

UNICESUMAR
ENGENHARIA CIVIL
PROGRAMAÇÃO PARA ENGENHARIA (NGER80_271)
ANDRÉ MARTINS OTOMURA

```
<i class=...  
<h3>Happy Clients</h3>  
</div>  
</div>  
<!-- end first count item -->  
<!-- second count item -->  
<div class="col-md-3 col-sm-6 col-xs-12 text-center wow fadeInDown"  
  data-wow-duration="500ms" data-wow-delay="200ms">  
  <div class="counters-item">  
    <div>  
      <span data-speed="3000" data-to="565">565</span>  
    </div>  
  </div>  
</div>
```

3ª AULA

```
80 @media  
81  
82 .navbar-inverse .navbar-nav>li>a {  
83   padding: 30px 10px;  
84 }  
85 .navbar-inverse .smaller .navbar-nav>li>a {  
86   padding: 20px 10px;  
87 }  
88 .navbar-inverse .navbar-nav>li {  
89   padding-right: 0;  
90 }  
91 .carousel-caption h2 {
```

Introdução ao software MATLAB: comandos e ferramentas básicas do software
Gráficos 2D e 3D
Funções e subalgoritmos em MATLAB

DICAS PARA SOLUCIONAR PROBLEMAS

1. ANOTAR OS DADOS FORNECIDOS PELO ENUNCIADO DO PROBLEMA E QUE O PROGRAMA UTILIZARÁ COMO ENTRADA (EX: NOME, VALOR, QUANTIDADE...);
2. ANOTAR TUDO O QUE PRECISA SER ENTREGUE NO FINAL, O QUE SERÁ APRESENTADO COMO RESPOSTA;
3. ESBOÇAR AS FÓRMULAS OU EQUAÇÕES QUE SERÃO UTILIZADAS PELO PROGRAMA PARA ENCONTRAR CADA UMA DAS RESPOSTAS NECESSÁRIAS;
4. FAZER UM LEVANTAMENTO DAS VARIÁVEIS NECESSÁRIAS PARA UTILIZAR AS FÓRMULAS E TAMBÉM PARA APRESENTAR OS RESULTADOS;
5. LISTAR O PASSO A PASSO DO QUE O PROGRAMA DEVERÁ EXECUTAR, NA ORDEM CERTA QUE POSSIBILITE CHEGAR AO RESULTADO;
6. ANALISAR E LISTAR OS COMANDOS DA LINGUAGEM DE PROGRAMAÇÃO REFERENTES AOS PASSOS LISTADOS ANTERIORMENTE;
- 7.

TÓPICOS	AULAS													
	04/jan	05/jan	08/jan	09/jan	10/jan	11/jan	12/jan	15/jan	16/jan	17/jan	18/jan	19/jan	22/jan	23/jan
Conceitos de hardware														
Conceitos e definições de algoritmos														
Diagrama de blocos e pseudocódigos														
Variáveis														
Constantes														
Comandos de atribuição														
Comandos de entrada e saída de dados														
Matrizes e vetores														
Estruturas condicionais (If - Else e Case) e suas implementações														
Estruturas de repetição (While e For) e suas implementações														
Funções e procedimentos														
Manipulação de arquivos														
Introdução ao software MATLAB: comandos e ferramentas básicas do software														
Gráficos 2D e 3D														
Funções e subalgoritmos em MATLAB														
Introdução ao Microsoft Excel: funções de busca e referência e funções lógicas														
Microsoft Excel para engenharia: atingir a meta de solver, cenários e tabela de dados														
Formulários e VBA														
Sistemas de equações lineares														
Zeros de equações algébricas e transcendent														
Interpolações e aproximações de funções														
Integração numérica														
Soluções numéricas de equações diferenciais														

Horário: de segunda a sexta-feira, das 13h00 até 17h30.
Local: Laboratório 8, Bloco 07
Intervalo1: 14h30 até 14h40
Intervalo2: 16h00 até 16h10

- Matrizes e vetores
- Estruturas condicionais (If - Else e Case) e suas implementações
- Estruturas de repetição (While e For) e suas implementações
- Funções e procedimentos
- Manipulação de arquivos

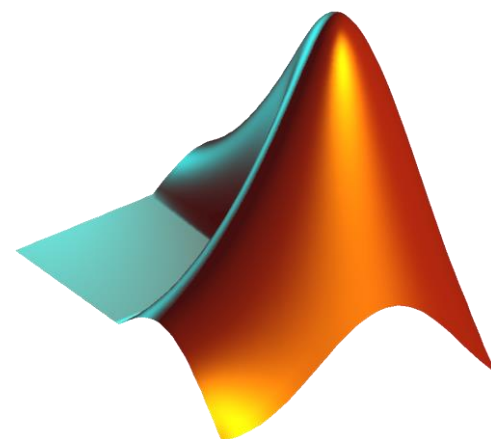
UNICESUMAR
ENGENHARIA CIVIL
PROGRAMAÇÃO PARA ENGENHARIA (NGER80_271)
ANDRÉ MARTINS OTOMURA

3ª AULA

Introdução ao software MATLAB: comandos e ferramentas básicas do software
Gráficos 2D e 3D
Funções e subalgoritmos em MATLAB

*"O MATLAB é uma poderosa linguagem usada na computação técnica.
O nome MATLAB vem do termo em inglês MATrix LABoratory
(Laboratório de Matrizes), pois as matrizes (ou arranjos) são o seu
elemento de dados básico."*

Métodos Numéricos para Engenheiros e Cientistas



MATLAB - INTERFACE

Unicesumar
Engenharia Civil
Programação para Engenharia
(NGER80_271)

MATLAB R2017a

HOME PLOTS APPS EDITOR PUBLISH VIEW

File Edit Breakpoints Run Run and Advance Run and Time

Current Folder: C:\Program Files\MATLAB\R2017a\bin

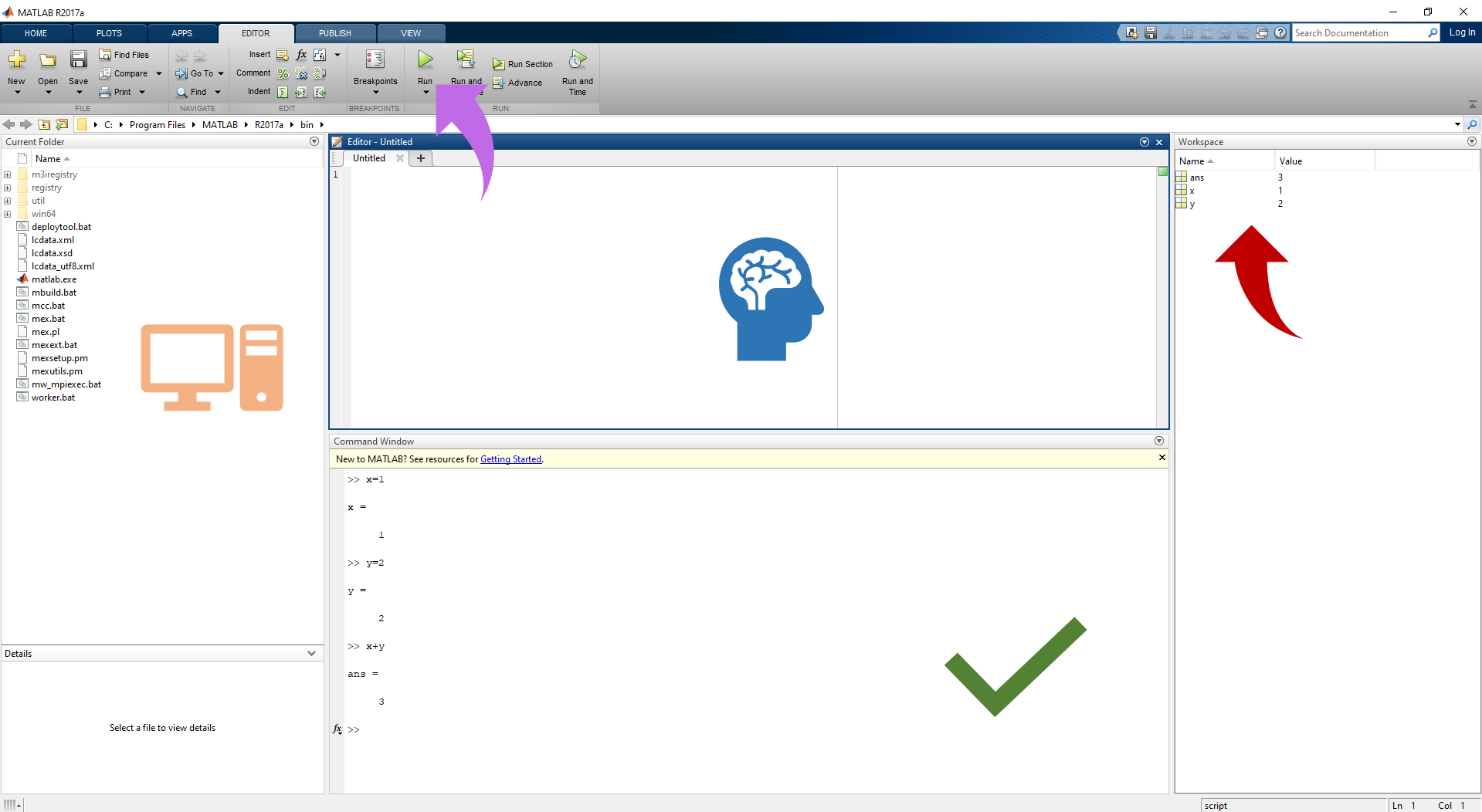
Workspace:

Name	Value
ans	3
x	1
y	2

Command Window:

```
>> x=1  
  
x =  
  
1  
  
>> y=2  
  
y =  
  
2  
  
>> x+y  
  
ans =  
  
3  
  
fx >>
```

script | Ln 1 Col 1



Operações aritméticas fundamentais com escalares

A maneira mais simples de usar o MATLAB é tratá-lo como uma calculadora. Com escalares, os símbolos das operações aritméticas são:

Operação	Símbolo	Exemplo	Operação	Símbolo	Exemplo
Adição	+	$5 + 3$	Divisão à direita	/	$5 / 3$
Subtração	-	$5 - 3$	Divisão à esquerda	\	$5 \setminus 3 = 3 / 5$
Multiplicação	*	$5 * 3$	Exponenciação	^	$5 \wedge 3$ (significa $5^3 = 125$)

Os exemplos a seguir mostram o uso da função `sqrt(x)` quando o MATLAB trabalha com escalares.

```
>> sqrt(64)
```

```
ans =  
    8
```

O argumento é um número.

```
>> sqrt(50 + 14*3)
```

```
ans =  
    9.5917
```

O argumento é uma expressão.

```
>> sqrt(54 + 9*sqrt(100))
```

```
ans =  
    12
```

O argumento inclui uma função.

```
>> (15 + 600/4)/sqrt(121)
```

```
ans =  
    15
```

A função é incluída em uma expressão.

Tabela A-2 Funções matemáticas elementares

Comando	Descrição	Exemplo	
sqrt (x)	Raiz quadrada	>> sqrt(81) ans = 9	
exp (x)	Exponencial (e^x)	>> exp(5) ans = 148.4132	
abs (x)	Valor absoluto	>> abs(-24) ans = 24	
log (x)	Logaritmo natural, na base e (ln)	>> log(1000) ans = 6.9078	
log10 (x)	Logaritmo na base 10	>> log10(1000) ans = 3.0000	
sin (x)	Seno de um ângulo x (em radianos)	>> sin(pi/6) ans =	>> sind(30) ans =
sind (x)	Seno de um ângulo x (em graus)	0.5000	0.5000
As demais funções trigonométricas são escritas da mesma maneira. As funções trigonométricas inversas são escritas simplesmente acrescentando “a” na frente dos comandos, como, por exemplo, asin(x).			

Fonte: Métodos Numéricos para Engenheiros e Cientistas

Tabela A-2 Funções matemáticas elementares

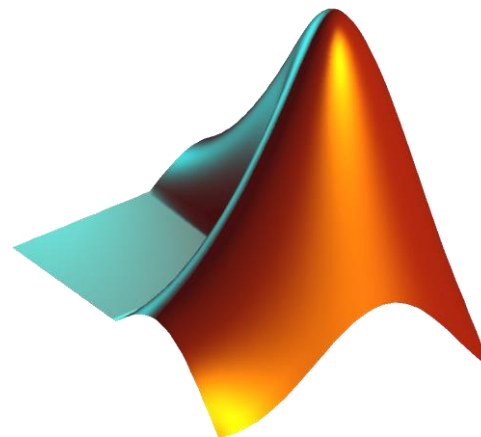
Comando	Descrição	Exemplo	
sqrt (x)	Raiz quadrada	>> sqrt(81) ans = 9	
exp (x)	Exponencial (e^x)	>> exp(5) ans = 148.4132	
abs (x)	Valor absoluto	>> abs(-24) ans = 24	
log (x)	Logaritmo natural, na base e (ln)	>> log(1000) ans = 6.9078	
log10 (x)	Logaritmo na base 10	>> log10(1000) ans = 3.0000	
sin (x)	Seno de um ângulo x (em radianos)	>> sin(pi/6) ans =	>> sind(30) ans =
sind (x)	Seno de um ângulo x (em graus)	0.5000	0.5000
As demais funções trigonométricas são escritas da mesma maneira. As funções trigonométricas inversas são escritas simplesmente acrescentando “a” na frente dos comandos, como, por exemplo, asin(x).			

Fonte: Métodos Numéricos para Engenheiros e Cientistas



“O arranjo é a forma fundamental que o MATLAB utiliza para armazenar e manipular dados. Um arranjo é uma lista de números organizados em linhas e/ou colunas. O arranjo mais simples (unidimensional) corresponde a uma linha ou uma coluna de números, o que é comumente chamado de vetor na ciência e na engenharia. Um arranjo mais complexo (bidimensional) é uma coleção de números organizados em linha e colunas, o que é chamando de matriz na ciência e na engenharia. Cada número presente em um vetor ou matriz é chamado de elemento.”

Fonte: Métodos Numéricos para Engenheiros e Cientistas



CRIANDO UM VETOR:

`x = [a b c d e]`

Ex.:

`x = [1 2 3 4 5]`

`y = [1,1 23,4 143,1]`

`z = ["nome" "apelido" "sobrenome"]`

```
>> ano = [1984 1986 1988 1990 1992 1994 1996]
```

Vetor linha formado com a digitação dos elementos.

```
ano =
```

```
    1984    1986    1988    1990    1992    1994    1996
```

```
>> pnt = [2; 4; 5]
```

Vetor coluna formado com a digitação dos elementos.

```
pnt =
```

```
    2
```

```
    4
```

```
    5
```

```
>> x = [1:2:13]
```

Vetor linha com espaçamento constante.

```
x =
```

```
    1    3    5    7    9   11   13
```

```
>> va = linspace(0,8,6)
```

Vetor linha com 6 elementos, primeiro elemento 0 e último elemento 8.

```
va =
```

```
    0    1.6000    3.2000    4.8000    6.4000    8.0000
```

CRIANDO UMA MATRIZ:

`x = [a b c; a2 b2 c2]`

Ex.:

`x = [1 2 3 4 5; 21 22 23 24 25]`

`y = [1,1; 23,4; 143,1]`

`z = ["nome"; "apelido"; "sobrenome"; "e-mail"]`

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$


```
>> a = [5 35 43; 4 76 81; 21 32 40]
```

Um ponto-e-vírgula é digitado entre cada linha.

```
a =
```

```
    5    35    43  
    4    76    81  
   21    32    40
```

```
>> cd = 6; e = 3; h = 4;
```

Definem-se as variáveis.

```
>> Mat = [e, cd*h, cos(pi/3); h^2, sqrt(h*h/cd), 14]
```

Elementos são fornecidos como expressões matemáticas.

```
Mat =
```

```
    3.0000   24.0000    0.5000  
   16.0000    1.6330   14.0000
```

Soma e subtração de arranjos

Com arranjos, as operações de soma, subtração e multiplicação seguem as regras da álgebra linear (ver Capítulo 2). As operações $+$ (adição) e $-$ (subtração) podem ser realizadas apenas com arranjos de mesmo tamanho (com o mesmo número de linhas e colunas). A soma ou a diferença de arranjos é obtida com a soma ou subtração de seus elementos correspondentes.

Em geral, se A e B são dois arranjos (por exemplo, matrizes (2×3)),

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

então, a matriz obtida com a soma de A e B é:

$$\begin{bmatrix} (A_{11} + B_{11}) & (A_{12} + B_{12}) & (A_{13} + B_{13}) \\ (A_{21} + B_{21}) & (A_{22} + B_{22}) & (A_{23} + B_{23}) \end{bmatrix}$$

```
>> VA = [8 5 4]; VB = [10 2 7];
```

Define dois vetores VA e VB.

```
>> VC = VA + VB
```

Define um vetor VC que é igual a VA+VB.

```
VC =
```

```
18  7 11
```

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

Define duas matrizes A e B.

```
A =
```

```
5  -3  8
```

```
9   2 10
```

```
B =
```

```
10  7  4
```

```
-11 15  1
```

```
>> C = A + B
```

Define uma matriz C que é igual a A+B.

```
C =
```

```
15  4 12
```

```
-2 17 11
```

```
>> C - 8
```

Subtrai 8 da matriz C.

```
ans =
```

```
7  -4  4
```

```
-10  9  3
```

8 é subtraído de cada elemento de C.

MATLAB – VETORES E MATRIZES

Multiplicação de arranjos

A operação de multiplicação $*$ é executada pelo MATLAB de acordo com as regras da álgebra linear (ver Seção 2.4.1). Isso significa que, se A e B são duas matrizes, a operação $A*B$ pode ser realizada apenas se o número de colunas da matriz A for igual ao número de linhas da matriz B . O resultado é uma matriz que tem o mesmo número de linhas de A e o mesmo número de colunas de B . Por exemplo, se A é uma matriz (3×2) e B é uma matriz (2×4) , então a operação $C=A*B$ tem como resultado uma matriz (3×4) :

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

Define duas matrizes A e B.

```
A =
```

```
2  -1
8   3
6   7
```

```
B =
```

```
4   9   1  -3
-5   2   4   6
```

```
>> C = A*B
```

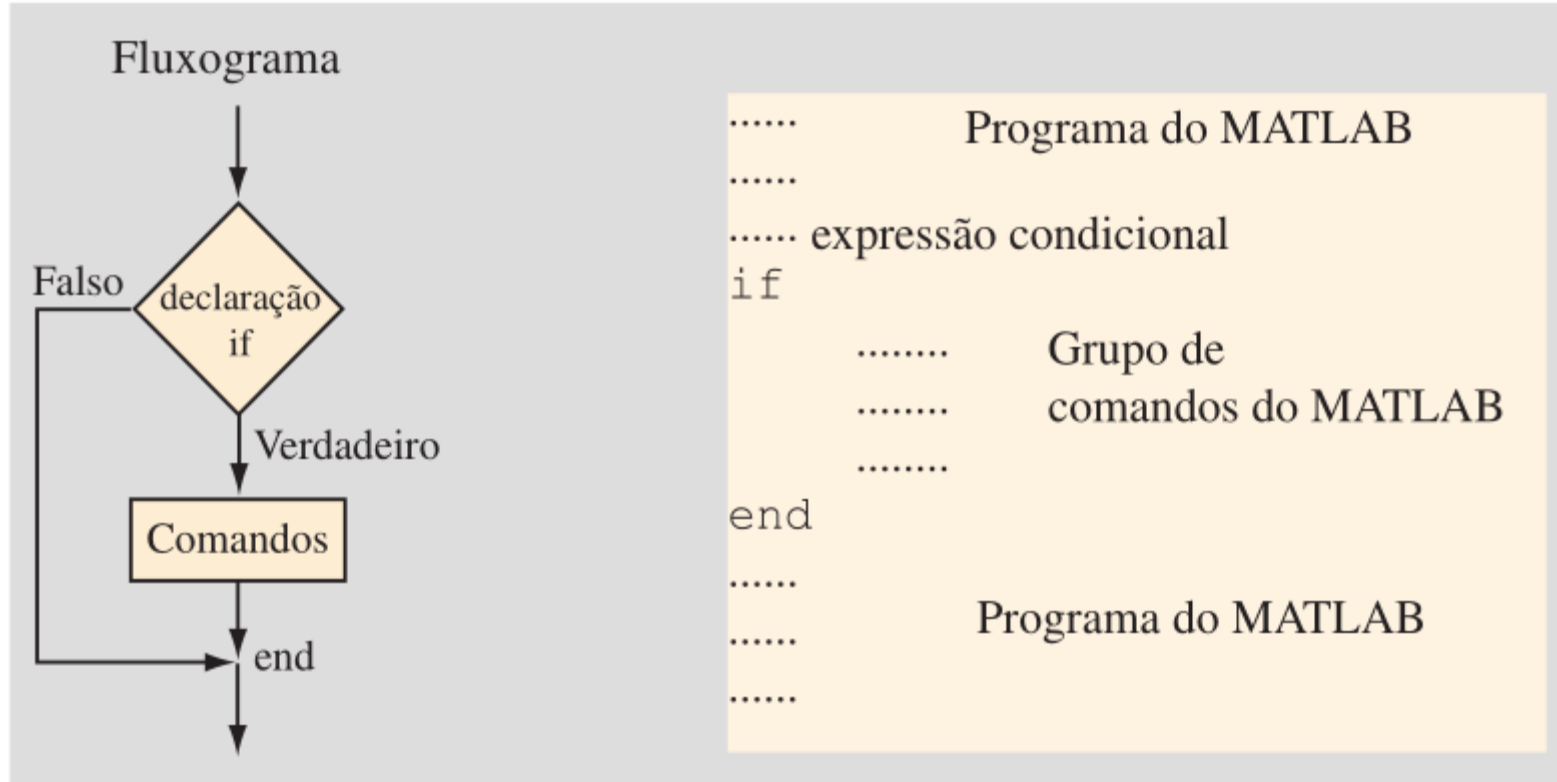
Multiplica A*B.

```
C =
```

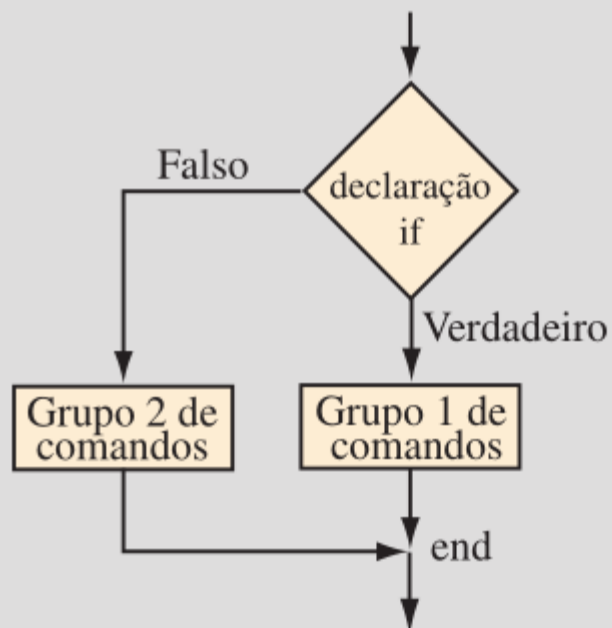
```
13  16  -2 -12
17  78  20  -6
-11 68  34  24
```

C é uma matriz (3 x 4).



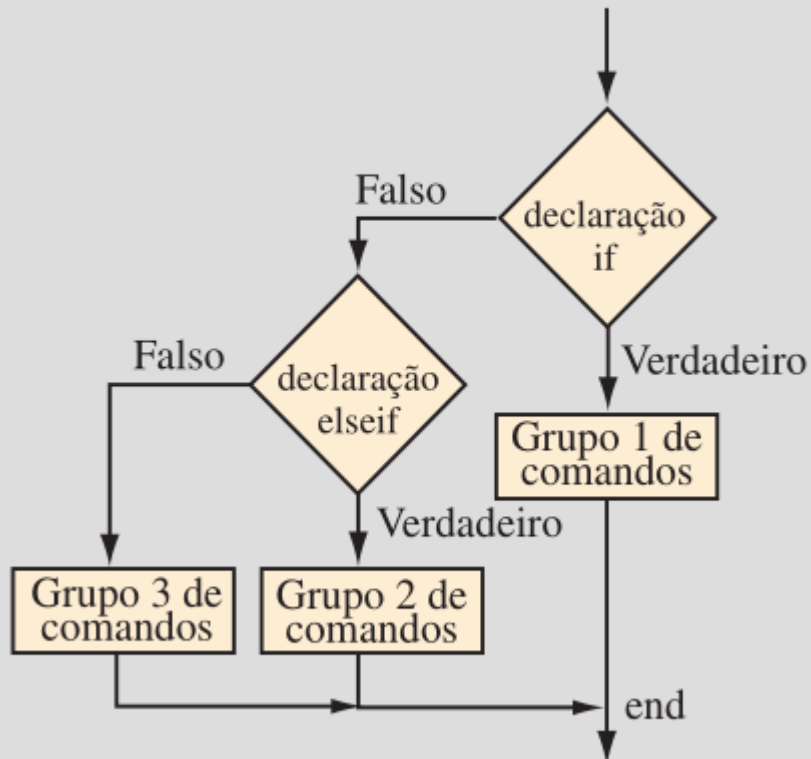


Fluxograma



```
..... Programa do MATLAB
.....
if expressão condicional
    ..... Grupo 1 de
    ..... comandos do MATLAB
else
    ..... Grupo 2 de
    ..... comandos do MATLAB
end
..... Programa do MATLAB
.....
```


Fluxograma

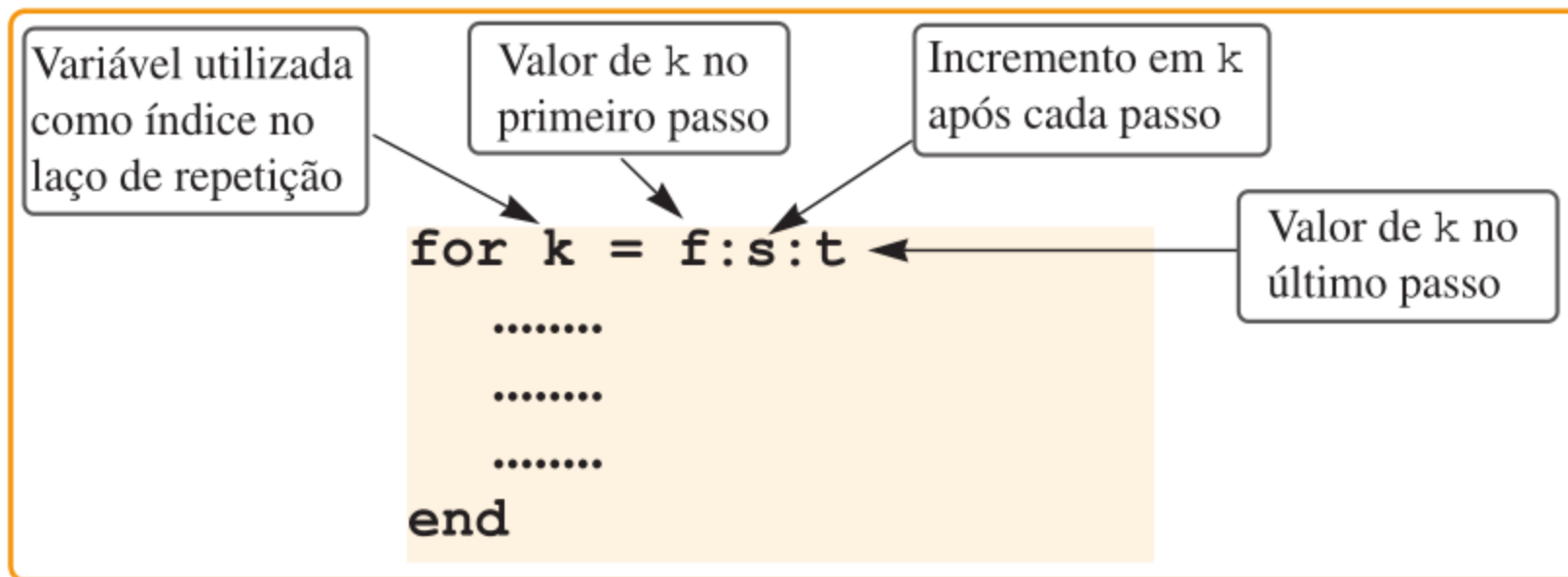


```

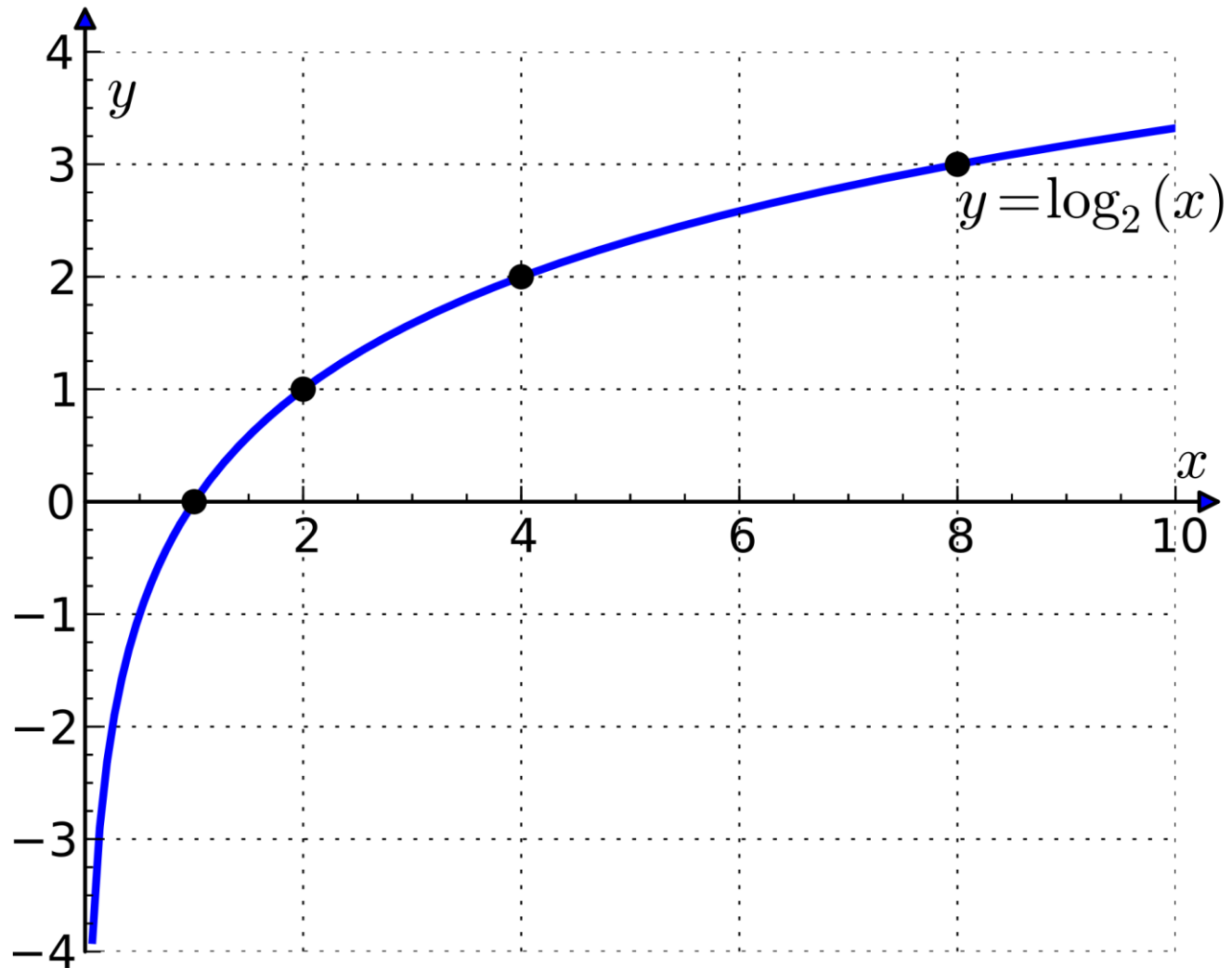
.....      MATLAB program.
.....

if expressão condicional
    .....  ]  Grupo 1 de
    .....   ]  comandos do MATLAB.
    .....   ]
elseif expressão condicional
    .....  ]  Grupo 2 de
    .....   ]  comandos do MATLAB.
    .....   ]
else
    .....  ]  Grupo 3 de
    .....   ]  comandos do MATLAB.
    .....   ]
end

.....      Programa do MATLAB.
.....
  
```





```
plot (x,y)  
xlabel('titulo_do_eixo_x')  
ylabel('titulo_do_eixo_y')  
title('titulo_do_grafico')
```

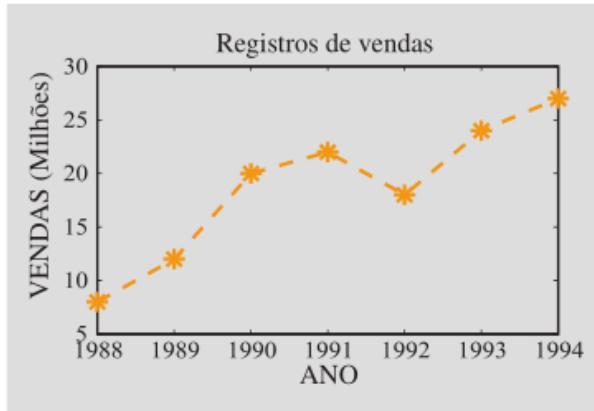


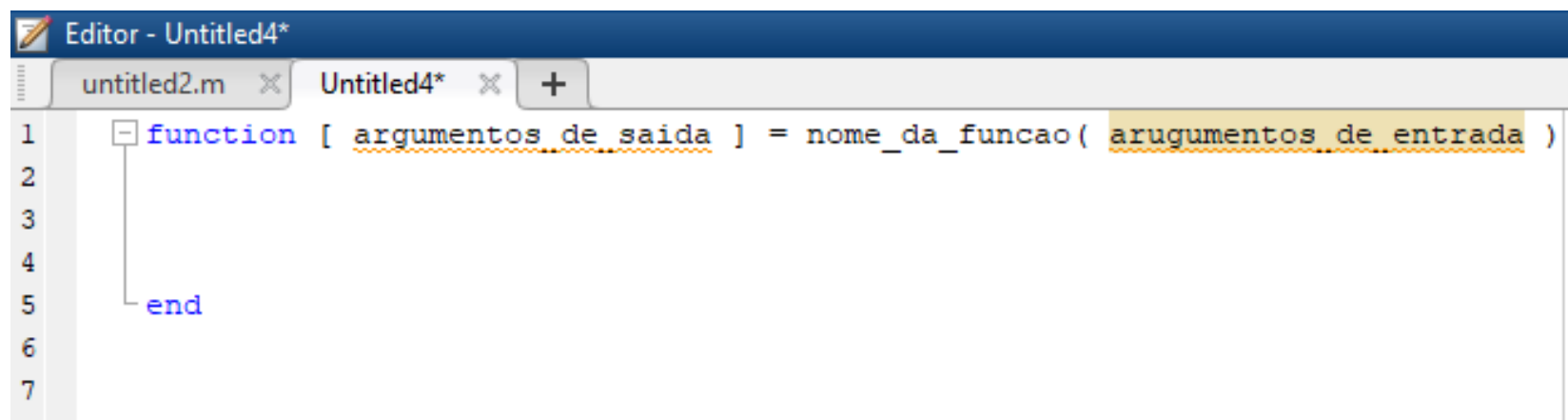
Figura A-6 Gráfico formatado.

Por exemplo, o programa listado abaixo produz o gráfico mostrado na Fig. A-6. Os dados são traçados com uma linha laranja tracejada usando asteriscos como marcadores. A figura inclui rótulos de eixos e um título.

```
>> ano=[1988:1:1994];  
>> vendas=[8 12 20 22 18 24 27];  
>> plot(ano,vendas,'--r*','linewidth',2,'markersize',12)  
>> xlabel('ANO')  
>> ylabel('VENDAS(Milhões)')  
>> title('Registros de Vendas')
```

```
x = [-3:1:3];  
y = [-3:1:2];  
[xx, yy] = meshgrid(x, y);  
zz = xx.^2 - yy.^2  
  
figure  
surf (xx, yy, zz);  
xlabel('titulo_do_eixo_x');  
ylabel('titulo_do_eixo_y');  
zlabel('titulo_do_eixo_z');  
title('titulo_do_grafico');
```



The image shows a screenshot of the MATLAB Editor window. The title bar reads "Editor - Untitled4*". Below the title bar, there are two tabs: "untitled2.m" and "Untitled4*", with a "+" button to add more. The "Untitled4*" tab is active. The editor area shows a function definition starting on line 1: `function [argumentos_de_saida] = nome_da_funcao(arugumentos_de_entrada)`. The word "function" is in blue. The output arguments `[argumentos_de_saida]` are underlined with a dashed line. The input arguments `arugumentos_de_entrada` are highlighted in yellow. The function ends on line 5 with the keyword `end` in blue. A line number margin on the left shows lines 1 through 7.

```
1 function [ argumentos_de_saida ] = nome_da_funcao( arugumentos_de_entrada )
2
3
4
5 end
6
7
```


AULA 02

Super exercício;
Fatorial;

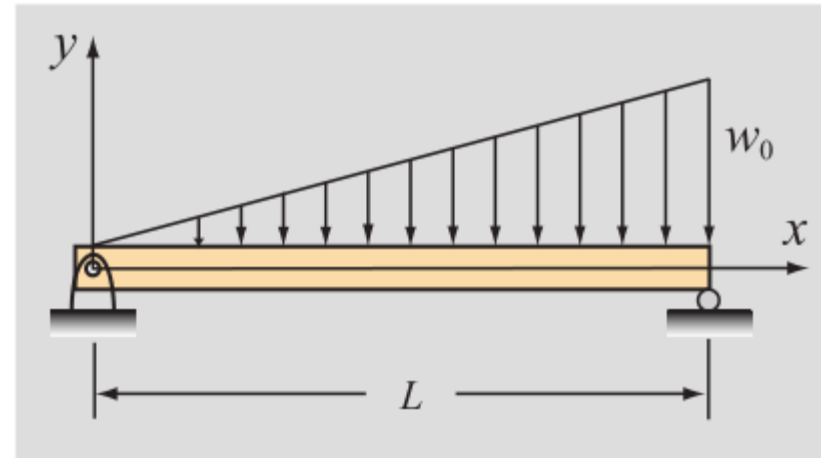
Plotagem

Reta

Parábola

A.12 Uma viga é presa de um lado e suportada por um rolamento em sua outra extremidade. A deflexão y no ponto x de uma viga carregada da forma mostrada é dada pela equação:

$$y = \frac{-w_0 x}{360 E I L} (3x^3 - 10L^2 x^2 + 7L^4)$$



onde E é o módulo elástico, I é o momento de inércia e L é o comprimento da viga. Para a viga mostrada na figura, $L = 6$ m, $E = 70 \times 10^9$ Pa (alumínio), $I = 9,19 \times 10^{-6}$ m⁴ e $w_0 = 800$ N/m.

Trace a deflexão y da viga em função de x .

