

Introdução ao Matlab

Operações Básicas

Operação	Símbolo	Exemplos
Adição, $a+b$	+	$5 + 6$
Subtração, $a-b$	-	$19 - 4.7$
Multiplicação, $a.b$	*	$5.02 * 7.1$
Divisão, $a \div b$	/ ou \	$45/5$ ou $5 \setminus 45$
Potência, a^b	^	3^4

Operações Lógicas

Operador Relacional	Descrição
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
==	igual a
!=	diferente de

Operador lógico	Descrição	Uso
&	E	Conjunção
	ou	Disjunção
~	Não	Negação

Constantes e Variáveis

>> 3*25 + 5*12

ans = 135

>> 1.23e-1

significa 0,123

Comando	Formato	Comentário
format short	33.5000	4 dígitos decimais (formato default)
format long	33.5000000000000000	16 dígitos
format short e	3.3500e+001	5 dígitos mais expoente
format long e	3.3500000000000000e+001	16 dígitos mais expoente
format hex	4040c00000000000	Hexadecimal
format bank	33.50	2 dígitos decimais
format +	+	positivo, negativo ou zero
format rat	67/2	Racional

Constantes e Variáveis

>> q1=3, p1=25, q2=5, p2=12

q1 = 3

p1 = 25

q2 = 5

p2 = 12

>> total=q1*p1+q2*p2

total = 135

Regras de construção das variáveis	Comentários/Exemplos
Variáveis com letras minúsculas e maiúsculas são diferentes, mesmo que consistam das mesmas letras.	Total, total, TOTAL e ToTaL são variáveis diferentes.
As variáveis podem consistir de até 19 caracteres	sdtf65erkjh3448bafg
As variáveis devem começar com uma letra e pode ser seguida de letras, números ou subscrito (_).	Var_2 X34 a_b_c

Variáveis Especiais

Variáveis especiais	Significado
ans	Variável usada para exibir os resultados
pi	Número 3,14159
eps	Menor número tal que, quando adicionado a 1, cria um número maior que 1 no computador.
flops	Armazena o número de operações em ponto flutuante realizadas.
inf	Significa infinito
NAN ou nan	Significa não é um número, por exemplo, 0/0.
i e j	Unidade imaginária [$\sqrt{-1}$].
nargin	Número de argumentos de entrada de uma função
nargout	Número de argumentos de saída de uma função
realmin	Menor número que o computador pode armazenar
realmax	Maior número que o computador pode armazenar

Comentário e pontuações

Símbolo	Função
,	Separar comandos dados em uma mesma linha.
;	Separar comandos dados em uma mesma linha. Se o último caractere da declaração é um ponto e vírgula , a impressão na tela é suprimida, mas a tarefa é realizada.
%	Todo e qualquer caracter depois do símbolo de porcentagem é tomado como comentário .
...	Pode-se continuar uma certa expressão na próxima linha usando um espaço em branco e três pontos , "...", ao final das linhas incompletas.

Variáveis literais

Uma variável pode conter uma cadeia de caracteres ao invés de um número

```
>> a = 'MATLAB'
```

```
a = MATLAB
```

Funções Matemáticas

- Matemática elementar;
- Funções especiais;
- Matrizes elementares e especiais;
- Decomposição e fatoração de matrizes;
- Análise de dados;
- Polinômios;
- Solução de equações diferenciais;
- Equações não-lineares e otimização;
- Integração numérica;
- Processamento de sinais.

Funções Matemáticas

- Por exemplo:

```
>> x = sqrt(2)/2
```

```
x = 0.7071
```

```
>> y = acos(x)
```

```
y = 0.7854
```

```
>> y_graus = y*180/pi
```

```
y_graus = 45.0000
```

<code>abs(x)</code>	valor absoluto de x .
<code>acos(x)</code>	arco cujo coseno é x
<code>asin(x)</code>	arco cujo seno é x .
<code>atan(x)</code>	arco cuja tangente é x .
<code>conj(x)</code>	conjugado complexo
<code>cos(x)</code>	coseno de x .
<code>cosh(x)</code>	coseno hiperbólico de x .
<code>exp(x)</code>	exponencial e^x .
<code>floor(x)</code>	arredondamento em direção ao $-\infty$

<code>gcd(x, y)</code>	máximo divisor comum de x e y .
<code>lcm(x, y)</code>	mínimo múltiplo comum de x e y .
<code>log(x)</code>	logaritmo de x na base e .
<code>log10(x)</code>	logaritmo de x na base 10.
<code>rem(x, y)</code>	resto da divisão de x por y .
<code>round(x)</code>	arredondamento para o inteiro mais próximo
<code>sign(x)</code>	função <i>signum</i>
<code>sin(x)</code>	seno de x .
<code>sinh(x)</code>	seno hiperbólico de x .
<code>sqrt(x)</code>	raiz quadrada de x .
<code>tan(x)</code>	tangente de x .
<code>tanh(x)</code>	tangente hiperbólica de x .

Números Complexos

>> $z1 = 3+4*i$

$z1 = 3.0000 + 4.0000i$

>> $z2 = 3+4*j$

$z2 = 3.0000 + 4.0000i$

>> $z1+z2$

$ans = 6.0000 + 8.0000i$

- No MATLAB, a conversão entre as formas polar e retangular de um número complexo utiliza as seguintes funções:
- real: parte real de um número complexo
- imag: parte imaginária de um número complexo
- abs: calcula o valor absoluto ou módulo de um número complexo
- angle: calcula o ângulo de um número complexo
- **Exemplo:**

```
>> x = 1-4i
x = 1.0000 - 4.0000i
>> a = real(x)
a = 1
>> b = imag(x)
b = -4
>> M = abs(x)
M = 4.1231
>> theta = angle(x)*180/pi
theta = -75.9638
```

VETORES

X=primeiro : último	Cria um vetor x começando com o valor primeiro , incrementando-se de 1(um) em 1(um) até atingir o valor último ou o valor mais próximo possível de último
X=primeiro:incremento:último	Cria um vetor x começando com o valor primeiro , incrementando-se do valor incremento até atingir o valor último ou o valor mais próximo possível de último
X=linspace(primeiro, último, n)	Cria um vetor x começando com o valor primeiro e terminado no valor último , contendo n elementos linearmente espaçados.
X=logspace(primeiro, último, n)	Cria um vetor x começando com o valor 10^{primeiro} e terminando no valor $10^{\text{último}}$, contendo n elementos logaritmicamente espaçados
X=[2 2*pi sqrt(2) 2-3j]	Cria um vetor x contendo os elementos especificados

Operações entre vetores

Seja $a = [a_1 \ a_2 \ \dots \ a_n]$, $b = [b_1 \ b_2 \ \dots \ b_n]$ e c um escalar

operação	expressão	resultado
adição escalar	$a+c$	$[a_1+c \ a_2+c \ \dots \ a_n+c]$
adição vetorial	$a+b$	$[a_1+b_1 \ a_2+b_2 \ \dots \ a_n+b_n]$
multiplicação escalar	$a*c$	$[a_1*c \ a_2*c \ \dots \ a_n*c]$
multiplicação vetorial	$a.*b$	$[a_1*b_1 \ a_2*b_2 \ \dots \ a_n*b_n]$
divisão	$a./b$	$[a_1/b_1 \ a_2/b_2 \ \dots \ a_n/b_n]$
potenciação	$a.^c$	$[a_1^c \ a_2^c \ \dots \ a_n^c]$
	$c.^a$	$[c^{a_1} \ c^{a_2} \ \dots \ c^{a_n}]$
	$a.^b$	$[a_1^{b_1} \ a_2^{b_2} \ \dots \ a_n^{b_n}]$

MATRIZES:

- Os elementos de cada linha da matriz são separados por espaços em branco ou vírgulas e as colunas separadas por ponto e vírgula, colocando-se colchetes em volta do grupo de elementos que formam a matriz. Por exemplo, entre com a expressão:

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

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

OPERAÇÕES COM MATRIZES

- Transposta;
- Adição;
- Subtração;
- Multiplicação;
- Divisão à direita;
- Divisão à esquerda;
- Exponenciação;

- **Funções:** o MATLAB possui algumas funções que se aplicam a matrizes como, por exemplo, as funções `size` (fornece o número de linhas e colunas de uma matriz) e `length` (fornece o maior valor entre o número de linhas e colunas).

função	descrição
<code>sum</code>	soma dos elementos
<code>prod</code>	produto dos elementos
<code>mean</code>	média aritmética
<code>std</code>	desvio padrão
<code>max</code>	maior elemento
<code>min</code>	menor elemento
<code>sort</code>	ordena em ordem crescente

Submatrizes

- Sendo B uma matriz 5x5 unitária, podemos defini-la através da seguinte função:

$$B = \begin{matrix} & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

>> B = ones (5,5)

- Sendo C uma matriz de zeros 3x4, podemos defini-la como:

$$C = \begin{matrix} & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 \end{matrix}$$

>> C=zeros(3,4)

Programação

- **Arquivos.m**
 - Para escrever um programa (ou arquivo .m) no MATLAB, escolha File na barra de menu. Dentro do menu File escolha New e selecione M-file. Abre-se, então, um editor de textos, onde pode-se escrever os comandos do MATLAB. Para editar um arquivo já existente, selecione a opção Open M-File, a partir do menu File. Os arquivos podem, também, ser editados fora do MATLAB utilizando qualquer editor de texto.

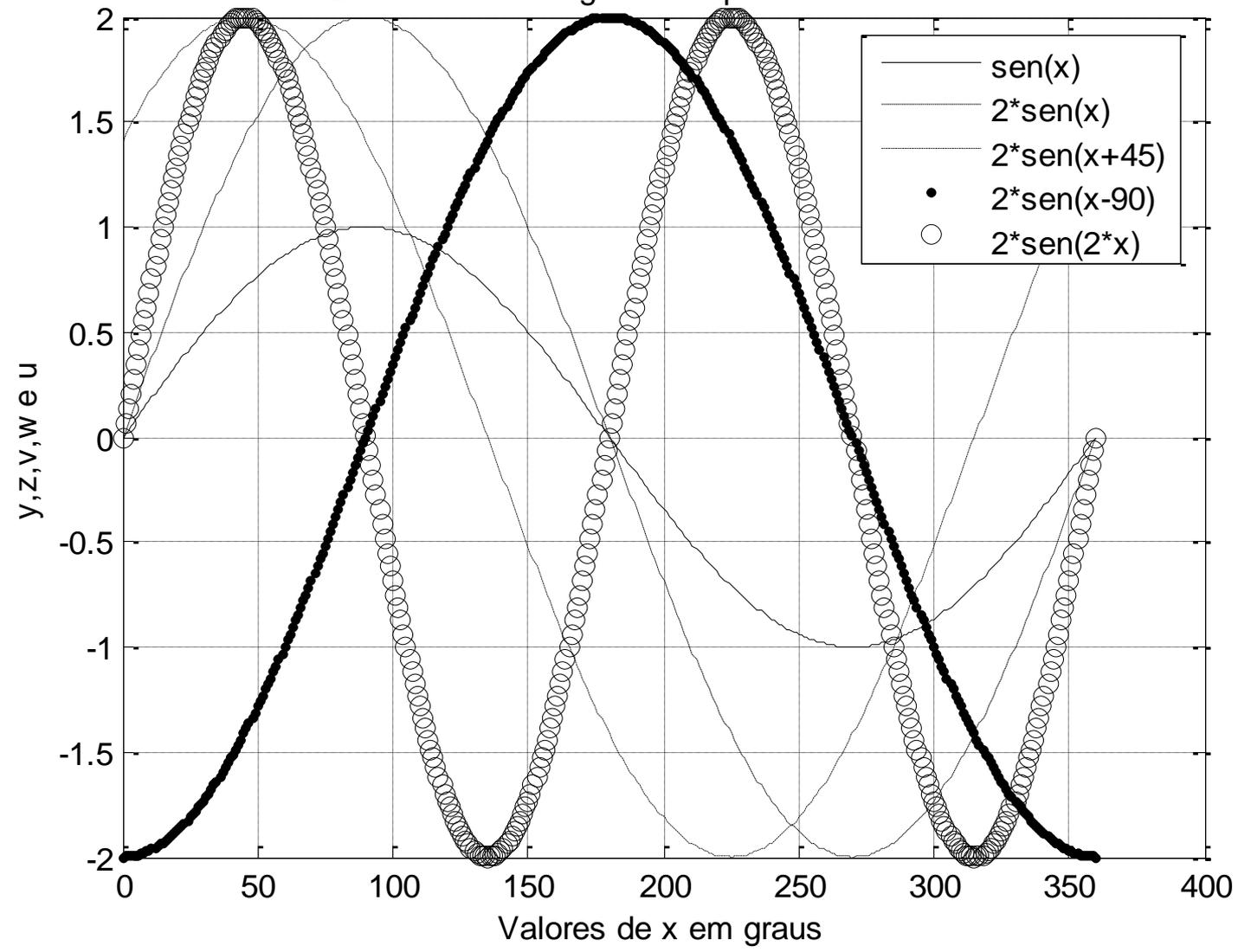
```

%=====
% Exemplo de programação no MATLAB
% Este exemplo plota uma função seno nas seguintes condições:
% sen(x)
% 2*sen(x)
% 2*sen(x+45)
% 2*sen(x-90)
% 2*sen(2*x)
%=====
x=0:360;
% Seno com amplitude A=1 e defasagem phi=0 graus
A = 1;
phi = 0;
y = A*sin(2*pi*x/360+2*pi*phi/360);
% Seno com amplitude A=2 e defasagem phi=0 graus
A = 2;
z = A*sin(2*pi*x/360+2*pi*phi/360);
% Seno com amplitude A=2 e defasagem phi=45 graus
phi = 45;
v = A*sin(2*pi*x/360+2*pi*phi/360);
% Seno com amplitude A= 2 e defasagem phi=-90 graus
phi = -90;
w=A*sin(2*pi*x/360+2*pi*phi/360);
% Seno com amplitude A= 2 e defasagem phi=0 graus
phi = 0;
u = A*sin(2*pi*2*x/360+2*pi*phi/360);
% Plotagem do resultado
plot(x,y,'k-',x,z,'k--',x,v,'k-.',x,w,'k.',x,u, 'ko')
grid
xlabel('Valores de x em graus')
ylabel('y,z,v,w e u')
title('Estudo de defasagem e amplitude de um seno')
legend('sen(x)', '2*sen(x)', '2*sen(x+45)', '2*sen(x-90)', '2*sen(2*x)')

```

- Uma vez escrito o programa, entre no menu **File** da janela do editor de textos e escolha a opção **Save as...** Nesta opção do menu, salve o programa como *prog1.m* no seu diretório de trabalho. Em seguida, volte à janela de comandos do MATLAB e use o comando *cd* ou a opção **Set Path...** do menu **File** para ir ao diretório onde o programa *prog1.m* foi salvo. Em seguida, digite o nome do arquivo (sem a extensão **.m**) para executar o programa:
- `>>prog1`

Estudo de defasagem e amplitude de um seno



Controles de Fluxo

Estruturas Condicionais

Uma estrutura condicional permite a escolha do grupo de comandos a serem executados quando uma dada condição for satisfeita ou não, possibilitando desta forma alterar o fluxo natural de comandos. Esta condição é representada por uma expressão lógica.

Estrutura if-end

A estrutura condicional mais simples do MA TLAB é:

```
if      <condição>  
      <comandos>  
end
```

Se o resultado da expressão lógica <condição> for 1 (verdadeiro) então a lista de <comandos> será executada. Se o resultado for 0 (falso) os <comandos > não serão executados. Por exemplo, considere o arquivo **estcond1.m** cujo conteúdo é:

```
a = input('Entre com o valor de a : ');  
if    a >= 0  
    b = sqrt(a)  
end
```

Para executá-lo basta fornecer o seu nome na área de trabalho

```
>> estcond1
```

```
Entre com o valor de a : 2
```

```
b = 1.4142
```

- Estrutura **if-else-end**

```
if      <condição>  
        <comandos 1>  
else  
        <comandos 0>  
  
end
```

Exemplo:

```
a = input('Entre com o valor de a : ');  
    if      a > 0  
        b = log(a)  
    else  
        b = exp(a)  
  
    end
```

quando executado resultará

```
>> estcond2
```

```
Entre com o valor de a : 5
```

```
b = 1.6094
```

- Estrutura **if-elseif-end**

```
if    <condição 1>  
      <comandos 1>  
elseif <condição 2>  
      <comandos 2>  
elseif <condição 3>  
      <comandos 3>  
  
      ...  
else  
      <comandos 0>  
  
end
```

- Exemplo

```
a = input('Entre com o valor de a : ');
```

```
    if      a <= -10
        b = exp(a)
    elseif a < 0
        b = 1/a
    elseif a <= 10
        b = a^2
    elseif a < 100
        b = sqrt(a)
    else
        b = 10
```

```
    end
```

quando executado resultará

```
>> estcond3
```

```
Entre com o valor de a : 4
```

```
b = 16
```

Estruturas de Repetição

- A estrutura de repetição faz com que uma seqüência de comandos seja executada repetidamente até que uma dada condição de interrupção seja satisfeita. O MATLAB possui duas estruturas de repetição: as estruturas **for-end** e a **while-end**.

- Estrutura **for-end**

A estrutura **for-end** permite que um grupo de comandos seja repetido um número específico de vezes. Sua sintaxe é:

```
for  <variável>=<arranjo>  
      <comandos>  
end
```

onde <variável> é a variável-de-controle que assume todos os valores contidos no vetor linha <arranjo> . Assim, o número de repetições da lista <comandos > é igual ao número de elementos no vetor <arranjo>.

- Exemplo: Crie o vetor $x=[0\ 36\ 72\ 108\ 144\ 180\ 216\ 252\ 288\ 324]$ através do comando **for** (exercicio1.m).

```
x(1)=0;
```

```
for n=2:10
```

```
    x(n)=x(n-1)+36;
```

```
end
```

```
x
```

```
>>exercicio1
```

```
x =
```

```
0 36 72 108 144 180 216 252 288 324
```

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

```
%estrep3.m
```

```
    for i=1:8
```

```
        for j=1:8
```

```
            A(i,j)=i+j;
```

```
            B(i,j)=i-j;
```

```
        end
```

```
    end
```

```
    C=A+B;
```

```
    A, B, C
```

```
>>estrep3
```

Subprograma function

- Um outro tipo de arquivo de roteiro é usado para o próprio usuário criar novas funções para o MatLab. Uma função é criada no MatLab como um arquivo **.m**, porém começando sempre com o seguinte cabeçalho:

```
function [variáveis de saída] = Nome_da_Função (variáveis de entrada)
```

Montagem das matrizes Y e Z

- Algoritmo para montagem da matriz Y(completa) em MATLAB.

% nb; número de barras

% nl; número de linhas

% ng; número de geradores

% ger; index generators

% de; origin index vector

% para; destiny index vector

% rl; line resistance vector

% xl; line reactance vector

% bl; line shunt vector

```
• %Matriz Y
• clear all
• dados6
•
• Y=sparse(nb,nb);
• t=a.*exp(j*phi);
• for line=1:nl
•     k1=de(line);
•     k2=para(line);
•     aux=-(1/(j*x1(line)));
•     Y(k1,k2)=Y(k1,k2)+conj(t(line))*aux;
•     Y(k1,k1)=Y(k1,k1)-(a(line)^2)*aux+j*bl(line)/2;
•     Y(k2,k2)=Y(k2,k2)-aux+j*bl(line)/2;
•     Y(k2,k1)=Y(k2,k1)+t(line)*aux;
• end
•
• for i=1:ng,
•     aux=1/(j*xg1(i));
•     Y(ger(i),ger(i))=Y(ger(i),ger(i))+aux;
• end
•
• Y1=Y
• Z1=inv(Y1)
```