

Apostila de Algoritmos

Curso de MATLAB

Prof.: Samuel Dias

1. Introdução ao MATLAB

Caros alunos, primeiramente você o que significa MATLAB? Não? Bom, MATLAB significar Laboratório de Matrizes.

É um programa utilizado para efetuar cálculos científicos e de engenharia.

Vantagens do Matlab:

- Facilidade de uso (padrão Windows, janelas de comando e edição independentes);
- Independência de Plataforma (Windows e Mac, infelizmente ainda não funciona em Linux);
- Funções Predefinidas (Média aritmética, desvio padrão, mediana, *toolbox*, fontes na *internet*);
- Interface gráfica de usuário (chamado de *gui* – *graphical user interface*).

Desvantagens do Matlab:

- É uma linguagem interpretada, por isso pode ser mais lento que linguagens compiladas;
- É um software pago e muito caro.

O Ambiente Matlab:

A unidade fundamental do matlab é a matriz.

Então caro aluno, não preciso nem lembrar o que é uma matriz.

Definição: é uma coleção de valores de dados organizados em linhas e colunas.

A vantagem de se trabalhar com matriz é que se pode acessar um determinado elemento de forma individual.

Veja o exemplo a seguir:

1. Digite a matriz $A = [2 \ 4 \ 6 \ 8 \ 10]$ na janela de comando do matlab. Encontre o valor armazenado em cada posição da matriz.

P. exemplo, $A(1,3) = 6$ ou $A(5) = 10$.

Agora, crie uma matriz contendo duas linhas e a mesma quantidade de colunas e faça a mesma operação que a realizada anteriormente.

A área de trabalho Matlab:

- Janela de comando: permite ao usuário inserir comandos pelo marcador de comandos (>>) que serão executados de imediato.

Ex.: Determine a área de um círculo de 2.5m de raio utilizando a janela de comando.

>> área = pi * 2.5^2

- Janela de Histórico: exibe uma lista de comandos que o usuário inseriu na janela de comandos. Para reexecutar qualquer comando, simplesmente clique duas vezes sobre ele com o botão esquerdo do mouse. E caso você queira excluir um comando presente na janela de histórico, clique sobre o mesmo com o lado direito do mouse e em seguida clique em “delete selection”.

- Janela de Edição: é usado para criar novos arquivos M ou modificar os existentes. É nesta janela que o usuário pode digitar todo um algoritmo construído.

Caro aluno é válido mencionar que um comentário é inserido no matlab através da utilização do símbolo “%” seguido do comentário.

Além disso, pode-se observar que um texto comentado fica escrito na cor verde, variáveis e números na cor preta e palavras-chave da linguagem em azul.

Finalmente, um arquivo M pode ser executado ao digitarmos seu nome na janela de comandos.

- Janela de Figuras: é usada para representar gráficos Matlab.

Professor, você poderia definir o que é um gráfico? É claro que sim! Veja.

Gráficos Matlab: é uma representação gráfica bidimensional ou tridimensional de dados, uma imagem ou uma GUI.

Veja exemplo de algoritmo:

```
% Este arquivo calcula o seno de x para 0 <= x <= 6
%
x = 0:0.1:6;
y = sin(x);
plot(x,y);
```

A workspace (espaço de trabalho):

Quando digitamos a expressão “z = 10” na janela de comandos, é criado automaticamente uma variável chamada de “z” e é armazenado o valor “10” na mesma.

Pode-se definir o *espaço de trabalho* como sendo a coleção de todas as variáveis e matrizes que podem ser usadas pelo usuário quando um comando, arquivo M ou função em particular está em execução.

Mas professor, a medida que estudamos a quantidade de variáveis vai aumentando. Como faço para apagar uma determinada variável ou todas ao mesmo tempo? Veja como é simples.

Para apagar uma ou mais variáveis, digite o código

```
>> clear var1 var2 ...
```

E caso você queira apagar todas as variáveis ao mesmo tempo, digite o código

```
>> clear all
```

Ixe professor, e se eu não lembrar desses comandos? O que faço? Calme, calme, rs! Selecione quem você quer apagar e em seguida digite a tecla “delete”.

Viu como foi fácil? Legal!

Alguns Comandos Importantes:

- clc => limpa a janela de comandos;
- clf => limpa a janela de figuras;

Caro aluno não se pode esquecer o comando “control + c” que possibilita a interrupção do programa em execução e devolve o marcador de comandos.

Veja exemplo:

```
a = 0;  
i = 0;  
while a < 1  
    i = i + 1  
end
```

Neste exemplo, observe que a condição $a < 1$ será sempre satisfeita, portanto a linha de comando “ $i = i + 1$ ” ficará sendo executada infinitamente. Observe que não foi colocado o sinal de “ ; ” nesta linha de comando, o que possibilita que os valores de “ i ” sejam ecoados na janela de comando.

Símbolos Matemáticos:

- + => adição;
- => subtração;
- * => multiplicação;
- / => divisão;
- ^ => potência.

Um exercício simples:

Utilize o matlab para calcular o volume de um cilindro de raio r e altura h. Suponha que o raio vale 0.1 m e a altura é de 0.5 m.

REVISÃO:

- Qual a função da janela de comandos? Edição? Figuras?
- O que é o espaço de trabalho?

- Como você pode limpar um conteúdo de um espaço de trabalho?
- A distância percorrida por uma bola em queda livre é dada pela equação:

$$x = x_0 + v_0 t + \frac{1}{2} a t^2$$

Utilize o matlab para calcular a posição da bola no tempo $t = 5s$, sendo $x_0 = 10m$, $v_0 = 15m/s$ e $a = -9.81m/s^2$.

- Encontre o valor da seguinte expressão:

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

2. Matlab Básico

Variáveis e Matrizes:

Como já foi dito anteriormente, a unidade básica no Matlab é a matriz.

E também já foi dito que uma matriz é uma coleção de valores organizados em linhas e colunas.

Também foi dito que podemos acessar um valor individual de uma matriz, seguido de índices entre parênteses que identificam a linha e coluna em particular.

Além disso, é válido lembrar que um escalar é interpretado no matlab como uma matriz que apresenta uma única linha e uma coluna.

Mas professor existe alguma diferença entre vetor e matriz? É claro que sim!

Veja, um vetor seria um tipo de matriz com apenas uma dimensão e a matriz propriamente dita apresenta duas ou mais dimensões.

Assim ficou moleza!

Para melhorar o entendimento, identifique as posições da seguinte matriz.

Além disso, veja que as dimensões da matriz podem ser encontradas através do comando “size(a)”. Que mostra os valores 4 e 5, como sendo a quantidade de linhas e colunas, respectivamente.

Uma dica!

Ao escrever um algoritmo é importante escolher nomes com significado para as variáveis. P. exemplo é interessante que se use a o nome “media” para receber o valor da média aritmética ao invés do nome ser “casa”.

Uma segunda dica!

O nome de uma variável não pode ser iniciada por um número, o nome da variável deve apresentar até 31 caracteres e caso se queira utilizar uma palavra composta, deve-se utilizar o *underline* “_”. P. exemplo, juro_composto.

Uma terceira dica!

O matlab diferencia letras maiúsculas das minúsculas, portando o nome da variável “endereço” é diferente de “Endereço”.

Uma quarta dica!

No matlab, variáveis podem ser criadas a qualquer momento simplesmente associando valores a elas o que não acontece em outras linguagens, como por exemplo o C.

Iniciando variáveis no matlab:

Existem três formas de se carregar valores em uma variável:

- Atribuindo o valor diretamente em uma declaração apropriada (z = 10);
- Carregar dados digitados diretamente pelo teclado;
- Ler dados de um arquivo.

Expressões de Atribuição:

Formato: $\text{var} = \text{expressão}$

Onde: var – nome da variável

Expressão – é uma constante, uma matriz, uma combinação de constantes, outras variáveis e operações matemáticas.

Ex.:

```
var = 40 * i;  
var2 = var / 5;  
array = [1 2 3 4];  
x = 1; y = 2;
```

*Caro aluno, no primeiro exemplo ($\text{var} = 40 * i$) o valor que será armazenado na variável “var” será $40i$ que é um número imaginário.*

*Veja também que podemos criar uma **matriz vazia**, por exemplo, $a = []$;*

Observe o seguinte exemplo:

```
a = [0 1+7] => a = [0 8];  
b = [a(2) 7 a] => b = [8 7 0 8];
```

Uma dica importante, se um elemento específico da matriz estiver definido e um ou mais elementos antes dele não, os elementos anteriores serão automaticamente criados e iniciados com o valor zero.

Exemplo: $c(2,3) = 5$;
 $d = [1 \ 2] \Rightarrow d(4) = 5$;
 $\Rightarrow d = [1 \ 2 \ 0 \ 5]$;

*Caro aluno vamos criar um momento de reflexão. É simples se criar pequenas matrizes onde o usuário pode definir termo a termo. Mas como você irá fazer se a matriz apresentar centenas ou milhares de elementos? E aí? E aí? Agora complicou professor ☹. Calma, calma. O matlab lhe ajuda através do **operador dois-pontos**.*

Formatação: prim:inc:ult;

Onde: - prim: primeiro termo;

- inc: valor do incremento;

- ult: último termo;

Exemplo: $x = 1:2:10 \Rightarrow x = 1 \ 3 \ 5 \ 7 \ 9$

Atenção: Se o valor do incremento for “um”, este valor pode ser omitido.

Outros exemplos: $\text{angulos} = (0.01:0.01:1)*\text{PI}$

Uma pergunta: Vocês podem me dizer a diferença entre $f = [1:4]$ e $g = [1:4]'$????

Funções Pré-definidas:

Função “zeros” – cria uma matriz de zeros do tamanho desejado.

Exemplos:

a = zeros(2);

b = zeros(2,3);

c = [1 2;3 4];

d = zeros (size(c));

Função “ones” – cria uma matriz de “uns” do tamanho desejado, possui as mesmas características da função “zeros”.

Função “eye” – cria uma matriz identidade do tamanho especificado pelo usuário, possuindo as mesmas características da função “zeros”.

Função “length” – retorna o comprimento de um vetor ou a dimensão maior de uma matriz bidimensional.

Função “size” – retorna dois valores, especificando o número de linhas e o de colunas.

Iniciando variáveis com entrada pelo teclado:

Como pode ser observado, pode-se efetuar a entrada de dados pelo teclado do computador e salvá-lo em uma variável. Para isso utiliza-se a função “input”.

Exemplo: recebe = input('Entre com um valor: ');

REVISÃO:

- Qual a diferença entre uma matriz e um vetor?
- Responda as questões seguintes considerando a matriz abaixo.

$$A = \begin{bmatrix} 1.1 & -3.2 & 3.4 & 0.6 \\ 0.6 & 1.1 & -0.6 & 3.1 \\ 1.3 & 0.6 & 5.5 & 0.0 \end{bmatrix}$$

- i) Qual o tamanho de A?
- ii) Qual o valor de A(2,3)?
- iii) Defina quais índices de todos os elementos apresentam valor igual a 0.6.

DESAFIO: Crie um algoritmo que mostre os índices de todos os elementos que apresentem valor igual a 0.6.

Submatrizes:

É possível selecionar e utilizar subconjuntos de matrizes no Matlab como se fosse uma matriz separada.

Ixe professor, então como se faz isso? Muito fácil, basta colocar uma lista de todos os elementos a serem selecionados entre parênteses após o nome da matriz. Veja o exemplo e note como é simples!

Exemplo 01:

Seja a matriz = [1.1 -2.2 3.3 -4.4 5.5], veja:

- impar = matriz ([1 4]); => impar = [1.1 -4.4];
- aleat = matriz (1:2:5); => aleat = [1.1 3.3 5.5];

Exemplo 02:

Seja a mat = [1 2 3; -2 -3 -4; 3 4 5], veja:

- col = mat (1 , :) => col = [1 2 3];

- col2 = mat (: , 1 : 2 : 3) => col2 = $\begin{bmatrix} 1 & 3 \\ -2 & -4 \\ 3 & 5 \end{bmatrix}$

Função “end”:

Quando utilizada no índice de uma matriz, a função *end* retorna o valor mais alto admitido por ele. Veja exemplos:

a) >> ar3 = [1 2 3 4 5 6 7 8];

>> ar3 (5: end) => ar3 = [5 6 7 8]

>> ar3 (end) => ar3 = 8

b) >> ar4 = [1 2 3 4; 5 6 7 8; 9 10 11 12];

>> ar4 (2 : end , 2 : end) => ar4 = $\begin{bmatrix} 6 & 7 & 8 \\ 10 & 11 & 12 \end{bmatrix}$

Caro alunos, podemos inserir valores em uma determinada posição de uma matriz já criada. Veja exemplo:

>> sub = [1 2 3 4; 5 6 7 8; 9 10 11 12];

>> sub (1:2, [1 4]) = [20 21; 22 23];

Ixe professor, achei muito interessante este procedimento!

Veja um outro exemplo:

>> sub2 = [1 2 3 4; 5 6 7 8; 9 10 11 12];

>> sub2 (1:2, 1:2) = 1

Agora vamos estudar algumas variáveis especiais pré-definidas no matlab:

Calcule o comprimento “cir” de uma circunferência de raio $r = 10$ cm.

```
>> cir = 2 * pi * 10
```

```
>> pi = 3;
```

```
>> cir = 2 * pi * 10
```

Podemos concluir que a variável pi possui um valor pré-estabelecido de 3.14 e este valor pode ser substituído pelo usuário ao seu critério.

Algumas funções especiais pré-definidas:

pi	Armazena π com 15 dígitos significativos
i, j	Contém o valor $i (\sqrt{-1})$
clock	Essa variável especial contém a data e a hora correntes na forma de um vetor linha de 6 elementos, contendo ano, mês, dia, hora, minuto e segundo
date	Contém a data corrente em formato de cadeia de caracteres, como “19-Feb-2010”

REVISÃO:

1. Dada a matriz c definida abaixo, determine o conteúdo das seguintes submatrizes:

$c = [1.1 \ -3.2 \ 3.4 \ 0.6; \ 0.6 \ 1.1 \ -0.6 \ 3.1; \ 1.3 \ 0.6 \ 5.5 \ 0.0]$

- a. $c(2, :)$
- b. $c(:, \text{end})$
- c. $c(1:2, 2:\text{end})$
- d. $c(6)$
- e. $c(4:\text{end})$
- f. $c(1:2, 2:4)$
- g. $c([1 \ 3], 2)$
- h. $c([2 \ 2], [3 \ 3])$

2. Determine o conteúdo da matriz “a” após a execução das seguintes declarações:

- a. `a = [1 2 3; 4 5 6; 7 8 9];`
`a ([3 1] , :) = a ([1 3] , :);`
- b. `a = [1 2 3; 4 5 6; 7 8 9];`
`a ([2 2] , :);`
- c. `a = [1 2 3; 4 5 6; 7 8 9];`
`a ([1 3] , :) = a ([2 2] , :);`

3. Determine o conteúdo da matriz “a” após a execução das seguintes declarações:

- a. `a = eye (3 , 3);`
`b = [1 2 3];`
`a (2 , :) = b;`
- b. `a = eye (3 , 3);`
`b = [4 5 6];`
`a (: , 3) = b';`
- c. `a = eye (3 , 3);`
`b = [7 8 9];`
`a (3 , :) = b ([3 1 2]);`

Exibindo dados de saída:

Caros alunos conforme já vimos se quisermos ecoar o valor de uma determinada variável ou expressão matemática na janela de comando, basta que não se coloque o “;” no final de uma sentença.

Mas professor existem outras formas de exibirmos dados para o usuário? É claro que sim. Vamos lá!

Saída estruturada usando a função “fprintf”:

Um exemplo de Estrutura:

```
fprintf ( ‘ O valor de pi = %f \n’ , pi)
```

O símbolo “%f” serve para inserir naquele ponto do texto o valor armazenado por uma variável, no exemplo o valor de π . Além disso, foi utilizado o símbolo “\n” que serve para saltar uma linha após a inserção do texto acima. E, finalmente, podemos efetuar a formatação do valor inserido na variável %f através da utilização, por exemplo, “%4.2f” que cria um campo com quatro caracteres de largura, incluindo dois dígitos depois do ponto decimal.

ATENÇÃO! ATENÇÃO! O comando “fprintf” somente exibe a parte real de um número.

Eita, eita. Não entendi foi nada!

Exemplo:

```
>> x = 2 * ( 1 - 2 * i ) ^ 3;  
>> fprintf ( 'fprintf:   x = %6.4f \n ' , x );
```

AH! Agora eu entendi! rs!

Arquivo de Dados:

No matlab existem essencialmente duas formas de salvar arquivos utilizando o comando “save”. A primeira forma consiste em gerar um arquivo .mat que tem a vantagem de guardar o nome e o tipo de cada variável. No entanto, caso queiramos compartilhar este arquivo entre dois computadores, necessariamente os dois devem ter instalados o matlab. A segunda forma consiste em gerar um arquivo .dat que tem a vantagem de poder ser aberto em qualquer editor de texto, no entanto possui a desvantagem de perder o nome e os tipos de variáveis.

Ih professor, agora ferrou! Lá em casa eu não to com o matlab instalado e eu tava querendo ter acesso aos valores coletados nas variáveis. Poxa, foi trabalho perdido tudo que fiz? ☹ Calma, calma. Você está estudando muito e não precisa ficar nervoso assim. O matlab te fornece a possibilidade de salvar arquivos de dados no formato ascii que pode ser aberto em qualquer computador com ou sem matlab. Sério? Você tá de brincadeira né professor? Olhe pro meu rosto cê acha que estou brincando? Hehehehe. Pois diz aí fêssor como nós pode salvar assim. Vamos lá! :P

Primeiramente o formato para gerar arquivos .mat:

Formato:

Save filename var1 var2 var3 ...

Onde – filename: é o nome do arquivo no qual as variáveis serão gravadas;

var1, var2, ...: são as variáveis que o usuário pretende salvar;

Salvando arquivos em formato ascii:

Exemplo:

```
>> x = [1.23 3.14 6.28; -5.1 7.00 0];
```

```
>> save x.dat x -ascii
```

Agora ta moleza! Salvar em formato ascii!

Professor agora fiquei com uma dúvida. Vamos supor que o meu algoritmo apresente umas cinqüenta variáveis e eu queira salvar todas, então eu preciso digitar o nome de todas uma a uma? A não ser que você queira fazer serviço de português você pode digitar o seguinte comando “save filename”. Oxe, só isso? É sim, ai o matlab salva todas as variáveis que estão neste momento na janela de trabalho. Legal, legal!

Bom professor, já aprendi a salvar arquivos no matlab agora como faço para abrir ou carregar um arquivo existente? Muito fácil! Veja:

Formato:

load filename

onde – filename é nome do arquivo a ser carregado.

Se o arquivo for do tipo .mat todas as variáveis no arquivo serão recuperadas, com os nomes e tipos idênticos aos anteriores.

Professor uma pergunta, pra abrir um arquivo em formato ascii tem algum segredo? Claro que não. Igual, igual e igual!

Operações com Escalares e Matrizes:

Operador de atribuição:

Exemplo:

```
>> ii = 5;  
>> ii = ii + 1;
```

Operações com Escalares:

Exemplo:

```
>> 2 ^ ( ( 8 + 2 ) / 5 ) = 2 ^ ( 10 / 5 )  
                        = 2 ^ 2  
                        = 4
```

Operações com Matrizes:

Podem ser realizadas de duas formas chamadas de *operação estrutural e operação matricial*.

- Operação estrutural: são operações entre matrizes executadas elemento a elemento.

Exemplo:

```
A = [1 3; 2 4];  
B = [-1 3; -2 1];  
Logo, A + B = [0 5; 1 5];
```

Neste tipo de operação entre matrizes devemos ter sempre matrizes com a mesma quantidade de linhas e colunas, caso contrário o matlab gera uma mensagem de erro.

Uma operação estrutural pode ser realizada também entre uma matriz e um escalar, veja exemplo:

```
A = [1 2; 3 4];  
Logo, A + 4 = [5 6; 7 8];
```


Tabela com as Operações Estruturais e Matriciais Comuns:

Soma estrutural = soma matricial	$a + b$
Subtração estrutural = subtração matricial	$a - b$
Multiplicação estrutural	$a .* b$
Multiplicação matricial	$a * b$
Divisão à direita estrutural	$a ./ b$
Divisão à esquerda estrutural	$a .\ b$
Divisão matricial* à direita	a / b
Divisão matricial* à esquerda	$a \backslash b$
Expoente estrutural	$a .^ b$

* - A divisão matricial é definida por $a * \text{inv}(b)$, onde $\text{inv}(b)$ é a inversa da matriz b .

- Operação Matricial: nas operações matriciais, como o próprio nome diz são relações baseadas em matrizes e as regras foram definidas quando fizemos o 1º ano do ensino médio.

É o novo!!!! Hehehehehe. 😊

Exercícios: Observe os resultados das seguintes expressões utilizando o software matlab. Seja $a = [1 \ 0; 2 \ 1]$, $b = [-1 \ 2; 0 \ 1]$, $c = [3; 2]$ e $d = 5$.

$a + b$	$a + c$
$a .* b$	$a + d$
$a * b$	$a .* d$
$a * c$	$a * d$

Hierarquia de Operações:

Freqüentemente muitas operações matemáticas são combinadas em uma única expressão. Veja exemplo:

$$\text{distancia} = 0.5 * \text{aceleração} * (\text{tempo} ^ 2)$$

$$\text{distancia} = (0.5 * \text{aceleração} * \text{tempo}) ^ 2$$

Nestes dois casos, tem-se resultados distintos apesar de termos a mesma expressão. Logo, são válidas as regras que vimos no colegial para calcular expressões matemáticas e a má interpretação pode levar a resultados desastrosos.

Exercício: As variáveis a, b, c e d foram iniciadas com os seguintes valores $a = 3$, $b = 2$, $c = 5$ e $d = 3$. Encontre os resultados:

- i. $saida = a * b + c * d$
- j. $saida = a * (b + c) * d$
- k. $saida = (a * b) + (c * d)$
- l. $saida = a ^ b ^ d$
- m. $saida = a ^ (b ^ d)$

Caro aluno, qual conclusão pôde chegar ao fazer este exercício? Portanto uma ótima revisão da 5ª série irá cair muito bem para quem tiver dúvida. Ótimo exercício de casa!

Revisão:

1. Assuma que a, b, c e d são definidas conforme a seguir, e calcule os resultados das seguintes operações se elas forem legais. Se uma operação for ilegal, explique o motivo.

$a = [2 \ 1; -1 \ 2]$, $b = [0 \ -1; 3 \ 1]$, $c = [1; 2]$ e $d = -3$.

- a) $result = a .* c$
- b) $result = a * [c \ c]$
- c) $result = a .* [c \ c]$
- d) $result = a + b * d$
- e) $result = a + b .* d$

2. Resolva para x a equação $Ax = B$, onde $A = [1 \ 2 \ 1; 2 \ 3 \ 2; -1 \ 0 \ 1]$ e $B = [1; 1; 0]$.

Outras funções pré-definidas no Matlab:

- Função “max”: como o próprio nome da função afirma, irá retornar o maior valor dentro de um vetor. Exemplo: `maxval = max ([1 -5 6 -3])`. No entanto, pode ser útil saber em qual posição do vetor está localizado este valor máximo para isso basta utilizar a seguinte formatação para o exemplo em questão: `[maxval pos] = max ([1 -5 6 -3])`.

Interessante professor esta informação!

Caro aluno, antes de continuarmos nosso estudo, vejamos um exemplo de atribuição de um vetor como entrada, produzindo um vetor de saída de acordo com uma função pré-definida. Veja:

```
>> x = [0 pi / 2 pi 3 * pi / 2 2 * pi]
>> y = sin ( x )
```

Produz um vetor de saída `y = [0 1 0 -1 0]`.

- Função “sqrt”: é uma função utilizada para calcular a raiz quadrada de um número qualquer. A vantagem desta função, no ambiente matlab, é que ela pode calcular a raiz quadrada de um número real ou imaginário. Fato este que em outras linguagens de programa, tipo C, ocasionaria erro.

*Incrível professor o poder desta ferramenta matemática que estamos aprendendo!
Valeu!*

Exemplo:

```
>> sqrt ( -2 )
ans = 0 + 1.4142i
```

Funções Matemáticas	Descrição
<code>abs (x)</code>	Calcula o módulo de x
<code>sin (x)</code>	Calcula o seno de x, em radianos
<code>cos (x)</code>	Calcula o cosseno de x, em radianos
<code>tan (x)</code>	Calcula a tangente de x, em radianos
<code>asin (x)</code>	Calcula o arco-seno de x
<code>acos (x)</code>	Calcula o arco-cosseno de x
<code>atan (x)</code>	Calcula o arco-tangente de x
<code>exp (x)</code>	Calcula e^x
<code>log (x)</code>	Calcula o logaritmo natural \log_e^x
<code>[valor , pos] = max (x)</code>	Retorna o maior valor do vetor x e a sua posição
<code>[valor , pos] = min (x)</code>	Retorna o menor valor do vetor x e a sua posição
<code>mod (x , y)</code>	Retorna o resto da divisão de x por y
<code>sqrt (x)</code>	Calcula a raiz quadrada de x

Eita professor, fiquei foi tonto com tanta função matemática. Calma, calma. Pode ter certeza que elas sempre serão bastante úteis no seu dia a dia. Legal?

Funções de Arredondamento	Descrição
ceil (x)	Arredonda x para o inteiro mais próximo em direção a mais infinito. Exemplo: >> ceil (3.1) = 4 >> ceil (-3.1) = -3
fix (x)	Arredonda x para o inteiro mais próximo em direção ao zero. Exemplo: >> fix (3.1) = 3 >> fix (-3.1) = -3
floor (x)	Arredonda x para o inteiro mais próximo em direção a menos infinito. Exemplo: >> floor (3.1) = 3 >> floor (-3.1) = -4
round (x)	Arredonda x para o inteiro mais próximo

Introdução a Diagramas:

Caro aluno, nos nossos primeiros dias de aula plotamos o gráfico da função seno e vimos que para tal, bastava-se criar dois vetores e utilizar a função “plot”. Com o intuito de prosseguirmos na construção de gráficos, veja exemplo.

Desenhe o gráfico da função $f(x) = x^2 - 10x + 15$ para valores de x inteiros variando de 0 a 10.

```
>> x = 0 : 1 : 10;
>> y = x.^2 - 10.*x + 15;
>> plot ( x , y );
```

Professor, eu achei bastante interessante termos revisado como faço para criar um gráfico. Mas to com uma idéia de deixar esse gráfico mais refinado, por exemplo, com título, legenda e linhas de grade. Tem como a gente fazer isso? Eu preciso responder que sim? Precisa professor. Então, ta certo. Da sim caro aluno pra fazer isso e muito mais. Vamos lá?

Aproveitando o mesmo desenho anterior, vamos inserir algumas linhas de comando e verificar a diferença encontrada.

```
>> x = 0 : 1 : 10;  
>> y = x.^2 - 10.*x + 15;  
>> plot ( x , y );  
>> title ('Gráfico de f (x)');  
>> xlabel ('x');  
>> ylabel ('y');  
>> grid on;
```

Taí professor que eu gostei! Mas cê pode ensinar mais alguma coisa sobre recursos gráficos? Claro que sim. Vamos lá!

Diagramas Múltiplos:

Agora iremos aprender a desenhar mais de um gráfico ao mesmo tempo através de um exemplo.

Seja a função $f(x) = \sin 2x$ e sua derivada $f'(x) = 2 \cos 2x$ plote o gráfico destas duas funções ao mesmo tempo.

```
>> x = 0 : pi / 100 : 2 * pi;  
>> y1 = sin ( 2 * x );  
>> y2 = 2 * cos ( 2 * x );  
>> plot ( x , y1 , x , y2 );
```

Professor, no resultado do exemplo anterior pude notar que os gráficos foram gerados nas cores verde e azul, escolhidas automaticamente pelo matlab. Como faço para especificar a cor diretamente e mais, como faço para alterar o estilo da linha? Calma, vamos responder por parte. Tudo bem?

Cores de linhas, estilos de linhas, estilos de marcadores e legendas:

Aproveitando o exemplo anterior, vamos alterar os gráficos desenhados através da seguinte linha de comando.

```
>> plot ( x , y1 , 'r--' , x , y2 , 'bo');
```

Eita professor, o gráfico ficou com uma outra cara.

Segue uma tabela contendo a cor, marcador e estilo da linha! Veja.

Cor	Marcador	Estilo de Linha
y – amarelo	. – ponto	- – sólido
m – magenta	o – círculo	: – 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 – triângulo (apontando para baixo)	
	^ – triângulo (apontando para cima)	
	< – triângulo (apontando para a esquerda)	
	> – triângulo (apontando para a direita)	
	p – pentágono	
	h – hexágono	
	<none> sem marcador	

Após os gráficos terem sido desenhados, pela observação dos resultados, não sei precisar de cara qual gráfico pertence a qual função. No entanto, pode-se criar legendas que facilitaram a interpretação dos mesmos. Para isso, insira a seguinte linha de comando. Veja exemplo:

```
>> x = 0 : pi / 100 : 2 * pi;
>> y1 = sin ( 2 * x );
>> y2 = 2 * cos ( 2 * x );
>> plot ( x , y1 , 'k-' , x , y2 , 'b--');
>> title ( ' Gráfico de f(x) = sen (2x) e sua derivada');
>> xlabel ( 'x' );
>> ylabel ( 'y' );
>> legend ( 'f(x)' , 'd/dx f(x)' );
>> grid on;
```

Exercícios:

1. Construa um algoritmo que faça a conversão de temperatura em graus Celsius para Fahrenheit.

Expressão: $\frac{c}{5} = \frac{f - 32}{9}$

2. Dada uma fonte de tensão de 120 volts com resistência interna de 50 ohms, alimenta uma carga de resistência RL. Encontre o valor da resistência RL que resultará na potência máxima possível fornecida pela fonte para a carga. Quanta potência será fornecida nesse caso? Desenhe também a potência fornecida para a carga como função da resistência de carga RL.

Seja: $Pot = Ri^2$ e $i = \frac{V}{Rs + RL}$.

Expressões de ramificação e projeto de programa:

*Caro aluno, até então vimos praticamente à programação do tipo sequencial, ou seja, os comandos são lidos linha após linha do início ao fim do código. No entanto, existem dois outros tipos de estrutura de programação: **ramificações** e **laços**. Estas duas estruturas serão tratadas nas próximas aulas. Então vamos deixar de blá blá e começar a estudar!*

Imaginem que você ao trabalhar precise escrever um programa para resolver um problema. Como você começa?

Para problemas pequenos o fato de nos tacarmos sobre o problema pode até dar certo, no entanto para problemas complicados do mundo real provavelmente o programador vai papocar! Lembre-se, calma e paciência é um primeiro princípio para desenvolver um ótimo algoritmo.

Operador Relacional:

São operadores que produzem um resultado verdadeiro (1) ou falso (0).

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

Exemplo:

- a) $3 < 4 \Rightarrow 1$
- b) $3 \leq 4 \Rightarrow 1$
- c) $3 == 4 \Rightarrow 0$
- d) $3 > 4 \Rightarrow 0$
- e) $4 \leq 4 \Rightarrow 1$
- f) $'A' < 'B' \Rightarrow 1$

Pode ser utilizado também em operações com matrizes, veja a seguir:

a) $A = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$ e $B = 0$.

$A > B \Rightarrow ?$

b) $A = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$ e $B = \begin{bmatrix} 1 & 0 \\ -2 & -1 \end{bmatrix}$

$A \geq B \Rightarrow ?$

Caro aluno, não confundir o operador de atribuição “=” com o operador de igualdade “==”. Ok? Cada um no seu quadrado, rs!

Outro Exemplo:

Variáveis				Relações		
A	B	nome	profissao	$A+1 \geq \text{sqrt}(B)$	$\text{Nome} \sim= \text{'Ana'}$	$\text{Profiss\~ao} == \text{'Medico'}$
3	16	'Miriam'	'Advogado'			
5	64	'Pedro'	'Medico'			
2.5	9	'Ana'	'Professor'			

Operador Igualdade (==) e diferença (~=):

O operador de igualdade retorna 1 quando os dois valores comparados são iguais, e 0 quando são diferentes. De maneira similar, o operador de não equivalência (~=) retorna 0 quando os dois valores são iguais, e 1 quando são diferentes.

ATENÇÃO: Ao trabalhar com comparadores, ter muito cuidado com números arredondados. Veja exemplo a seguir:

```
>> a = 0;
>> b = sin (pi);
>> a ==b
    ans =
         0
```

A princípio poderíamos achar que a comparação de igualdade iria retornar o valor 1, porém foi retornado o valor zero.

Operador Lógico:

São operadores que produzem um resultado lógico.

Operador	Operação
&	E lógico
	Ou lógico
xor	Ou exclusivo lógico
~	Não lógico

Tabela verdade para operadores lógicos:

Entradas		E	Ou	Xor	Não
I1	I2	I1 & I2	I1 I2	(I1, I2)	~I1
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Mais fácil do que isso só fazendo a disciplina de novo..kkkkkkkkkkkkkk!

Exemplo: Dado as variáveis valor1 = 1, valor2 = 0 e valor3 = -10, entenda os seguintes resultados apresentados a seguir.

- ~valor1 => 0
- valor1 | valor2 => 1
- valor1 & valor2 => 0
- valor1 & valor2 | valor3 => 1
- valor1 & (valor2 | valor3) => 1
- ~(valor1 & valor3) => 0

Revisão:

Assuma que a, b, c e d sejam conforme abaixo, e avalie as seguintes expressões:

a = 20; b = -2; c = 0; d = 1;

a > b	a == b
b > d	a & b > c
a > b & c > d	~~b

$$a = 2; b = \begin{bmatrix} 1 & -2 \\ -1 & 10 \end{bmatrix}; c = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}; d = \begin{bmatrix} -2 & 1 & 2 \\ 0 & 1 & 0 \end{bmatrix}$$

~(a > b)	c <= d
a > c & b > c	

$a = 2$; $b = 3$; $c = 10$; $d = 0$;

$a*b^2 > a*c$ $d \mid b > a$	$(d \mid b) > a$
---------------------------------	------------------

Construindo os primeiros algoritmos:

- 1) Fazer um programa que o usuário entre com dois números e imprimir a soma.
- 2) Fazer um programa que o usuário entre com dois números e imprimir a média aritmética.
- 3) Com base na questão 2, calcular a média ponderada com pesos 5 e 7.
- 4) Fazer um programa que lê os lados de um retângulo, calcula sua área e em seguida imprime o valor no monitor.
- 5) Fazer um programa que lê os valores de uma matriz quadrada de ordem 2, calcula o seu determinante e em seguida imprime seu valor no monitor.
- 6) Com base na questão 5, calcular o determinante para uma matriz quadrada de ordem qualquer.

- 7) Programa que lê três notas e calcula a média ponderada entre elas. Sendo que a primeira nota possui peso um, a segunda peso dois e a terceira peso três. Imprima a média no monitor.
- 8) Dada a sequência 1, 3, 5, 7, 9,... Fazer um programa que determine o vigésimo terceiro termo dessa sequência.
- 9) Baseado na sequência da questão anterior, elaborar um programa que determine qualquer termo dessa sequência.
- 10) Fazer um programa que calcula a diagonal principal de um paralelogramo e imprima seu valor no monitor.
- 11) Elaborar um programa que calcula o período de um pêndulo simples e imprima seu valor no monitor.
- 12) Programa que lê dois números inteiros e mostra o quociente e o resto da divisão entre eles.

Estrutura IF:

Forma:

if condição 01 comando 01; comando 02; Comando n; end	if condição 01 comando 01; comando 02; comando n; elseif condição 02 comando 01; comando 02; comando n; else comando 01; comando 02; comando n; end
---	---

Exercício:

- 01) Programa que calcula a média ponderada de três notas, sendo que a primeira possui peso um, a segunda peso dois e a terceira peso três. Ao ser determinada a média, mostrar situação do aluno.

Média > 7 ⇒ Situação: Aprovado
3 < Média < 7 ⇒ Situação: Prova Final
Média < 3 ⇒ Situação: Reprovado

- 02) Fazer um programa que leia três valores inteiros, determine e imprima o menor deles.
- 03) Programa que efetua a leitura de um número inteiro e identifica se este é par ou ímpar.
- 04) Dados três valores x, y, z, verificar se eles podem ser os comprimentos dos lados de um triângulo e, se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever uma mensagem.

Revisão Matemática:

- Condição de existência de um triângulo (cada lado deve ser menor do que a soma dos outros dois lados).
- Equilátero: os três lados iguais.
- Isósceles: dois lados iguais e um diferente.
- Escaleno: os três lados diferentes.

05) Foi adquirida uma estufa a ser utilizada em um laboratório de simulação climática e o engenheiro encontrou algumas incompatibilidades técnicas entre o aparelho com o nosso dia a dia.

a) A estufa possui como escala de temperatura fahrenheit. Criar um programa que faça a conversão para graus Celsius.

b) A mesma não possui um sistema de segurança para altas temperaturas, o que poderia ocasionar danos físicos a sua estrutura física. Elaborar um programa que mande mensagem de aviso para o engenheiro, caso a temperatura fique acima de 150 °C.

06) Um dos fatores que ocasionam problemas ambientais é o elevado crescimento populacional de pessoas em determinadas regiões do planeta, pois causam poluições das águas, ar e solo, bem como se aumenta o consumo dos recursos naturais. Criar um programa que calcula a densidade demográfica.

07) Criar um programa que transforma um valor em segundos para minutos.

08) Um fotosensor é um equipamento utilizado na detecção de velocidade de veículos automotores. Construir um algoritmo que envia ao motorista mensagens de acordo com sua velocidade de trafego. Para velocidades de até 110 km/h envia a mensagem “bom motorista”, acima dos 110 e até os 140 km/h envia a mensagem “você será multado com 3 pontos na carteira” e para velocidades superiores a 140 km/h a mensagem “você será multado com 7 pontos na carteira”.

09) Dada a expressão $ax^2 + bx + c = 0$, construir um algoritmo que informa para o usuário se a referida expressão possui raízes reais e diferentes, reais e iguais e imaginárias de acordo com os valores de a , b e c . Retorne também ao usuário a solução da equação.

10) Avaliar a função $f(x, y)$ para quaisquer valores de x e y especificados pelo usuário.

$$f(x, y) = \begin{cases} x + y, & x \geq 0 \text{ e } y \geq 0 \\ x + y^2, & x \geq 0 \text{ e } y < 0 \\ x^2 + y, & x < 0 \text{ e } y \geq 0 \\ x^2 + y^2, & x < 0 \text{ e } y < 0 \end{cases}$$

Estrutura de Repetição:

Laço for:

O laço **for** é o controlador de fluxo mais simples e usado na programação MATLAB. Analisando a expressão

```
>> for i=1:5
    X(i)=i^2;
end
```

pode-se notar que o laço **for** é dividido em três partes:

- A primeira parte (**i=1**) é realizada uma vez, antes do laço ser inicializado.
- A segunda parte é o teste ou condição que controla o laço, (**i<=5**).
- Esta condição é avaliada; se verdadeira, o corpo do laço (**X(i)=i^2**) é executado.

A terceira parte acontece quando a condição se torna falsa e o laço termina.

O comando **end** é usado como limite inferior do corpo do laço.

É comum construções em que conjuntos de laços **for** são usados principalmente com matrizes:

```
For i=1:8
    for j= 1:8,
        A(i,j)= i+j;
        B(i,j)= i-j;
    end
end
C=A + B;
```

Caro aluno, veja algumas variações do laço de repetição for:

- i. O incremento da função pode ser um valor diferente da unidade.

Exemplo:

```
for i = 1:2:10
    declaração 01;
    declaração 02;
    ...
    Declaração n;
end
```

ii. Quando a expressão de controle é um vetor.

Exemplo:

```
for ii = [5 9 7]
    declaração 01;
    declaração 02;
    ...
    declaração n;
end
```

Neste exemplo, o primeiro índice é o valor 5, o segundo valor 9 e o último valor 7.

iii. Quando a expressão de controle é uma matriz.

Exemplo:

```
for ii = [1 2 3; 4 5 6]
    declaração 01;
    declaração 02;
    ...
    declaração n;
end
```

Neste exemplo, o primeiro índice é o vetor [1; 4], o segundo o vetor [2; 5] e o último [3; 6].

Exercício 01: Utilizando o laço for, faça um algoritmo que calcula o fatorial de um número positivo n.

```
>> n = input('Digite Numero: ');
    fat = 1;
    for i = 1 : n
        fat = fat * i;
    end
```

Exercício 02: Construa um algoritmo que calcula os quadrados e as raízes cúbicas dos números de 1 até 100.

```
>> for i = 1:100
    quad = i^2
end
```

Exercício 03: 1) Crie o vetor x=[0 36 72 108 144 180 216 252 288 324] através do comando **for**.

```
>> x(1) = 0;
    for n = 2:10
        x(n) = x(n-1) + 36;
    end
x
```


Exercício 04: Plote 360 pontos de um período da função $y=\sin(2\pi x/360)$, usando o loop *for*.

```
>> for x=1:360
        y(x)=sin(2*pi*x/360);
    end
    plot(y)
```

Lação while:

No laço **while** apenas a condição é testada. Por exemplo na expressão

```
a = 1;

b = 15;
while a<b
    clc
    a = a+1
    b = b-1
    pause(1)
end
disp('fim do loop')
```

A condição **a<b** é testada. Se ela for verdadeira o corpo do laço, será executado.

Então a condição é retestada, e se verdadeira o corpo será executado novamente.

Quando o teste se tornar falso o laço terminará, e a execução continuará no comando que segue o laço após o **end**.

Exemplo 01: Construa o vetor $y = [64\ 32\ 16\ 8\ 4\ 2\ 1]$, usando o loop *while*.

```
>> num=128;
    n=0;
    while num>1
        num=num/2;
        n=n+1;
        y(n)=num;
    end
```

Um exemplo: Implemente um algoritmo que leia um conjunto de medidas e calcule a média aritmética e o desvio-padrão do conjunto de dados de entrada.

Revisão Matemática:

$$\text{Desvio-padrão: } s = \sqrt{\frac{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2}{N(N-1)}}$$

O matlab fornece comandos específicos para calcular a média aritmética através do comando “mean” e desvio-padrão através do comando “std”. *Ah professor, assim é fácil demais! Tem nem graça! Rs.*

Caro aluno, veja que nesse exemplo se gravarmos apenas um valor o programa irá retornar um valor “NaN” referente a divisão por zero presente na expressão que define o desvio-padrão, portanto o algoritmo deve ser alterado para enviar uma mensagem de erro caso aconteça esta situação. Agora é com vocês!

Exercícios:

01. Criar um algoritmo que leia os limites inferior e superior de um intervalo e imprimir todos os números pares no intervalo aberto e seu somatório.
02. Apresentar os quadrados dos números inteiros de 15 a 200.
03. Apresentar o total da soma obtida dos cem primeiros números inteiros.
04. Apresentar todos os números divisíveis por 4 que sejam menores que 200.
05. Elaborar um programa que efetue a leitura sucessiva de valores numéricos e apresente no final o total do somatório, a média e o total de valores lidos. O programa deve fazer as leituras dos valores enquanto o usuário estiver fornecendo valores positivos. Ou seja, o programa deve parar quando o usuário fornecer um valor negativo.
06. Elaborar um programa que efetue a leitura de valores positivos inteiros até que um valor negativo seja informado. Ao final devem ser apresentados o maior e menor valores informados pelo usuário.
07. Ler um número do teclado e imprimir todos os números de 1 até o número lido.
08. Solicitar um número entre 1 e 4. Se a pessoa digitar um número diferente, mostrar a mensagem "entrada inválida" e solicitar o número novamente. Se digitar correto mostrar o número digitado.