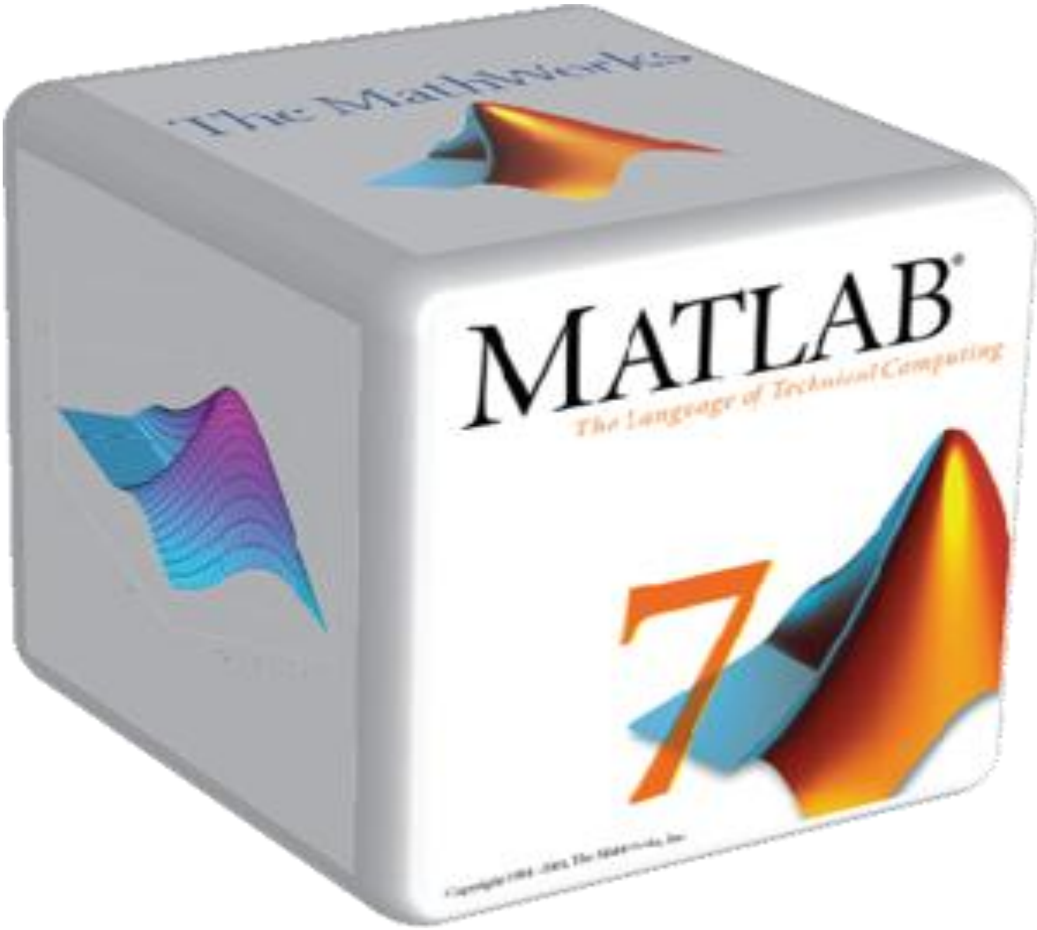
Noções de Matlab



Índice

Capitulo – 07

Capitulo – 08

Capitulo – 09

Capitulo – 10

Capitulo – 11

Capitulo – 12

Capítulo – 01

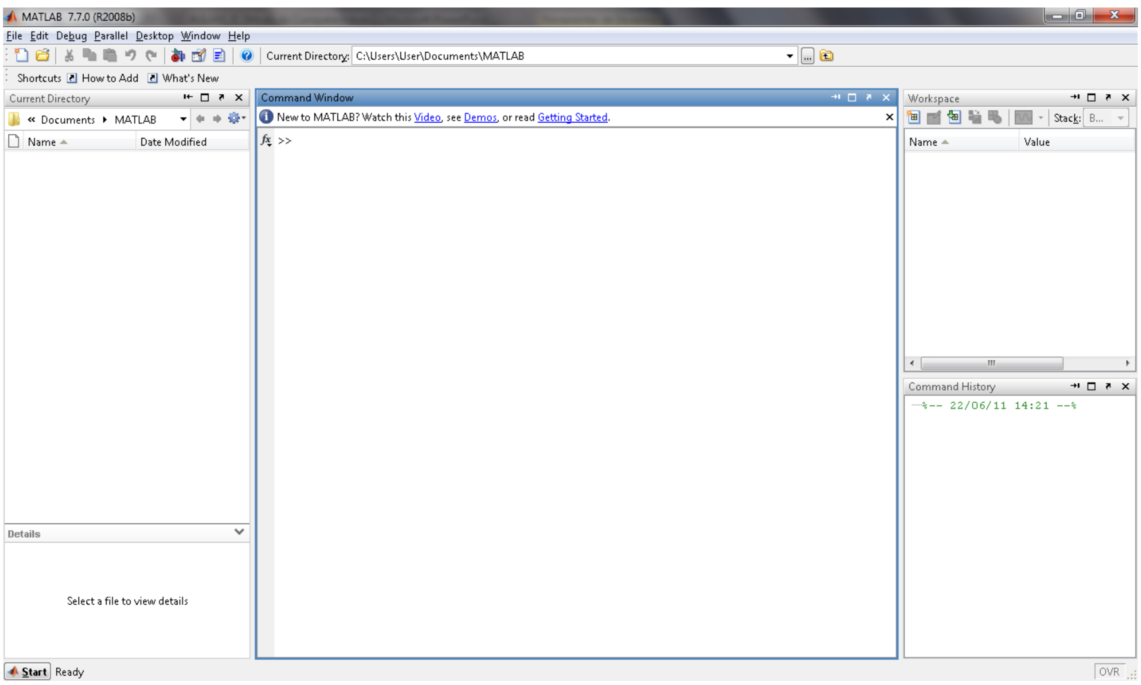
Introdução

O Matlab (abreviatura de MATrix LABoratory - Laboratório de Matrizes) é um software de simulação matemática que realiza operações matriciais, constrói gráficos em duas ou três dimensões, auxilia no processamento de sinais, além de manipular outras funções especializadas.

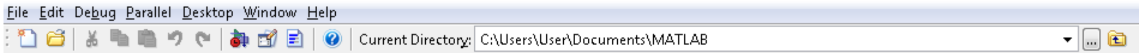
Ele trabalha com uma linguagem de programação de alto-nível, em um ambiente interativo, para o desenvolvimento de algoritmos, análise e visualização de dados e computação numérica. Próprio para as áreas técnicas e científicas, o software tem funções de tratamento numérico de alto desempenho, capazes de resolver problemas computacionais técnicos de forma mais eficiente do que as tradicionais linguagens de programação.
Além do ambiente interativo, outra facilidade do Matlab é a possibilidade de execução de arquivos texto contendo uma seqüência de instruções definidas pelo usuário. Esses arquivos texto, que têm extensão '.m', podem ser criados e editados dentro ou fora do seu ambiente.

Ambiente

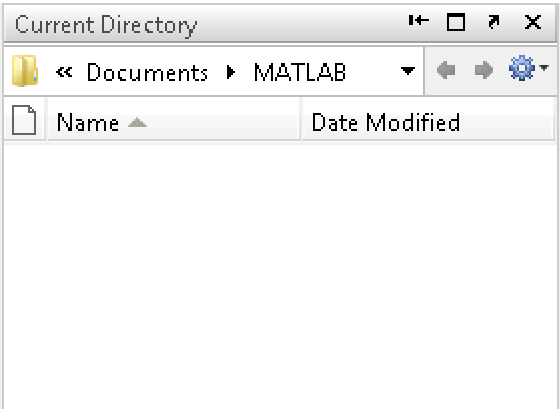
Inicializando o software, tem-se acesso o seguinte ambiente, como mostra a figura a seguir:



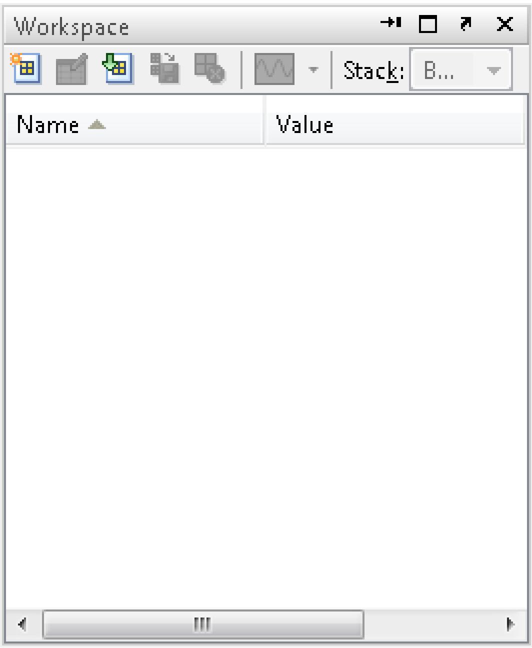
Compreendendo as divisões do ambiente numa fase inicial tem-se:



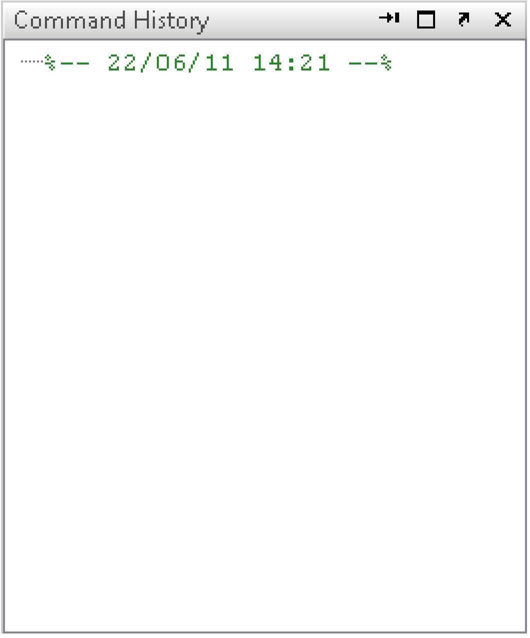
Na parte superior da tela encontra-se o menu suspenso, bem como o chamado SpeedBar, isto é, os ícones de acesso rápido, além do Current Directory que é o caminho onde o desenvolvimento feito em Matlab será gravado ou até mesmo lido algo que já se encontre disponível.



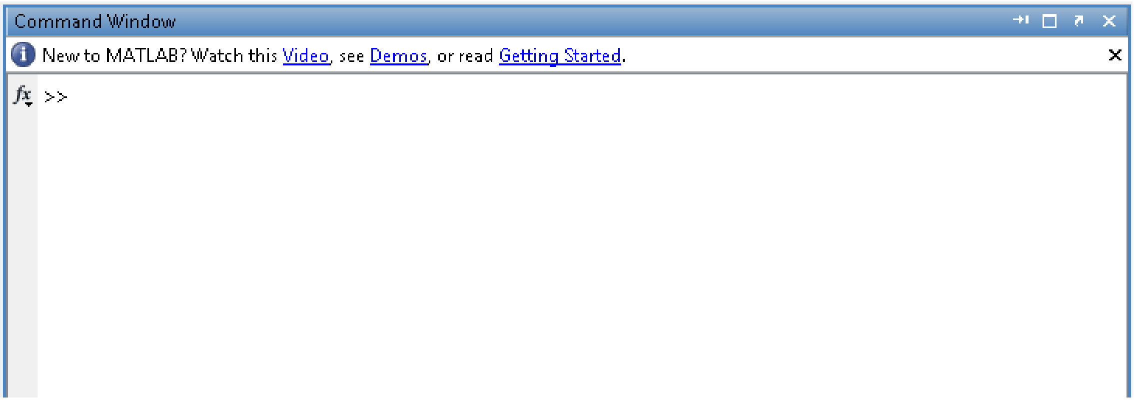
A coluna que se apresenta do lado esquerdo exibe os eventuais arquivos que estariam disponíveis para futuros acessos, especificando o nome do arquivo e sua data de gravação.



Do lado direito existe uma área denominada Workspace que caracteriza a memória de armazenamento, isto é, as variáveis (name) e seus respectivos valores(value).



Embaixo tem-se o Command History que registrará o histórico dos comandos executados.



A janela Command Window é a janela de interação, assim sendo, é a mais importante nesse instante. Cabe agora a seguinte pergunta:

“Quando sei que está pronto para uso?”

Assim que o **prompt** fique disponível, isto é, fx >>

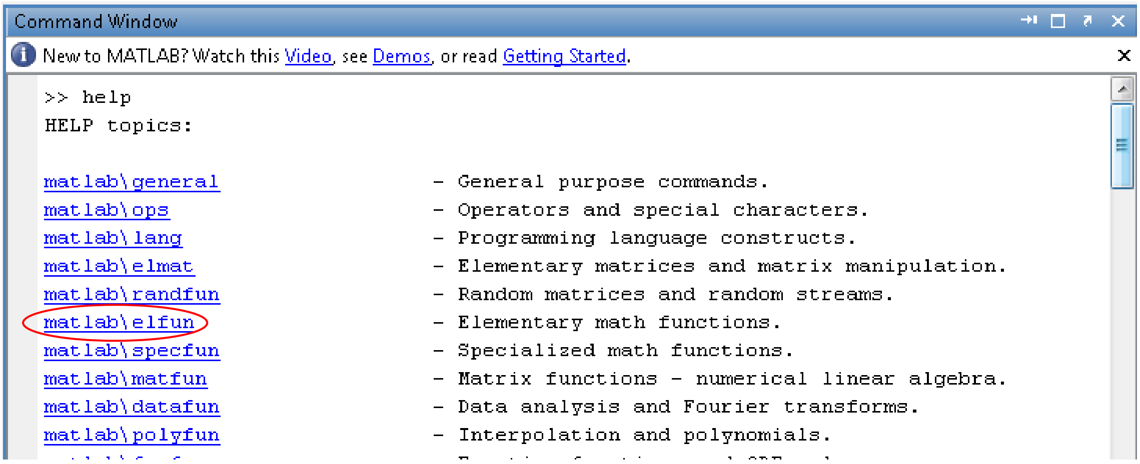
Todo comando deve ser finalizado teclando-se ***Enter***.

Para encerrar uma sessão de trabalho do **MATLAB** digita-se o comando ***exit*** (ou ***quit***) no *prompt* do programa.

IMPORTANTE: O Matlab é um software sensitive case, isto é, faz diferença entre letras MAIUSCULAS e MINUSCULAS. Assim sendo, seus comandos devem ser digitados sempre em letras MINÚSCULAS.

Ambiente de Ajuda

Comando **help**

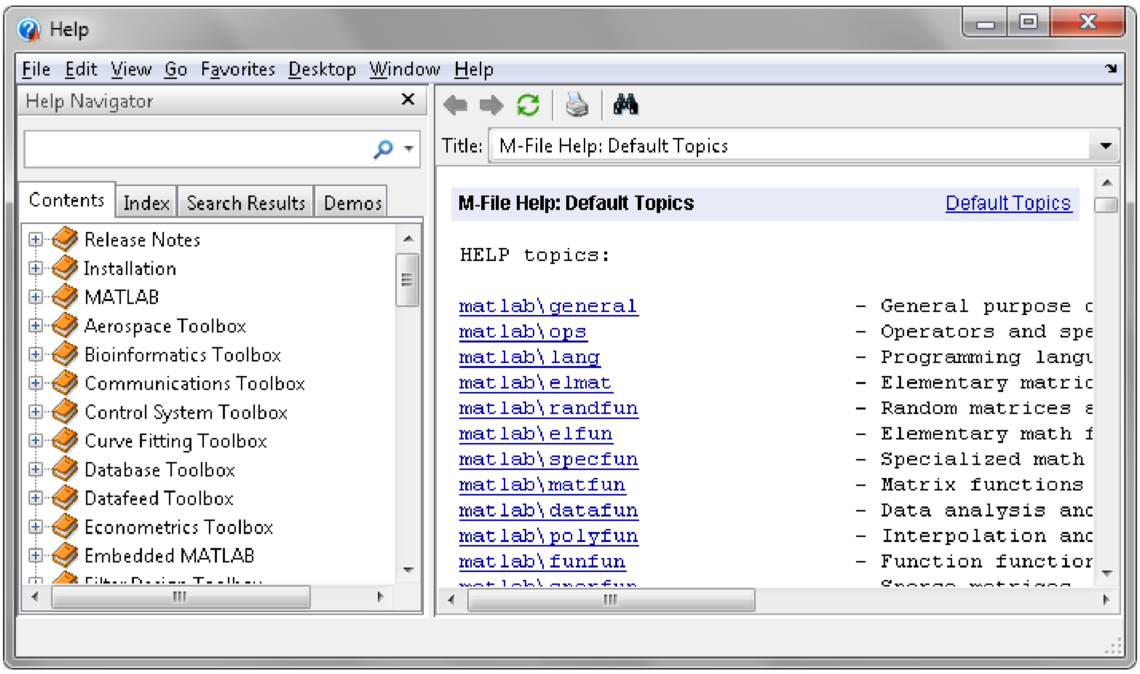


Apresenta uma listagem de todos os pacotes disponíveis.

Para se ter ajuda sobre um pacote específico ou sobre um comando ou função específica, deve-se combinar o comando *help* e o nome do pacote, comando ou função de interesse.

Por exemplo: >> help elfun

Comando **helpwin**



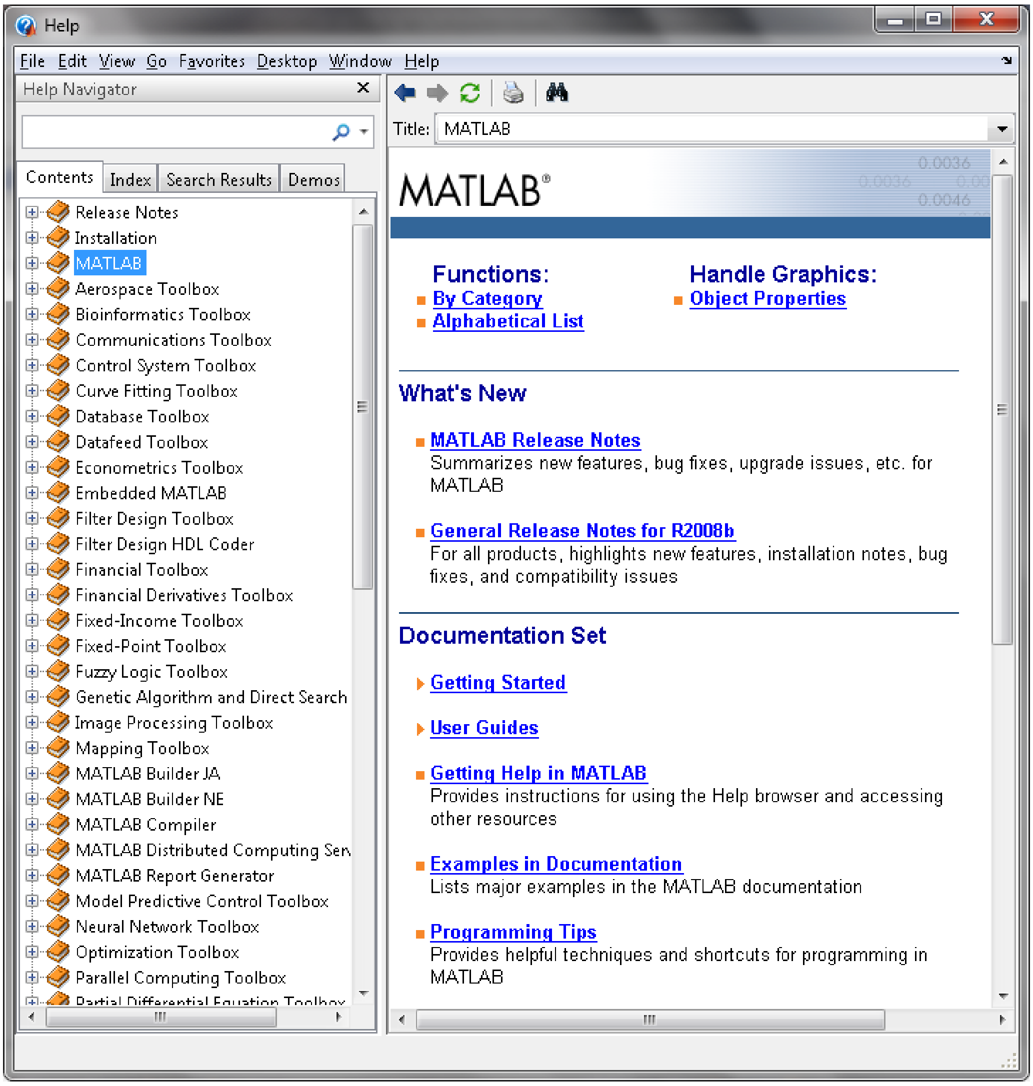
Abre-se uma nova janela textual de ajuda.

Combinando-se o ***double-click*** sobre um dos itens obtêm-se os vários níveis de ajuda de um pacote específico e de um comando ou função de interesse.

O nome de uma função pode ser introduzido no campo superior esquerdo para se ter diretamente uma ajuda deste comando.

No ***pull-down*** *menu* à direita deste campo encontram-se os tópicos relacionados com a função que está sendo explicada no momento.

Comando **helpdesk**



Dispara-se um programa de navegação instalado no seu computador (*Netscape*, *Internet Explorer*, etc) com um ambiente de ajuda mais completo que utiliza a linguagem de hipertexto.

Neste ambiente de ajuda é possível obter uma listagem das funções:

Por bloco de especificidade (MATLAB/Functions/by Subject) ou

Por ordem alfabética (MATLAB/Functions/by Index).

A ajuda específica de um determinado comando pode ser feito diretamente no campo abaixo de *Go to MATLAB function*. Além disto, estão disponíveis vários documentos explicativos (*Online Manuals*).

Operações Básicas

A primeira forma de se utilizar o **MATLAB** é como uma calculadora científica através da digitação de comandos diretamente no seu ***prompt.*** Assim sendo, vamos discutir inicialmente a notação das operações básicas, a saber:

Notação Matemática	Notação MATLAB	Prioridade Natural
+	**+**	**3º**
-	**-**	**3º**
. ou x	*****	**2º**
÷ ou /	**/**	**2º**
x^y	**x ^ y**	**1º**

Vamos propor inicialmente a seguinte cálculo: **-1+5**, executando no MatLab, temos:

Nota-se que o resultado da operação foi atribuído à variável *ans*

```
>> -1+5
ans =
     4
>> |
```

***ans*wer = resposta**

Para melhorar esse tipo de notação poderíamos executar o mesmo cálculo da seguinte forma:

```
>> resultado = -1+5

resultado =

     4
```

Números decimais são representados **por ponto**, isto é, ½ =**0.5**, então efetue:

0.5 + 3.2 – 1.3

```
>> resultado = 0.5+3.2-1.3

resultado =

    2.4000
```

Analise as expressões e veja como foram feitas as operações:

```
>> resultado = 32/2^4/2

resultado =

     1
```

32/2^4/2
32/16/2
2/2
1

```
>> resultado = 32/2^(4/2)

resultado =

     8
```

32/2^(4/2)
32/2^2
32/4
8

Uma outra forma de se fazer estes cálculos seria trabalhar com variáveis auxiliares, como mostrado a seguir:

Consideremos:

v1=3,

v2=5,

v3=12,

v4=4.

Calcule:

v1*v2+v3/v4

No MATLAB

```
» v1=3,v2=5,v3=12,v4=4

v1 =

     3

v2 =

     5

v3 =

    12

v4 =

     4

» resultado = v1*v2+v3/v4

resultado =

    18
```

v1*v2+v3/v4

3*5+12/4

15+12/4

15+3

18

Com base nesses conceitos execute no Matlab as seguintes expressões:

1. $-3-5+2$
2. $(2-5+1-8+10-7)+(3-9-1)-(3-7+8)$
3. $1-6-[4-6+2-9+3-(2+10-7-5)+2-6]$
4. $(-8-9-5+22-3)\div(-1-1-1)$
5. $(-2+5-6)^2$
6. $(1-37-8)^2-(-1-2-3)^2$
7. $\left(\frac{1}{3}+\frac{2}{5}-\frac{3}{2}+1\right)^2$
8. $(0{,}2-0{,}5+2{,}5)(0{,}5-3-2{,}5)+(5-7-10+2)-(-3)^2$
9. $\dfrac{2-3-11+4-1+6-(7-6+3)+14}{(-3)(-5)}$
10. $\frac{3}{4}+\frac{2}{3}-\left(\frac{1}{2}\right)^2-\frac{11}{5}+3{,}5$
11. $\left(2^{-\frac{3}{2}}+1\right)^{-1}$
12. Fazendo x=2,1 calcule:

$$\frac{(2x^2-2)(x^3+3x)}{(x+1)^2-2}+x$$

Bom trabalho...

Capítulo – 02

Exponenciais

Notação Matemática	Notação no MATLAB
$\sqrt{X}$	sqrt(x)
$\sqrt[n]{x^m} = x^{\frac{m}{n}}$	x ^ (m/n)
e^x	exp(x)

Exemplos – Efetuem os cálculos

$$\left(\sqrt[3]{2^2}\right)^2 = (2^{\frac{2}{3}})^2 \longrightarrow$$

```
» resultado = (2^(2/3))^2
resultado =
    2.5198
```

$$3 \cdot 2^{\sqrt{3}} \cdot 2^{-\sqrt{3}} \longrightarrow$$

```
» x=sqrt(3)
x =
    1.7321
» resultado = 3*2^x*2^(-x)
resultado =
    3.0000
```

$$\left(\frac{5^{\sqrt{2}+\sqrt{3}}}{25^{\sqrt{2}-\sqrt{3}}}\right)^{\sqrt{3}} \longrightarrow$$

```
» x=sqrt(2)
x =
    1.4142
» y=sqrt(3)
y =
    1.7321
» resultado = (5^(x+y)/25^(x-y))^y
resultado =
    3.7897e+004
```

$$\frac{\sqrt[3]{e^2}+1}{1-e^2} \longrightarrow$$

```
» x=exp(2)

x =

    7.3891

» resultado = (x^(1/3) + 1)/(1-x)

resultado =

   -0.4614
```

Logaritmos

Notação Matemática	Notação no MATLAB
$\log_2^{(x)}$	log2(x)
$\log(x) = \log_{10}^{(x)}$	log10(x)
$\ln(x) = \log_e^{(x)}$	log(x)

Exemplos:

$$3^{1+\log 4} \longrightarrow$$

```
» resultado = 3^(1+log10(4))

resultado =

    5.8127
```

$$5^{2-\ln 6} \longrightarrow$$

```
» resultado = 5^(2-log(6))

resultado =

    1.3982
```

Considere x=2, y=3 e z =5, com essas informações calculem os próximos exercícios:

$$\log_2\left(\frac{2x}{x^2+y^2}\right)$$

```
» x=2,y=3,z=5

x =

     2


y =

     3


z =

     5

» resultado = log2(2*x/(x^2+y^2))

resultado =

   -1.7004
```

$$\frac{3.\log_2^{(x+y)} - 2.\log(x+z) + 4\ln z^2}{5}$$

```
>> x=2;y=3;z=5;

>> a=log2(x+y);

>> b=log10(x+z);

>> c=log(z^2);

>> resultado=(3*a-2*b+4*c)/5

resultado =

    3.6302
```

Com base nesses conceitos execute no Matlab as seguintes expressões:

1. $\sqrt{2} \cdot \sqrt{15} \cdot \sqrt{30}$

2. $\dfrac{\sqrt{2} \cdot \sqrt[3]{2}}{\sqrt[4]{2}}$

3. $\left(\sqrt{12} - 2\sqrt{27} + 3\sqrt{75}\right)\sqrt{3}$

4. $\sqrt{2} \cdot \sqrt{2+\sqrt{2}} \cdot \sqrt{2+\sqrt{2+\sqrt{2}}} \cdot \sqrt{2-\sqrt{2+\sqrt{2}}}$

5. $\dfrac{e+\sqrt{e^2-1}}{e-\sqrt{e^2-1}} - \dfrac{e-\sqrt{e^2-1}}{e+\sqrt{e^2-1}}$

6. $\left(\dfrac{2^{\sqrt{27}} \cdot 8^{\sqrt{e}}}{4^{-\sqrt{3}}}\right)^{\frac{\sqrt{3}}{2}}$

Fazendo a=5 e b=2, então:

7. $\log_2^{\left(\frac{2e}{a^2-b^2}\right)}$

8. $\log\left(\sqrt[3]{\dfrac{a}{b^2\sqrt{e}}}\right)$

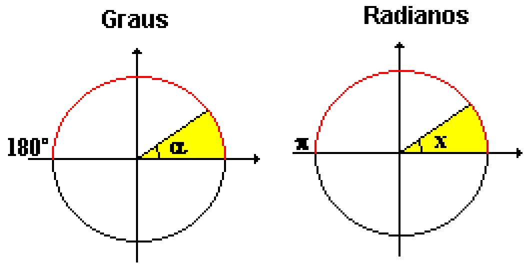
9. $\ln\left(\dfrac{a^2\sqrt{b}}{a(a+b)^2}\right)$

10. $\dfrac{1}{2}\log\left(a^2+b^2\right) - \left[\dfrac{1}{3}\log_2^{(a+b)} - \ln(a-b)\right]$

Capítulo – 03

Funções Trigonométricas

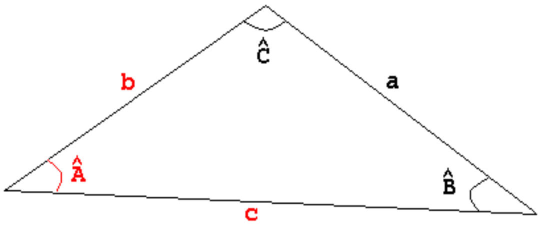
A princípio um conceito é fundamental, quando se trata valor trigonométrico no computador, só é admitido valor em radianos, assim sendo, se deve recordar a mudança de unidades de graus para radianos, isto é:



Então $\left.\begin{array}{l} 180^\circ \longrightarrow \pi \\ \alpha \longrightarrow x \end{array}\right\}$ $x = \dfrac{\alpha \cdot \pi}{180}$

Notação Matemática	Notação MATLAB
sen(x)	sin(x)
cos(x)	cos(x)
$tg(x) = \dfrac{sen(x)}{\cos(x)}$	tan(x)
$\cos\sec(x) = \dfrac{1}{sen(x)}$	1/sin(x)
$\sec(x) = \dfrac{1}{\cos(x)}$	1/cos(x)
$\cot g(x) = \dfrac{1}{tg(x)} = \dfrac{\cos(x)}{sen(x)}$	1/tan(x)
arccos(x)	acos(x)
arcsen(x)	asin(x)
arctg(x)	atan(x)

Resolver um triângulo conhecendo um lado ***a*** e os dois ângulos adjacentes a ele (***B e C***), então:

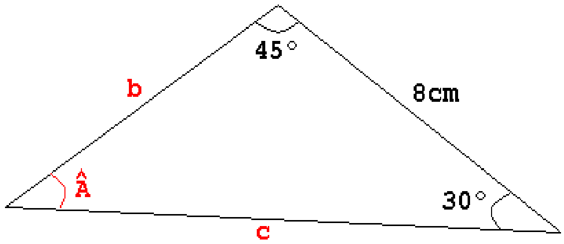


$$\hat{A} = 180° - \left(\hat{B} + \hat{C}\right)$$

$$b = \frac{a.sen(\hat{B})}{sen(\hat{A})}$$

$$c = \frac{a \cdot sen(\hat{C})}{sen(\hat{A})}$$

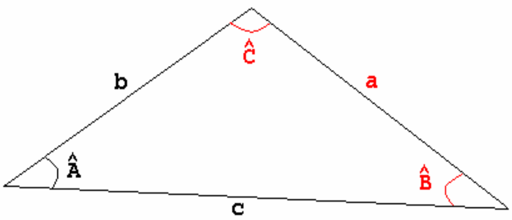
Aplicando numericamente tem-se:




```
>> a=8;cgr=45;bgr=30;
>> agr=180-(bgr+cgr);
>> ard=agr*pi/180;brd=bgr*pi/180;crd=cgr*pi/180;
>> b=a*sin(brd)/sin(ard); c=a*sin(crd)/sin(ard);
```

>> agr	>> b	>> c
agr =	b =	c =
105	4.1411	5.8564

Resolver um triângulo conhecendo dois lados (***b*** e ***c***) e o ângulo que eles formam (***A***).

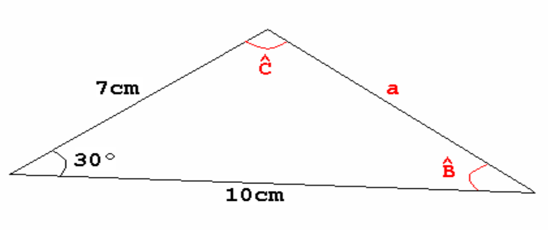


$$a = \sqrt{b^2 + c^2 - 2bc \cdot \cos(A)}$$

$$b^2 = a^2 + c^2 - 2ac \cdot \cos(B) \Rightarrow \cos(B) = \frac{a^2 + c^2 - b^2}{2ac} \Rightarrow B = \arccos\left(\frac{a^2 + c^2 - b^2}{2ac}\right)$$

$$c^2 = a^2 + b^2 = 2ab \cdot \cos(C) \Rightarrow \cos(C) = \frac{a^2 + b^2 - c^2}{2ab} \Rightarrow C = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Aplicando numericamente tem-se:



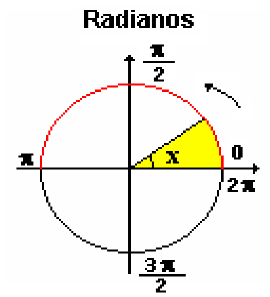

```
>> b=7;c=10;agr=30;
>> ard=agr*pi/180;
>> a=sqrt(b^2+c^2-2*b*c*cos(ard));
>> brd=acos((a^2+c^2-b^2)/(2*a*c));
>> crd=acos((a^2+b^2-c^2)/(2*a*b));
>> bgr=brd*180/pi; cgr=crd*180/pi;
```

```
>> a

a =

    5.2684
```

```
>> bgr

bgr =

   41.6312
```

```
>> cgr

cgr =

  108.3688
```

Vejamos agora como traçar os primeiros gráficos no MatLab, utilizando as funções trigonométricas em uma volta completa na circunferência trigonométrica, no sentido anti-horário, isto é, [0,2∏].



Para fazermos um gráfico no computador precisamos definir um passo bem pequeno de tal sorte que o computador seja capaz de traçar o gráfico com precisão.

Nesse caso temos a variável x variando de 0 até 6.2838, então para atender as necessidades computacionais podemos entender que se o passo fosse 1, teríamos apenas os valores: 0 – 1 – 2 – 3 – 4 – 5 – 6.

Então fixaremos o passo como sendo 0.1, então teremos:

0 – 0.1 – 0.2 – 0.3 -.... – 6.0 – 6.1 – 6.2

Assim teremos uma gama muito maior de pontos para obter a precisão do desenho da função.

Como definir o passo?

>> x=0:0.1:6.28;

Cabe destacar que:

0 (zero) define o valor inicial de x

0.1 define o passo de variação do x

6.28 define o valor final de x

Qual o comando capaz de traçar gráfico no Matlab

plot(x,y)

Com base nesses conceitos tracemos o gráfico y=sen(x)

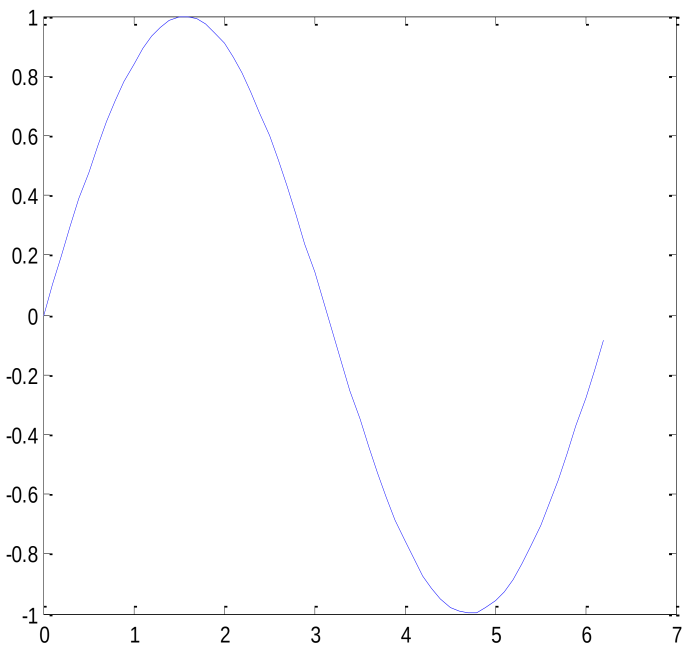
Então no command window digita-se:

>> x=0:0.1:6.28;

>> y=sin(x);

>> plot(x,y)

Tem-se então o seguinte resultado:



Com base nesses conceitos execute no MatLab os seguintes exercícios:

1. Calcule os valores das funções trigonométricas para os ângulos notáveis, isto é, 30º, 45º e 60º

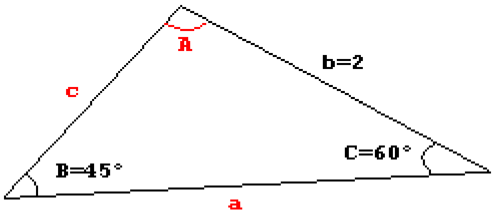
2. Sabendo que $a = 2\sqrt{3}, \quad b = 2, \quad c = 4$ calcule o valor de S, sabendo que

   S = arccos(a/c) – arcsen(b/c)

3. Nos conceitos trigonométricos uma relação é tida como fundamental, isto é, $sen^2(x)+cos^2(x) = 1$, então calcule: cos(x), tg(x), cotg(x), sec(x) e cossec(x), sabendo que sen(x)=4/5 e que x pertence ao II quadrante.

4. Dado o triangulo com suas medidas,

   calcule os valores sinalizados em vermelho.



5. Trace o gráfico da função:

   y=cos(x), com 0≤x≤2π

Capítulo – 04

Matrizes

Para criar uma variável onde é armazenada uma matriz, basta escrever os elementos da matriz entre colchetes [...], sendo os elementos de uma mesma linha da matriz separados por vírgula (ou espaço em branco) e as linhas separadas por ponto e vírgula. Por exemplo, para armazenar a matriz

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Longrightarrow$$

```
>> A=[1,2,3;4,5,6]

A =

     1     2     3
     4     5     6
```

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

A =

     1     2     3
     4     5     6
```

Acessar os elementos de uma matriz usando os comandos

- O elemento que está na 2ª linha e 3ªcoluna
  >> A(2,3)
  ans =
  6

- A 2ª linha da matriz
  >> A(2,:)
  ans =
  4 5 6

- A 1ª coluna da matriz
  >> A(:,1)
  ans =
  1
  4

- Exibir a sub-matriz formada pela 1ª e 2ª colunas da matriz A.

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

- Utilizando a matriz A, gere uma matriz B colocando mais uma coluna com os números 7 e 8 respectivamente:

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

- Utilizando a matriz A, gere uma matriz C colocando mais uma linha com os números 7, 8 e 9 respectivamente:

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

Adição de matrizes

Dadas duas matrizes $A=(a_{ij})_{mxn}$ e $B=(b_{ij})_{mxn}$ chama-se adição A+B a matriz $C=(c_{ij})_{mxn}$ tal que $c_{ij}=a_{ij}+b_{ij}$ para todo i e todo j. Isto significa que a adição de duas matrizes A e B do tipo mxn é uma matriz C do mesmo tipo em que cada elemento é a soma dos elementos correspondentes em A e B.

Exemplo: Dadas as matrizes A e B, calcule a adição de A por B

$$A=\begin{bmatrix}1 & 7 & 2\\ 5 & -1 & -2\end{bmatrix} \quad e \quad B=\begin{bmatrix}3 & 21 & 6\\ 15 & -3 & -6\end{bmatrix}$$

$$A+B=?$$

Então no Command Window, digita-se:

```
>> A=[1 7 2; 5 -1 -2];

>> B=[3 21 6; 15 -3 -6];

>> ADICAO = A+B
```

ADICAO =

4 28 8

20 -4 -8

Subtração de matrizes

Dadas duas matrizes $A=(a_{ij})_{mxn}$ e $B=(b_{ij})_{mxn}$ chama-se subtração A+B a matriz $C=(c_{ij})_{mxn}$ tal que $c_{ij}=a_{ij}-b_{ij}$ para todo i e todo j. Isto significa que a adição de duas matrizes A e B do tipo mxn é uma matriz C do mesmo tipo em que cada elemento é a soma dos elementos correspondentes em A e B.

Exemplo: Dadas as matrizes A e B, calcule a adição de A por B

$$A=\begin{bmatrix} 1 & 7 & 2 \\ 5 & -1 & -2 \end{bmatrix} \quad e \quad B=\begin{bmatrix} 3 & 21 & 6 \\ 15 & -3 & -6 \end{bmatrix}$$

$$A-B=?$$

Então no Command Window, digita-se:

```
>> A=[1 7 2; 5 -1 -2];

>> B=[3 21 6; 15 -3 -6];

>> SUBTRACAO = A-B

SUBTRACAO =

  -2  -14   -4

 -10    2    4
```

Multiplicação de um escalar por uma matriz

Dado um número k e uma matriz $A=(a_{ij})_{mxn}$ chama-se produto k.A a matriz $B=(b_{ij})_{mxn}$ tal que $b_{ij}=k.a_{ij}$ para todo i e todo j. Isto significa que multiplicar uma matriz A por um número k é construir uma matriz B formada pelos elementos de A todos multiplicados por k.

Exemplo: Dadas a matriz A e o número k = 3, calcule k.A

$$k = 3 \qquad e \qquad A = \begin{bmatrix} 1 & 7 & 2 \\ 5 & -1 & -2 \end{bmatrix}$$

Então se digita no Command Window:

```
>> A=[1 7 2; 5 -1 -2];

>> k=3;

>> RESULTADO = k*A

RESULTADO =

    3   21    6

   15   -3   -6
```

Produto entre Matrizes

Dadas duas matrizes $A=(a_{ij})_{mxn}$ e $B=(b_{jk})_{nxp}$ chama-se produto A.B a matriz $C=(c_{ik})_{mxp}$

tal que: $c_{ik} = a_{i1} \cdot b_{1k} + a_{i2} \cdot b_{2k} + a_{i3} \cdot b_{3k} + ... + a_{in} \cdot b_{nk} = \sum_{j=1}^{n} a_{ij} \cdot b_{jk}$

para todo i ∈ {1,2,...,m} e todo k ∈ {1,2,..,p}

Exemplo: Dadas as matrizes A e B, calcule A.B

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad e \qquad B = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

Então se digita no Command Window:

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

>> B=[7;8;9];

>> PRODUTO = A*B

PRODUTO =

   50

  122
```

Matriz Transposta

Dada uma matriz $A = (a_{ij})_{mxn}$ chama-se transposta de A, a matriz $A^t = (a'_{ji})_{nxm}$ tal que $a'_{ji} = a_{ij}$ para todo i e todo j. Isto significa que, as colunas de A^t são ordenadamente iguais às linhas de A.

Notação da matriz transposta A^t no MATLAB é: **A.'**

Exemplos: Dadas as matrizes A e B, calcule A^t e B^t.

$$A = \begin{bmatrix} 1 & 4 \\ 7 & 2 \end{bmatrix} \Rightarrow A^t = ?$$

Então se digita no Command Window:

```
>> A=[1 4;7 2];

>> TRANSPOSTA = A.'

TRANSPOSTA =

     1     7

     4     2
```

$$B = \begin{bmatrix} 1 & 2 & 5 \\ -1 & 7 & 2 \end{bmatrix} \Rightarrow B^t = ?$$

Então se digita no Command Window:

```
>> B=[1 2 5; -1 7 2];

>> TRANSPOSTA = B.'

TRANSPOSTA =

     1    -1

     2     7

     5     2
```

Matrizes Especiais

eye(m,n) - gera matriz com m-linhas e n-colunas com valor unitário nos elementos de índices iguais e zero para os demais elementos. Por exemplo:

```
>> A=eye(2,3)

A =

    1    0    0

    0    1    0

>> I=eye(3,3)

I =

    1    0    0

    0    1    0

    0    0    1
```

Matematicamente conhecida como **Matriz Identidade** (I_3) toda matriz quadrada aonde na diagonal principal os elementos são sempre 1 e o resto é igual a zero.

zeros(m,n) - gera matriz nula com m-linhas e n-colunas.

```
>> A=zeros(2,3)

A =

    0    0    0

    0    0    0

>> B=zeros(2,2)

B =

    0    0

    0    0

>> C=zeros(1,4)
C =
    0    0    0    0
```

Matriz Inversa

Seja A uma matriz quadrada de ordem n. Dizemos que A é matriz inversível se existir uma matriz $B=A^{-1}$ tal que $A.B=B.A=I_n$. Se A não é inversível, dizemos que A é uma matriz singular.

Notação da matriz inversa A^{-1} no MATLAB é: **inv(A)**

Exemplos: Dadas as matrizes A e B, calcule A^{-1} e B^{-1}.

$$A = \begin{bmatrix} 5 & 6 \\ 4 & 5 \end{bmatrix} \Rightarrow A^{-1} = ?$$

```
>> A=[5 6;4 5];
>> INVERSA = inv(A)
INVERSA =
   5.0000  -6.0000
  -4.0000   5.0000
```

$$B = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 4 \end{bmatrix} \Rightarrow B^{-1} = ?$$

```
>> B=[1 0 1; 1 2 3; 1 2 4];
>> INVERSA = inv(B)
INVERSA =
   1.0000   1.0000  -1.0000
  -0.5000   1.5000  -1.0000
        0  -1.0000   1.0000
```

Sistemas Lineares

É um conjunto de m (m≥1) equações lineares, nas incógnitas $x_1, x_2, x_3, \dots, x_n$. Assim o sistema S é linear:

$$S\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad\quad \vdots \quad\quad\quad \vdots \quad\quad\quad\quad \vdots \quad\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n = b_m \end{cases}$$

Lembrando a definição de produto de matrizes, notemos que o sistema linear S pode ser escrito na forma matricial.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix} \quad \Rightarrow \quad \begin{aligned} & A \cdot X = B \\ & A^{-1} \cdot A \cdot X = A^{-1} \cdot B \\ & I.X = A^{-1} \cdot B \\ & X = A^{-1} \cdot B \end{aligned}$$

Notação no MatLab para esse tipo de situação:

X=inv(A)*B ou simplesmente **X = A\B**

Resolva o seguinte sistema linear:

$$\begin{cases} x + y + z = 6 \\ x - y - z = -4 \\ 2x - y + z = 1 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ 2 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 1 \end{bmatrix} \Rightarrow A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ 2 & -1 & 1 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} 6 \\ -4 \\ 1 \end{bmatrix}$$

Então:

```
>> A=[1 1 1; 1 -1 -1; 2 -1 1];
>> B=[6; -4; 1];
>> X=inv(A)*B

X =

    1
    3
    2
```

Determinantes

Consideremos o conjunto das matrizes quadradas de elementos reais. Seja M uma matriz de ordem n desse conjunto. Chamamos determinante da matriz M (det M) o número que podemos obter operando com os elementos de M

Notação do determinante da matriz M no MATLAB é: **det(M)**

Exemplos: Dadas as matrizes A e B, calcule det A e det B.

$$A = \begin{bmatrix} 5 & 6 \\ 4 & 5 \end{bmatrix} \Rightarrow \det A = ?$$

```
>> A=[5 6; 4 5];
>> DETERMINANTE = det(A)
DETERMINANTE =
    1
```

$$B = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 4 \end{bmatrix} \Rightarrow \det B = ?$$

```
>> B=[1 0 1; 1 2 3; 1 2 4];
>> DETERMINANTE = det(B)
DETERMINANTE =
    2
```

Com base nos conceitos desenvolvidos faça os exercícios a seguir:

1. Dadas as matrizes

$$A=\begin{bmatrix}1 & 5 & 7\\ 3 & 9 & 11\end{bmatrix},\quad B=\begin{bmatrix}2 & 4 & 6\\ 8 & 10 & 12\end{bmatrix}\quad e \quad C=\begin{bmatrix}0 & -1 & -5\\ 1 & 4 & 7\end{bmatrix}$$

Calcule: A+B+C, A-B+C, A-B-C e –A+B-C

2. Sendo $A=\begin{bmatrix}2 & 1\\ 3 & 2\\ -4 & 0\end{bmatrix}$ e $B=\begin{bmatrix}1 & 2\\ 3 & -3\\ 2 & 1\end{bmatrix}$ Calcule 2A-B

3. Calcule o produto das matrizes:

$$\begin{bmatrix}1 & -1 & 5 & 0\\ 2 & 3 & 7 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 1\\ 2 & 1\\ 3 & 1\\ 1 & 1\end{bmatrix}$$

$$\begin{bmatrix}1 & -1\\ 2 & 2\\ 3 & 4\end{bmatrix}\cdot\begin{bmatrix}1 & 2 & 3\\ 4 & -5 & 1\end{bmatrix}$$

4. Dada a matriz $X=\begin{bmatrix}1 & 2 & 5\\ -1 & 7 & 2\end{bmatrix}$ Calcule a sua transposta.

5. Se $Y=\begin{bmatrix}1 & 1 & 1\\ 2 & 3 & 4\end{bmatrix}$ calcule Z = ½.Y^t

6. Dadas as matrizes a seguir, calcule suas inversas:

$$A=\begin{bmatrix}2 & 5\\ 1 & 3\end{bmatrix}\quad e \quad B=\begin{bmatrix}1 & 9 & 5\\ 3 & 1 & 2\\ 6 & 4 & 4\end{bmatrix}$$

7. Determine a matriz X, tal que: $\begin{bmatrix}2 & 2\\ 5 & 5\end{bmatrix}+\begin{bmatrix}1 & 2\\ 3 & 5\end{bmatrix}\cdot X=\begin{bmatrix}1 & 7\\ 2 & 7\end{bmatrix}$

8. Resolva os seguintes sistemas lineares:

$$\begin{cases}5x+2y+3z=2\\ 3x+y+4z=-1\\ 4x-3y+z=3\end{cases}$$

$$\begin{cases} 3x + 5y + 2z = 26 \\ x - 7y + z = -16 \\ 5x - y + 3z = 14 \end{cases}$$

9. Calcule o valor de:

$$\det A = \begin{vmatrix} 1 & 1 & 9 \\ 1 & 8 & 7 \\ 1 & 5 & 3 \end{vmatrix}$$

$$\det B = \begin{vmatrix} 3 & 2 & 4 & 3 \\ 5 & 7 & 2 & 4 \\ 2 & 4 & 5 & 3 \\ 2 & 3 & 0 & 7 \end{vmatrix}$$

$$\det C = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{vmatrix} + \begin{vmatrix} 1 & 2 & 4 & 5 \\ 0 & 1 & 0 & 0 \\ 1 & 3 & 0 & 1 \\ 1 & 4 & 2 & 1 \end{vmatrix}$$

Capítulo – 05

Polinômios

Algumas vezes precisamos calcular as raízes de um polinômio. Isto pode ser feito no MATLAB através do comando **roots**. Veja o seguinte exemplo:

Calcule os zeros do seguinte polinômio: $p(x) = x^2 + 5x + 6$. Então:

```
>> p=[1 5 6];

>> raizes=roots(p)

raizes =

  -3.0000

  -2.0000
```

Em algumas situações temos as raízes e precisamos obter o polinômio, para tanto, utliza-se o comando **poly**. Veja o exemplo:

Sabendo que o polinômio tem duas raízes, a saber, -3 e -2 quais são os seus coeficientes? Então:

```
>> raizes=[-3 -2];

>> coeficientes=poly(raizes)

coeficientes =

    1    5    6
```

Como são duas raízes distintas, temos: $p(x)=x^2+5x+6$

Para obter o valor do polinômio num determinado ponto tem-se o comando **polyval**. Veja o exemplo: Sabendo que $p(x) = x^2 + 5x + 6$, calcule o valor no ponto x=7. Então:

```
>> p=[1 5 6];
>> polyval(p,7)
ans =
   90
```

Operações polinomiais

Para entender as operações que se seguem utilizaremos o seguinte exemplo:
Sejam os polinômios $p(x) = x^4 – 3x^2 +5x – 30$ e $w(x) = 2x^4 –7x^3 +2x –15$. Calcule:

Adição

Se a dimensão de **p** é igual à dimensão de **w**, a adição será dada por: c = p + w

Se a dimensão de **p** for diferente da dimensão de **w**, devemos:

- Preencher com zeros os coeficientes das potências que faltam em um polinômio para este igualarem em dimensão com outro
- A adição será dada por: c = p + w

```
>> p=[1 0 -3 5 -30];
>> w=[2 -7 0 2 -15];
>> Adicao=p+w
Adicao =
    3   -7   -3    7   -45
```

Portanto $p(x)+w(x)=3x^4-7x^3-3x^2+7x-45$

Multiplicação

Para multiplicar um polinômio pelo outro, utiliza-se o comando **conv(p,w)** entendendo que será feita a multiplicação do polinômio **p** pelo polinômio **w**. Então:

```
>> p=[1 0 -3 5 -30];
>> w=[2 -7 0 2 -15];
>> multiplicacao = conv(p,w)
multiplicacao =
    2   -7   -6   33  -110   204   55  -135   450
```

Portanto $p(x).w(x) = 2x^8-7x^7-6x^6+33x^5-110x^4+204x^3+55x^2-135x+450$

Divisão

Para efetuar a divisão entre polinômios, tem-se o seguinte procedimento

[q,r] = deconv (p,w)

Cuja resposta consta de duas variáveis**:**

q: é o quociente da divisão de p por w.

r: é o resto da divisão de p por w.

Então:

```
>> p=[1 0 -3 5 -30];
>> w=[2 -7 0 2 -15];
>> [q,r]=deconv(p,w)
q =
   0.5000
r =
        0    3.5000   -3.0000    4.0000  -22.5000
```

Derivada

Para determina a derivada de um polinômio, utiliza-se o comando **polyder (p)** onde serão exibidos os coeficientes do polinômio que representam a derivada. Então:

```
>> p=[1 0 -3 5 -30];
>> derivada = polyder(p)
derivada =
    4     0    -6     5
```

Portanto o polinômio $p(x) = x^4 – 3x^2 +5x – 30$ *tem como derivada* $p´(x)=4x^3-6x+5$

Capítulo – 06

Gráficos

Para traçar um gráfico, deve-se fundamentar na notação das funções, isto é, toda função é uma relação binária de A em B, portanto, toda função é um conjunto de pares ordenados.
Geralmente, existe uma sentença aberta y=f(x) que expressa a lei mediante a qual, dado x∈A, determina-se y∈B tal que (x,y)∈f, então:

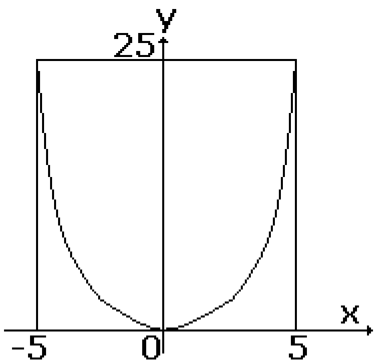
f = { (x,y) | x ∈ A, y ∈ B e y = f(x) }

Isto significa que, dados os conjuntos A e B, a função f tem a lei de correspondência y = f(x).

Assim sendo, apoiado nesse conceito, entendemos que devemos variar a abscissa (x), para determinarmos a ordenada (y) e desta forma traçar o gráfico.

Por exemplo: Tracemos o gráfico $y = x^2$
Matematicamente tem-se

x	y
-5	25
-4	16
-3	9
-2	4
-1	1
0	0
1	1
2	4
3	9
4	16
5	25



Para implementar o exemplo no Matlab a abscissa deve variar de -5 até 5, então:
```
>>x=-5:5;
```
Porém para calcular y devemos elevar x ao quadrado, então:
```
>>y=x^2;
??? Error using ==> mpower
Matrix must be square.
```
Isso nos obriga a entender um novo conceito, isto é:

Operações elemento – por – elemento

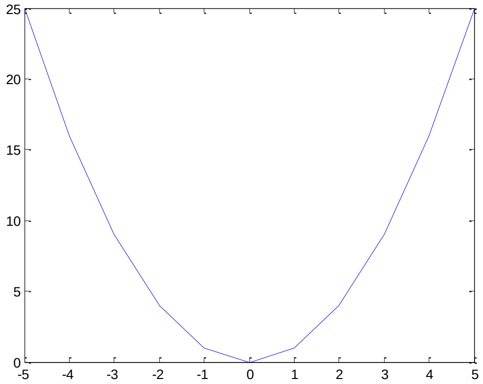
Ao contrário das operações tradicionais como multiplicação de matriz, são operações entre elementos. Para tanto, dispomos de uma notação especial, como mostra o quadro a seguir:

Símbolo	Operação
.*	Multiplicação
./ ou .	Divisão
.^	Potência

Utilizando essa notação não estaremos multiplicando a matriz X_{1x11} por X_{1x11} sendo isso que provocou o erro através da sintaxe >> y=x^2.

O correto seria >> y=x.^2, onde teríamos a seguinte multiplicação:
$x_{11}.x_{11}$, $x_{12}.x_{12}$, $x_{13}.x_{13}$, $x_{14}.x_{14}$, ...
(-5).(-5), (-4).(-4), (-3).(-3), (-2).(-2), ...

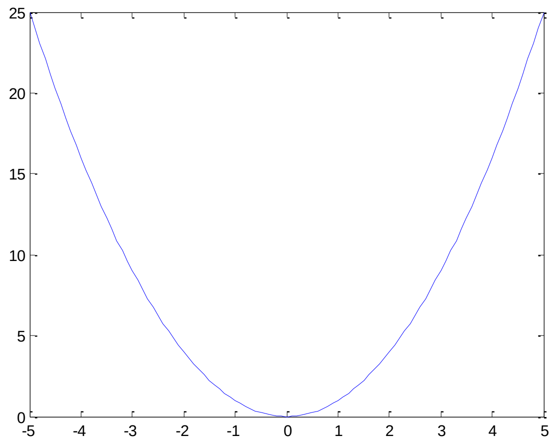
Assim sendo, o gráfico da função $y = x^2$, em Matlab ficaria:
>> x=-5:5;
>> y=x.^2;
>> plot(x,y)
Com o seguinte resultado:



Ainda assim, nota-se que a curva não tem muita precisão, pois o passo como não foi definido, assume o valor 1. Para que a curva fique mais precisa deve-se definir um passo diferente de 1, como é exemplificado a seguir:

>> x=-5:0.1:5;
>> y=x.^2;
>> plot(x,y)

Tem-se:



Como traçar um gráfico, variando o estilo da linha, as marcações dos pontos e a cor. Para isso existe um padrão, a saber:

Cor		**Marcador**		**Estilo**	
Código	Descrição	Código	Descrição	Código	Descrição
y	Amarelo	**.**	Ponto	**-**	Sólido
m	Magneta	**o**	Circulo	**:**	Pontilhado
c	Ciano	**x**	x	**-.**	Ponto-traço
r	Vermelho	**+**	mais	**--**	Tracejado
g	Verde	*****	asterisco	**<none>**	Sem linha
b	Azul	**s**	quadrado		
w	Branco	**d**	losango		
k	Preto	**v**	Triang.p/baixo		
		^	Triang.p/cima		
		<	Triang.p/esquerda		
		>	Triang.p/direita		
		p	Pentágono		
		h	Hexágono		
		<none>	Sem marcador		

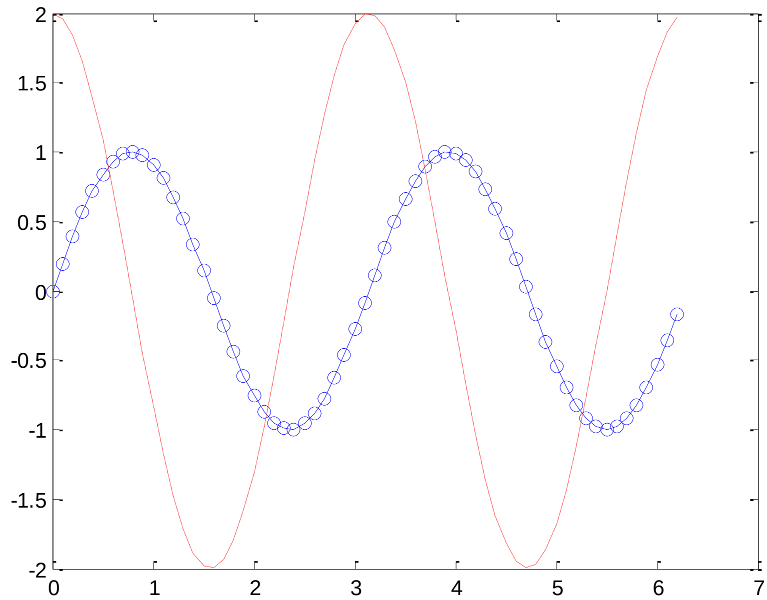
Para explorar os recursos apresentados, vejamos o seguinte exemplo:
Dada a função y=sen(2x), calcule a derivada e trace o gráfico de função, bem como da derivada.

$$y = sen(2x) \rightarrow função$$
$$y' = 2\cos(2x) \rightarrow derivada$$

Em Matlab vem:

```
>> x=0:0.1:2*pi;
>> y1=sin(2*x);
>> y2=2*cos(2*x);
>> plot(x,y1,'o-',x,y2,'--r')
```

Cujo resultado é:

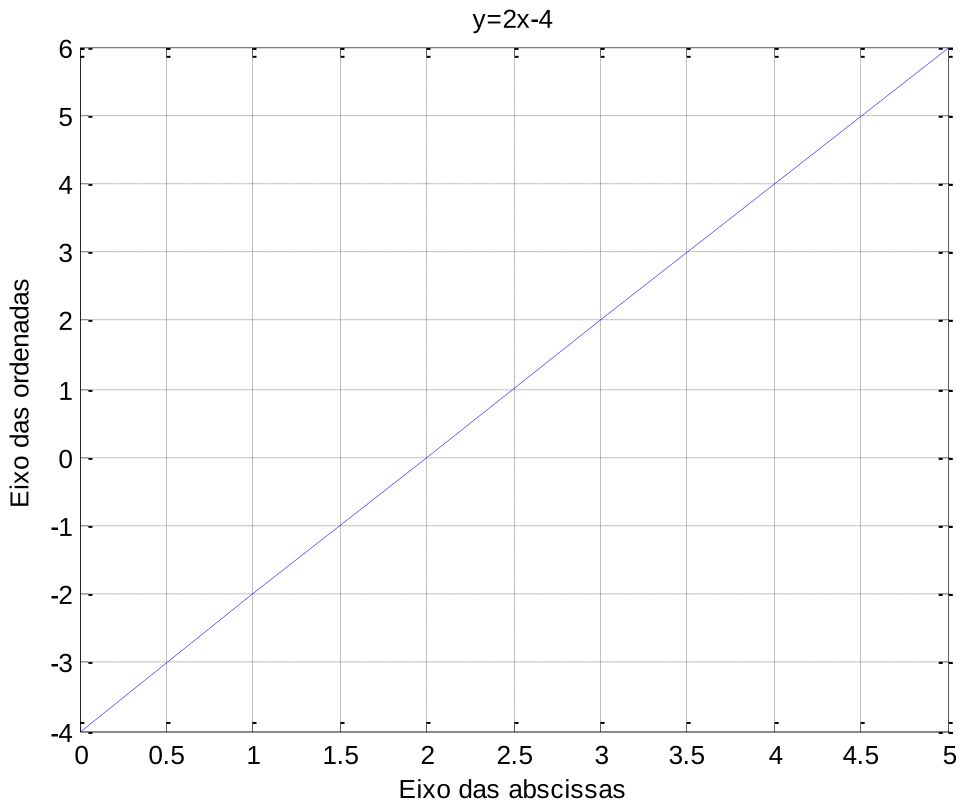


Para refinar o gráfico temos alguns comandos, a saber:

Como definir o eixo da abscissa com sua respectiva identificação
xlabel('texto');
Como definir o eixo da ordenada com sua respectiva identificação
ylabel('texto');
Como definir o título do gráfico com sua respectiva identificação
title('texto');
Como ativar as linhas de grade para referenciar o traçado
grid on;
Como desativar as linhas de grade para referenciar o traçado
grid off;

Para explorar esses comandos, façamos o gráfico da reta y=2x-4, com x ϵ [0,5].

```
>> x=0:5;
>> y=2*x-4;
>> plot(x,y)
>> xlabel('Eixo das abscissas');
>> ylabel('Eixo das ordenadas');
>> title('y=2x-4');
>> grid on;
```



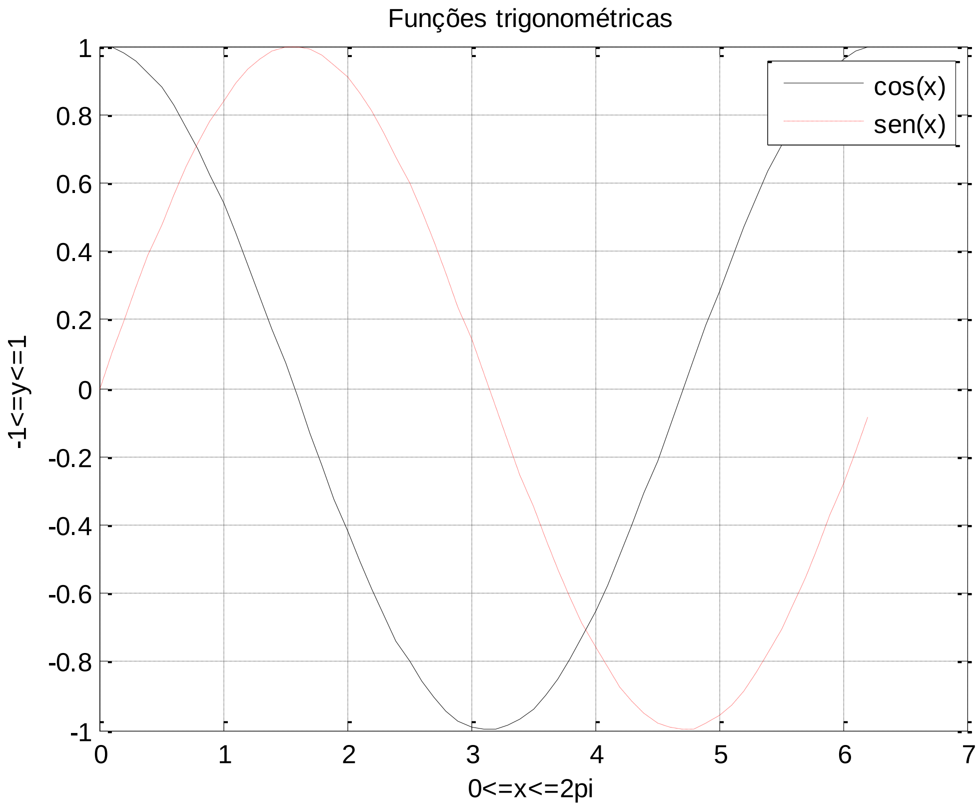
Como ativar/desativar a legenda em um gráfico?

Ativando legenda - comando: legend(‘texto’,’texto’,...’texto’,posição)
Desativando legenda – comando: legend off

Valor	Posição	Valor	Posição	Valor	Posição
0	Escolha automática	**2**	Canto superior esquerdo	**4**	Canto inferior direito
1	Canto superior direito	**3**	Canto inferior esquerdo	**-1**	À direita do desenho

Exemplo: y=cos(x) e y=sen(x)

```
>> x=0:0.1:2*pi;
>> y1=cos(x);
>> y2=sin(x);
>> plot(x,y1,'k',x,y2,':r');
>> grid on
>> xlabel('0<=x<=2pi');
>> ylabel('-1<=y<=1');
>> title('Funções trigonométricas');
>> legend('cos(x)','sen(x)')
```



Como traçar vários gráficos com distintos comandos plot?

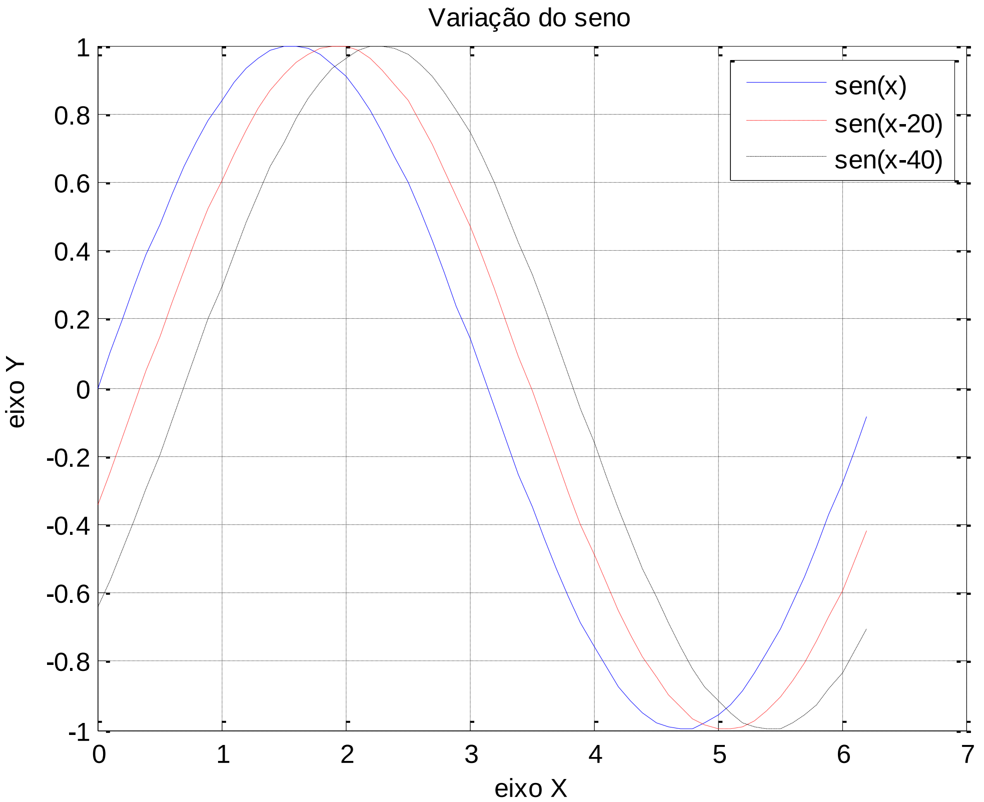
Para atender essa necessidade se tem o seguinte comando:

hold on – fixa as definições para os próximos gráficos subsequentes
hold off – desativa as definições

Para tanto trace os gráficos de cada função a seguir:

y=sen(x), y=sen(x-20^0) e y=sen(x-40^0), então:

```
>> x=0:0.1:2*pi;
>> y1=sin(x);
>> plot(x,y1);
>> hold on;
>> y2=sin(x-20*pi/180);
>> plot(x,y2,'r--');
>> y3=sin(x-40*pi/180);
>> plot(x,y3,'k-.');
>> grid on;
>> xlabel('eixo X');
>> ylabel('eixo Y');
>> title('Variação do seno');
>> legend('sen(x)','sen(x-20)','sen(x-40)');
```

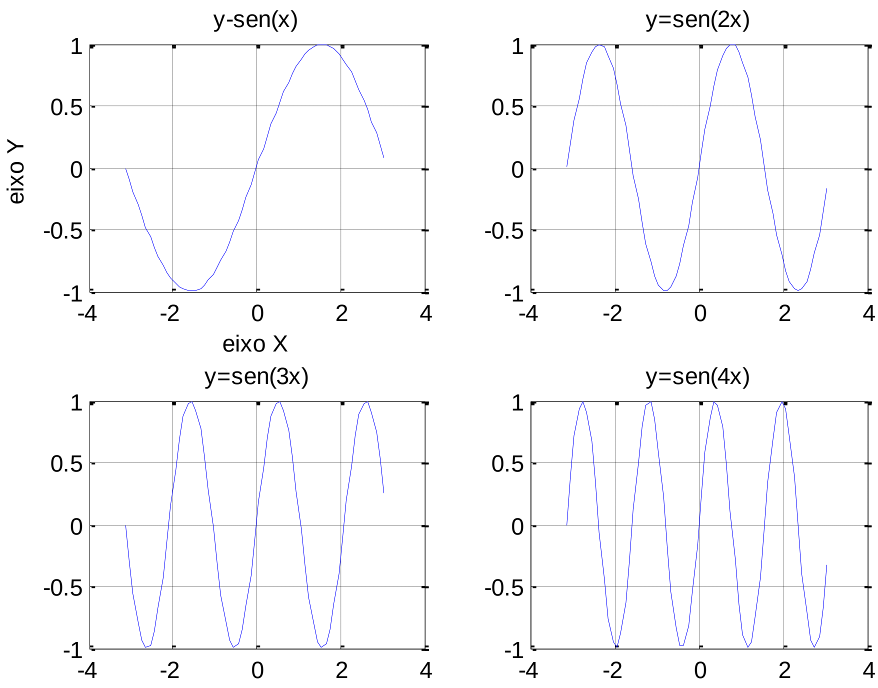


Como gerar vários gráficos cada qual em seu plano, isto é, sub-gráficos.

Comando: subplot(linha,coluna,plano em foco)

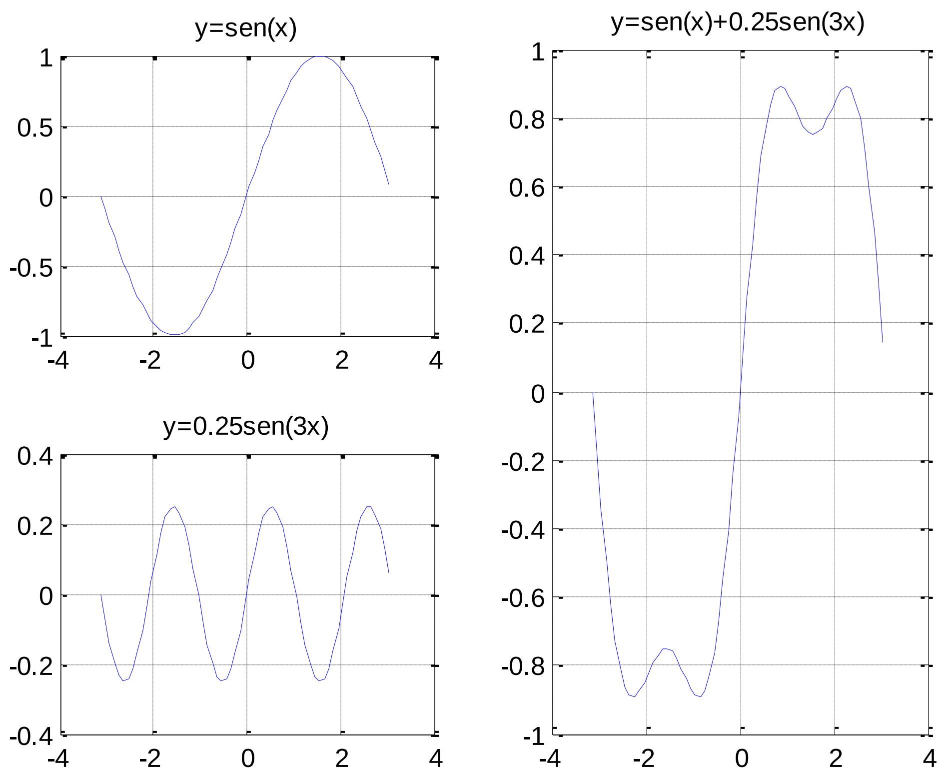
Exemplo: Vamos traçar os gráficos: sen(x), sen(2x),sen(3x) e sen(4x), com x∈[-π,π].

```
>> x=-pi:0.1:pi;
>> subplot(2,2,1); plot(x,sin(x));
>> grid on; title('y=sen(x)');
>> xlabel('eixo X'); ylabel('eixo Y');
>> subplot(2,2,2); plot(x,sin(2*x));
>> grid on; title('y=sen(2x)');
>> subplot(2,2,3); plot(x,sin(3*x));
>> grid on; title('y=sen(3x)');
>> subplot(2,2,4); plot(x,sin(4*x));
>> grid on; title('y=sen(4x)');
```



E se nos sub-gráficos a quantidade não for par?
Exemplo: Trace os seguintes gráficos y=sen(x), y=0.25sen(3x) e y=sen(x)+0.25sen(3x).

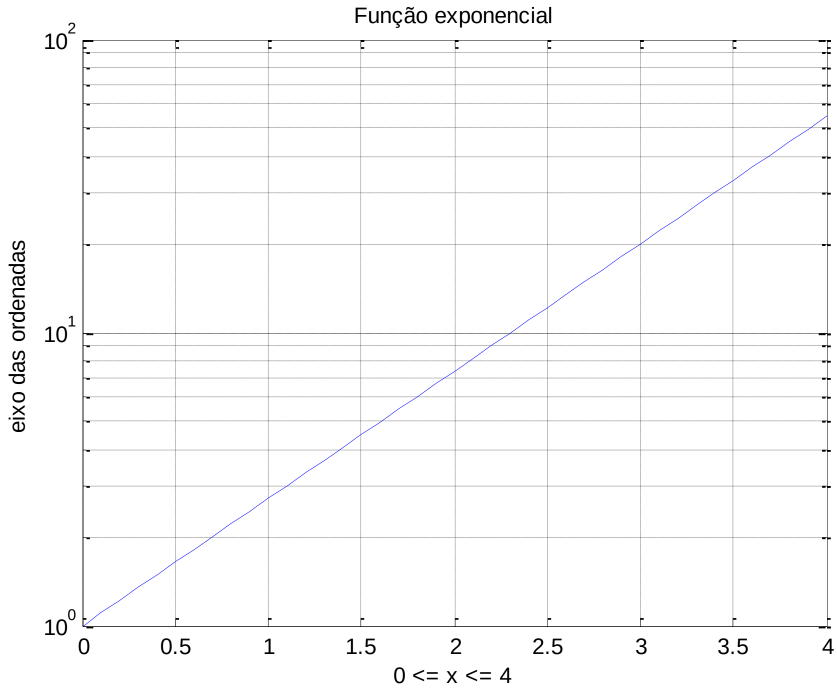
```
>> x=-pi:0.1:pi;
>> subplot(2,2,1); plot(x,sin(x));
>> grid on, title('y=sen(x)');
>> subplot(2,2,3); plot(x,0.25*sin(3*x));
>> grid on, title('y=0.25sen(3x)');
>> subplot(1,2,2); y=sin(x)+0.25*sin(3*x);
>> plot(x,y); grid on;
>> title('y=sen(x)+0.25sen(3x)');
```



Quando temos dados que variam em uma ampla gama de valores positivos, pode-se utilizar o recurso gráfico **monolog**, isto é, escala logarítmica, muito utilizado em várias situações da área de Engenharia. Para tanto, façamos o gráfico da função:

$$y = e^x \qquad com \qquad x \in [0,4] \qquad passo = 0.1$$

```
>> subplot(1,1,1);
>> x=0:0.1:4;
>> y=exp(x);
>> semilogy(x,y);
>> grid on;
>> xlabel('0 <= x <= 4');
>> ylabel('eixo das ordenadas');
>> title ('Função exponencial');
```



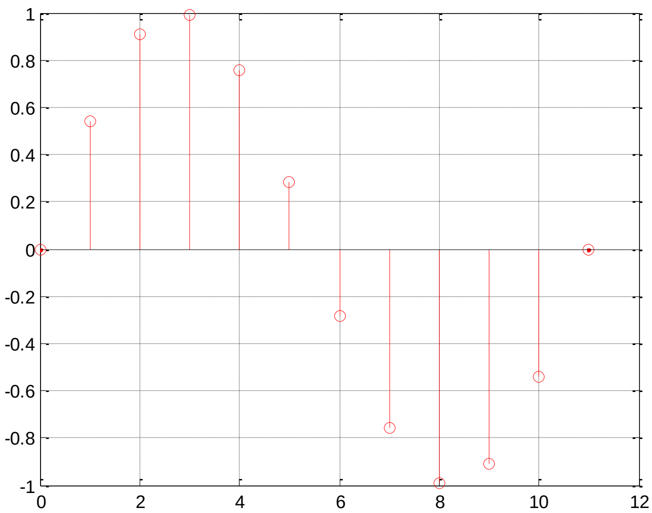
Como traçar um gráfico formado por pontos discretos?

Comando: stem(valores discretos, função)

Exemplo: Para n={0,1,2,3,4,5,6,7,8,9,10,11} (valores discretos), trace o gráfico da função:

$$y = sen\left(\frac{2\Pi n}{11}\right)$$

```
>> n=0:11;
>> y=sin(2*pi*n/11);
>> stem(n,y,'r');
>> grid on;
```



Com base nos conceitos desenvolvidos faça os exercícios a seguir:

1.Trace o gráfico das funções lineares a seguir sabendo que x Є [-10,10]

a) $y = \dfrac{2x-3}{2}$ b) $y = \dfrac{4-3x}{2}$

2.Tracem num mesmo gráfico as funções: $y = x^2 - 1 \quad e \quad y = -x^2 + 1$

Sabendo que x Є [-3,3] com passo 0.1. Sendo que a primeira função deverá estar na cor vermelha, circulo e sólido, enquanto que a segunda função deverá estar na cor preta e pontilhada.

$$y = x^2 = (1-\sqrt{3})x - \sqrt{3}$$

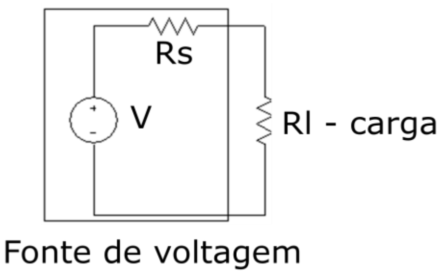
3.Trace o gráfico da função quadrática

com x Є [-2,2] com passo 0.1, defina também o eixo da abscissa, da ordenada e o título do gráfico.

4.Trace o gráfico y=2^x para x Є [-3,3] com passo 0.1, definindo o eixo da abscissa, ordenada, ativando a grade, fixada as definições trace o gráfico de $y = \log_2^x$ e gere a legenda.

5.Utilizando a escala molog construa o gráfico de temperatura (ºF)x(ºK), isto é, sabendo que a temperatura em (ºF) varia de -110 a 212 calcule a temperatura em (ºK), sabendo que $k = \frac{5}{9}(f - 32) + 273.15$

6.Sabendo que a fonte de voltagem V=120V com resistência interna Rs de 50 Ω e que a resistência de carga Rl varia de 1 Ω até 100 Ω. Faça o gráfico da resistência de carga (Rl) pela carga em potência máxima, sabendo que:



$$I = \frac{V}{R_{TOT}} = \frac{V}{Rs - Rl}$$

$$P = I^2 \cdot Rl$$

Capítulo – 07

Arquivos M

Matlab pode executar uma seqüência de declarações armazenada em arquivos chamados "**Arquivos M**" (devido à extensão "**.m**"). Muito do trabalho com Matlab está na criação e refinamento de arquivos M.

Os arquivos M podem ser **scripts** (roteiros) ou **funções**.

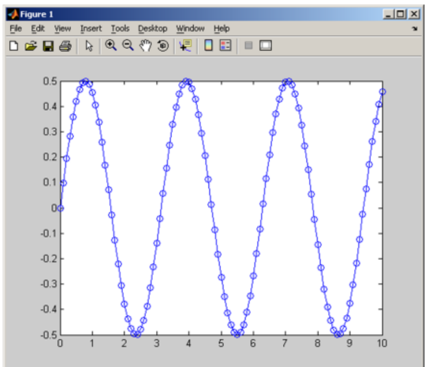
Um **script** consiste em uma seqüência de comandos do Matlab. Se o arquivo tem um nome, por exemplo, **teste.m**, então o comando **teste** fará o Matlab executar os comandos declarados no arquivo **teste.m**. **<u>Todas as variáveis em um script são globais</u>** e mudam os valores das variáveis de mesmo nome no ambiente da sessão de Matlab atual.

Arquivos funções fornecem flexibilidade ao Matlab. Pode-se criar novas funções específicas para o problema. Tais funções têm o mesmo status de qualquer outra função do Matlab. **<u>Em uma função as variáveis são locais</u>**.

Vamos inicialmente imaginar uma situação: “precisamos traçar o gráfico da função y=sen(x).cos(x) com x∈[0,10] e passo = 0.1”. Mas gostaríamos de guardar essa tarefa para utilizar em outra atividade.

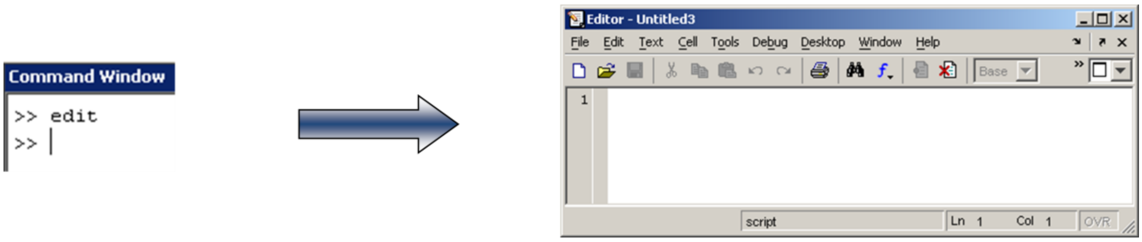
Command Window

```
>> x=0:0.1:10;
>> y=sin(x).*cos(x);
>> plot(x,y,'bo-')
>>
>> y=sin(x)*cos(x);
```

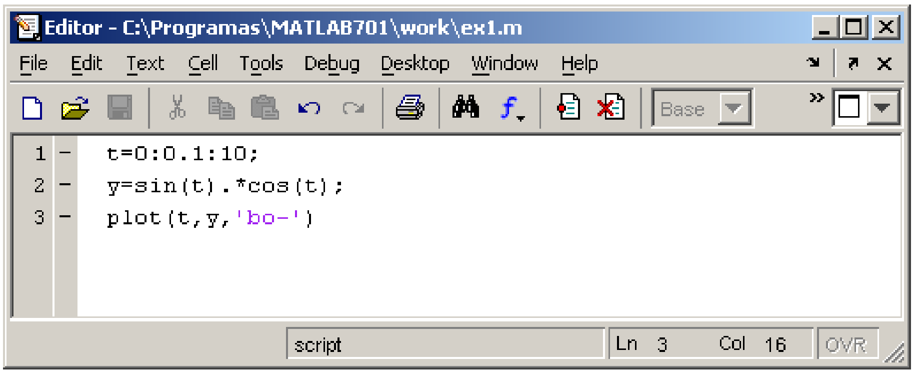


Como podemos observar a tarefa foi completada, mas está armazenada em memória, isto é, se fecharmos o Matlab teremos que digitar todos os códigos acima novamente. Para melhorar a performance iremos criar um arquivo na forma de script.

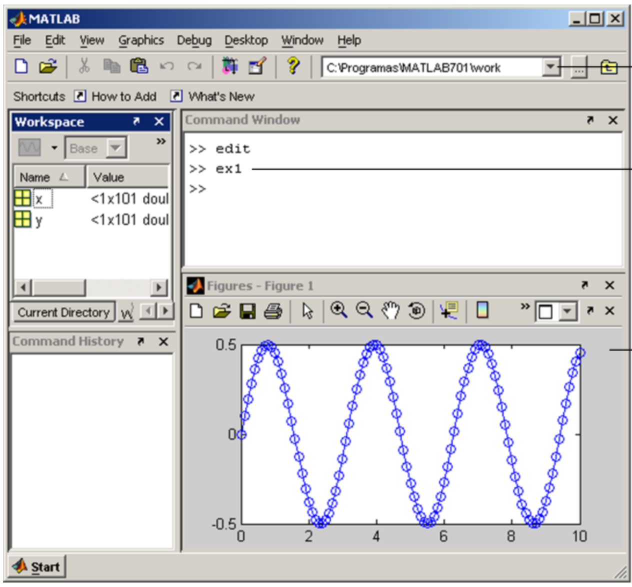
Então inicialmente no **command window** podemos ativar o editor nativo, porém um arquivo .m pode ser criado usando-se qualquer editor de texto.



Com o editor disponível, basta digitar os comandos e depois gravá-lo com o nome **ex1.m**, isto é:



Uma vez armazenado o script para poder acioná-lo, basta fazer o seguinte:



Local aonde foi gravado e script

Digita-se o nome do script e pressiona-se ENTER.

Resultado da execução do script

Agora mesmo que o Matlab seja desligado o script está armazenado para novas execuções, não sendo necessário editar o código, basta escrever o nome do script.

Para melhorar a implementação de um script em MatLab, discutiremos alguns comandos, a saber:

Comando: **%**

Finalidade: Utilizado para gerar comentário em uma linha do script.

Comando: **disp(‘texto’)**

Finalidade: Exibir uma mensagem para o usuário

Comando: **<variável> = input(‘texto’);**

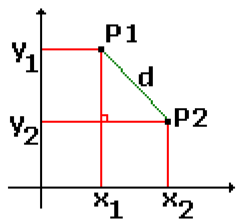
Finalidade: Atribuir para a variável um determinado valor em tempo de execução.

Comando: **fprintf(‘texto e caracteres de formatação de resultado’,<variável>);**

Finalidade: Exibir o resultado obedecendo a uma determinada formatação.

Caracteres de formatação de resultado

Caracter	Descrição
%d	Exibe valor como inteiro
%e	Exibe valor em formato exponencial
%f	Exibe valor em formato de ponto flutuante
%g	Exibe valor em formato de ponto flutuante ou exponencial o que for mais curto
%c.df	Exibe uma quantidade de casas numéricas (c) e fixa quantidade de casas decimais (d)
\n	Muda o prompt de linha.



Com base nesses conceitos vamos fazer um programa que:

Calcula a distância entre dois pontos (x1,y1) e (x2,y2) em um plano de coordenadas cartesianas, para tanto temos a seguinte equação: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Editor - C:\Programas\MATLAB701\work\distancia.m

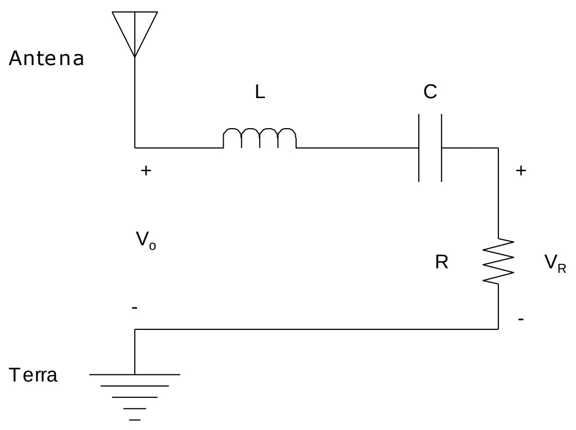
```
% Exibe a finalidade do programa
disp('Distancia entre dois pontos')
% entrada de dados P1 e P2
x1 = input('Dê a abscissa do primeiro ponto = ');
y1 = input('Dê a ordenada do primeiro ponto = ');
x2 = input('Dê a abscissa do segundo ponto = ');
y2 = input('Dê a ordenada do segundo ponto = ');
% calculo da distancia
d = sqrt((x1-x2)^2+(y1-y2)^2);
% exibe o valor da distancia
fprintf('A distancia vale %6.4f \n',d);
```

Então executando vem:

Command Window

```
>> distancia
Distancia entre dois pontos
Dê a abscissa do primeiro ponto = 3
Dê a ordenada do primeiro ponto = 2
Dê a abscissa do segundo ponto = 6
Dê a ordenada do segundo ponto = 10
A distancia vale 8.5440
```

Façamos então os seguintes scripts:



1. Observando a versão simplificada da parte frontal de um receptor de rádio AM, faça um programa interativo que calcula a carga resistiva através da fórmula abaixo. Para tanto, deve ser fornecido os seguintes dados, o Indutor (L), o Capacitor (C), a Resistência (R), a Voltagem (Vo) e a Freqüência (f). Então calcule ω = 2.π.f e

$$V_R = \frac{R}{\sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2}} V_o$$

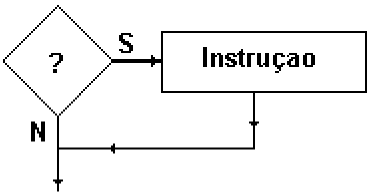
2. Fazer um script que calcula valores para a função *f* (*x*) = a.e^{-2x+b} e trace o gráfico *x versus f* (*x*), a partir dos valores inicial e final de *x* (*xi* e *x f, tendo 100 valores de x entre xi e xf*) e dos parâmetros *a* e *b*.

Capítulo – 08

Comando If

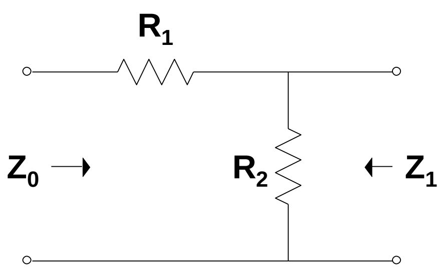
O comando **if** permite que se utilize *comandos condicionais* no MATLAB, tanto no Command Window como no interior de scripts ou funções.

Vejamos agora a variação do comando IF

Fluxograma	Algoritmo	Sintaxe no MatLab
	Se < condição > < instrução > Fim	if *<condição>* *<instrução>* end

Para exemplificar sua utilização façamos o seguinte programa:

Dada as impedâncias (Z0 e Z1) do circuito, calcule as resistências e a perda mínima, desde que Z0 > Z1. Conforme figura e formulas a seguir:



$$R_1 = z_0\sqrt{1-\frac{z_1}{z_0}} \qquad R_2 = \frac{z_1}{\sqrt{1-\frac{z_1}{z_0}}} \qquad PM = 20\cdot\log\left(\sqrt{\frac{z_0}{z_1}}+\sqrt{\frac{z_0}{z_1}-1}\right)$$

Editor - C:\Programas\MATLAB701\work\perda_minima.m

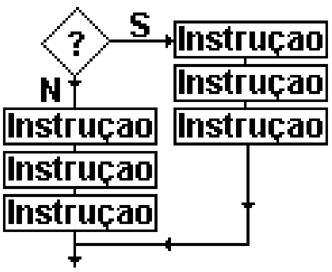
```
%Calculo das resistencias e perda mínima de um circuito
disp('Cálculo das resistências e perda minima de um circuito')
z0 = input('Dê a impedancia Z0 em Ohms = ');
z1 = input('Dê a impedancia Z1 em Ohms = ');
if z0>=z1
    r1=z0*sqrt(1-z1/z0);
    r2=z1/sqrt(1-z1/z0);
    pm=20*log10(sqrt(z0/z1)+sqrt(z0/z1-1));
    fprintf('R1 em Ohms = %f \n',r1);
    fprintf('R2 em Ohms = %f \n',r2);
    fprintf('Perda Mínima em dB = %f \n',pm);
end
```

Executando temos:

Command Window

```
>> perda_minima
Cálculo das resistências e perda minima de um circuito
Dê a impedancia Z0 em Ohms = 2
Dê a impedancia Z1 em Ohms = 3
>>
>> perda_minima
Cálculo das resistências e perda minima de um circuito
Dê a impedancia Z0 em Ohms = 3
Dê a impedancia Z1 em Ohms = 2
R1 em Ohms = 1.732051
R2 em Ohms = 3.464102
Perda Mínima em dB = 5.719475
>>
```

If...then...else

Fluxograma	Algoritmo	Sintaxe em MatLab
	Se <condição> <instrução> : <instrução> Senão < instrução > : < instrução > Fim	if *<condição>* *<instrução>* : <instrução> else *<instrução>* : <instrução> end

Para exemplificar sua utilização façamos o seguinte programa:

Suponha que um táxi esteja passando entre dois edifícios. Considere que a variável *d* contenha a distância do veículo ao edifício mais próximo. Se o carro estiver a 10 metros do edifício, a velocidade é calculada usando a seguinte equação:
$v = 0{,}425 + 0{,}00175d^2$.
Se o táxi estiver a uma distância maior que 10 metros, use a equação a seguirá:
$v = 0{,}625 + 0{,}12d - 0{,}00025d^2$

Programando temos:

```
Editor - C:\Programas\MATLAB701\work\velocidade.m
%Cálculo da velocidade do carro entre dois prédios
disp('Calculo da velocidade de um carro entre dois prédios')
d = input('Dê distancia em metros = ');
if d<=10
    v = 0.425 + 0.00175*d^2;
else
    v = 0.625 + 0.12*d - 0.00025*d^2;
end
fprintf('Velocidade em m/s² = %f \n',v);
```

Executando:

```
Command Window
>> velocidade
Calculo da velocidade de um carro entre dois prédios
Dê distancia em metros = 10
Velocidade em m/s² = 0.600000
>>
>> velocidade
Calculo da velocidade de um carro entre dois prédios
Dê distancia em metros = 30
Velocidade em m/s² = 4.000000
>>
```

Condição

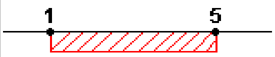
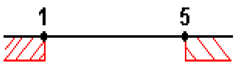
É a sintaxe da pergunta formulada na condicional para tanto, usa-se operadores e estes se dividem em dois grandes grupos, a saber:

Operadores relacionais

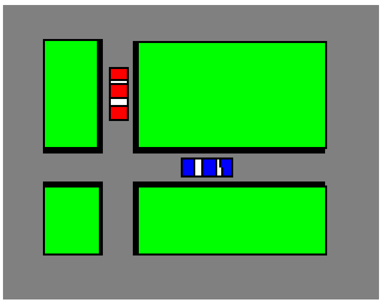
Estes operadores permitem que se estabeleça uma relação entre variáveis ou variáveis e constantes, assim sendo, utiliza-se os seguintes símbolos conforme tabela a seguir:

Notação Matemática	Notação em MatLab	Exemplos	
=	==	A == B	X == 2
>	>	A > B	X > 2
<	<	A < B	X < 2
≥	>=	A >= B	X >= 2
≤	<=	A <= B	X <= 2
≠	~=	A ~= B	X ~=2

Operadores lógicos

Exemplo	Notação Matemática	Notação MatLab
	$x \leq 1$ ou $x \geq 5$	(x <= 1) | (x >= 5)
	$1 \leq x \leq 5$	Não existe
	$x \geq 1$ e $x \leq 5$	(x >= 1) & (x <= 5)

Para exemplificar sua utilização façamos o seguinte programa:



Dada a velocidade inicial (m/s), aceleração (m/s²) e o intervalo de tempo (s). Calcule a velocidade (m/s) e o tipo de movimento, para tanto, se aceleração for zero a velocidade será igual à velocidade inicial e o tipo de movimento será uniforme. Porém, uma vez calculada a velocidade pela formula abaixo, deve ser analisada a aceleração se esta for maior que zero para qualquer tipo de velocidade o tipo de movimento será acelerado senão retardado. $v = v_0 t + \frac{a\sqrt{t}}{2}$

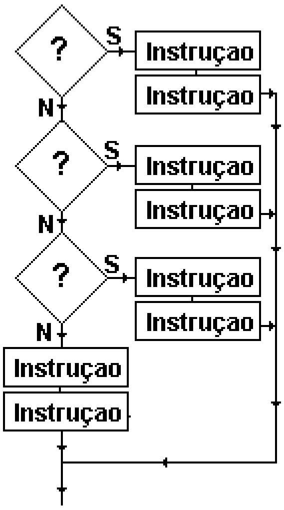
Editor - C:\Programas\MATLAB701\work\movimento.m

```
%Calculo da velocidade e classificação do tipo de movimento
disp ('Calculo da velocidade e classificação do tipo de movimento')
vo = input('Dê a velocidade inicial em (m/s) = ');
a = input('Dê a aceleração em (m/s²) = ');
t = input('Dê o intervalo de tempo em (s) = ');
if a==0
    mt = 'Uniforme';
    v=vo;
else
    v = vo*t+(a*sqrt(t))/2;
    if (v>0 & a>0) | (v<0 & a>0)
        mt = 'Acelerado';
    else
        mt = 'Retardado';
    end
end
fprintf ('Velocidade em (m/s) %f \n',v)
fprintf (mt);
```

Executando:

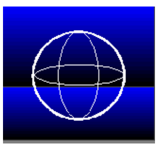
```
Calculo da velocidade e classificação do tipo de movimento
Dê a velocidade inicial em (m/s) = 10
Dê a aceleração em (m/s²) = 5
Dê o intervalo de tempo em (s) = 3
Velocidade em (m/s) 34.330127
Acelerado
>>
>> movimento
Calculo da velocidade e classificação do tipo de movimento
Dê a velocidade inicial em (m/s) = 10
Dê a aceleração em (m/s²) = 0
Dê o intervalo de tempo em (s) = 2
Velocidade em (m/s) 10.000000
Uniforme
>>
>> movimento
Calculo da velocidade e classificação do tipo de movimento
Dê a velocidade inicial em (m/s) = 10
Dê a aceleração em (m/s²) = -2
Dê o intervalo de tempo em (s) = 3
Velocidade em (m/s) 28.267949
Retardado
```

Condicional encadeada:

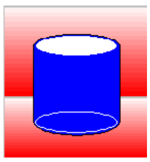
Fluxograma	Algoritmo	Sintaxe em MatLab
	Se <condição> <instrução> : <instrução> Senão Se < instrução > : < instrução > Senão Se < instrução > : < instrução > Senão < instrução > : < instrução > Fim	if *<condição>* *<instrução>* : <instrução> elseif <condição > *<instrução>* : <instrução> elseif <condição > *<instrução>* : <instrução> else *<instrução>* : <instrução> End

Façamos um programa utilizando esse recurso.

Tomemos três corpos de prova, a saber, um esférico outro cilíndrico e outro paralelogramo e calculemos o seu volume de acordo com a escolha do usuário. Isto é:
Se a escolha for 1 será esférico,
Se a escolha for 2 será cilíndrico e
Se a escolha for 3 será paralelogramo,
Qualquer outro valor deve ser desconsiderado com a respectiva mensagem
Uma vez feita a escolha deverá ser solicitadas as medidas necessárias para o cálculo do volume, a saber:

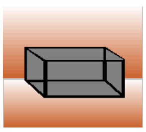


$$esf\acute{e}rico = \frac{4}{3} \cdot \pi \cdot r^3$$



$$cilindrico = \pi \cdot r^2 \cdot h$$

$$paralelogramo = h \cdot c \cdot l$$



```
%Volume do corpo de prova
disp('+-------------------------+')
disp('| CORPO DE PROVA          |')
disp('| 1. Esférico             |')
disp('| 2. Cilindrico           |')
disp('| 3. Paralelogramo        |')
disp('+-------------------------+')
escolha = input('Faça sua escolha: ');
if escolha == 1
   disp(' ')
   disp('Corpo de prova esférico')
   disp(' ')
   r = input('Dê o raio do corpo esférico em (cm) = ');
   v = (4/3)*(pi*r^3);
   fprintf ('Volume do corpo em (cm³) %f \n',v)
elseif escolha == 2
   disp(' ')
   disp('Corpo de prova cilindrico')
   disp(' ')
   r = input('Dê o raio do corpo cilindrio em (cm) = ');
   h = input('Dê a altura do corpo cilindrico em (cm) =');
   v = pi*r^2*h;
   fprintf ('Volume do corpo em (cm³) %f \n',v)
elseif escolha == 3
   disp(' ')
   disp('Corpo de prova paralelogramo')
   disp(' ')
   h = input('Dê a altura do corpo paralelogramo em (cm) = ');
   c = input('Dê o comprimento do corpo paralelogramo em (cm) = ');
   l = input('Dê a largura do corpo paralelogramo em (cm) = ');
   v = c*l*h;
   fprintf ('Volume do corpo em (cm³) %f \n',v)
else
  fprintf('Escolha fora da faixa')
end
```

Executando, temos:

```
>> corpo
+-------------------------+
| CORPO DE PROVA          |
| 1. Esférico             |
| 2. Cilindrico           |
| 3. Paralelogramo        |
+-------------------------+
Faça sua escolha: 1

Corpo de prova esférico

Dê o raio do corpo esférico em (cm) = 3
Volume do corpo em (cm³) 113.097336
```

```
>> corpo
+-------------------------+
| CORPO DE PROVA          |
| 1. Esférico             |
| 2. Cilindrico           |
| 3. Paralelogramo        |
+-------------------------+
Faça sua escolha: 2

Corpo de prova cilindrico

Dê o raio do corpo cilindrio em (cm) = 2
Dê a altura do corpo cilindrico em (cm) =4
Volume do corpo em (cm³) 50.265482
```

```
>> corpo
+-------------------------+
| CORPO DE PROVA          |
| 1. Esférico             |
| 2. Cilindrico           |
| 3. Paralelogramo        |
+-------------------------+
Faça sua escolha: 3

Corpo de prova paralelogramo

Dê a altura do corpo paralelogramo em (cm) = 3
Dê o comprimento do corpo paralelogramo em (cm) = 4
Dê a largura do corpo paralelogramo em (cm) = 2
Volume do corpo em (cm³) 24.000000
```

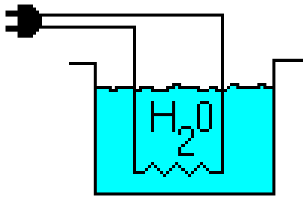
```
>> corpo
+-------------------------+
| CORPO DE PROVA          |
| 1. Esférico             |
| 2. Cilindrico           |
| 3. Paralelogramo        |
+-------------------------+
Faça sua escolha: 5
Escolha fora da faixa
```

Faça os seguintes programas

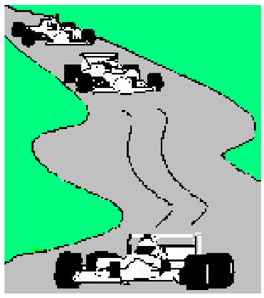
1. Nesse programa de ser fornecido o fluido (em litros), a temperatura inicial e final (em ºC), com relação ao aquecedor terá a potência (em W). Com essas informações deve-se calcular o tempo de aquecimento da água. Para tanto, leva-se em consideração que o fluido e os dados do aquecedor devem ser positivos, caso não seja deve exibir uma mensagem para o usuário. Para efetuar o cálculo solicitado devem-se levar em conta as seguintes fórmulas:

   $$Q = A \cdot 1000(TF - TI) \qquad E = Q \cdot 4{,}18 \qquad T = \frac{E}{P}$$

   Onde Q é o calor, E é a energia e T é o tempo.



2. Nesse programa de serem fornecidos os seguintes dados do ensaio, a saber: espaço inicial (em m), velocidade inicial (em m/s), aceleração (em m/s²) e o tempo (em s), com esses dados irão calcular o espaço percorrido (em m) e a velocidade (em m/s).Para tanto devemos levar em consideração a aceleração for igual a zero devemos efetuar o seguinte calculo (1) senão (2), como formulas abaixo:



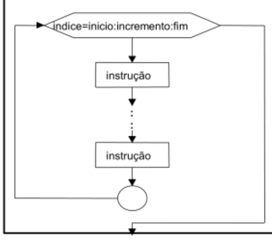
   (1) $\quad s = s_0 + v_0 \cdot t \qquad v = v_0$

   (2) $\quad s = s_0 + v_0 \cdot t + \dfrac{a \cdot t^2}{2} \qquad v = v_0 + a \cdot t$

Capítulo – 09

Estrutura de Repetição

Laço For...End

Fluxograma	Algoritmo	Sintaxe no MatLab
	Para <indice>=<inicio>:<incremento>:<fim> <instrução>; : <instrução>; Fim	For <indice>=<inicio>:<incremento>:<fim> <instrução>; : <instrução>; End

Observação importante:

A ausência do incremento implica que cada execução do laço seja somado 1 e o valor inicial deve ser sempre menor que o valor final.

Para exemplificar a utilização do laço for, façamos um exemplo que calcula o fatorial de um número natural qualquer. Isto é:

Sabemos matematicamente que:

0!=1 1!=1 n! = n.(n-1).(n-2)....3.2.1

```
%Calculo de um fatorial
disp('Cálculo de um fatorial')
n = input('Dê um número maior ou igual a zero = ');
if n < 0
    disp('Valor incompatível...');
else
    if n==0
        fprintf('Seu fatorial é 1 \n')
    elseif n==1
        fprintf('Seu fatorial é 1 \n')
    else
        num=1;
        for k = 1:n
            num = num*k;
        end
        fprintf('Seu fatorial é %d \n',num)
    end
end
```

Executando:

```
>> fatorial
Cálculo de um fatorial
Dê um número maior ou igual a zero = 0
Seu fatorial é 1
>>
>> fatorial
Cálculo de um fatorial
Dê um número maior ou igual a zero = 1
Seu fatorial é 1
>>
>> fatorial
Cálculo de um fatorial
Dê um número maior ou igual a zero = 5
Seu fatorial é 120
```

Calcule a distância percorrida por um projétil quando ele é lançado com uma velocidade inicial em m/s e ângulo inicial ***a*** em graus entre 5º e 85º, com incremento de 10º, então desenhe a trajetória do percurso. Assumiremos que não há resistência do ar.

Para efetuar os cálculos, devemos levar em consideração:

$$v_{Ox} = v_o \cdot \cos(a)$$

$$v_{Oy} = v_o \cdot sen(a)$$

$$Ins\tan te\_\max = \frac{v_{Oy}}{g}$$

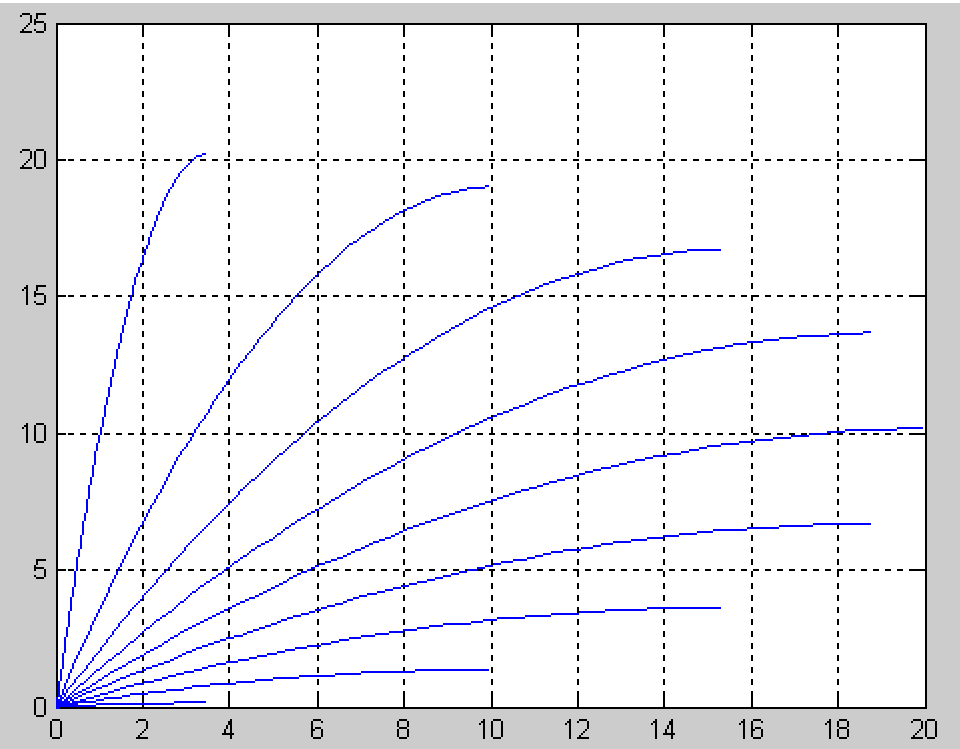
$$x(t) = v_{Ox} \cdot t$$

$$y(t) = v_{Oy} \cdot t - \frac{1}{2} \cdot g \cdot t^2$$

$$g = 9.81 (m/s^2)$$

```
%lançamento parabólico
disp('Gráfico do lançamento parabólico')
g = 9.81;
v = input('Dê a velocidade do projetil = ');
for a = 5:10:85
    ard = a * pi / 180;
    vOX = v * cos(ard);
    vOY = v * sin(ard);
    max = vOY/g;
    % zera as variáveis x e y
    x = zeros(1,50);
    y = zeros(1,50);
    for t = 1:50
      tempo = (t-1) * max/50;
      x(t) = vOX * tempo;
      y(t) = vOY * tempo - 1/2 * g * tempo^2;
    end
    plot(x,y);
    if a == 5
        hold on;
    end
end
grid on;
```

Executando:

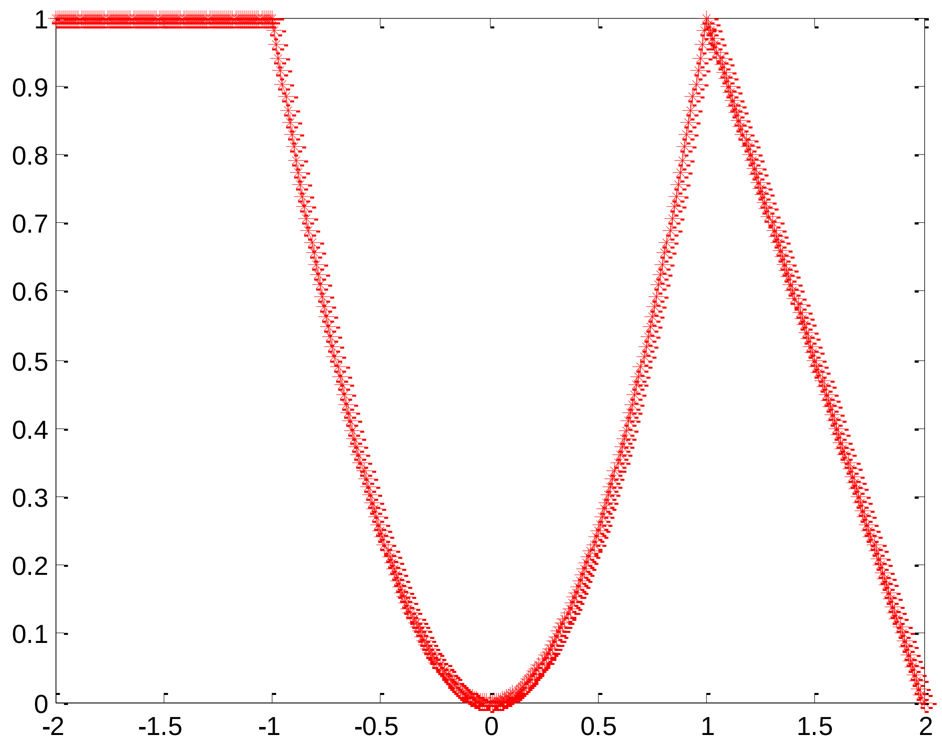


Façamos um programa que nos permita calcular valores para a função e traçar seu respectivo gráfico.

$$f(x) = \begin{cases} 1, se & x < -1 \\ x^2, se & -1 \le x \le 1 \\ -x+2, se & x > 1 \end{cases}$$

```
% traça o gráfico de uma função com intervalo
clear all;
clc;
n=0;
for t=-2:0.01:2
    n=n+1;
    x(n)=t;
    if x(n)<-1
      y(n) = 1;
    elseif x(n)>=-1 & x(n)<=1
      y(n) = x(n)^2;
    else
      y(n) = -x(n)+2;
    end
end
plot(x,y,'r*-')
```

Executando:



Vamos criar um programa que gere uma tabela de conversão de ºC (Celsius) para ºF (Farenheit) e ºK (Kelvin), entendendo que os ºC irão variar de 25 a 40 graus com incremento 1.

```
%converte temperatura em celsius para farenheit
clear all;
clc;
disp('Gera tabela de conversão de Celsius');
disp(' ');

fprintf('°C \t °F \t °K \n')

for C=25:40

    F = 1.8 * C + 32;
    k = 273.15+C;

    fprintf('%2.0f \t %5.1f \t %8.3f \n',[C F k])

end
```

Executando:

Gera tabela de conversão de Celsius para Farenheit

ºC	ºF	ºK
25	77.0	298.150
26	78.8	299.150
27	80.6	300.150
28	82.4	301.150
29	84.2	302.150
30	86.0	303.150
31	87.8	304.150
32	89.6	305.150
33	91.4	306.150
34	93.2	307.150
35	95.0	308.150
36	96.8	309.150
37	98.6	310.150
38	100.4	311.150
39	102.2	312.150
40	104.0	313.150

Conversão número para string e vice-versa

Para executar essa tarefa devemos conhecer alguns conceitos importantes, a saber:

num2str – é uma função responsável por converter um número em string
Sintaxe: num2str(número)

str2num – é uma função responsável por converte string em um número
Sintaxe: str2num(string)

Observação: para compor uma string que será exibida, mesclando string fixas com valores convertidos, devemos gerar uma variável string, concatenando as informações, isto é: <variável> = ['string fixa' string convertida];

Para utilizar esses conceitos, faça um programa que gere a tabuada de um número qualquer.

```
%tabuada de um número
clc;
clear all;
msg = 'Dê o número para gerar a tabuada = ';
n = input(msg);
for i = 1:10
    m = n*i;
    tabuada = [num2str(n) 'x' num2str(i) ' = ' num2str(m) '\n'];
    fprintf(tabuada);
end
```

Executando:

Dê o número para gerar a tabuada = 7
7x1 = 7
7x2 = 14
7x3 = 21
7x4 = 28
7x5 = 35
7x6 = 42
7x7 = 49
7x8 = 56
7x9 = 63
7x10 = 70

Faça agora os seguintes programas:

1.Dada a função abaixo, faça um programa que trace o seu respectivo gráfico:

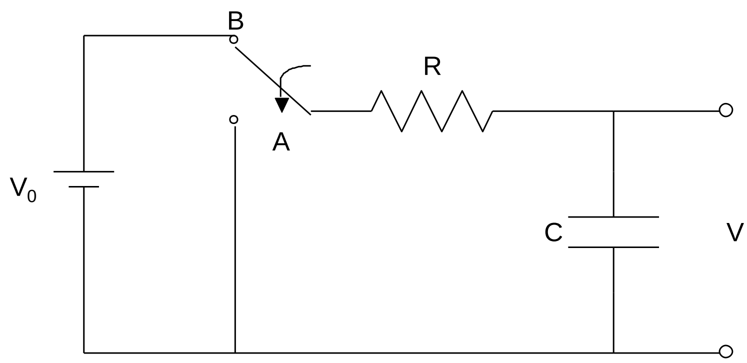
$$f(x) = \begin{cases} 4^{ex+2} & para \quad -6 \le x < -2 \\ x^2 & para \quad -2 \le x < 0 \\ (x+6.5)^{1/3} & para \quad 0 \le x \le 4 \end{cases}$$

2.Faça um programa que determine a soma dos m primeiros termos da série de Leibniz, aonde a quantidade m deve ser informada pelo usuário.

$$\sum_{n=0}^{m} (-1)^n \frac{1}{2n+1} \qquad com \quad (n = 0,1,2,\ldots,m)$$

3.Façamos um programa que gere uma tabela da tensão sobre o capacitor, em função do tempo, durante a descarga, para tanto, deverá ser fornecida as seguintes informações: V_0 a tensão inicial do capacitor, R a resistência do resistor e C a capacitância do capacitor.

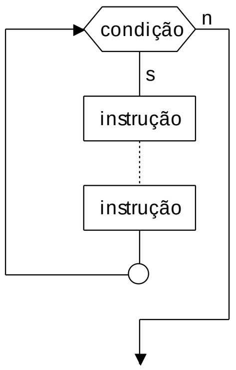
$$V = V_0 e^{\frac{-t}{RC}}$$



Capítulo – 10

Comando While

O loop *while* é uma importante estrutura para repetição de um grupo de comandos quando a condição especificada for verdadeira. O formato geral para esta estrutura de controle é:

Fluxograma	Algoritmo	Sintaxe no MatLab
	Enquanto < condição > < instrução >; : : < instrução >; Fim	while <*condição*> <*instrução*>; : : <*instrução*>; end

Se a expressão for verdadeira, então o grupo de comandos A é executado. Depois destes comandos serem executados, a condição é novamente questionada. Se for verdadeira, o grupo de comandos é novamente executado. Quando a condição for falsa, o controle pula para o comando posterior ao comando *end*. As variáveis modificadas no grupo de comandos A devem incluir as variáveis na expressão, ou o valor da expressão nunca será mudado. Se a expressão for verdadeira (ou é um valor não-nulo), o loop torna-se um loop infinito. (Lembre-se que você pode usar ^*c* para sair um loop infinito).

Como primeira aplicação, façamos um programa que traça uma função do segundo grau, aonde o usuário determina o intervalo da abscissa e os coeficientes da função. Isto é:

$$\lim_{inicial} \leq x \leq \lim_{final}$$

$$f(x) = ax^2 + bx + c$$

$$\lim_{inicial} = ?$$

$$\lim_{final} = ?$$

$$a = ?$$

$$b = ?$$

$$c = ?$$

```
%Traçar o gráfico de função quadrática num intervalo determinado
resp = 's';
while resp == 's'
   clear all;
   clc;
   disp('Função quadrática y=A.x²+B.x+C');
   disp(' ');
   li = input('Dê o limite inicial para x = ');
   lf = input('Dê o limite final para x = ');
     if li>=lf
     disp('LIMITE errado...');
     pause;
     resp = 's';
     else
       p = input('Dê a variação de x ');
       if p<=0
         disp('PASSO errado...');
         pause;
         resp = 's';
       else
         a = input('Dê o coeficiente A da função ');
         if a==0
           disp('Coeficiente A errado...');
           pause;
           resp = 's';
         else
           b = input('Dê o coeficiente B da função ');
           c = input('Dê o coeficiente C da função ');
           x = li:p:lf;
           y = a*x.^2 + b*x + c;
           plot(x,y,'*r-')
           resp = input('Nova função? (s/n)? ');
         end
       end
     end
end
```

Executando:

Função quadrática y=A.x²+B.x+C

Dê o limite inicial para x = 2

Dê o limite final para x = 1

LIMITE errado...

Função quadrática y=A.x²+B.x+C

Dê o limite inicial para x = 1

Dê o limite final para x = 2

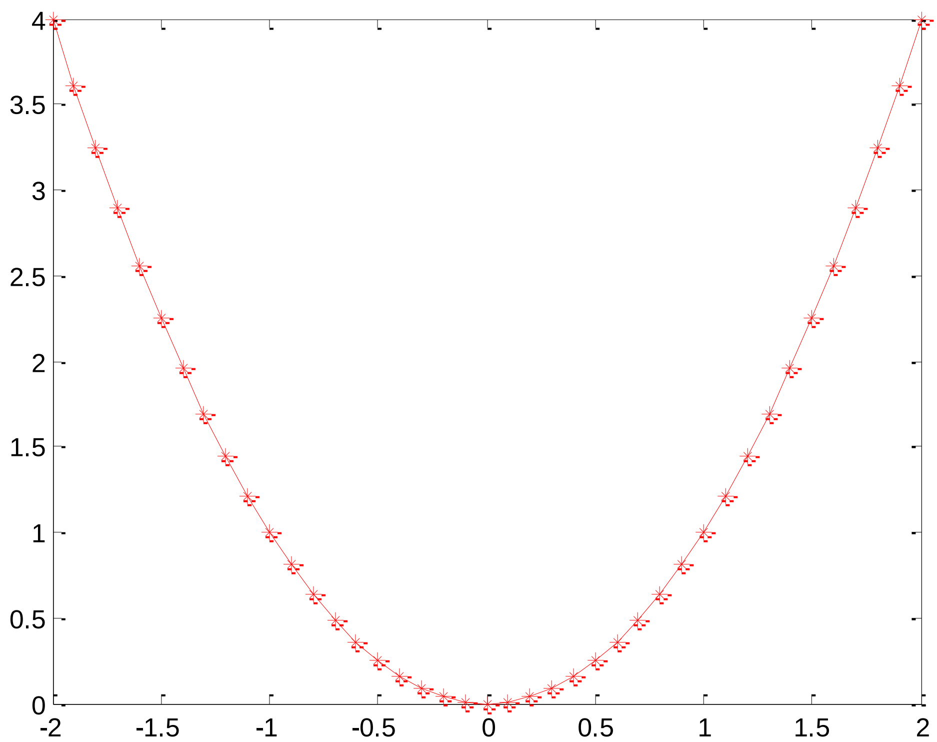
Dê a variação de x 0.1

Dê o coeficiente A da função 0

Coeficiente A errado...

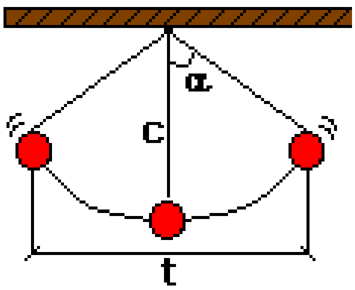
Função quadrática y=A.x²+B.x+C

Dê o limite inicial para x = -2
Dê o limite final para x = 2
Dê a variação de x 0.1
Dê o coeficiente A da função 1
Dê o coeficiente B da função 0
Dê o coeficiente C da função 0
Nova função? (s/n)?



Obs.: a resposta é uma string assim sendo se a resposta for sim digite ‘s’ caso a resposta seja não digite ‘n’

Vamos fazer um programa para calcular o período e o deslocamento de um pendulo conforme escolha do usuário, para tanto, será necessário fornecer o comprimento do pendulo em cm, o ângulo em graus e o tempo em segundos.



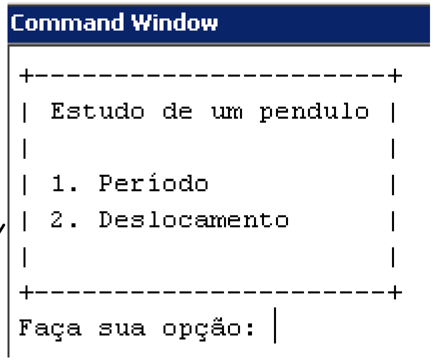
Os dados deverão ser consistidos para garantir sua validade operacional.
Para a confecção do programa devemos levar em consideração os seguintes conceitos:

A escolha deve ser 1 para **período** e 2 para **deslocamento**, qualquer outro valor deve ser ignorado.
O comprimento e o tempo deve ser um valor positivo.
Então:

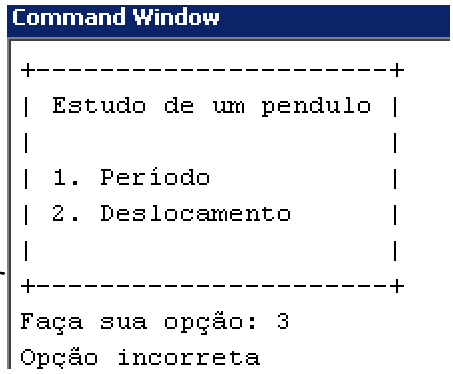
$$Periodo = p = 2 \cdot \pi . \sqrt{\frac{c}{980}}$$

$$Deslocamento = d = \alpha \cdot \cos\left(2 \cdot \pi \cdot \frac{t}{p}\right)$$

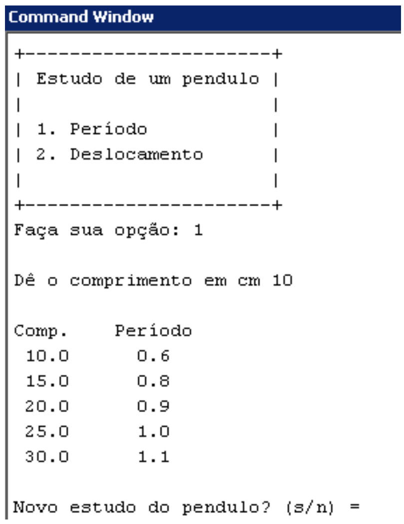
```
% Periodo e deslocamento de um pendulo
resp = 's';
while resp == 's'
   clc;
   clear all;
   disp('+--------------------------------+');
   disp('|      ESTUDO DE UM PENDULO      |');
   disp('|                                |');
   disp('|      1. Período                |');
   disp('|                                |');
   disp('|      2. Deslocamento           |');
   disp('|                                |');
   disp('+--------------------------------+');
   disp(' ');
   escolha = input('Faça sua opção: ');
   disp(' ');
   if escolha ~= 1 & escolha ~= 2
      fprintf('Opção errada.... \n');
      pause;
      resp = 's';
      continue;
   else
      if escolha == 2
        a = input('Dê o angulo em graus = ');
        t = input('Dê o tempo em segundos = ');
        if t <= 0
          fprintf('Tempo errado..... \n');
          pause;
          resp = 's';
        else
          c = input('Dê o comprimento em metros = ');
          if c<=0
            fprintf('Comprimento errado... \n');
            pause;
            resp='s';
          else
            c = input('Dê o comprimento em metros = ');
            if c<=0
              fprintf('Comprimento errado... \n');
              pause;
              resp='s';
            end
          end
        end
      disp(' ');
      if escolha == 1
         fprintf ('Comp. \t Periodo \n');
      else
```



```
Command Window
+----------------------+
| Estudo de um pendulo |
|                      |
| 1. Período           |
| 2. Deslocamento      |
|                      |
+----------------------+
Faça sua opção: |
```



```
Command Window
+----------------------+
| Estudo de um pendulo |
|                      |
| 1. Período           |
| 2. Deslocamento      |
|                      |
+----------------------+
Faça sua opção: 3
Opção incorreta
```



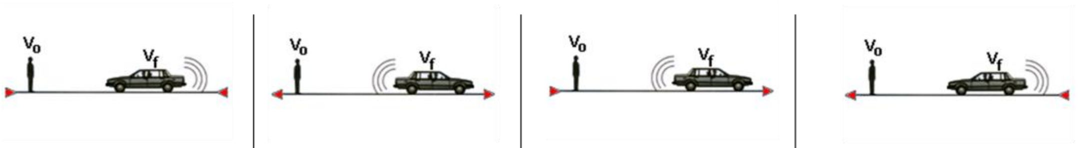
```
Command Window
+----------------------+
| Estudo de um pendulo |
|                      |
| 1. Período           |
| 2. Deslocamento      |
|                      |
+----------------------+
Faça sua opção: 1

Dê o comprimento em cm 10

Comp.    Período
 10.0      0.6
 15.0      0.8
 20.0      0.9
 25.0      1.0
 30.0      1.1

Novo estudo do pendulo? (s/n) =
```

```
          fprintf ('Ang \t Deslocamento \n');
       end
       if c>0 | (escolha==2 & t>0)
         controle = 0;
         while controle <= 4
           p = 2*pi*sqrt(c/980);
           if escolha ==1
             fprintf ('%5.1f \t %8.3f \n',[c p]);
             c = c+5;
           else
             ard = a*pi/180;
             d = ard*cos(2*pi*(t/p));
             fprintf ('%4.0f \t %8.3f \n',[a d]);
             a = a+10;
           end
           controle = controle + 1;
         end
       end
       disp(' ');
       resp = input('Novo estudo do pendulo? (s/n) = ');
       end
    end
end
```

Command Window

```
+----------------------+
| Estudo de um pendulo |
|                      |
| 1. Período           |
| 2. Deslocamento      |
|                      |
+----------------------+
Faça sua opção: 2

Dê o comprimento em cm 10
Dê o angulo em graus 30
Dê o tempo em segundos 5

Ang       Deslocamento
  30        0.4
  40        0.5
  50        0.6
  60        0.8
  70        0.9

Novo estudo do pendulo? (s/n) =
```

Façamos um programa que estudo o efeito DOPPLER. Para tanto, devemos levar em consideração a velocidade da fonte (V_f) em m/s, velocidade do observador (V_o) em m/s e a freqüência da fonte (f) em kHz. Ao estudar o efeito DOPPLER, temos que escolher como a fonte e o observador se comportam, isto é, se o efeito é convergente, divergente, sentido da fonte ou sentido do observador.
Então temos as seguintes possibilidades:



Convergente | Divergente | Sentido da fonte | Sentido do observador

Assim é possível calcular a freqüência ouvida pelo observador à medida que a velocidade do observador aumenta ou diminui de uma unidade em relação à fonte de acordo com o caso em estudo. Cabe destacar que estamos consideração o meio de propagação o ar cuja velocidade é de 340 m/s e cuja temperatura é de 25ºC. Com base nessas informações vamos gerar uma tabela da velocidade e freqüência em relação ao observador.

```
%Efeito DOPPLER
clear all;
clc;
resp = 's';
while resp == 's'
    clc;
    clear all;
    disp (' ');
```

```
disp ('+--------------------------+');
disp ('|      Efeito DOPPLER      |');
disp ('|                          |');
disp ('| 1. Convergente           |');
disp ('| 2. Divergente            |');
disp ('| 3. Sentido da fonte      |');
disp ('| 4. Sentido do observador |');
disp ('|                          |');
disp ('+--------------------------+');
disp (' ');
escolha = input('Faça sua opção: ');
if escolha ~=1 && escolha ~=2 && escolha ~= 3 && escolha ~= 4
    fprintf('Opçao errada \n');
    pause;
    resp='s';
else
    disp(' ');
    f = input('Dê a frequência da fonte em (kHz) ');
    if f<=0
        fprintf ('FREQUÊNCIA errada...');
        pause;
        resp = 's';
        continue;
    end
    vf = input('Dê a velocidade da fonte em (m/s) ');
    if vf<=0
        fprintf('VELOCIDADE DA FONTE errada...');
        pause;
        resp = 's';
        continue;
    end
    vo = input('Dê a velocidade do observador em (m/s) ');
    if vo<=0
        fprintf('VELOCIDADE DO OBSERVADOR errada...');
        pause;
        resp = 's';
        continue;
    end
    disp(' ');
    fprintf ('V_Observador \t F_Observador \n');
    controle = 0;
    while controle <=4
        if escolha == 1
            fo = f*((340+vo)/(340 - vf));
            fprintf('%10.4f \t %12.4f \n',[vo fo]);
            vo = vo+1;
        elseif escolha == 2
            fo = f*((340-vo)/(340 + vf));
            fprintf('%10.4f \t %12.4f \n',[vo fo]);
            vo = vo-1;
            if vo<0
                break;
            end
        elseif escolha == 3
            fo = f*((340+vo)/(340 + vf));
            fprintf('%10.4f \t %12.4f \n',[vo fo]);
            vo = vo+1;
        elseif escolha == 4
            fo = f*((340-vo)/(340 - vf));
            fprintf('%10.4f \t %12.4f \n',[vo fo]);
            vo = vo-1;
```

```
                if vo<0
                    break;
                end
            end
            controle = controle + 1;
        end
        disp(' ');
        resp = input('Novo estudo (s/n) = ');
    end
end
```

Façamos um programa para exibir a velocidade média do fluxo na seção total de água corrente de um canal.

Para tanto, será necessário receber as seguintes informações: a profundidade média hidráulica em (m), a diferença de superfície em (m), o comprimento do curso da água em (m) e o coeficiente de Manning. Deverá ser perguntado se existe variação da profundidade média hidráulica se a resposta for não deve assumir o valor zero, caso haja deverá ser fornecido esse valor em (m) sendo o mesmo acrescido na profundidade média hidráulica fornecida inicialmente fazendo dez interações gerando uma tabela da profundidade pela velocidade média em cada simulação.

Ao final deve haver a pergunta se haverá novo ensaio. Caso a resposta seja sim retorne ao inicio do programa e se a resposta for não encerre o programa.

Para tanto temos as seguintes fórmulas:

$$aux = \frac{D}{C} \qquad Veloc\_Média = \frac{\sqrt{(P \cdot aux)^3}}{N}$$

Onde:
D = Diferença de superfície
C = Comprimento do curso da água;
P = Profundidade média hidráulica;
N = Coeficiente de Manning.

Capítulo – 11

Formatos Numéricos

Quando o MATLAB mostra um resultado numérico ele segue certas regras.

No caso de nenhum formato estar definido, se um resultado é um número inteiro, o MATLAB mostra como um inteiro.

```
>> resultado = 2*3

resultado =

     6
```

Quando um resultado é um número real, o MATLAB mostra uma aproximação com até quatro casas decimais.

```
>> resultado = 3/2

resultado =

    1.5000
```

Se os dígitos significativos estiverem fora desta faixa, o MATLAB mostra o resultado em notação científica.

```
>> resultado = 0.00010011

resultado =

  1.0011e-004
```

Você pode definir um formato diferente. Vejamos agora algumas possibilidades:

Comando: **format short** → exibe o resultado com 4(quatro) dígitos decimais (formato padrão). Veja o exemplo:

```
>> format short
>> resultado = exp(1)

resultado =

    2.7183
```

Comando: **format short g** → exibe o resultado com 5(cinco) dígitos decimais com ou sem expoente.Veja o exemplo:

```
>> format short g
>> resultado = exp(1)

resultado =

       2.7183
```

Comando: **format short e** → exibe o resultado com 5(cinco) dígitos decimais mais expoente.Veja o exemplo:

```
>> format short e
>> resultado = exp(1)

resultado =

  2.7183e+000
```

Comando: **format long** → exibe o resultado com 14 (quartoze) dígitos decimais. Veja o exemplo:

```
>> format long
>> resultado = exp(1)

resultado =

   2.71828182845905
```

Comando: **format long g** → exibe o resultado com 15 (quinze) dígitos decimais,com ou sem expoente. Veja o exemplo:

```
>> format long g
>> resultado = exp(1)

resultado =

          2.71828182845905
```

Comando: **format long e** → exibe o resultado com 15 (quinze) dígitos decimais mais expoente. Veja o exemplo:

```
>> format long e
>> resultado = exp(1)

resultado =

    2.718281828459046e+000
```

Comando: **format bank** → exibe o resultado com 2(dois) dígitos decimais. Veja o exemplo:

```
>> format bank
>> resultado = exp(1)

resultado =

          2.72
```

Comando: **format compact** → exibe o resultado suprimindo as linhas adicionais. Veja o exemplo:

```
>> format compact
>> resultado = exp(1)
resultado =
          2.72
```

Comando: **format loose** → exibe o resultado restabelecendo as linhas adicionais. Veja o exemplo:

```
>> format loose
>> resultado = exp(1)

resultado =

          2.72
```

Comando: **format rat** → exibe a razão aproximada entre inteiros pequenos. Veja o exemplo:

```
>> format rat
>> resultado = exp(1)

resultado =

     1457/536
```

Variáveis permanentes

Existem algumas variáveis que são intrínsecas ao MATLAB e que não podem ser apagadas. Algumas são interessantes:

ans	Resposta mais recente, que não foi atribuída a nenhuma variável.	flops	Contador de operações matemáticas
eps	Precisão da máquina	NaN	Not a Number (indeterminação)
realmax	Maior número de ponto flutuante	inf	Infinito
realmin	Menor número de ponto flutuante	computer	Tipo de computador
pi	3.14159265358979	why	Resposta sucinta
i, j	Unidade imaginária	version	Versão do Matlab

Funções matemáticas comuns:

Função	Descrição
abs(x)	Modulo ou valor absoluto de x
cosh(x)	Coseno hiperbólico de x
gcd(x,y)	MDC dos inteiros x e y
imag(x)	Parte imaginária de um número complexo
lcm(x,y)	MMC dos inteiros x e y
max(x)	Retorna o máximo valor do vetor x
mean(x)	Retorna a média aritmética do vetor x
min(x)	Retorna o menor valor do vetor x
real(x)	Parte real de um número complexo
round(x)	Arredonda o valor de x
sinh(x)	Seno hiperbólico de x
std(x)	Retorna o desvio padrão do vetor x

Capítulo – 12

Limite

Seja I um intervalo aberto ao qual pertence o número real ***a***. Seja f uma função definida para x Є I – {a}. Dizemos que o limite de f(x), quando x tende a ***a***, é L e escrevemos:

$$\lim_{x \to a} f(x) = L \Leftrightarrow \left(\forall \varepsilon > 0,\ \exists \delta > 0 \ \middle|\ 0 < |x-a| < \delta \Rightarrow |f(x) - L| < \varepsilon \right)$$

Em Matlab temos a seguinte sintaxe:

Operação Matemática	Comando no MatLab
$\lim\limits_{x \to 0} f(x)$	limit(f)
$\lim\limits_{x \to a} f(x)$	limit(f,x,a) ou limit(f,a)
$\lim\limits_{x \to a-} f(x)$	limit(f,x,a,'left')
$\lim\limits_{x \to \infty} f(x)$	limit(f,x,a,'right')

Para executar o limite no MatLab se faz necessário definir a variável que será calculada, para tanto, temos o seguinte comando:

syms <variavel> <variavel> ... <variavel>

Obs.: É possível declarar apenas uma ou no caso de haver mais de uma declara-se a variável separada por espaço.

Exemplos:

$$\lim_{x \to 0} \frac{sen(x)}{x}$$

```
>> syms x

>> resultado = limit(sin(x)/x)

resultado =

1
```

Definição de Derivada

Seja f uma função definida em um intervalo aberto I e x_0 um elemento de I. Chama-se derivadas de f no ponto x_0 o limite $\lim\limits_{x \to x_0} \dfrac{f(x) - f(x_0)}{x - x_0}$ se este existir e for finito.

A derivada de f no ponto x_0 pode ser indicada das seguintes formas:

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} \quad ou$$

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \quad ou$$

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Exemplos:

$$\lim_{x \to 1} \frac{x^2 - 5x + 4}{|x - 1|} \longrightarrow$$

```
>> resultado = limit((x^2-5*x+4)/abs(x-1),x,1)

resultado =

NaN ——→ ∄ (não existe)
```

$$\lim_{x \to 1-} \frac{x^2 - 5x + 4}{|x - 1|} \longrightarrow$$

```
>> resultado = limit((x^2-5*x+4)/abs(x-1),x,1,'left')

resultado =

3
```

$$\lim_{x \to 1+} \frac{x^2 - 5x + 4}{|x - 1|} \longrightarrow$$

```
>> resultado = limit((x^2-5*x+4)/abs(x-1),x,1,'right')

resultado =

-3
```

$$\lim_{x \to \infty} \frac{3 - 2x}{5x + 1} \longrightarrow$$

```
>> resultado = limit((3-2*x)/(5*x+1),x,inf)

resultado =

-2/5
```

No Matlab temos a seguinte sintaxe:

***diff*(f, var, ord)**, onde:
- **f**: função literal de uma ou mais variáveis
- **var**: variável em relação a qual deseja-se derivar
- **ord**: ordem da derivada a ser calculada

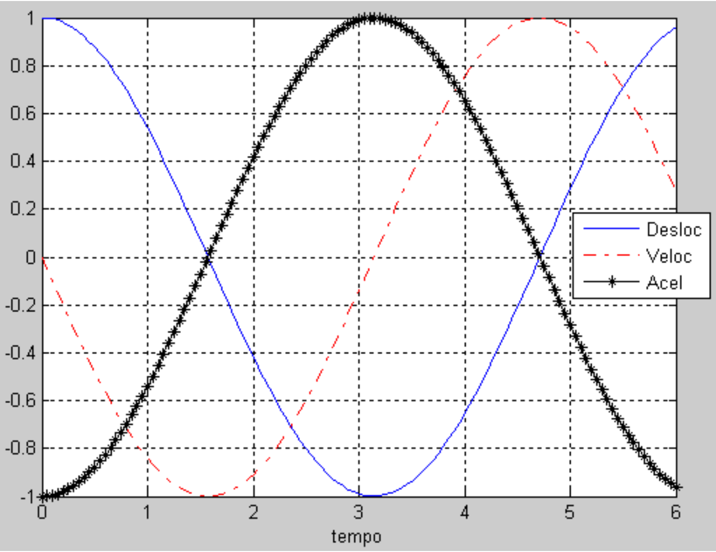
Assim, tem-se:

***diff*(f)** → derivada de **f** em relação a **x**
***diff*(f,s)** → derivada de **f** em relação a **s**
***diff*(f,2)** → derivada de **f** duas vezes em relação a **x**
***diff*(f,s,2)** → derivada de **f** em relação a **s** duas vezes

Aplicação: Um móvel desloca-se sobre um segmento de reta obedecendo à equação horária s = cos(t) através do sistema internacional de unidades, então determine a equação da velocidade e da aceleração. Lembrando que: v(t)=s'(t) e a(t)=v'(t). Faça então um programa que trace o gráfico das três equações para 0≤t ≤6.

Pelo Command Window

```
>> syms t;
>> s = cos(t);
>> v = diff(s);
>> a = diff(v);
>> v

v =

-sin(t)


>> a

a =

-cos(t)
```



Gerando as linhas de código, vem:

```
syms t
S = cos(t);
V = diff(S);
A = diff(V);
%ezplot(M); hold on; ezplot(P); hold off;
T = 0:0.05:6;
ST = double(subs(S,t,T));
VT = double(subs(V,t,T));
AT = double(subs(A,t,T));
plot(T,ST,'b',T,VT,'r-.',T,AT,'k*-')
title(' ')
legend('Desloc','Veloc','Acel')
xlabel('tempo'); grid
```

Exemplos:

$f(x) = sen\left(\frac{x^3}{x^2+1}\right)$

```
>> syms x
>> derivada = diff(sin(x^3/(x^2+1)))

derivada =

cos(x^3/(x^2+1))*(3*x^2/(x^2+1)-2*x^4/(x^2+1)^2)
```

$\longrightarrow f'(x) = \cos\left(\frac{x^3}{x^2+1}\right)\cdot\left(\frac{3.x^2}{x^2+1} - \frac{2.x^4}{(x^2+1)^2}\right)$

$f(x) = sen^2(x).\ln x^3$

```
>> derivada = diff((sin(x))^2*log(x^3))

derivada =

2*sin(x)*log(x^3)*cos(x)+3*sin(x)^2/x
```

$\longrightarrow f'(x) = 2.sen(x).\ln(x^3).\cos(x) + \frac{3.sen^2(x)}{x}$

$f(x) = \cos(x) - \frac{1}{3}\cos^3(x)$

```
>> derivada = diff(cos(x)-(1/3)*(cos(x)^3))

derivada =

-sin(x)+cos(x)^2*sin(x)
```

$\longrightarrow f'(x) = sen(x) + \cos^2(x).sen(x)$

$f(x) = \sec(x) + e^{6x}$

```
>> syms x e
>> derivada = diff(sec(x)+e^(6*x))

derivada =

sec(x)*tan(x)+6*e^(6*x)*log(e)
```

$\longrightarrow f'(x) = \sec(x).tg(x) + 6e^{6x}\ln(e)$

Calcule a derivada de segunda ordem da função $f(x)=x^3$.

```
>> syms x
>> derivada = diff(x^3,x,2)

derivada =

6*x
```

$\longrightarrow f''(x) = 6x$

Calcule a derivada de primeira ordem da função $f(x,y)=x^2+y^2$ (

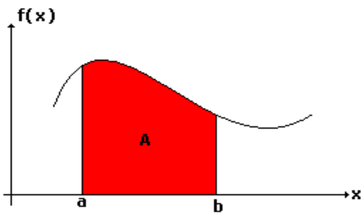
```
>> syms x y
>> derivada_x = diff(x^2+y^2,x,1)

derivada_x =

2*x
```

$\longrightarrow f'(x) = 2x$

Calcule a derivada de primeira ordem da função $f(x,y)=x^2+y^2$ (

```
>> syms x y
>> derivada_y = diff(x^2+y^2,y,1)

derivada_y =

2*y
```

$\longrightarrow f'(y) = 2y$



Integral

Integral Vamos agora procurar um processo para calcular a integral de f em [a,b] sem termos que recorrer à definição. Para tanto, consideremos f contínua e não negativa em [a,b].

O número $\int_a^b f(x)dx$ representa a área A sob o gráfico de f(x) no intervalo [a,b]. Isso é:

No **MatLab** são dadas **três funções** para calcular a área sob a curva num intervalo finito. A saber:

1º. A integração numérica usando o processo chamado de quadratura, no MATLAB, pode ser feita através da seguinte sintaxe:

quad(fun,a,b) onde fun é a função contínua
a é o limite inferior do intervalo de integração
b é o limite superior do intervalo de integração.

Integração numérica usando a regra de Simpson recursiva

Exemplo: Calcule a área sob o gráfico de f(x)=x² - 5x + 9, x∈[1,4]

Matematicamente temos:

$$A = \int_1^4 x^2 - 5x + 9 = \Big|_1^4 \frac{x^3}{3} - 5\frac{x^2}{2} + 9x =$$

$$= \left(\frac{4^3}{3} - 5\frac{4^2}{2} + 9.4\right) - \left(\frac{1^3}{3} - 5\frac{1^2}{2} + 9.1\right) = \frac{52}{3} - \frac{41}{6} = \frac{21}{2} = 10.5$$

No Matlab, utilizando a função quad, temos:

```
>> F = @(x)x.^2-5*x+9;
>> Q = quad(F,1,4);
>> integral = Q

integral =

   10.5000
```

Para definir a função cujos valores de cálculo serão definidos a posteriores utiliza-se a seguinte forma sintática:

@(x) função;

Aonde são tratados os possíveis valores de x, sendo obrigatório o seu uso nessa função.

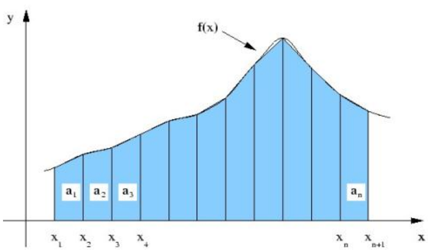
ou

```
>> integral = quad(@(x) x.^2-5*x+9,1,4)

integral =

   10.5000
```

2º. Aproxima a integral sobre a função, pelo somatório das áreas dos trapézios, no MATLAB, pode ser feita através da seguinte sintaxe:

z = trapz(x,y)

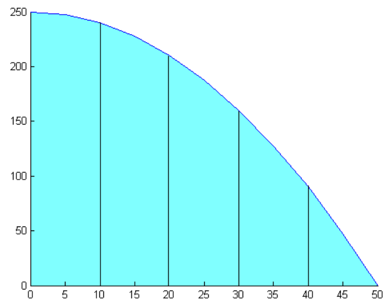
Onde: x = intervalo fechado da integração
y = função de integração.



$$A = \int_a^b f(x)dx = \sum_{i=1}^{n} f(\overline{x_i})\Delta_i x$$

Exemplo: Faça uma estimativa da área A sob o gráfico de $f(x) = 250 - \frac{x^2}{10}$

com 0 ≤ x ≤ 50, dividindo o intervalo [0,50] em sub-intervalos de comprimento 10.



Matematicamente seria:

$$A = \int_0^{50} 250 - \frac{x^2}{10}$$

No Matlab tem-se:

```
>> x=0:10:50;
>> y=250-x.^2/10;
>> z=trapz(x,y)

z =

        8250
```

3º. Outra notação para o cálculo da integral de uma função no MATLAB pode ser feita através das seguintes sintaxes:

int(**f**) é a integral indefinida da função **f** em relação à variável **x**
int(**f**,**a**,**b**) é a integral definida da função **f** em relação à variável **x** de **a** até **b**

Exemplos:

$\int x.\cos(x)dx =$ ⇒

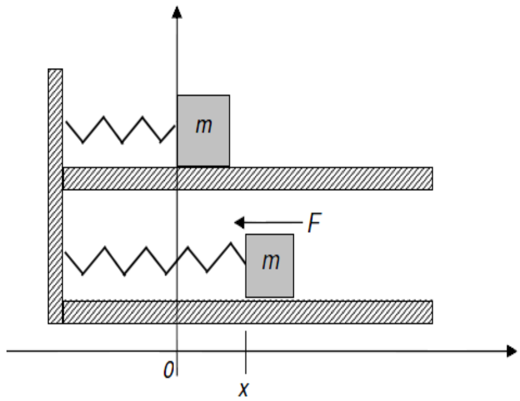
```
>> integral = int('x*cos(x)')

integral =

cos(x)+x*sin(x)
```

$\int_0^1 \frac{x^3}{\sqrt{1-x}} dx = ?$ ⇒

```
>> integral = int('x^3/sqrt(1-x)',0,1)

integral =

32/35
```

Exercício

Faça um programa que estude o sistema massa-mola. Para tanto devemos receber via teclado as seguintes informações: a constante elástica da mola em (N/m), o deslocamento máximo da mola em (cm) e a massa do bloco ligado à mola em (g). Calcula-se então a Energia total, a força na mola, a energia potencial, a energia cinética e a velocidade da massa. Fazendo então em subgráficos as seguintes representações: deslocamento versus energia cinética, deslocamento versus energia potencial, deslocamento versus energia total e deslocamento versus velocidade da massa. Para tanto temos as seguintes sentenças:



$$Fisicamente \quad temos:$$

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

$$velocidade = v = \frac{dx}{dt} = \int adt$$

$$deslocamento = x = \int vdt$$

$$Sistema \quad massa-mola:$$

$$Força = f = -kx = ma \therefore kx = m\frac{d^2x}{dt^2} \therefore a = \frac{d^2x}{dt^2} = -\frac{k}{m}x$$

$$Energia \quad potencial = E_p = \frac{1}{2}kx^2$$

$$Energia \quad cinética = E_c = \frac{1}{2}mv^2$$