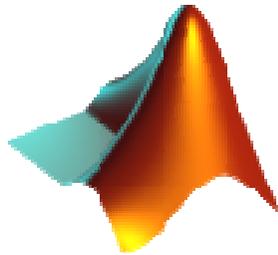


# Aula 1 - Introdução ao MatLab

O que é MatLab?



- MatLab (=“Matrix laboratory”) é um *ambiente* para computação científica e técnica desenvolvido pela empresa MathWorks.
- O ambiente integra uma **linguagem de programação** (de alto nível, tipo script) e **ferramentas de visualização** (gráficos, etc.) além de **pacotes computacionais** pré-instalados (bibliotecas, pacotes de manipulação simbólica, etc.).

# Abrindo o MatLab: Elementos básicos

The screenshot shows the MATLAB 7.8.0 (R2009a) environment with several windows and annotations:

- Current Directory:** Shows the file explorer with a folder named "IntroducaoMatLab" modified on 2/15/16 10:01 AM.
- Command Window:** Contains the following commands and outputs:

```
>> 2+2
ans =
    4

>> 4/5
ans =
    0.8000

>> sqrt(2)
ans =
    1.4142

>> exp(1)
ans =
    2.7183

fx >>
```
- Workspace:** Shows a table with one variable:

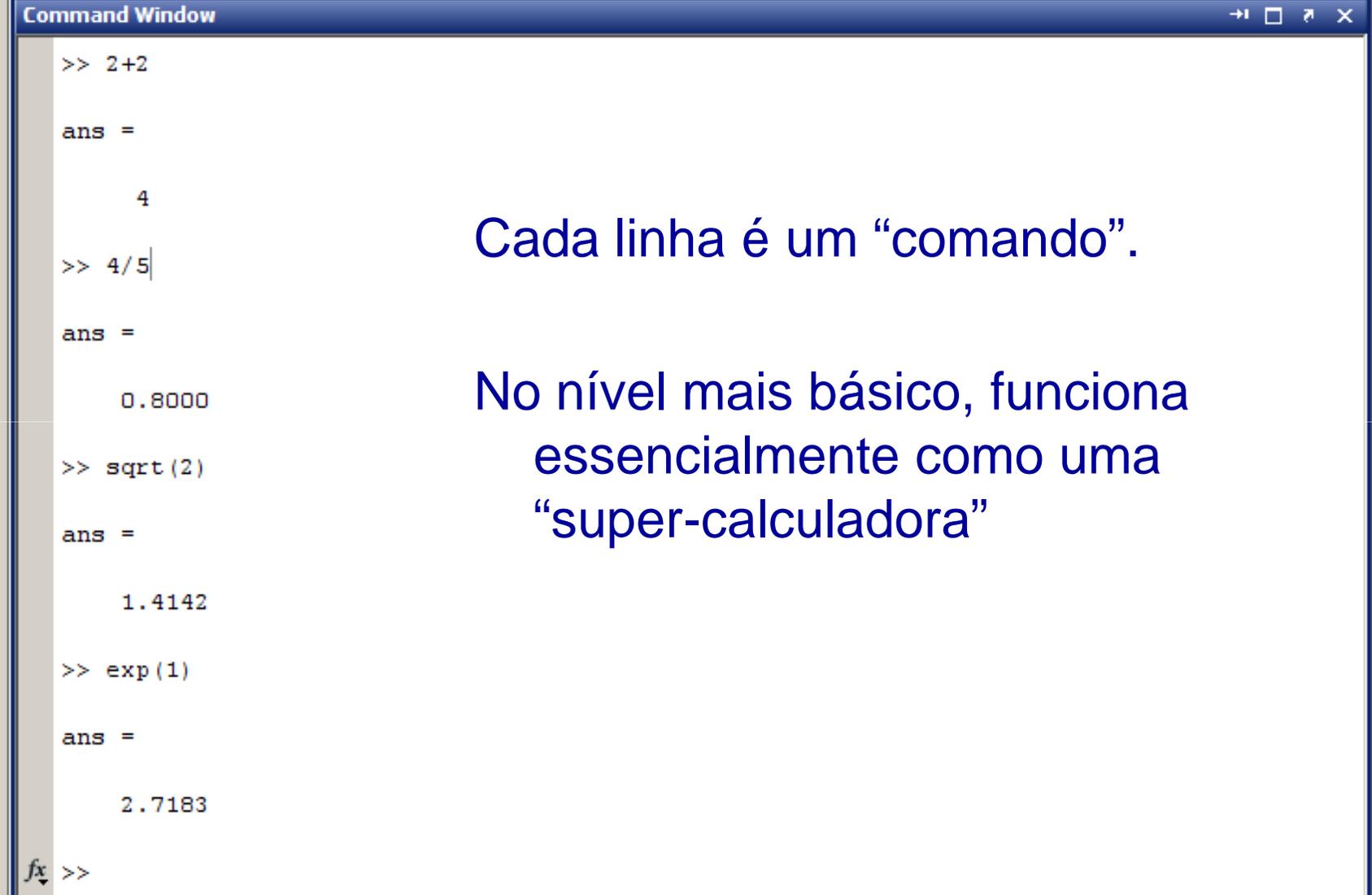
Name	Value
ans	2.7183
- Command History:** Lists the following commands:

```
Vetor=1:10:1
Vetor=1:2:10
Vetor=0:2:10
Vetor=0:1:10
2/11/16 4:44 PM --%
Vetor=0:1:10
Vetor2=0:2:10
1+1
2/11/16 5:37 PM --%
2/15/16 10:00 AM --%
1+1
sqrt(2)
Exp(1)
exp(1)
clear
2+2
4/5
sqrt(2)
exp(1)
```

Annotations in red text:

- Diretório:** Points to the Current Directory window.
- Janela de comando:** Points to the Command Window.
- Variáveis:** Points to the Workspace window.
- Histórico:** Points to the Command History window.

# Básico do MatLab: Janela de comando

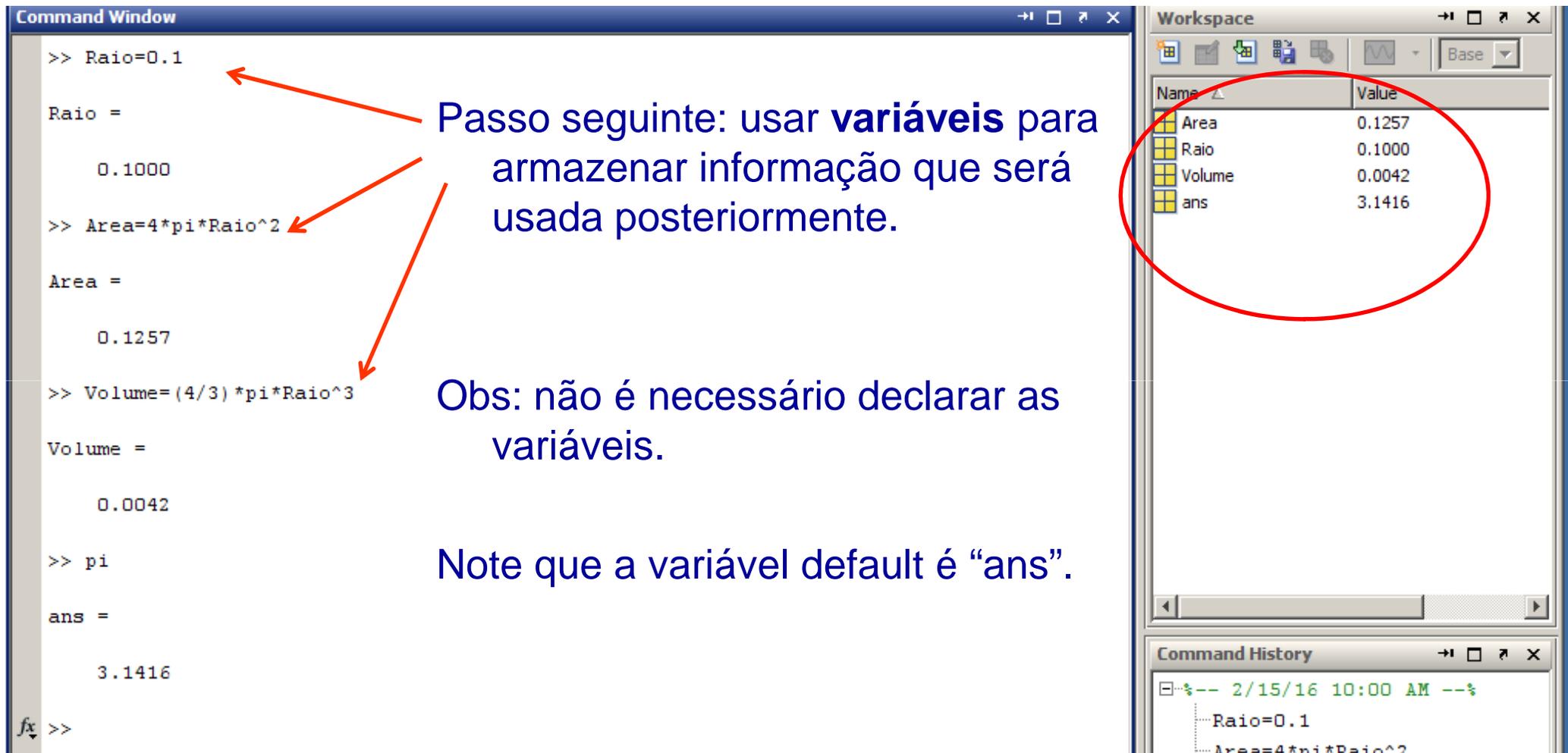


```
Command Window
>> 2+2
ans =
    4
>> 4/5
ans =
    0.8000
>> sqrt(2)
ans =
    1.4142
>> exp(1)
ans =
    2.7183
fx >>
```

Cada linha é um “comando”.

No nível mais básico, funciona essencialmente como uma “super-calculadora”

# Básico do MatLab: Variáveis



The image shows a MATLAB Command Window on the left and a Workspace window on the right. The Command Window contains the following code and output:

```
>> Raio=0.1
Raio =
    0.1000
>> Area=4*pi*Raio^2
Area =
    0.1257
>> Volume=(4/3)*pi*Raio^3
Volume =
    0.0042
>> pi
ans =
    3.1416
fx >>
```

The Workspace window shows the following variables and their values:

Name	Value
Area	0.1257
Raio	0.1000
Volume	0.0042
ans	3.1416

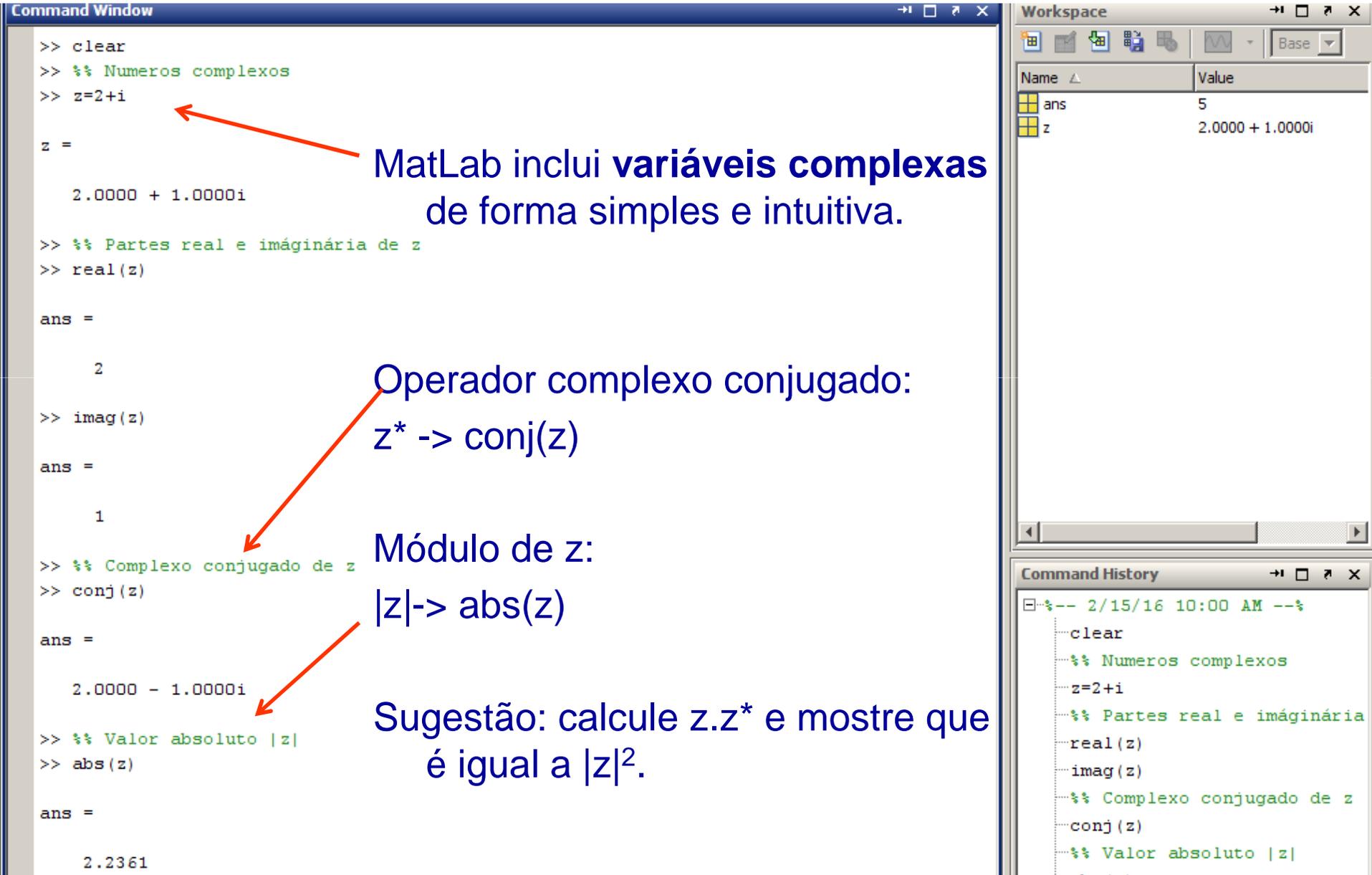
Annotations in the image include red arrows pointing from the text to the corresponding code lines in the Command Window. A red circle highlights the Workspace window.

Passo seguinte: usar **variáveis** para armazenar informação que será usada posteriormente.

Obs: não é necessário declarar as variáveis.

Note que a variável default é "ans".

# Básico do MatLab: Variáveis complexas



The screenshot shows the MATLAB Command Window and Workspace. The Command Window contains the following code and output:

```
>> clear
>> %% Numeros complexos
>> z=2+i

z =

    2.0000 + 1.0000i

>> %% Partes real e imaginária de z
>> real(z)

ans =

     2

>> imag(z)

ans =

     1

>> %% Complexo conjugado de z
>> conj(z)

ans =

    2.0000 - 1.0000i

>> %% Valor absoluto |z|
>> abs(z)

ans =

    2.2361
```

The Workspace window shows the following variables:

Name	Value
ans	5
z	2.0000 + 1.0000i

The Command History window shows the following commands:

```
clear
%% Numeros complexos
z=2+i
%% Partes real e imaginária de z
real(z)
imag(z)
%% Complexo conjugado de z
conj(z)
%% Valor absoluto |z|
abs(z)
```

Annotations in the image:

- An arrow points from the text "MatLab inclui **variáveis complexas** de forma simples e intuitiva." to the line `z=2+i` in the Command Window.
- An arrow points from the text "Operador complexo conjugado:  $z^* \rightarrow \text{conj}(z)$ " to the line `conj(z)` in the Command Window.
- An arrow points from the text "Módulo de z:  $|z| \rightarrow \text{abs}(z)$ " to the line `abs(z)` in the Command Window.
- An arrow points from the text "Sugestão: calcule  $z \cdot z^*$  e mostre que é igual a  $|z|^2$ ." to the line `abs(z)` in the Command Window.

# Usando o "Help" do MatLab

The screenshot shows the MATLAB 7.8.0 (R2009a) environment. The main window displays the 'Current Directory' and 'Command Window'. The 'Help' window is open, showing search results for 'complex numbers'. The 'Help Navigator' pane on the left lists search results, with 'Working with Complex Numbers' selected. The main content area displays the title 'Embedded MATLAB Coding Style :: Working with the Embedded MATLAB Subset (Embedded MATLAB™)' and the section 'Working with Complex Numbers'. The text explains that the Embedded MATLAB subset supports complex numbers and operations, and provides examples of how to create local complex variables by assignment. The examples include:

```
x = 5 + 6i; % x is a complex number by assignment.  
y = 7 + 8j; % y is a complex number by assignment.  
x = complex(5,6); % x is the complex number 5 + 6i.
```

The text also discusses how to specify and use complex variables in Embedded MATLAB functions, providing an example of a variable declaration and assignment:

```
x = 1 + 2i; % x is declared a complex variable.  
y = int16(x); % Real and imaginary parts of y are int16.  
x = 3; % x now has the value 3 + 0i.
```

Conflicts can occur from operations with real operands that can have complex results. For example, the following code generates an error:

```
z = 3; % Sets type of z to double (real)  
z = 3 + 2i; % ERROR: cannot recast z to complex
```

The following is a possible workaround that you can use if you know that a variable can be assigned a complex number:

```
m = complex(3); % Sets m to complex variable of value 3 + 0i  
m = 5 + 6.7i; % Assigns a complex result to a complex number
```

Finally, it notes that cases in which a function can return a complex number for a real argument are handled individually for each

Annotations on the image include:

- A red circle around the 'Help' menu item in the top toolbar, with an arrow pointing to it.
- A red circle around the 'Help Navigator' search bar, with an arrow pointing to it.
- The text 'ou digite "F1"' in red, pointing to the Help icon.
- The text 'Pesquise!' in red, pointing to the search bar.
- The text 'Vários exemplos' in red, pointing to the examples in the Help window.

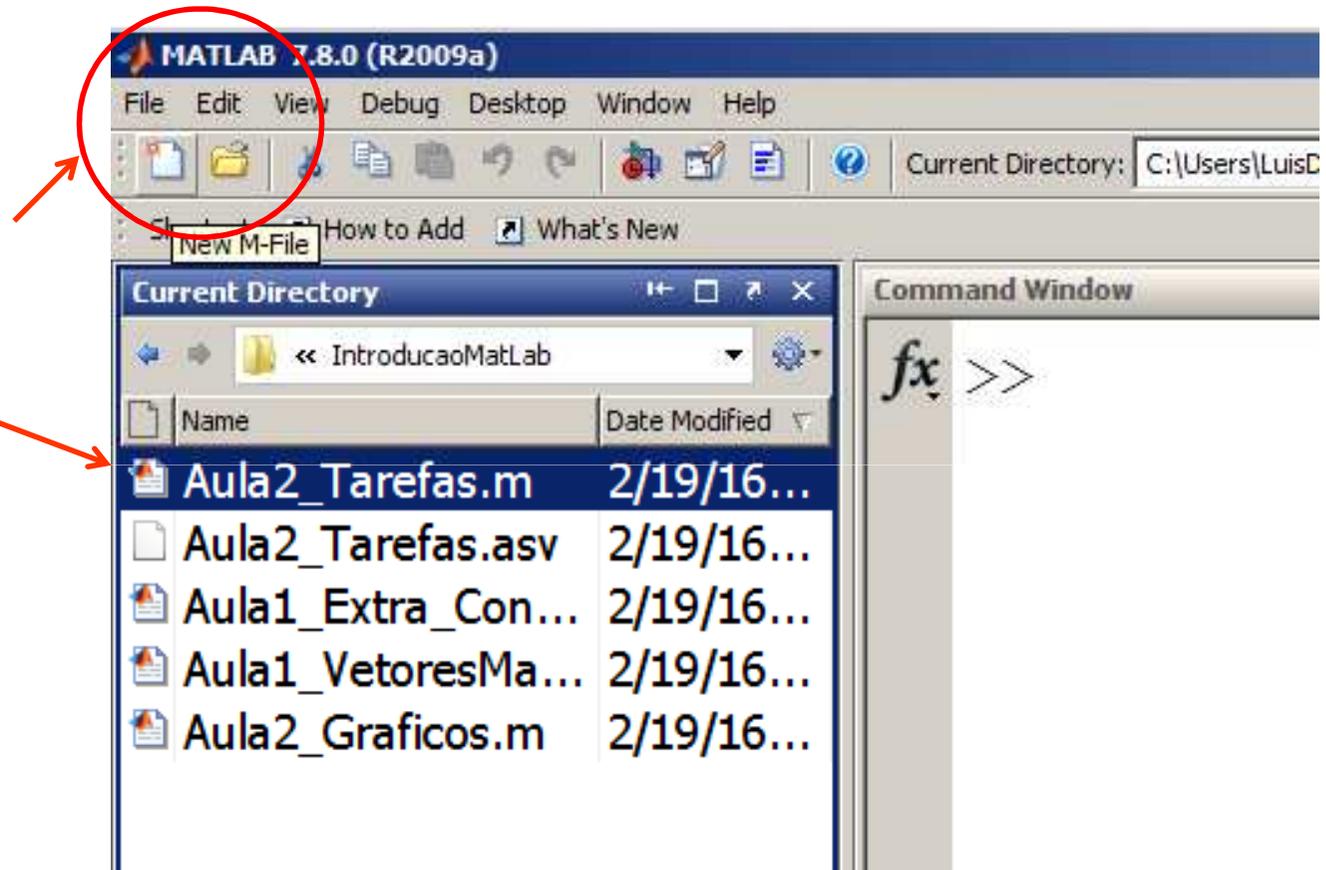
# Tarefas da aula 1

Lembrando: toda aula haverá tarefas!! (20% da média final!!!)

- Tarefa 1: Calcule o seno, o cosseno, a tangente, a raiz quadrada e a raiz cúbica de  $\pi/9$ .
- Tarefa 2: Dados dois números complexos  $z_1=5+8i$  e  $z_2=6+4i$ , calcule:
  - $z_1 \cdot z_2$
  - $|z_1+z_2|^2$
  - o ângulo de fase de  $z_1+z_2$ .

# Usando “m-files” para gravar as Tarefas.

Na prática do dia-a-dia, é mais conveniente utilizar arquivos texto (com extensão “.m”) para passar os comandos ao invés de usar a janela de comando.

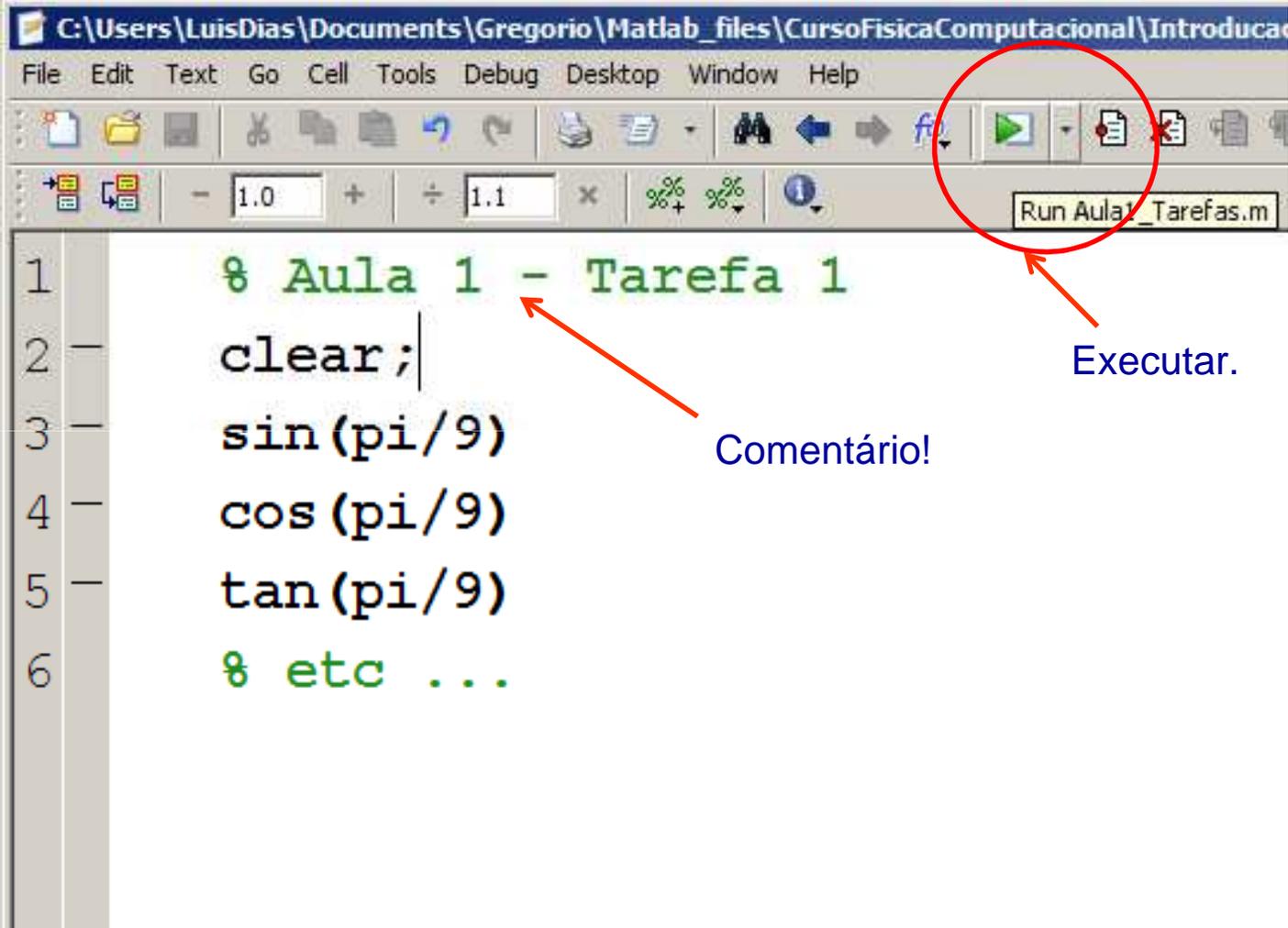


Os arquivos “.m” são mostrados no diretório corrente.

# Usando “m-files”.

Digitamos os comandos (um por linha) e depois executamos a sequência “de uma vez”.

Caso ocorra algum erro, podemos corrigir diretamente no arquivo e executar novamente.



The screenshot shows the MATLAB IDE interface. The title bar indicates the file path: C:\Users\LuisDias\Documents\Gregorio\Matlab\_files\CursoFisicaComputacional\Introducao. The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations and execution. A red circle highlights the 'Run' button (a green play icon) and the 'Run' button in the Command Window (labeled 'Run Aula1\_Tarefas.m'). An arrow points from the text 'Executar.' to this button. The script editor shows the following code:

```
1 % Aula 1 - Tarefa 1
2 clear;
3 sin(pi/9)
4 cos(pi/9)
5 tan(pi/9)
6 % etc ...
```

An arrow points from the text 'Comentário!' to the first line of code, which is a comment. The Command Window at the bottom is empty.

# Usando “m-files”.

Variáveis armazenadas

Os resultados dos comandos (aqueles sem “;” no final da linha) aparecem na janela de comando.

```
>>
ans =
0.3420
ans =
0.9397
ans =
0.3640
```

```
% Aula 1 - Tarefa 1
clear;
sin(pi/9)
cos(pi/9)
tan(pi/9)
etc ...
```

Name	Value
ans	0.3640

O último resultado fica armazenado na variável “ans”.

# Básico do MatLab: Vetores e Matrizes

“**Tudo** (ou quase tudo) no MatLab são matrizes”

The image shows a MATLAB Command Window and Workspace. The Command Window displays the following code and output:

```
>> %% Usando "%" antes do comando escrevemos um comentário (será ignorado...)
>> %% "clear" limpa as variáveis e etc.
>> %% Definindo um vetor (array)
>> Vetor=[0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5]

Vetor =

Columns 1 through 8
    0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000

Columns 9 through 11
    4.0000    4.5000    5.0000

>> %% Acessando o i-ésimo elemento do vetor: Vetor(i)
>> Vetor(7)

ans =

    3

>> Vetor(10)

ans =

    4.5000
```

The Workspace window shows the following variables:

Name	Value
Vetor	<1x11 double>
ans	4.5000

The Command History window shows the following commands:

```
clear
%% Um vetor em 1D (array)
clear
%% Usando "%" antes do comando escrevemos um comentário (será ignorado...)
%% Definindo um vetor (array)
Vetor=[0 1 2 3 4 5 6 7 8 9]
%% Acessando o i-ésimo elemento do vetor: Vetor(i)
```

Annotations in the image:

- An arrow points from the text "Vetores: criados com “[ ]” e espaços entre os elementos." to the code `Vetor=[0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5]`.
- An arrow points from the text "Acessamos o i-ésimo elemento do vetor com “( )”" to the code `Vetor(7)`.
- Another arrow points from the same text to the code `Vetor(10)`.

# Criando vetores com o operador ":"

The screenshot shows the MATLAB Command Window and Workspace. The Command Window displays the following code and output:

```
>> clear
>> %% O Operador ":" nos permite gerar vetores com valores igualmente espaçados
>> %% de forma mais prática.
>> %% Por exemplo:
>> Vetor2=0:0.5:5

Vetor2 =

Columns 1 through 8
    0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000

Columns 9 through 11
    4.0000    4.5000    5.0000

>> %% Notacao: Vetor=Inicial:Passo:Final
>> Vetor2(7)

ans =

    3

>> Vetor2(10)

ans =

    4.5000
```

The Workspace window shows the following variables:

Name	Value
Vetor2	<1x11 double>
ans	4.5000

The Command History window shows the following commands:

```
clear
%% Usando "%" antes do com
clear
%% Usando "%" antes do com
%% "clear" limpa as variáv
%% Definindo um vetor (arr
Vetor=[0 0.5 1 1.5 2 2.5
%% Acessando o i-ésimo ele
```

**Vetores regulares:** podem ser criados com o operador ":"  
"Vetor=Inicial:Passo:Final"

Obs: se o passo for omitido, o MatLab assume "1".

**Acessamos o i-ésimo elemento do vetor com "("**

# Tarefas da aula 1 (cont).

Lembrando: toda aula haverá tarefas!! (20% da média final!!!)

- Tarefa 3: Gere uma sequência de números pares com início em 4 e término em 100.
- Tarefa 4: Gere uma sequência que comece em  $-\pi$  e acabe em  $+\pi$  com um passo de  $\pi/15$ .