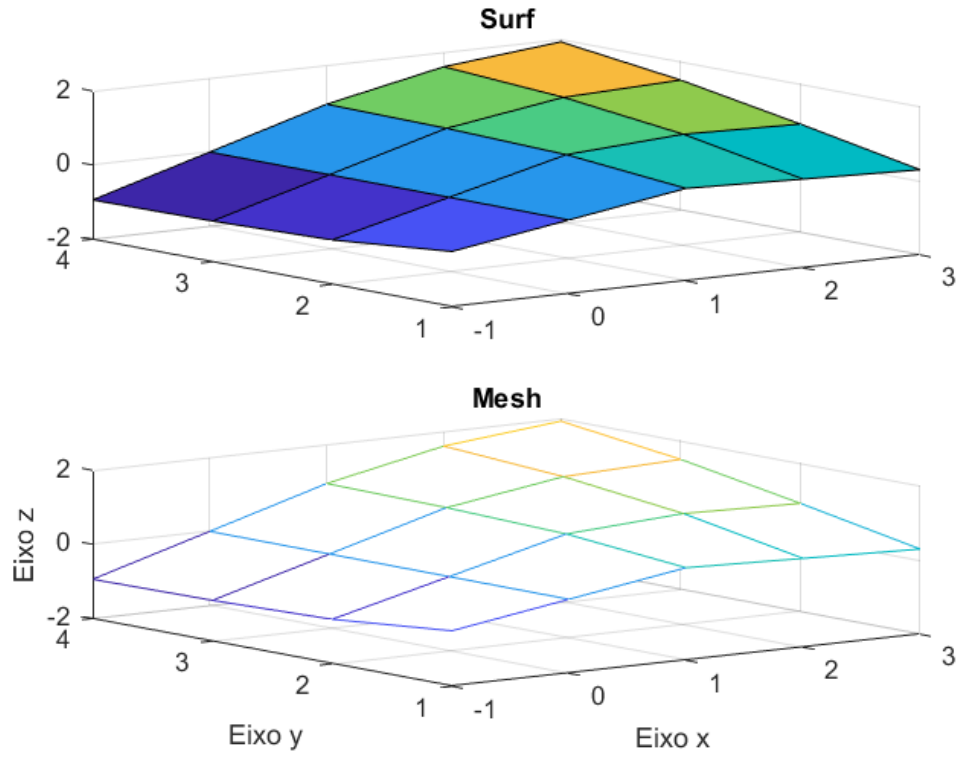
```
Z =

   -0.5000         0    0.5000    0.4000    0.3000
   -0.8000         0    0.8000    1.0000    0.9231
   -0.9000         0    0.9000    1.3846    1.5000
   -0.9412         0    0.9412    1.6000    1.9200
```

Gráficos 3D - Usando surf ou mesh

- Usando as funções surf ou mesh

São funções parecidas, de modo que a única diferença entre elas é o fato de uma ser preenchida e a outra não

```
%Exemplo grids e surf
clear variables;
x = -1:.1:3;
y = 1:.1:4;
[X, Y] = meshgrid(x, y);
Z = X.*Y.^2./(X.^2 + Y.^2);
subplot(2, 1, 1);
surf(X, Y, Z);
title('Surf');
subplot(2, 1, 2);
mesh(X, Y, Z);
title('Mesh');
```

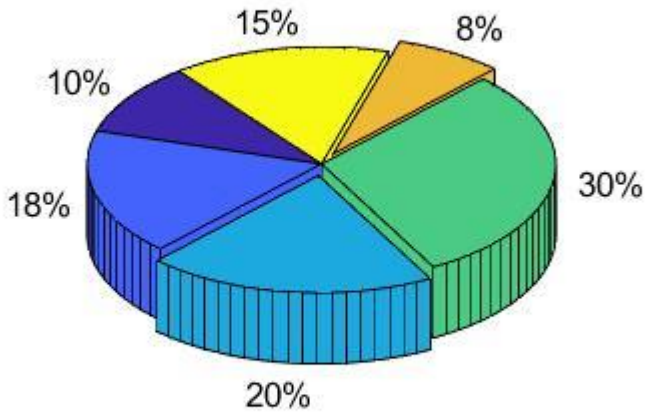


Gráficos 3D - Outros Gráficos 3D

- Código

```
%Plot de Pizza 3-D (pie3)
%% Configurações do Plot
img = figure('Name', 'Resposta Total', 'Color', 'white');
%pie3(V) em que V é o vetor contendo as quantidades que
%serão mostradas no gráfico
V = [4 7 8 12 3 6];
Exploded = [0 0 1 0 1 0];
axis equal
pie3(V, Exploded)
%Obs: Usamos o vetor Exploded para destacar os pedaços
% correspondentes a 8 e 3.
%Obs: Como o Gráico em pizza é uma distribuição, ele não
%possui eixos
```

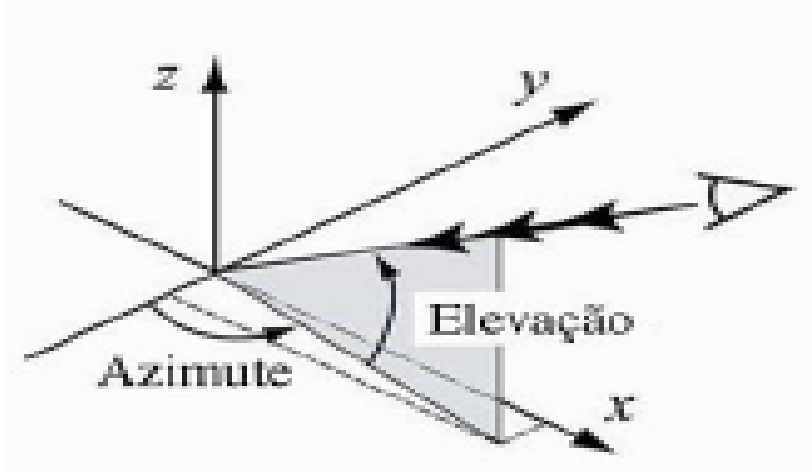
- Saída

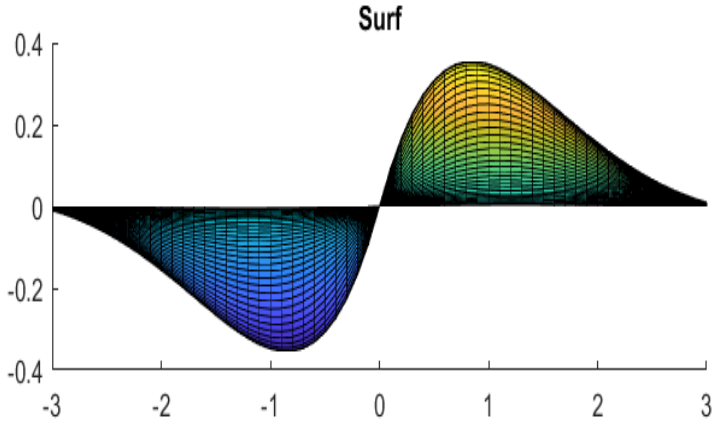


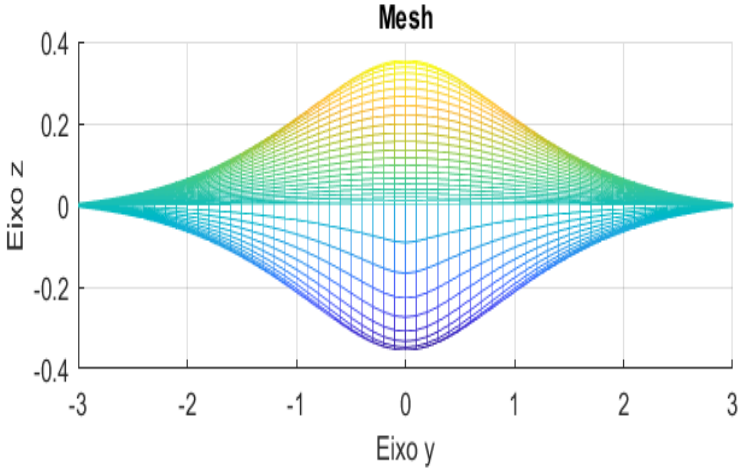
Gráficos 3D - O Comando View

- view(azimutal, elevação)

O comando view: Controla a direção da observação do gráfico. O controle é feito através da especificação em termos dos ângulos azimutal e o ângulo de elevação.







Comandos Importantes

- figure('propriedade', 'valor');

Comandos	Descrição
set(obj, prop, esp1,...);	Altera alguma propriedade de um dado objeto
gcf	Retorna a figura atual
'Color', 'Cor'	Altera a cor da figure
'NumberTitle'	Remove a numeração da figure
Colormap	Altera o padrão de cores do surf3