Notas de Aula
MatLab - 1

Routo Terada

www.ime.usp.br/~rt

Depto. C. da Computação - USP

Bibliografia:

E. Y. Matsumoto, MatLab6 Fundamentos de Programação, Edit. Érica, 2000

K. Chen et al., Mathematical explorations with MatLab, Cambridge University Press 1999

D. Hanselman et al., MatLab 5 -- Guia do Usuário, Editora Makron 1999

conteúdo

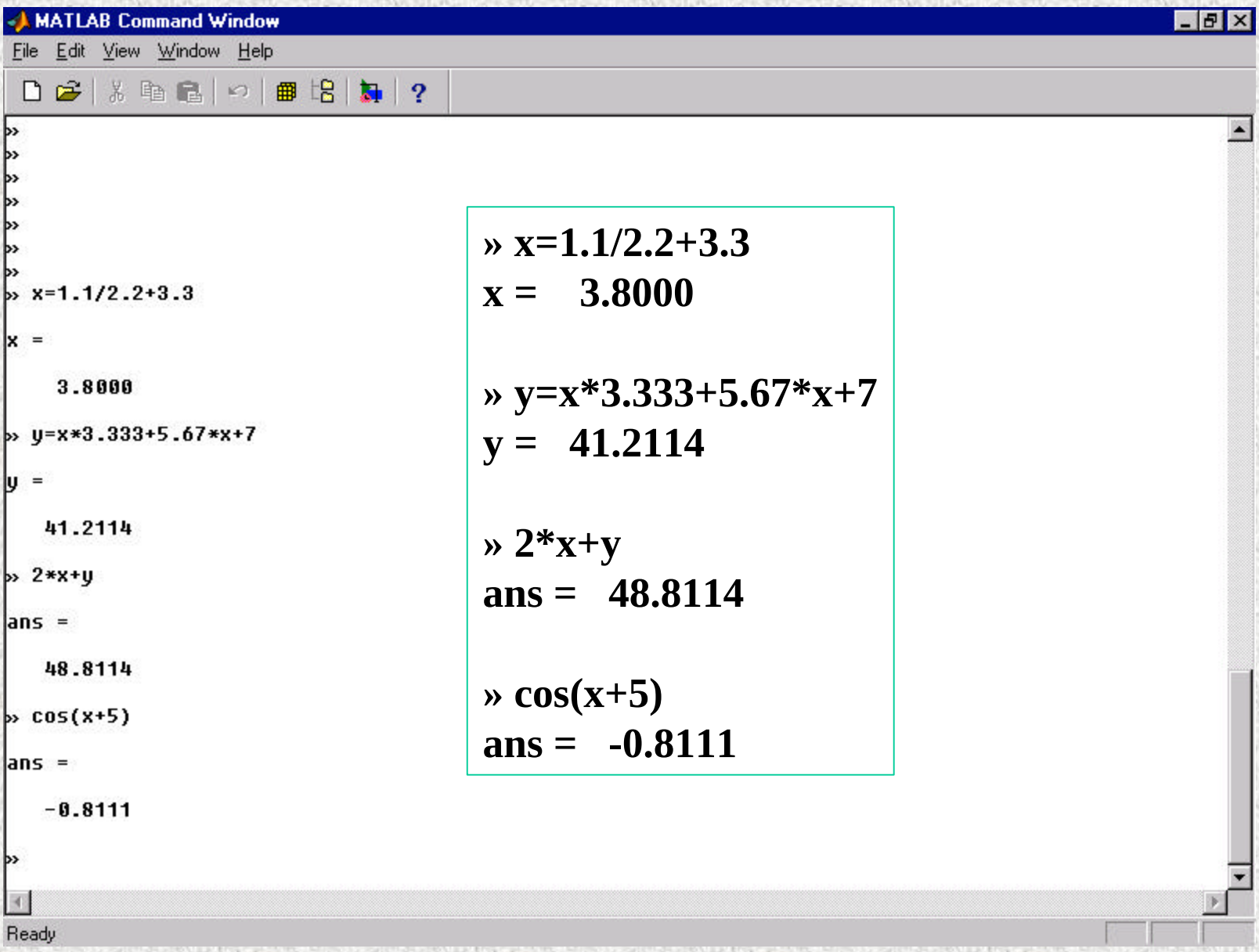
- **Workspace:variáveis, help, números complexos, save e load, who.**
- **Funções matemáticas elementares. Número variável de algarismos significativos. Aritmética sobre números complexos.**
- **Arquivos M: edição, funções básicas. Exemplos de ToolBox.**
- **Vetores e matrizes: submatriz, geração de matriz aleatória, operações aritméticas sobre matrizes, solução de sistemas lineares.**
- **Execução de comandos MAPLE.**
- **Operações relacionais e lógicas.**
- **Funções de data e hora.**

MatLab -- Matrix Laboratory (Cleve Moler, U. N. Mexico, ´80)

1. **Diversos ToolBoxes de projetos**
2. **Inclusos**
   LINPACK (Álgebra Linear), EISPACK (autovalores)
3. **Linguagem MatLab para programar usando ToolBox**

Prompt no WorkSpace é »
Para sair use comando quit
Na janela WorkSpace
- **não podemos editar os comandos**
- **resultado do cálculo é atribuído à variável ans (de *ans*wer)**

Seta para cima: para repetir comando anterior



» x=1.1/2.2+3.3
x = 3.8000

» y=x*3.333+5.67*x+7
y = 41.2114

» 2*x+y
ans = 48.8114

» cos(x+5)
ans = -0.8111

Matemática elementar

operação	*símbolo*	*exemplo*
soma	+	5+2.2=7.2000
subtração	-	3.3-8.8=12.1000
multiplicação	*	3.14*1.111=3.4885
→ divisão	/ ou \	44/2 =2\44=22
potenciação	^	2^6=64
raiz quadrada	sqrt()	sqrt(2.2)=1.4832
chão	floor()	floor(-2.7)=-3
teto	ceil()	ceil(-2.7)=-2
max div comum	gcd()	gcd(4,8)=4
logaritmo natural	log()	log(2.2)=0.7885
logaritmo na base 10	log10()	log10(2.2)=0.3424
resto de divisão	rem(,)	rem(711,7)=4
arredondam.	round()	round(-2.1)= -2
		roun(-2.7)= -3
sinal	sign()	sign(-2.7)= -1

operação	*símbolo*	*exemplo*
soma	+	5+2.2=7.2000
subtração	-	3.3-8.8=12.1000
multiplicação	*	3.14*1.111=3.4885
divisão	/ ou \	44/2 =2\44=22
potenciação	^	2^6=64
raiz quadrada	sqrt ()	sqrt (2.2)=1.4832
chão	floor()	floor(-2.7)=-3
teto	ceil ()	ceil (-2.7)=-2
max div comum	gcd ()	gcd (4,8)=4
logaritmo natural	log()	log(2.2)=0.7885
logaritmo na base 10	log10()	log10(2.2)=0.3424
resto de divisão	rem (,)	rem (711,7)=4
arredondam .	round()	round(-2.1)= -2
		round(-2.7)= -3
sinal	sign()	sign(-2.7)= -1
seno	sin()	sin(3.1)=0.0416
cosseno	cos ()	cos (3.1)=-0.9991
tangente	tan()	tan(3.1)=-0.0416

1. **help** ***nomecom*** **para obter info. do comando** ***nomecom***

2. **who e whos listam variáveis do WorkSpace**

3. **clear limpa as variáveis do WorkSpace**

4. **save** ***nomearq*** **salva o WorkSpace com nome** ***nomearq***

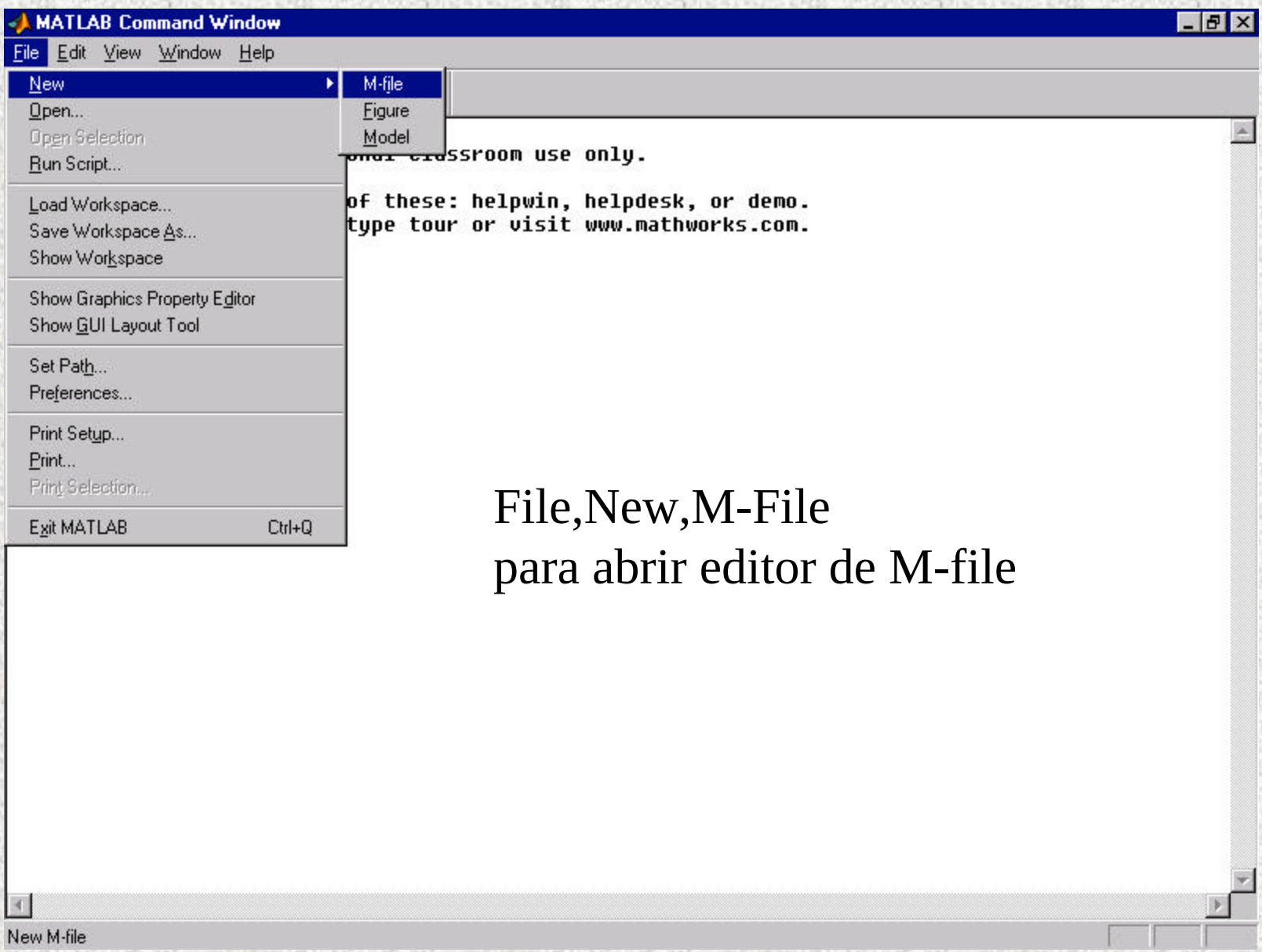
5. **load** ***nomearq*** **-- faz o contrário de save**

6. **Comando terminado com pto.-e-vírgula: resultado não sai na tela**

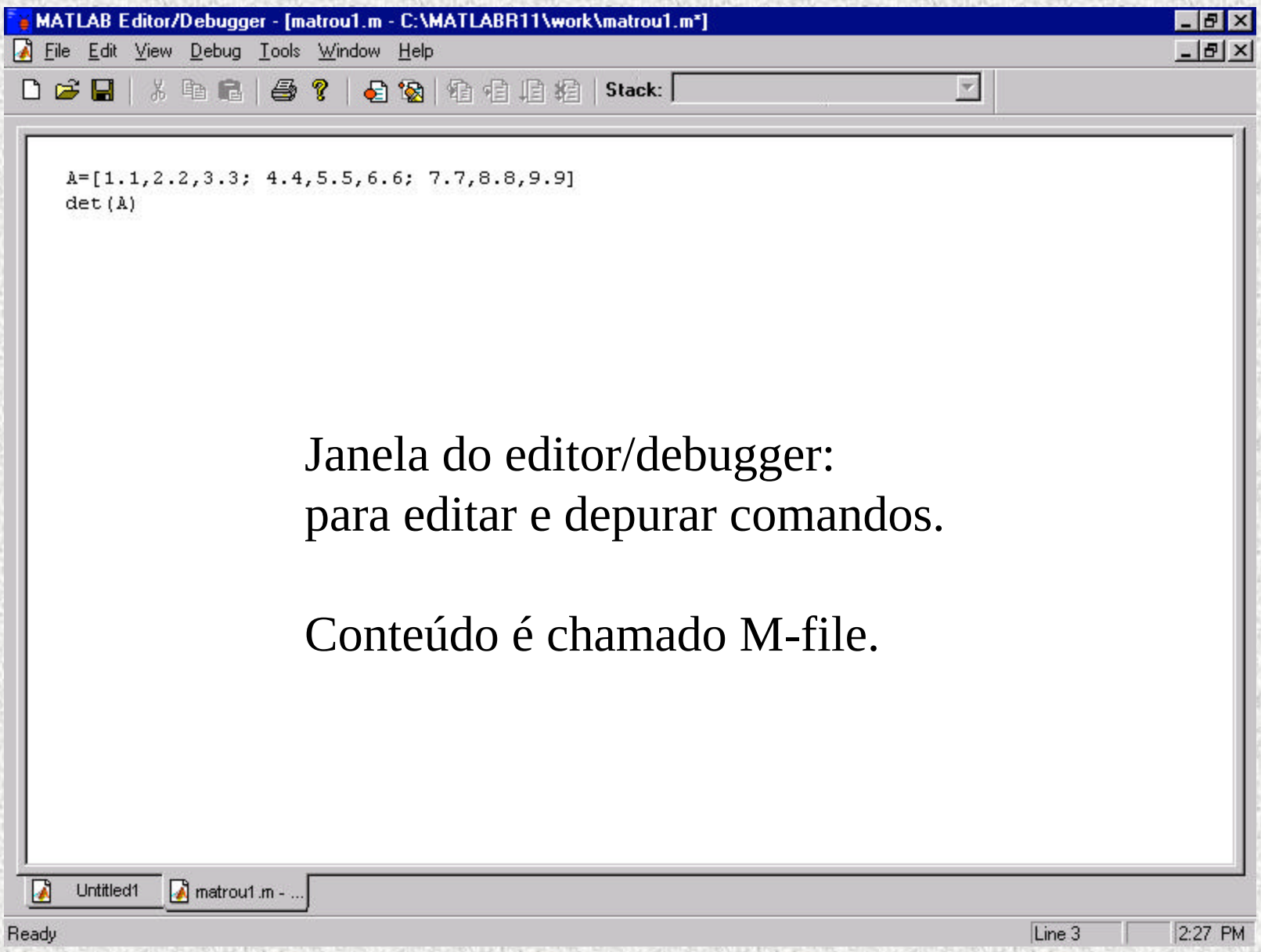
7. **Vírgula para separar vários comandos na mesma linha**

8. **Para continuar um comando na linha seguinte, digitar ... no fim da primeira linha**

9. **O sinal % indica que o resto da linha é um comentário**

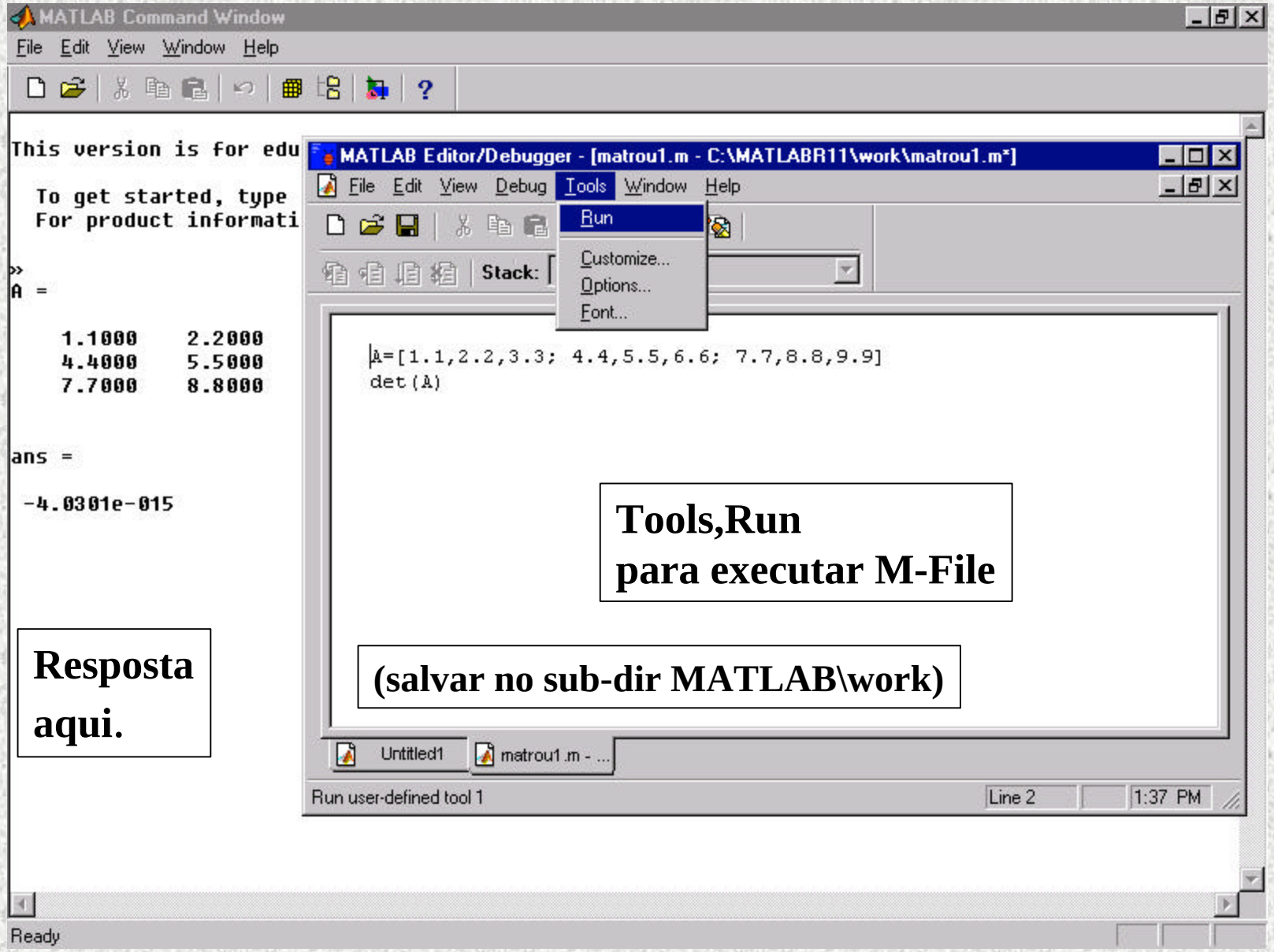


File,New,M-File
para abrir editor de M-file

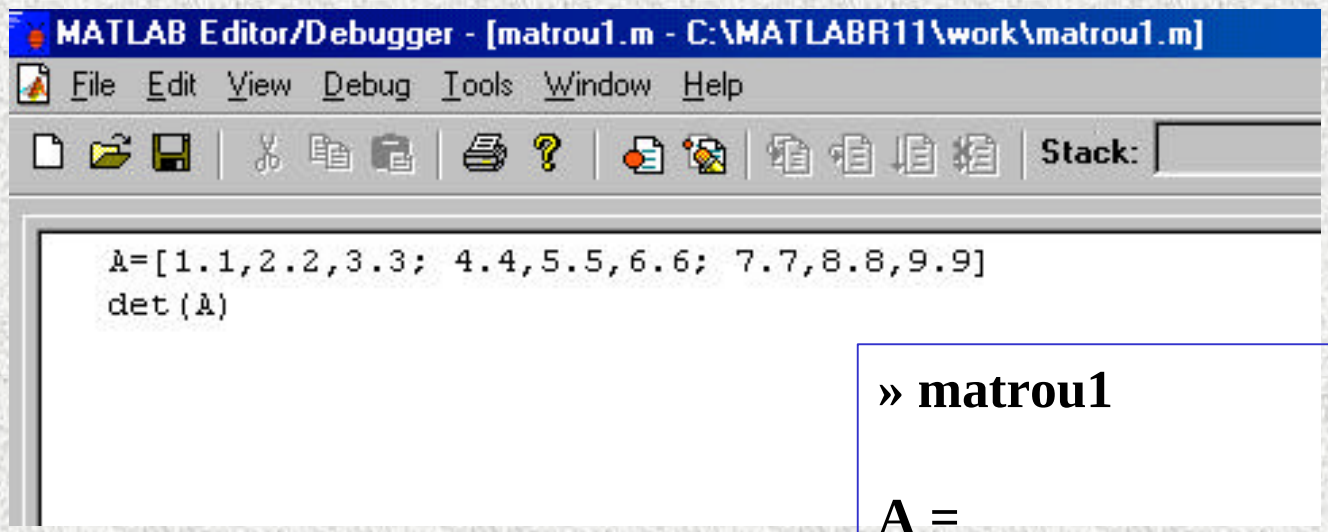


Janela do editor/debugger:
para editar e depurar comandos.

Conteúdo é chamado M-file.

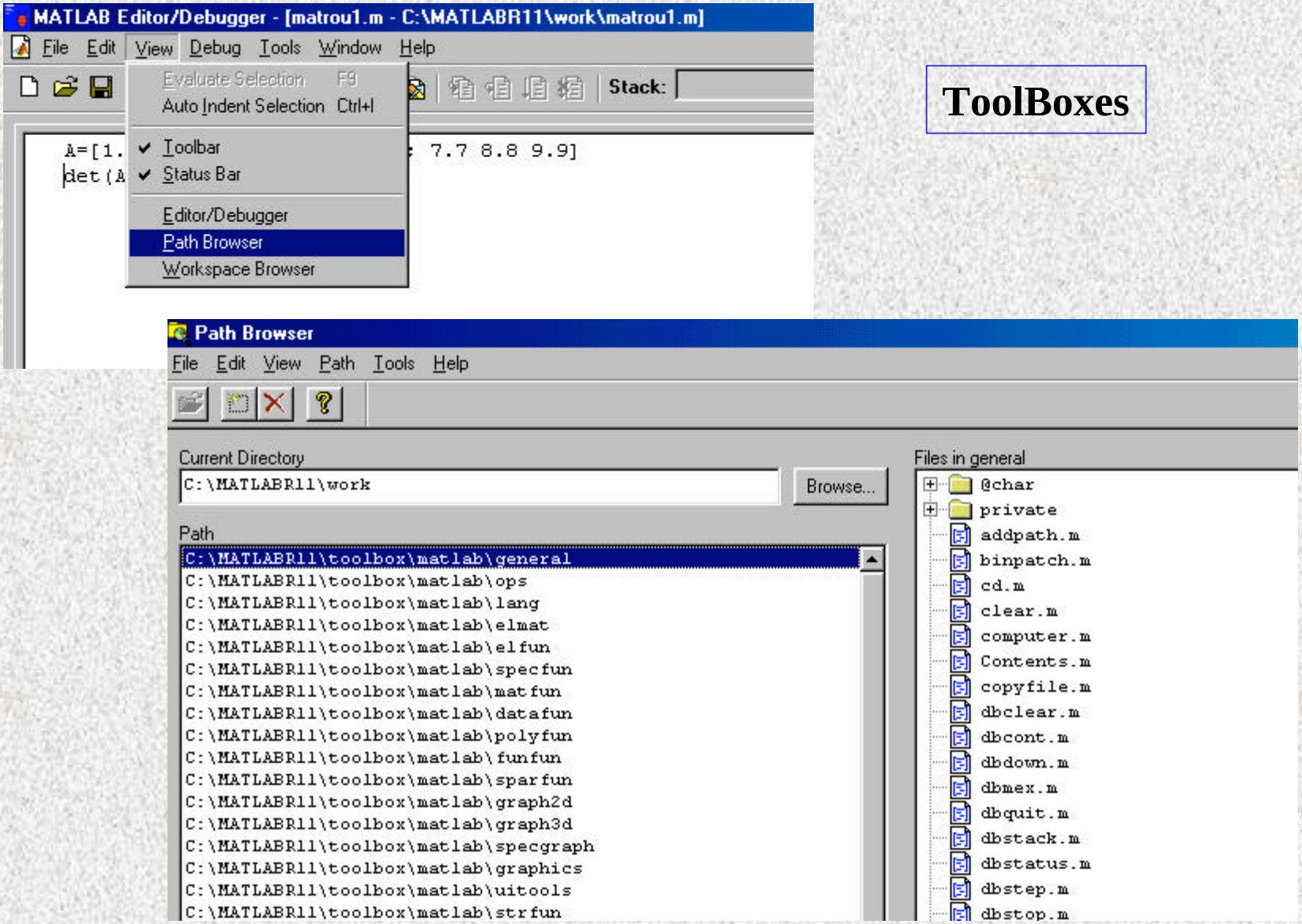

MATLAB Command Window
File Edit View Window Help
This version is for edu
To get started, type
For product informati
>>
A =
1.1000 2.2000
4.4000 5.5000
7.7000 8.8000
ans =
-4.0301e-015
MATLAB Editor/Debugger - [matrou1.m - C:\MATLABR11\work\matrou1.m*]
File Edit View Debug Tools Window Help
Run
Customize...
Options...
Font...
Stack:
A=[1.1,2.2,3.3; 4.4,5.5,6.6; 7.7,8.8,9.9]
det(A)
Tools,Run
para executar M-File
(salvar no sub-dir MATLAB\work)
Resposta aqui.
Untitled1
matrou1.m - ...
Run user-defined tool 1
Line 2
1:37 PM
Ready

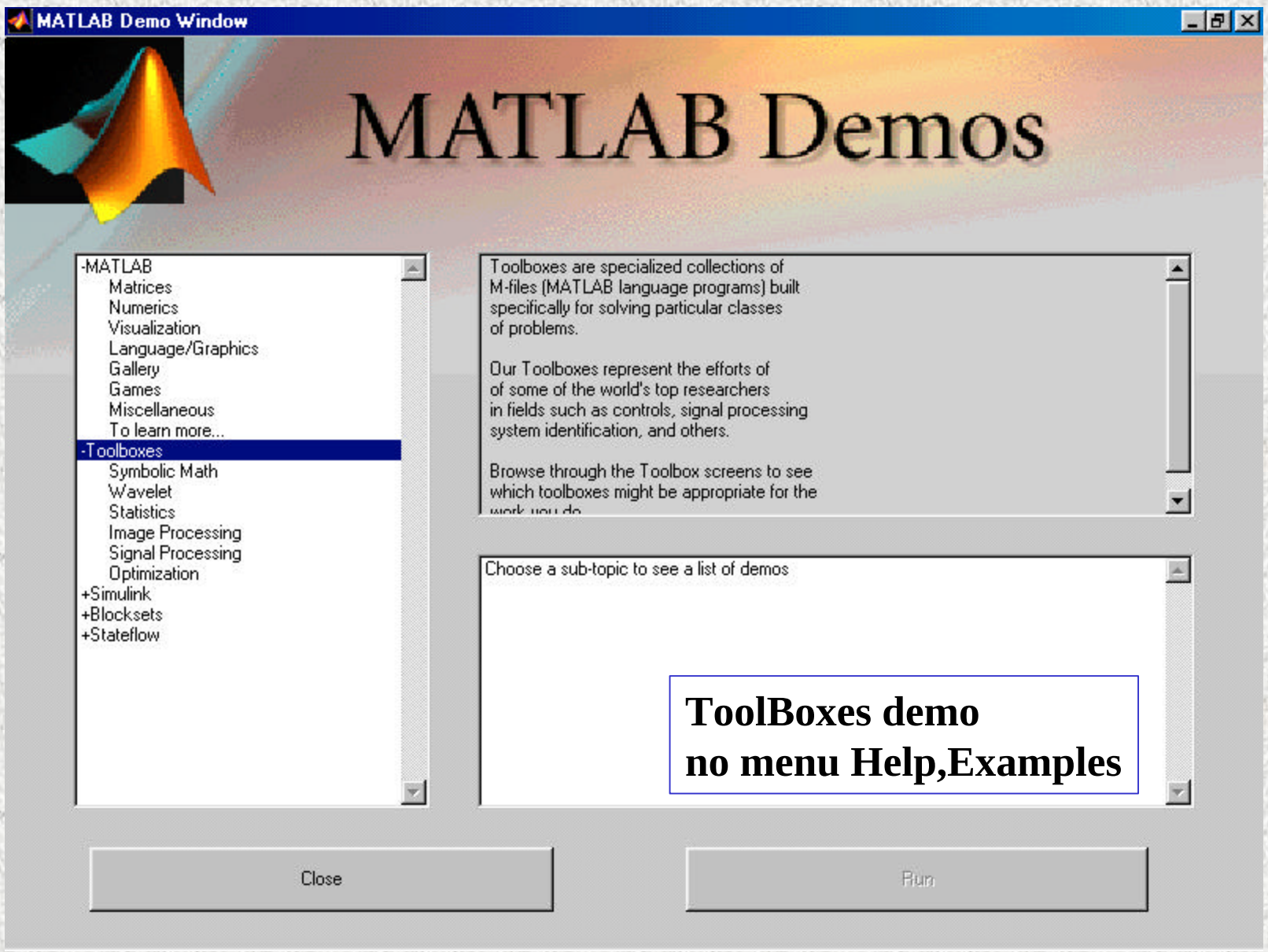
O M-file chamado *matrou1.m* estando gravado no subdiretório \work, pode-se executá-lo digitando o seu nome.

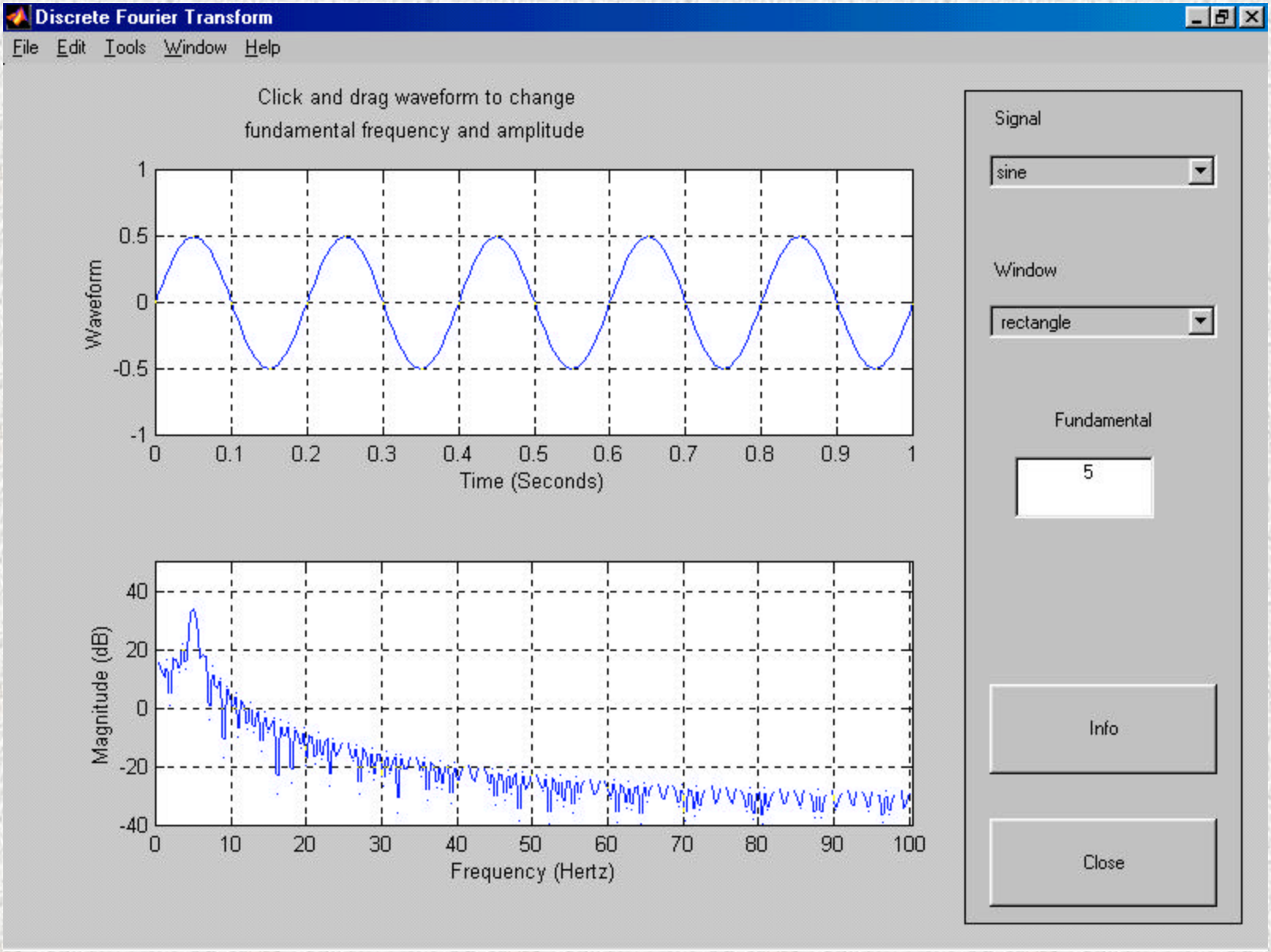



```
» matrou1

A =
    1.1000    2.2000    3.3000
    4.4000    5.5000    6.6000
    7.7000    8.8000    9.9000

ans =
 -4.0301e-015
```

ToolBoxes


MATLAB Editor/Debugger - [matrou1.m - C:\MATLABR11\work\matrou1.m]
File Edit View Debug Tools Window Help
Evaluate Selection F9
Auto Indent Selection Ctrl+I
Toolbar
Status Bar
Editor/Debugger
Path Browser
Workspace Browser
Stack:
A=[1.
7.7 8.8 9.9]
det(A
Path Browser
File Edit View Path Tools Help
Current Directory
C:\MATLABR11\work
Browse...
Path
C:\MATLABR11\toolbox\matlab\general
C:\MATLABR11\toolbox\matlab\ops
C:\MATLABR11\toolbox\matlab\lang
C:\MATLABR11\toolbox\matlab\elmat
C:\MATLABR11\toolbox\matlab\elfun
C:\MATLABR11\toolbox\matlab\specfun
C:\MATLABR11\toolbox\matlab\matfun
C:\MATLABR11\toolbox\matlab\datafun
C:\MATLABR11\toolbox\matlab\polyfun
C:\MATLABR11\toolbox\matlab\funfun
C:\MATLABR11\toolbox\matlab\sparfun
C:\MATLABR11\toolbox\matlab\graph2d
C:\MATLABR11\toolbox\matlab\graph3d
C:\MATLABR11\toolbox\matlab\specgraph
C:\MATLABR11\toolbox\matlab\graphics
C:\MATLABR11\toolbox\matlab\uitools
C:\MATLABR11\toolbox\matlab\strfun
Files in general
@char
private
addpath.m
binpatch.m
cd.m
clear.m
computer.m
Contents.m
copyfile.m
dbclear.m
dbcont.m
dbdown.m
dbmex.m
dbquit.m
dbstack.m
dbstatus.m
dbstep.m
dbstop.m


MATLAB Demo Window
MATLAB Demos
-MATLAB
Matrices
Numerics
Visualization
Language/Graphics
Gallery
Games
Miscellaneous
To learn more...
-Toolboxes
Symbolic Math
Wavelet
Statistics
Image Processing
Signal Processing
Optimization
+Simulink
+Blocksets
+Stateflow
Toolboxes are specialized collections of M-files (MATLAB language programs) built specifically for solving particular classes of problems.
Our Toolboxes represent the efforts of of some of the world's top researchers in fields such as controls, signal processing system identification, and others.
Browse through the Toolbox screens to see which toolboxes might be appropriate for the
Choose a sub-topic to see a list of demos
ToolBoxes demo
no menu Help,Examples
Close
Run


Discrete Fourier Transform
File Edit Tools Window Help
Click and drag waveform to change
fundamental frequency and amplitude
Waveform
Time (Seconds)
Magnitude (dB)
Frequency (Hertz)
Signal
sine
Window
rectangle
Fundamental
5
Info
Close

Número p (pi), função digits(), e função vpa()

```
» pi % número pi é pré-definido
ans =
   3.1416

» digits(32)  % aumenta número de dígitos significativos
» x=vpa(pi)   % função vpa() para considerar 32 dígitos
x =
 3.1415926535897932384626433832795

» 2*x
ans =
 6.2831853071795864769252867665590
```

comandos
format short e format long

após comando format long:
número de dígitos significativos
aumenta para cerca de 16
format short: restaura para cerca de 5

```
» format long
» realmin
ans =    2.225073858507201e-308
» realmax
ans =    1.797693134862316e+308
» format short
» realmin
ans =  2.2251e-308
» realmax
ans =  1.7977e+308
```

```
» format long
» pi
ans =3.14159265358979
» format short
» pi
ans =3.1416
» format long
» 1/3
ans =0.33333333333333
» format short e, 1/3
ans =3.3333e-001
» format long e, 1/3
ans =3.33333333333333e-001
```

O tipo básico do MatLab é uma matriz de números *complexos*

Uma matriz real é uma matriz que possui a parte imaginária de todos os elementos iguais a zero

Um vetor linha é uma matriz 1 por n

Um vetor coluna é uma matriz n por 1

Um escalar é uma matriz 1 por 1

O apóstrofe ' transpõe uma matriz

i ou j representa a raiz quadrada de -1:

```
» i
ans =       0 + 1.0000i
» j
ans =       0 + 1.0000i
» i*i
ans =    -1
» (1.1-2.2j)*(6.1+3j)
ans =  13.3100 -10.1200i
» (1.1-2.2j)*(6.1+3j)/(2+4.1j)
ans =  -0.7147 - 3.5950i
```

função	*calcula*
abs(z)	**valor absoluto (módulo) de valor complexo z**
angle(z)	**ângulo do complexo z**
conj(z)	**conjugado de z**

```
» z=2+1.4j
z =   2.0000 + 1.4000i
» abs(z), angle(z)
ans =    2.4413
ans =    0.6107
» cos(z), sin(z)
ans =  -0.8951 - 1.7316i
ans =   1.9558 - 0.7925i
```

```
» z=2+1.4j
z =   2.0000 + 1.4000i
» modulo=abs(z), ang=angle(z)
modulo =    2.4413
ang =    0.6107
» modulo*cos(ang), modulo*sin(ang)
ans =     2          % parte real
ans =    1.4000  % parte imaginária
```

```
» B=[1 2; 3 4], C=[5; 6], D=[7; 8; 9], E=[B C; D']

B =
     1     2
     3     4

C =
     5
     6

D =
     7
     8
     9

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

É possível blocar matrizes de dimensões compatíveis.

Cuidado **ao concatenar matrizes com os espaços em branco pois estes são equivalentes a vírgulas separando elementos.**

[1, 2+3]==[1 5] mas [1,2 +3]==[1 2 3]

Funções para gerar matrizes**:**

zeros(m,n)	**matriz m por n de zeros**
ones(m,n)	**matriz m por n de uns**
rand(m,n)	**matriz m por n de aleatórios em [0,1]**

```
» rand(3,5)

ans =
    0.9501    0.4860    0.4565    0.4447    0.9218
    0.2311    0.8913    0.0185    0.6154    0.7382
    0.6068    0.7621    0.8214    0.7919    0.1763
```

A(i,j) é o elemento na linha i e coluna j da matriz A
i=1,2,…m e j=1,2,…n (índ. começa com 1)
A(i,:) é a i-ésima linha
A(:,j) é a j-ésima coluna
Vetor i:p:s é o vetor [i, i+p, i+2p, … i+np] com i+np £ s

```
2:2:14
ans =    2    4    6    8   10   12   14
» 2:2:15
ans =    2    4    6    8   10   12   14
» 2:2:16
ans =    2    4    6    8   10   12   14   16
```

Matriz de 3 ou mais índices (só depois da versão 5.0)

Função cat(): concatena matrizes bi-dimensionais para formar tri-

```
» a1=[1 1; 0 0]
a1 =
     1     1
     0     0
» a2=[0 0; 2 2]
a2 =
     0     0
     2     2
» a3=[0 3; 3 0]
a3 =
     0     3
     3     0
```

```
» A=cat(3,a1,a2,a3)
A(:,:,1) =
     1     1
     0     0
A(:,:,2) =
     0     0
     2     2
A(:,:,3) =
     0     3
     3     0
```

matriz a1 c/ índ. 1

matriz a2 c/ índ. 2

matriz a3 c/ índ. 3

```
» A(1,2,3)
ans =     3
» A(1,1,3)
ans =     0
```

diag(A) **é o vetor diagonal principal de A, se A é uma matriz quadrada**

eye(n) **é a matriz identidade n por n**

```
» A=[1 2; 3 4], diag(A)

A =
     1     2
     3     4

ans =

     1
     4

» eye(4)

ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Se vetlin e vetcol são vetores de índices, A(vetlin,vetcol) é a submatriz de A com índices de linha em vetlin e índices de coluna em vetcol

```
A =
    1.1000    2.2000    3.3000
    4.4000    5.5000    6.6000
    7.7000    8.8000    9.9000

» vetlin=[3 1],vetcol=[2 3], A(vetlin, vetcol)
vetlin =
     3     1
vetcol =
     2     3

ans =
    8.8000    9.9000
    2.2000    3.3000
```

+ - * / ^ **são as operações de soma, subtração, produto, divisão e potência sobre matrizes**

\ **é divisão à esquerda; x=A\b fornece solução de Ax=b**

```
» d=2\8, c=8/2
d =     4
c =     4
```

```
A =
    9.1000    2.2000    3.3000
    4.4000    9.5000    6.6000
    7.7000    8.8000    9.9000
» b=[1 2 3], x=A\b', A*x
b =
     1     2     3
x =
    0.0000
    0.0000
    0.3030
ans =
    1.0000
    2.0000
    3.0000
```

inv(A)	**inversa da matriz A**
.* .^ ./ .	**versão pontual das operações** **prod, pot div à dir, div à esq** **i.e., op executada elemento e elemento**

```
» a=[3 3 3], b=[-1 0 1], c=a.*b

a =
     3     3     3

b =
    -1     0     1

c =
    -3     0     3
```

A=22 define A como matriz 1 por 1

A(2,3)=44 redefine como matriz 2 por 3

A(:,2)=[] elimina coluna 2 pois [] denota matriz vazia

```
» A=22

A =
    22

» A(2,3)=44

A =
    22     0     0
     0     0    44

» A(:,2)=[]

A =
    22     0
     0    44
```

Para executar qualquer comando MAPLE

>> maple(' comando-maple; ')

```
» maple('f:=x +2*t*y -t -2*t^3')

ans =

f := x+2*t*y-t-2*t^3

» maple('g:=diff(f,t)')

ans =

g := 2*y-1-6*t^2

» maple('h:=solve(g,y)')

ans =

h := 1/2+3*t^2
```

Formato dos valores simbólicos - função sym()

Função sym permite escolher um formato:

'r' de racional (default)

'f' de flutuante: produto de algarismos hexadecimais com potência de 2

'e' de estimativa de erro

'd' de decimal

```
» x=sym(1/3,'r')
  x =1/3
 » x=sym(1/3,'f')
   x ='1.5555555555555'*2^(-2)
 » x=sym(1/3,'e')
  x =1/3-eps/12
 » x=sym(1/3,'d')
   x =.33333333333333314829616256247390
```

variável pré-definida
eps=2.2204e-016

operadores relacionais são

< <= > >= == ~=

que retornam 0 (falso) ou 1 (verdadeiro)

operadores unários

~	**é não**
&	**é E**
|	**é OU**

```
» resp=22<=100
resp =     1

» t=resp>4
t =     0

» u= (t==0)&(resp ~= 0)
u =     1
```

Funções de data e hora

datestr() resulta uma string

```
» t=now;
» datestr(t,2) % mm/dd/aa
ans =10/14/00
» datestr(t,5) % mm
ans =10
» datestr(t,7) % dd
ans =14
» datestr(t,10) % aaaa
ans =2000
» datestr(t,15) %HH:MM
ans =12:38
```

Ver: help datestr

clock() resulta um vetor de inteiros

```
» T=clock, ano=T(1),mes=T(2),dia=T(3),hora=T(4),min=T(5),seg=T(6)
T =  1.0e+003 *
   2.0000    0.0100    0.0140    0.0120    0.0590    0.0055
ano =       2000
mes =    10
dia =    14
hora =    12
min =    59
seg =    5.5400
```

weekday() resulta dia da semana

```
» weekday(now)
ans =     7
» weekday('21-Dec-1994')
ans =     4
```

eomday() = último dia do mês

```
» eomday(1999,3)
ans =    31
```

```
» cal=calendar(1999,3)
cal =    0    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   26   27
    28   29   30   31    0    0    0
     0    0    0    0    0    0    0
```

```
» calendar(1999,3)
                Mar 1999
   S    M   Tu    W   Th    F    S
   0    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   26   27
  28   29   30   31    0    0    0
   0    0    0    0    0    0    0
```