



INSTITUTO FEDERAL
Espírito Santo
Campus Serra

Engenharia de Controle e Automação

Cálculo Numérico

Aula 2 - Introdução ao Octave/Matlab

Prof. Hilário Tomaz Alves de Oliveira

hilario.oliveira@ifes.edu.br

Agenda

- Introdução
 - Matlab
 - Octave
- Octave
 - Instalação
 - Comandos Básicos
 - Scripts
 - Funções
 - Programação Estruturada

Lista de Exercícios

- **Lista de Implementação 1** – Introdução ao Octave.
 - Disponível no Ambiente Virtual.
- Data Entrega: **22/03/2020** até as **23:55**.

Matlab

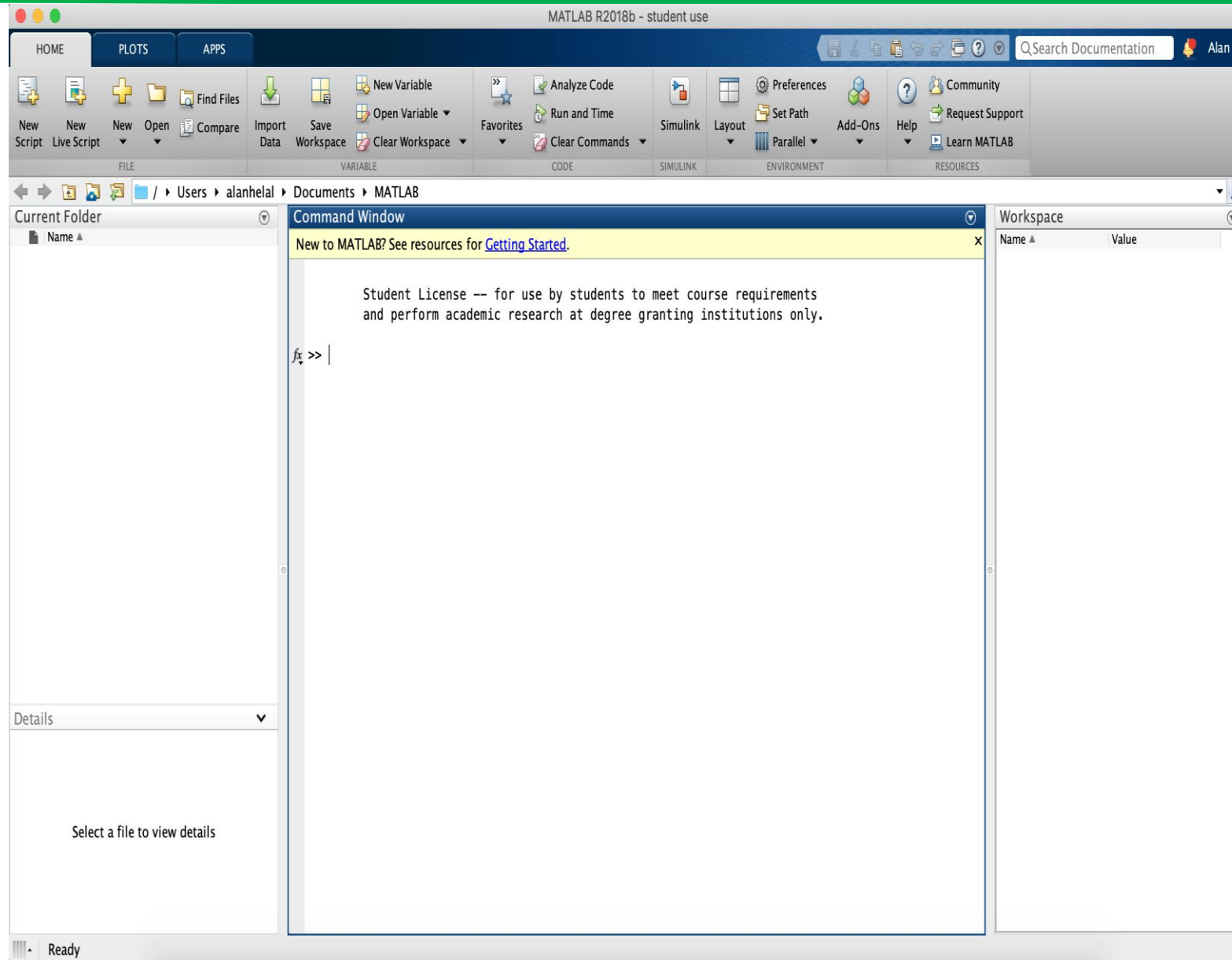
- O **MATLAB®** é uma plataforma de programação projetada especificamente para engenheiros e cientistas.
- O MATLAB possui recursos de programação (semelhante a linguagem C), porém voltada para processamento numérico intensivo.



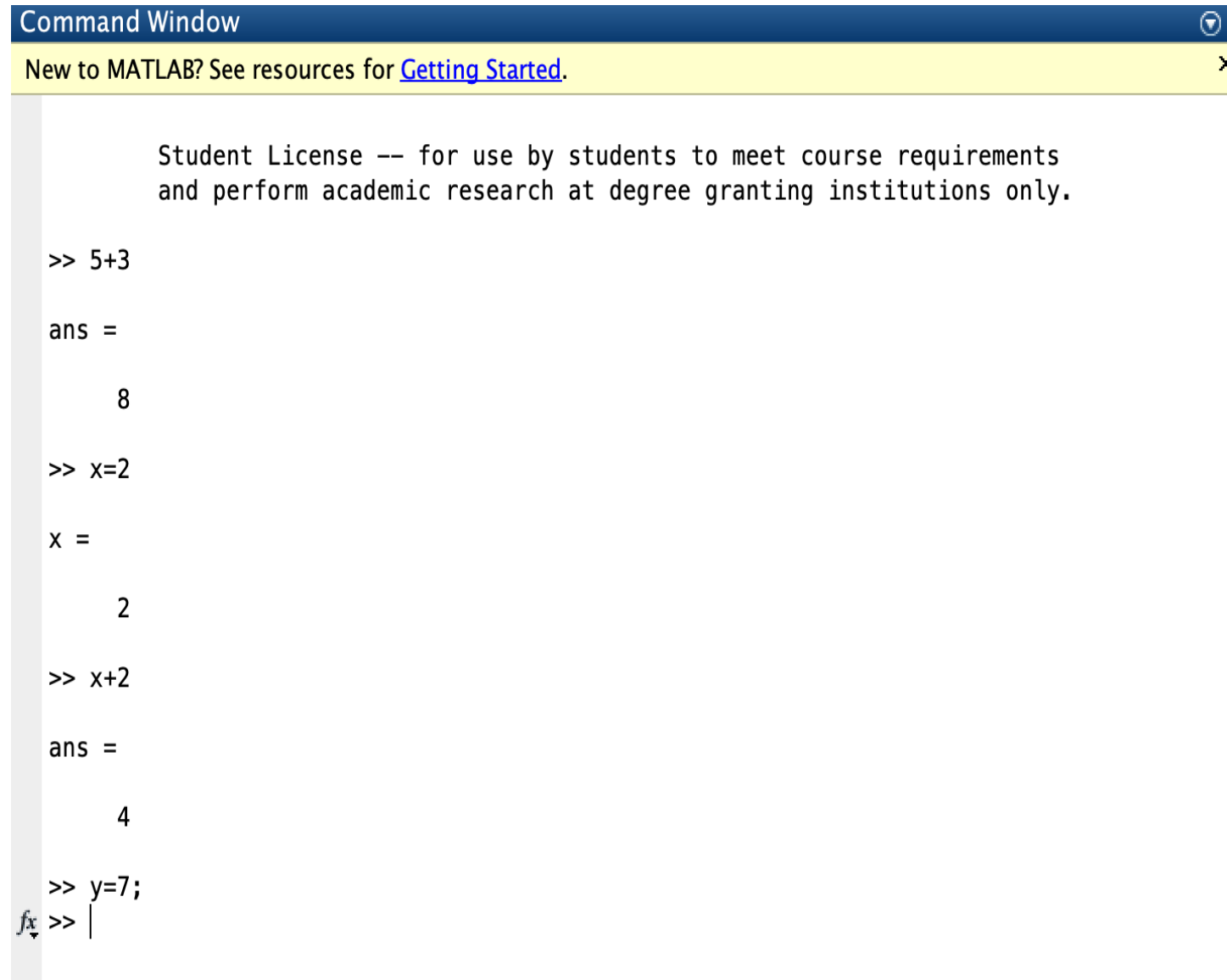
Matlab

- O **Matlab** pode ser adquirido diretamente do site da MathWorks.
 - Site: <https://www.mathworks.com/products/matlab.html>
- Preços:
 - Student Suite → **USD 55.00**
 - Home → **USD 95**
 - Standard → **USD 940 (Anual)**
- Possui uma versão de avaliação de 30 dias.

Matlab – Tela Principal



Matlab – Tela Principal



```
Command Window
New to MATLAB? See resources for Getting Started.

Student License -- for use by students to meet course requirements
and perform academic research at degree granting institutions only.

>> 5+3

ans =

     8

>> x=2

x =

     2

>> x+2

ans =

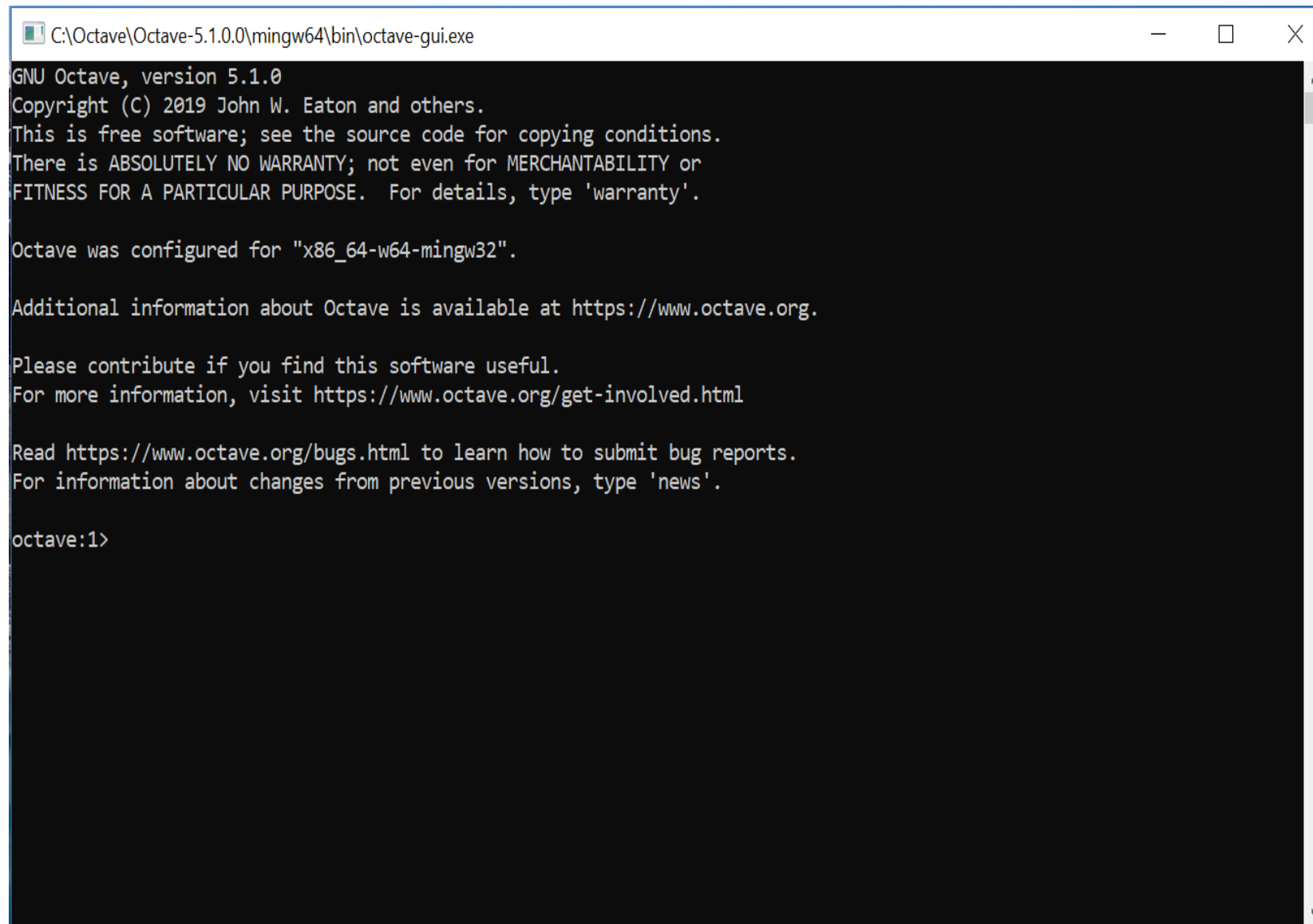
     4

>> y=7;
fx >> |
```


Octave

- Octave é um "Matlab de código aberto".
 - Site: <https://www.gnu.org/software/octave/>
- O GNU Octave é uma linguagem de alto nível, destinada principalmente a cálculos numéricos.
 - Ele fornece uma interface gráfica ou via linha de comando conveniente para realizar processamento numérico.
 - A linguagem é em grande parte compatível com o MatLab.

Octave – Linha de Comando



The image shows a screenshot of the GNU Octave 5.1.0 command-line interface (CLI) window. The window title bar indicates the path to the executable: `C:\Octave\Octave-5.1.0.0\mingw64\bin\octave-gui.exe`. The main content area is a black terminal window with white text. It displays the GNU Octave version (5.1.0), copyright information (© 2019 John W. Eaton and others), and a disclaimer stating that the software is free but provided without warranty. It also mentions the configuration for "x86_64-w64-mingw32". Several URLs are provided for more information, including the Octave website, a page for getting involved, and a page for submitting bug reports. The prompt `octave:1>` is visible at the bottom of the terminal window.

```
C:\Octave\Octave-5.1.0.0\mingw64\bin\octave-gui.exe
GNU Octave, version 5.1.0
Copyright (C) 2019 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

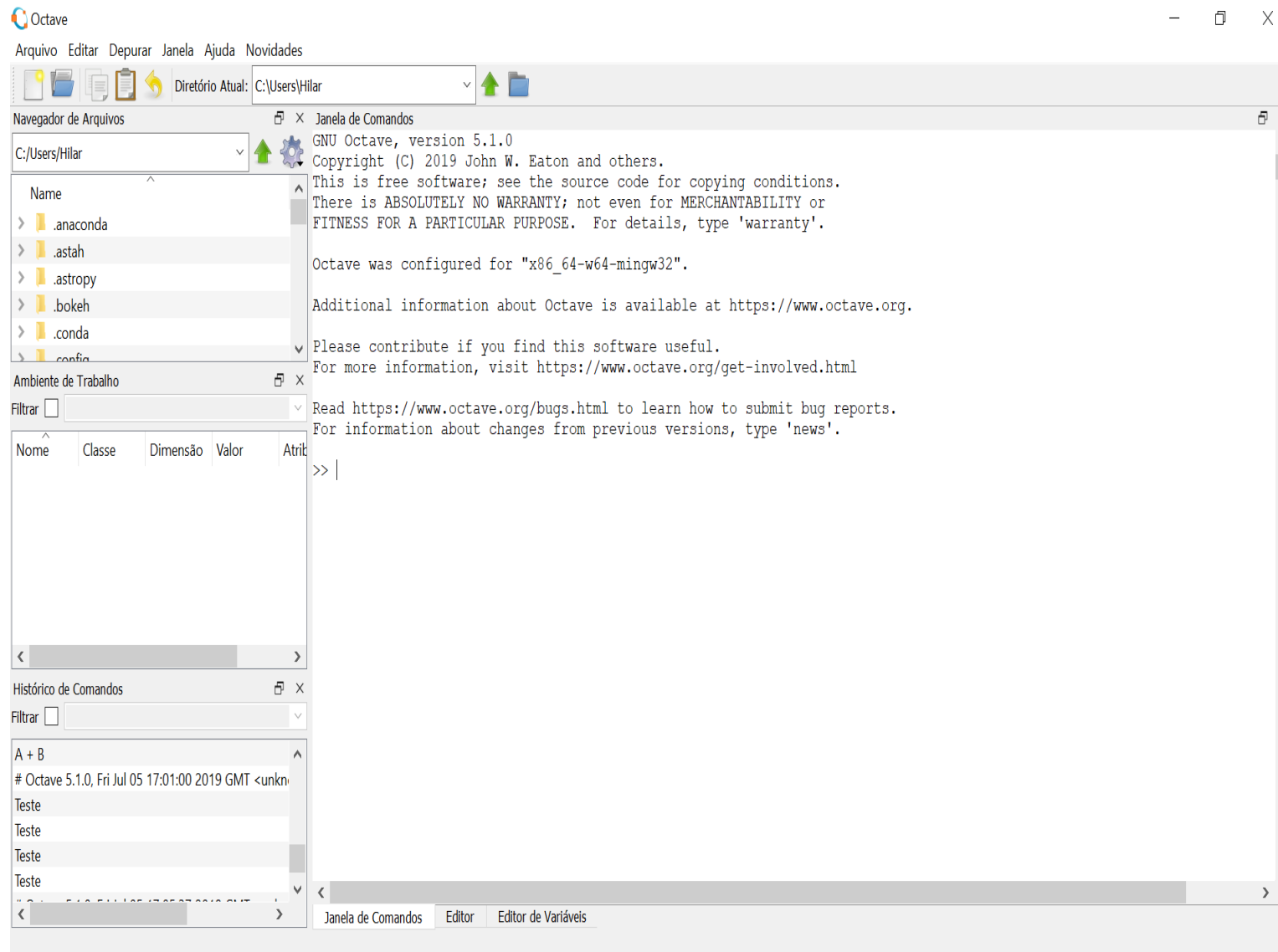
Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

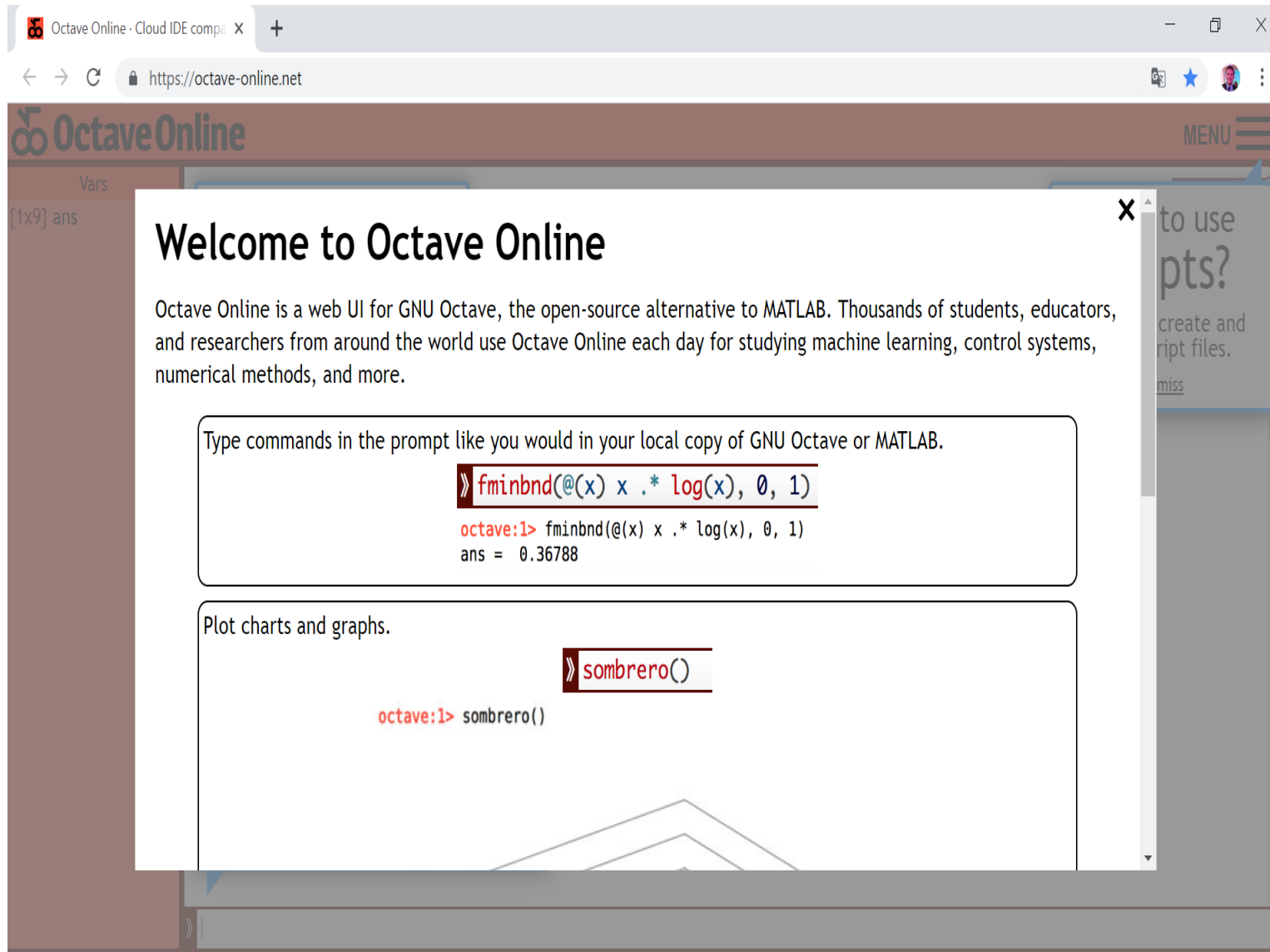
Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

octave:1>
```

Octave – Interface Gráfica



Octave – Versão Online



The screenshot shows a web browser window with the URL `https://octave-online.net`. The page features the Octave Online logo and a navigation menu. A central white box contains the following text:

Welcome to Octave Online

Octave Online is a web UI for GNU Octave, the open-source alternative to MATLAB. Thousands of students, educators, and researchers from around the world use Octave Online each day for studying machine learning, control systems, numerical methods, and more.

Type commands in the prompt like you would in your local copy of GNU Octave or MATLAB.

```
» fminbnd(@(x) x .* log(x), 0, 1)
octave:1> fminbnd(@(x) x .* log(x), 0, 1)
ans = 0.36788
```

Plot charts and graphs.

```
» sombrero()
octave:1> sombrero()
```

On the right side of the interface, there is a sidebar with a "MENU" button and a partially visible text box that says "to use pts?" and "create and ript files."

Octave e Matlab

- O **Octave** e o **Matlab** são ambos **linguagens de alto nível e ambientes de programação** matemática para:
 - Visualização;
 - Programação em geral;
 - **Computação numérica**: álgebra linear, otimização, controle, estatísticas, processamento de sinal e imagem, etc.

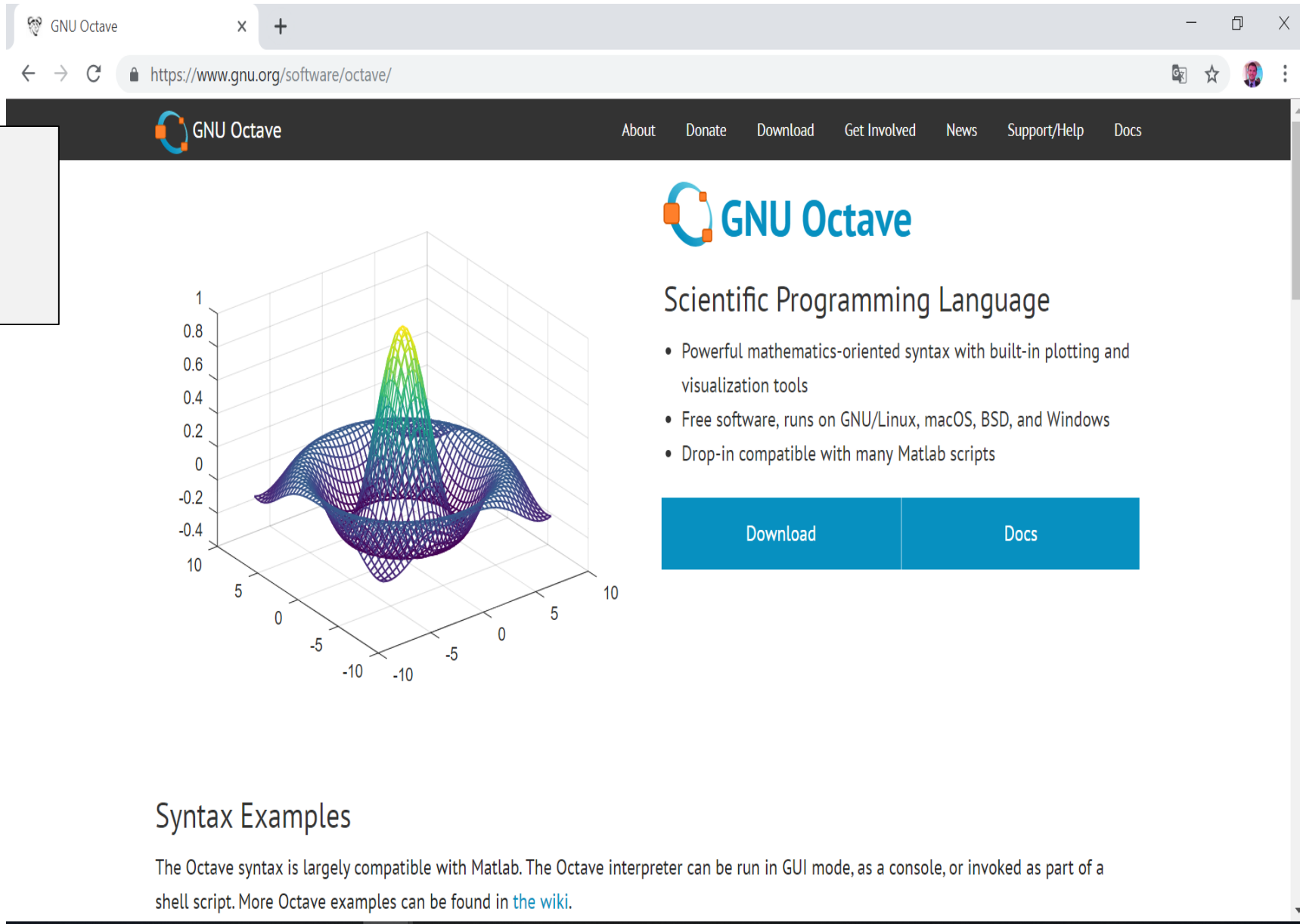
Octave e Matlab

- Comparação **Matlab** e **Octave**:
 - Matlab é mais flexível/avançado/poderoso/dispendioso.
 - Octave é gratuito (licença **GPL**).
 - Existem pequenas diferenças na sintaxe.
- **OBS:** O **Octave**/**Matlab** podem ser lentos.

Octave - Instalação

Octave – Passo 1 Instalação

1. Entrar no site do Octave.



The screenshot shows the GNU Octave website in a web browser. The browser's address bar displays the URL <https://www.gnu.org/software/octave/>. The website's header includes the GNU Octave logo and navigation links: About, Donate, Download, Get Involved, News, Support/Help, and Docs. The main content area features a 3D wireframe plot of a mathematical function, a large blue button labeled 'Download', and a smaller blue button labeled 'Docs'. Below the buttons, the text 'Scientific Programming Language' is displayed, followed by a bulleted list of features: 'Powerful mathematics-oriented syntax with built-in plotting and visualization tools', 'Free software, runs on GNU/Linux, macOS, BSD, and Windows', and 'Drop-in compatible with many Matlab scripts'. At the bottom of the page, the 'Syntax Examples' section begins with the text: 'The Octave syntax is largely compatible with Matlab. The Octave interpreter can be run in GUI mode, as a console, or invoked as part of a shell script. More Octave examples can be found in [the wiki](#).'

GNU Octave

About Donate Download Get Involved News Support/Help Docs

GNU Octave

Scientific Programming Language

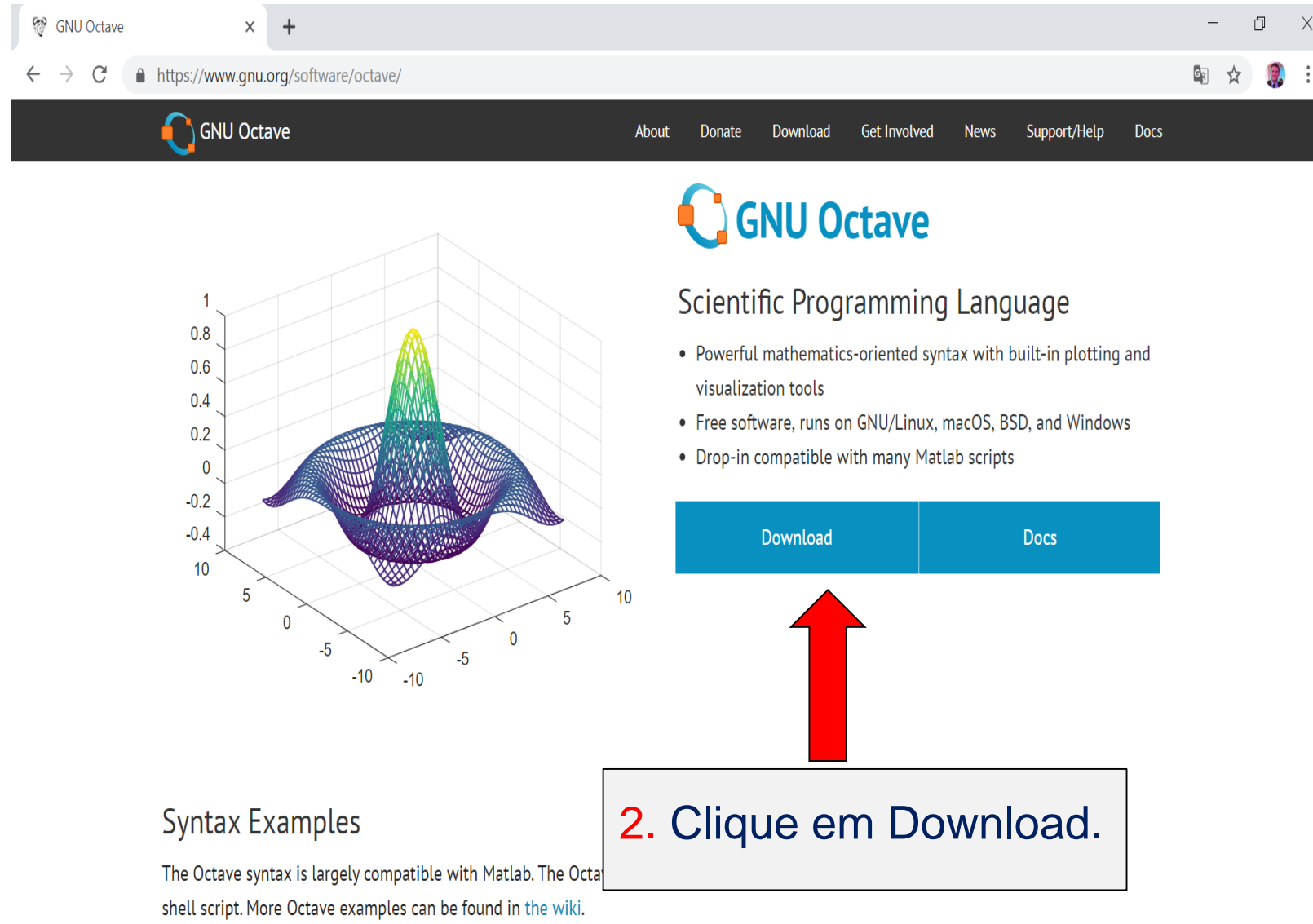
- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

Download Docs

Syntax Examples

The Octave syntax is largely compatible with Matlab. The Octave interpreter can be run in GUI mode, as a console, or invoked as part of a shell script. More Octave examples can be found in [the wiki](#).

Octave – Passo 2 Instalação



The screenshot shows the GNU Octave website in a web browser. The browser's address bar displays <https://www.gnu.org/software/octave/>. The website's navigation bar includes links for About, Donate, Download, Get Involved, News, Support/Help, and Docs. The main content area features the GNU Octave logo, the title "Scientific Programming Language", and a list of features: "Powerful mathematics-oriented syntax with built-in plotting and visualization tools", "Free software, runs on GNU/Linux, macOS, BSD, and Windows", and "Drop-in compatible with many Matlab scripts". Below this list are two buttons: "Download" and "Docs". A large red arrow points from a text box at the bottom to the "Download" button. To the left of the main content is a 3D plot of a mathematical function. At the bottom left, there is a section titled "Syntax Examples" with a brief description of Octave's compatibility with Matlab.

GNU Octave

About Donate Download Get Involved News Support/Help Docs

GNU Octave

Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

Download Docs

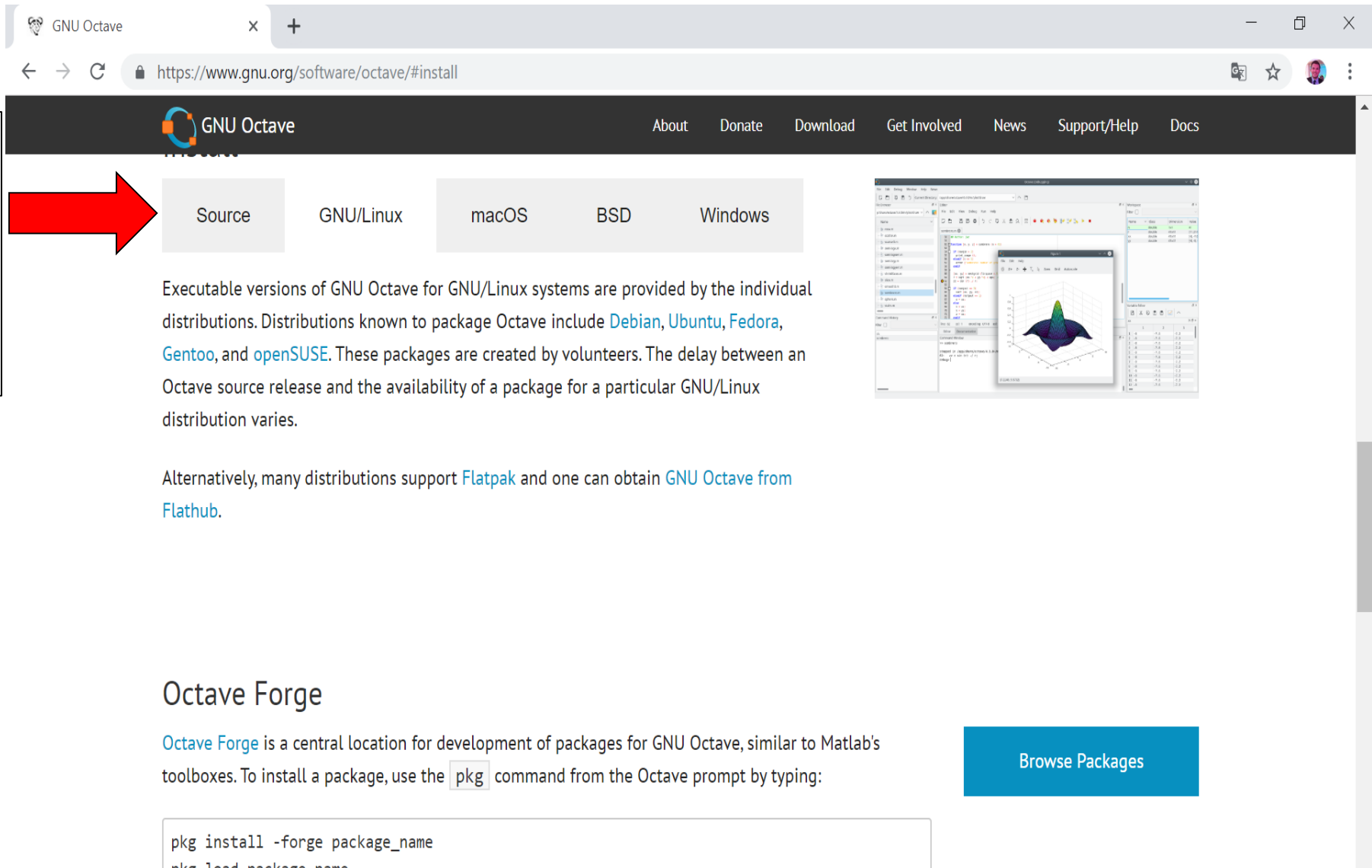
2. Clique em Download.

Syntax Examples

The Octave syntax is largely compatible with Matlab. The Octave shell script. More Octave examples can be found in [the wiki](#).

Octave – Passo 3 Instalação

3. Selecione o seu Sistema Operacional



The screenshot shows the GNU Octave website's installation page. A red arrow points to the 'Source' tab in the navigation bar. The page content includes a paragraph about GNU/Linux distributions, a paragraph about Flatpak, and a section for Octave Forge with a 'Browse Packages' button. A code block at the bottom shows the commands to install and load a package from Octave Forge.

GNU Octave

Source GNU/Linux macOS BSD Windows

Executable versions of GNU Octave for GNU/Linux systems are provided by the individual distributions. Distributions known to package Octave include [Debian](#), [Ubuntu](#), [Fedora](#), [Gentoo](#), and [openSUSE](#). These packages are created by volunteers. The delay between an Octave source release and the availability of a package for a particular GNU/Linux distribution varies.

Alternatively, many distributions support [Flatpak](#) and one can obtain [GNU Octave from Flathub](#).

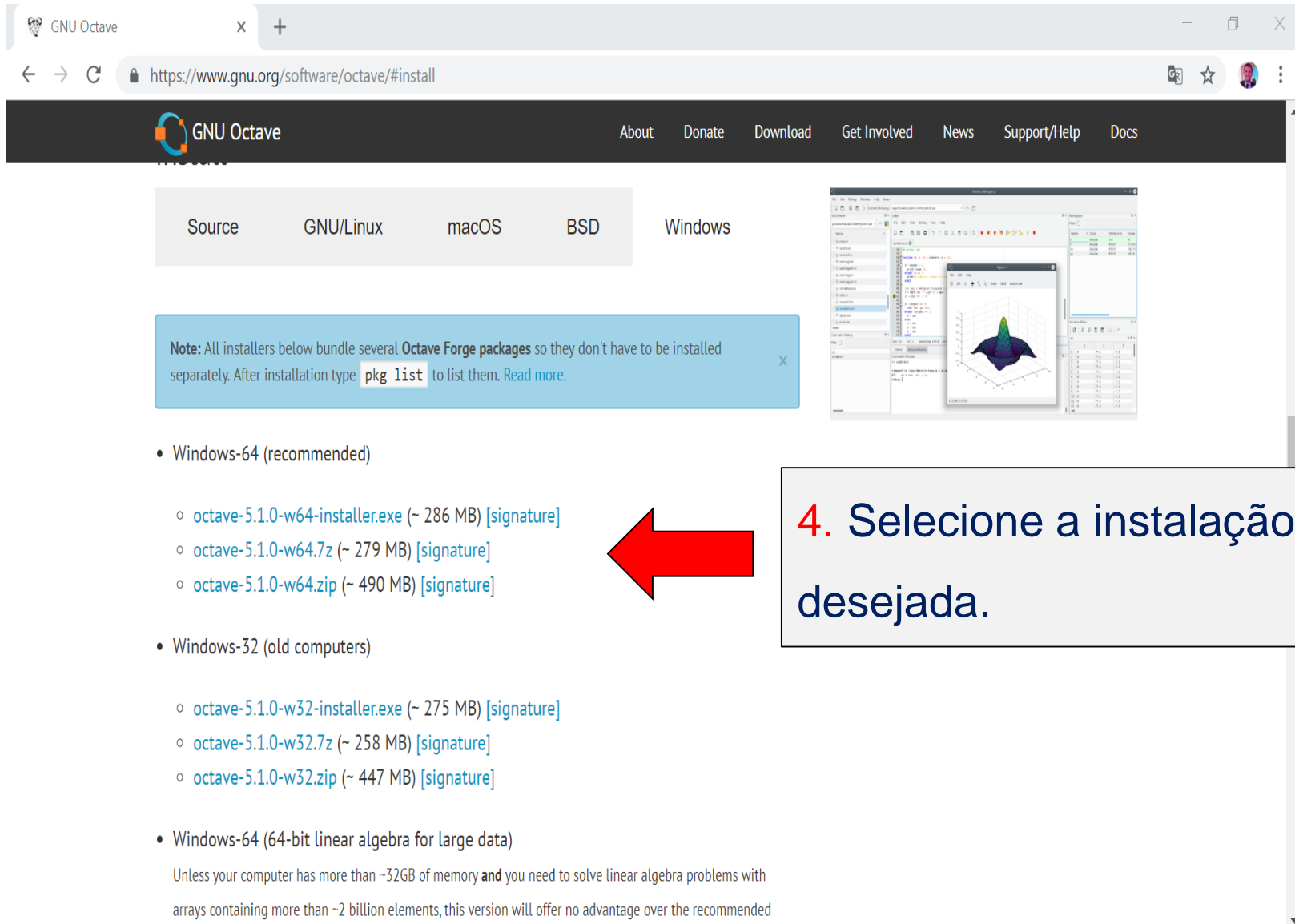
Octave Forge

[Octave Forge](#) is a central location for development of packages for GNU Octave, similar to Matlab's toolboxes. To install a package, use the `pkg` command from the Octave prompt by typing:

```
pkg install -forge package_name  
pkg load package_name
```

Browse Packages

Octave – Passo 4 Instalação



GNU Octave

https://www.gnu.org/software/octave/#install

About Donate Download Get Involved News Support/Help Docs

Source GNU/Linux macOS BSD Windows

Note: All installers below bundle several **Octave Forge** packages so they don't have to be installed separately. After installation type `pkg list` to list them. [Read more.](#)

- Windows-64 (recommended)
 - [octave-5.1.0-w64-installer.exe](#) (~ 286 MB) [\[signature\]](#)
 - [octave-5.1.0-w64.7z](#) (~ 279 MB) [\[signature\]](#)
 - [octave-5.1.0-w64.zip](#) (~ 490 MB) [\[signature\]](#)
- Windows-32 (old computers)
 - [octave-5.1.0-w32-installer.exe](#) (~ 275 MB) [\[signature\]](#)
 - [octave-5.1.0-w32.7z](#) (~ 258 MB) [\[signature\]](#)
 - [octave-5.1.0-w32.zip](#) (~ 447 MB) [\[signature\]](#)
- Windows-64 (64-bit linear algebra for large data)

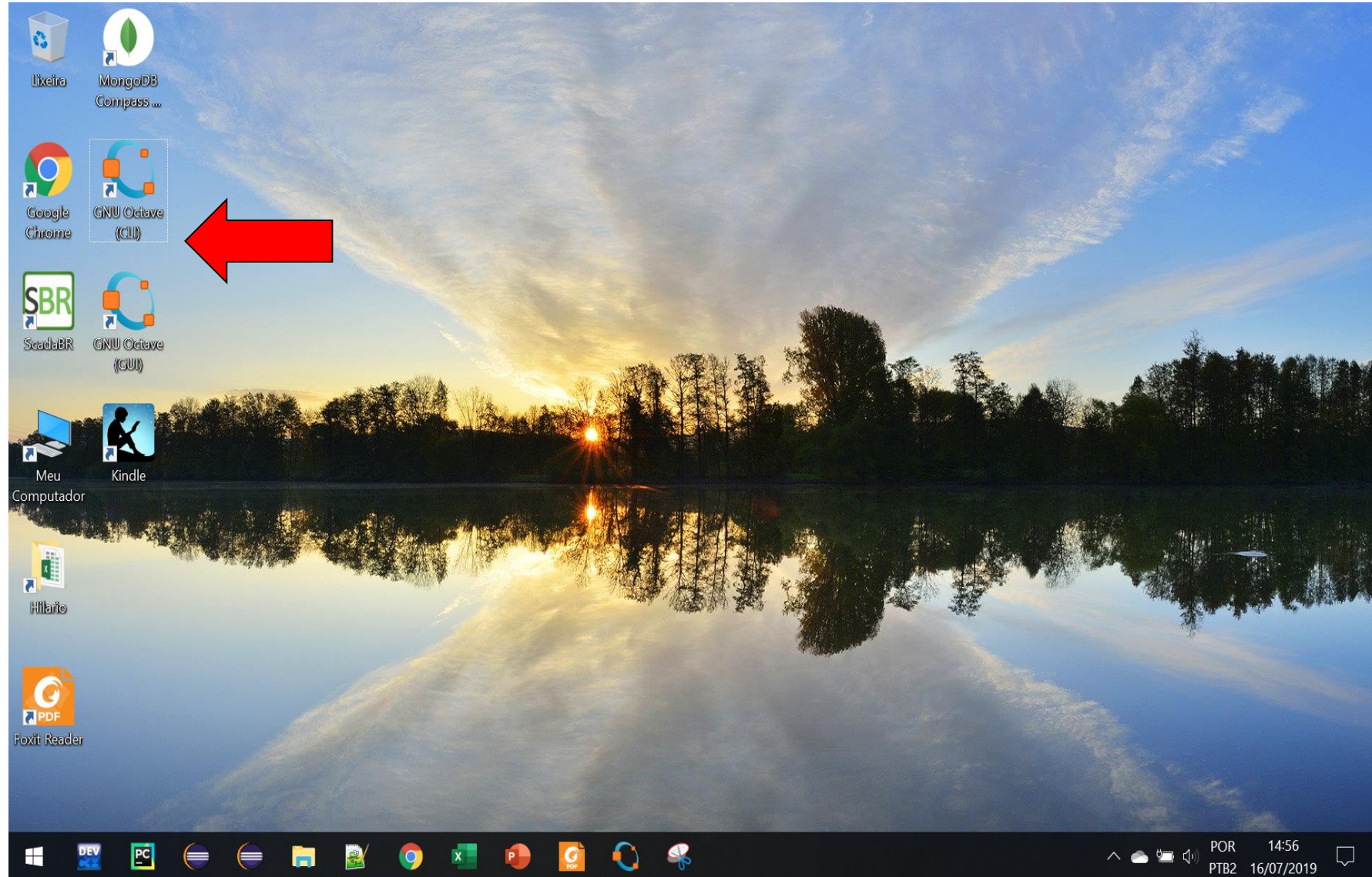
Unless your computer has more than ~32GB of memory **and** you need to solve linear algebra problems with arrays containing more than ~2 billion elements, this version will offer no advantage over the recommended

4. Selecione a instalação desejada.

Octave – Instalação

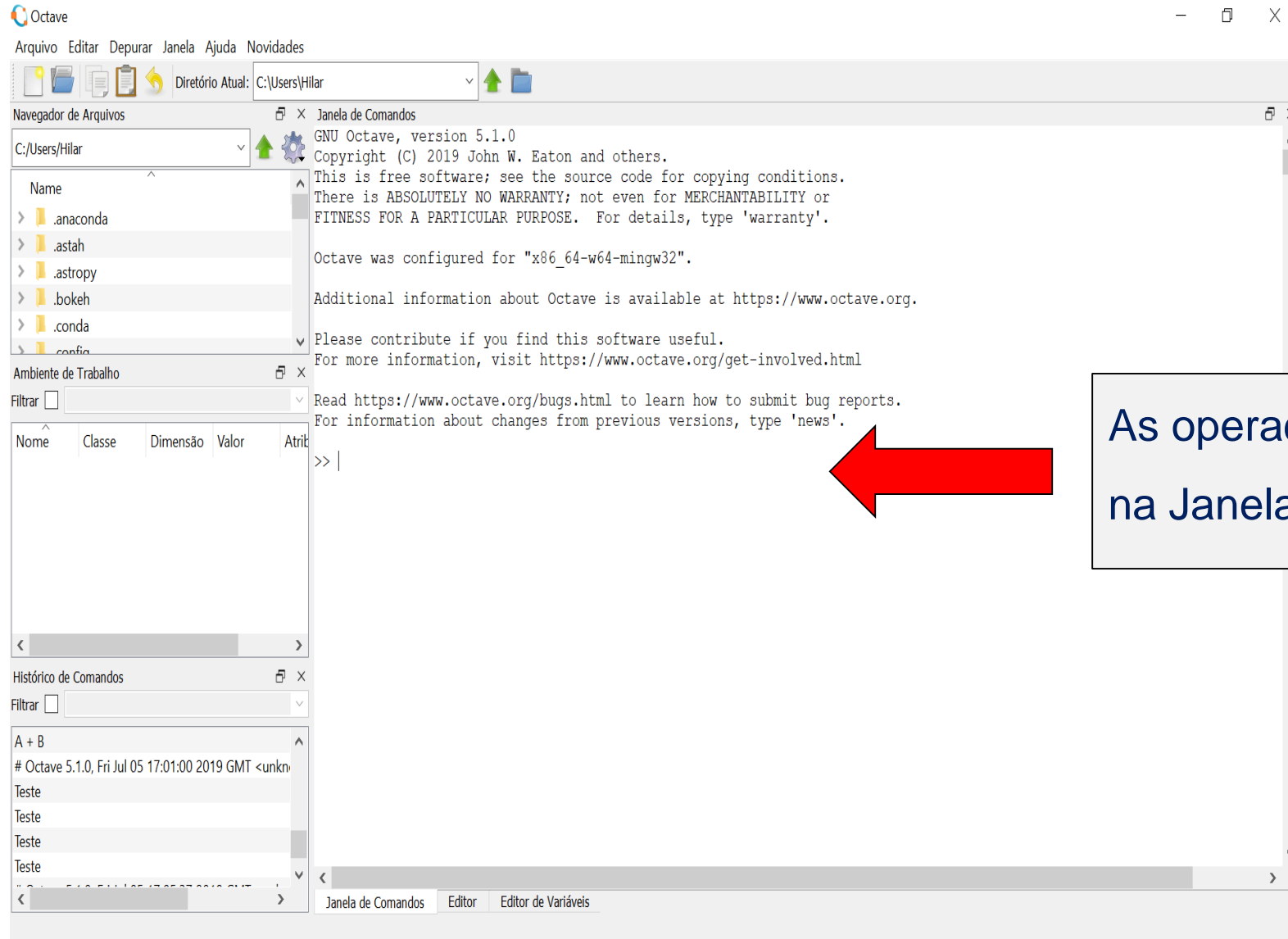
- Após ter completado o download do instalador:
 1. Execute o instalador;
 2. Next > Next >;
 3. Escolha a pasta em que queres instalar o Octave;
 4. Next;
 5. Aguarda um pouco a instalação;
 6. Finalmente “Finish”.
- Se tudo correu bem deverá ter aparecido um ícone na Área de Trabalho.

Octave – Instalação



Octave – Comandos Básicos

Octave – Operações Básicas

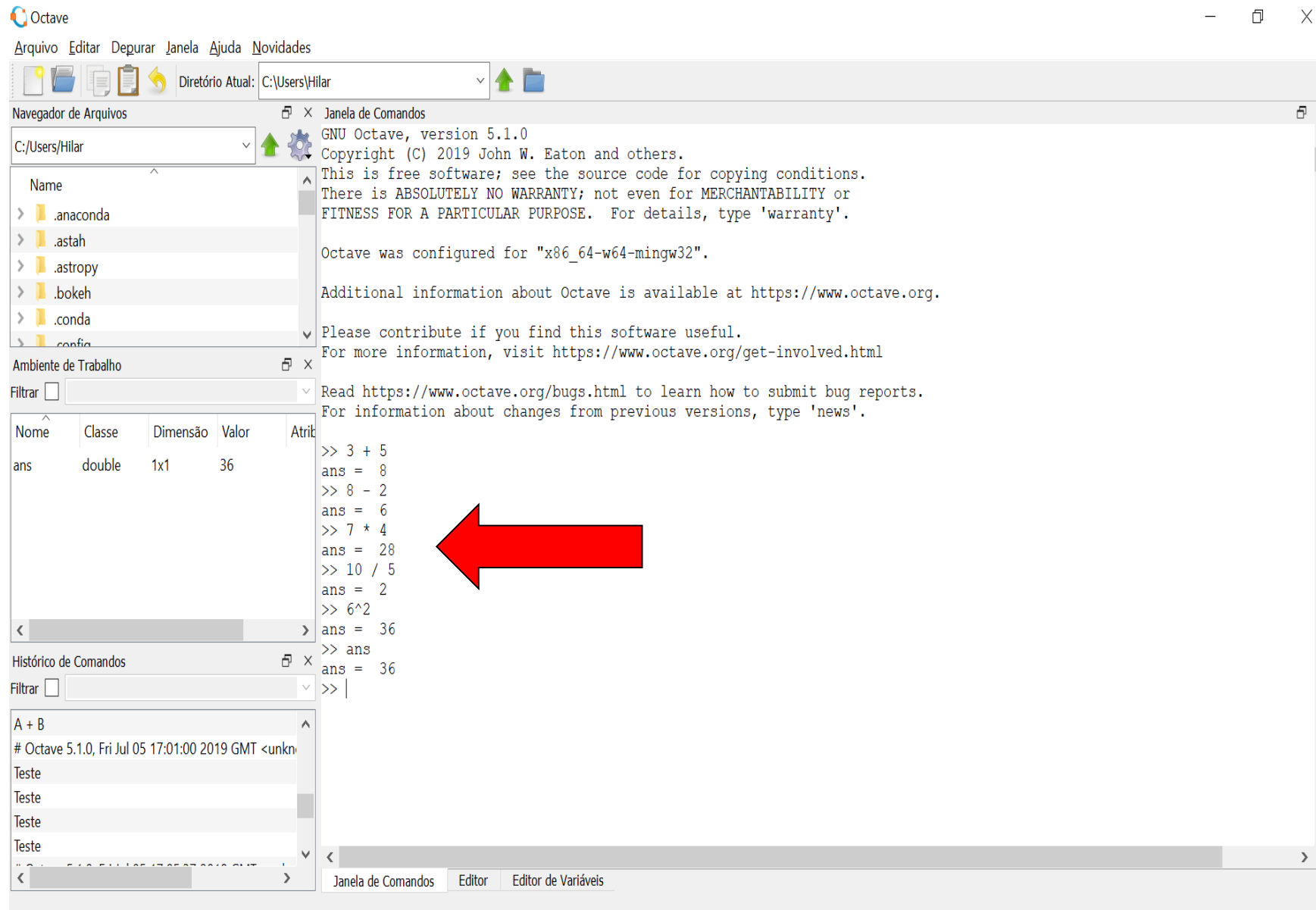


As operações são realizadas na Janela de Comandos.

Octave – Operações Básicas

- Operações Básicas:
 - Soma ($+$) $\rightarrow 3 + 5$
 - Subtração ($-$) $\rightarrow 8 - 2$
 - Multiplicação ($*$) $\rightarrow 7 * 4$
 - Divisão ($/$) $\rightarrow 10 / 5$
 - Exponenciação ($^$) $\rightarrow 6 ^ 2$
- Implicitamente o Octave atribui o resultado da operação a uma variável chamada de “ans”.

Octave – Operações Básicas



Exercício 1

1. Na Janela de Comandos faça um exemplo usando as quatro operações básicas mais a exponenciação para cada um dos números a seguir:

a) 4 e 2

b) 21 e 7

c) 25 e 5



Octave – Operações Básicas

- Podemos escrever uma **expressão complexa** a partir dos operadores básicos:

```
>> 2^2 + 2*3 - 4/2  
>> ans = 8
```

- **OBS:** É importante observar a ordem de precedência dos operadores.

Exercício 2

2. Na Janela de Comandos verifique qual o resultado das seguintes operações:

a) $4 + 2 * 3$

b) $2^3 * (4 - 2) / 3 * 2$

c) $2^2/4 - (16/2)^2$



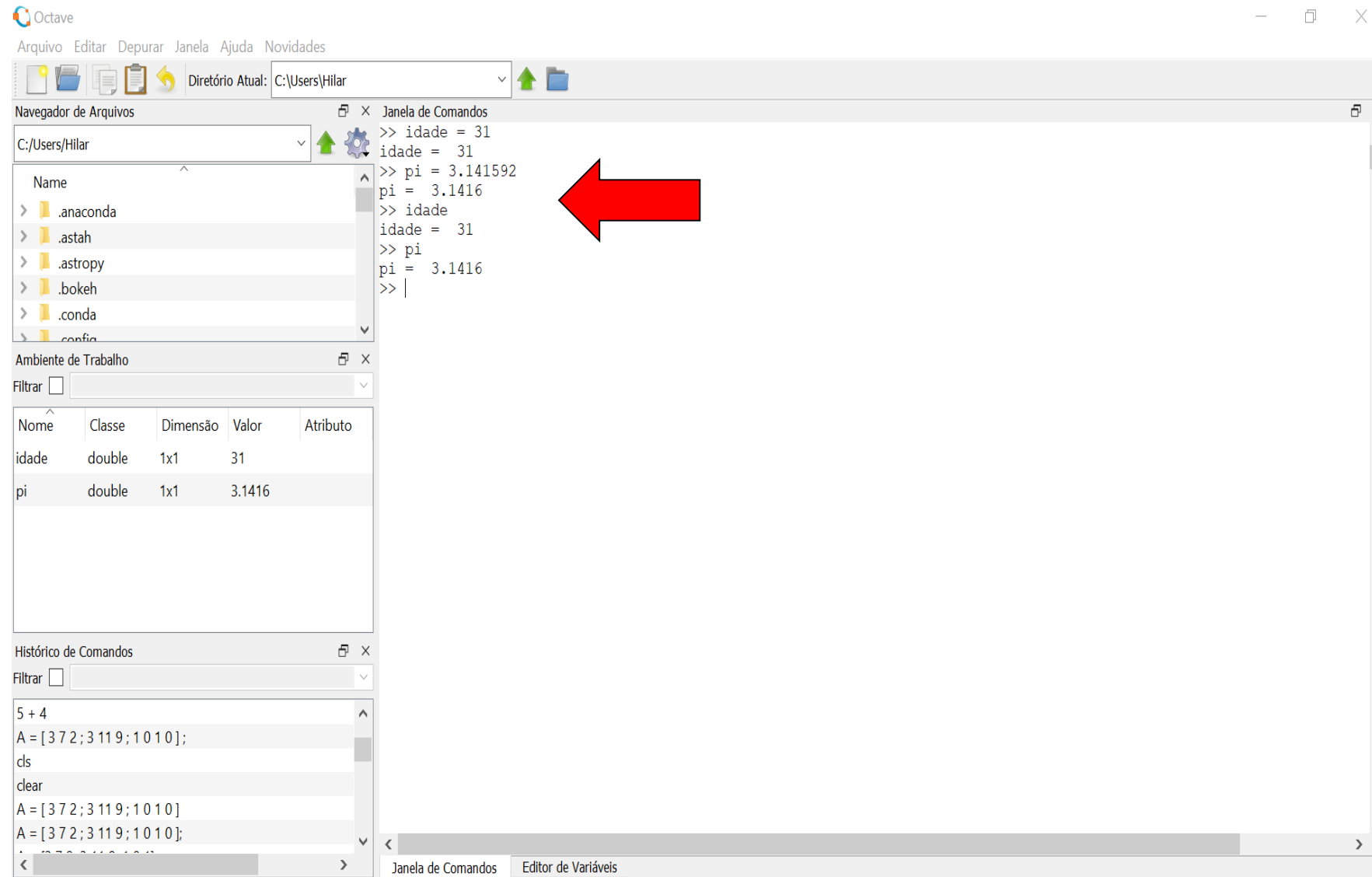
Octave – Variáveis

- É possível atribuir valores a **variáveis** através da **Janela de Comandos**.
- Para **definir uma variável** basta **digitar o nome** que deseja para a variável seguido do **operador "="** e então **o valor que se deseja armazenar**.

```
>> idade = 31
```

```
>> salario = 2555.20
```

Octave – Variáveis



The screenshot displays the Octave GUI interface. The 'Janela de Comandos' (Command Window) is active, showing the following commands and their outputs:

```
>> idade = 31
idade = 31
>> pi = 3.141592
pi = 3.1416
>> idade
idade = 31
>> pi
pi = 3.1416
>> |
```

A red arrow points to the command window output. The 'Ambiente de Trabalho' (Workspace) pane shows the current variables:

Nome	Classe	Dimensão	Valor	Atributo
idade	double	1x1	31	
pi	double	1x1	3.1416	

A blue arrow points to the workspace table. The 'Histórico de Comandos' (Command History) pane shows the following commands:

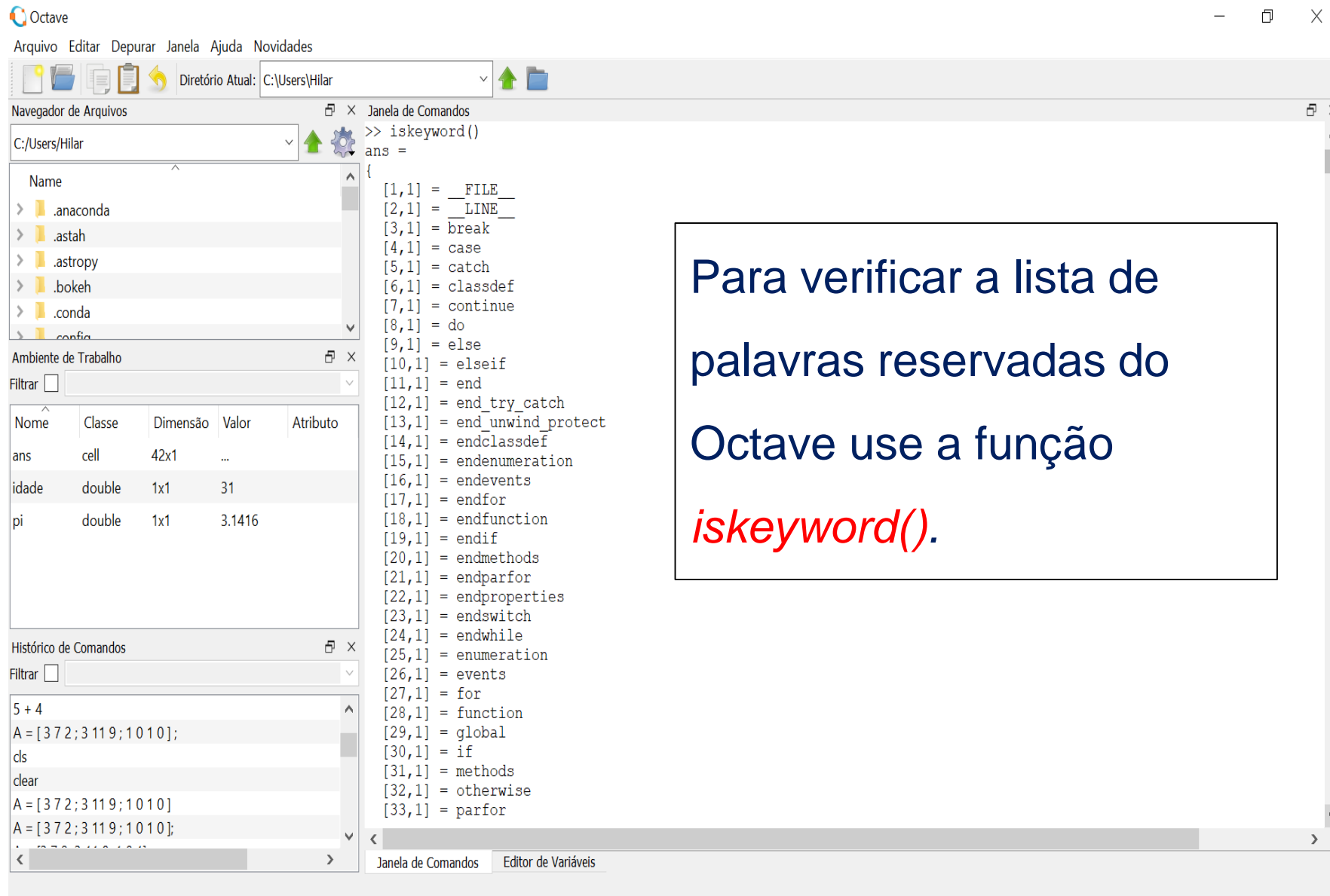
```
5 + 4
A = [3 7 2; 3 11 9; 10 10];
cls
clear
A = [3 7 2; 3 11 9; 10 10]
A = [3 7 2; 3 11 9; 10 10];
< 10 7 2 3 11 9 10 10 >
```

The bottom status bar indicates the current view is 'Janela de Comandos' and 'Editor de Variáveis'.

Octave – Variáveis

- O nome de uma variável deve respeitar as seguintes condições:
 - Não pode começar com um número.
 - Exemplo: 1idade.
 - Não pode ser uma palavra reservada* do Octave.
 - Não pode conter espaço e nem caracteres especiais.
 - Exemplo: quatro lados, lado*2, lado&diagonal.

Octave – Palavras reservadas



The screenshot shows the Octave GUI with the following components:

- Menu Bar:** Arquivo, Editar, Depurar, Janela, Ajuda, Novidades.
- Toolbar:** Includes icons for file operations and a dropdown menu showing the current directory: C:\Users\Hilar.
- Navegador de Arquivos:** A tree view showing the file system structure under C:\Users\Hilar.
- Ambiente de Trabalho:** A table displaying workspace variables.
- Histórico de Comandos:** A list of previously executed commands.
- Janela de Comandos:** The main command window showing the execution of `iskeyword()`.

The **Ambiente de Trabalho** table contains the following data:

Nome	Classe	Dimensão	Valor	Atributo
ans	cell	42x1	...	
idade	double	1x1	31	
pi	double	1x1	3.1416	

The **Janela de Comandos** shows the following output for the command `>> iskeyword()`:

```
ans =  
{  
 [1,1] = _FILE_  
 [2,1] = _LINE_  
 [3,1] = break  
 [4,1] = case  
 [5,1] = catch  
 [6,1] = classdef  
 [7,1] = continue  
 [8,1] = do  
 [9,1] = else  
 [10,1] = elseif  
 [11,1] = end  
 [12,1] = end_try_catch  
 [13,1] = end_unwind_protect  
 [14,1] = endclassdef  
 [15,1] = endenumeration  
 [16,1] = endevents  
 [17,1] = endfor  
 [18,1] = endfunction  
 [19,1] = endif  
 [20,1] = endmethods  
 [21,1] = endparfor  
 [22,1] = endproperties  
 [23,1] = endswitch  
 [24,1] = endwhile  
 [25,1] = enumeration  
 [26,1] = events  
 [27,1] = for  
 [28,1] = function  
 [29,1] = global  
 [30,1] = if  
 [31,1] = methods  
 [32,1] = otherwise  
 [33,1] = parfor
```

Para verificar a lista de
palavras reservadas do
Octave use a função
iskeyword().

Octave – Variáveis

- Um ponto importante é que o Octave é *case sensitive*, ou seja, ele diferencia letras maiúsculas de minúsculas.
 - Por exemplo:
 - idade é diferente de IDADE
 - Idade é diferente de IDADE

Exercício 3

3. Declare **duas variáveis** contendo **dois números** e realize as **quatro operações básicas** mais **a exponenciação** com essas duas variáveis, armazenando o resultado também em uma variável.



Octave – Tipos e Variáveis

- Matrizes (reais e complexas).
- Vetores → é uma matriz com uma linha e n colunas.
- Strings → um vetor de n caracteres.
- Estruturas → tipos compostos.
- Escalares → é uma matriz de dimensão 1x1.
 - Inteiros → é um double.
 - Booleano → é um inteiro (1 = verdadeiro e 0 = falso).

Octave – Tipos e Variáveis

- Exemplos:

- Escalar:

```
>> idade = 31
```

- String:

```
>> nome = 'Hilario'
```

- Boolean:

```
>> iraChover = true
```

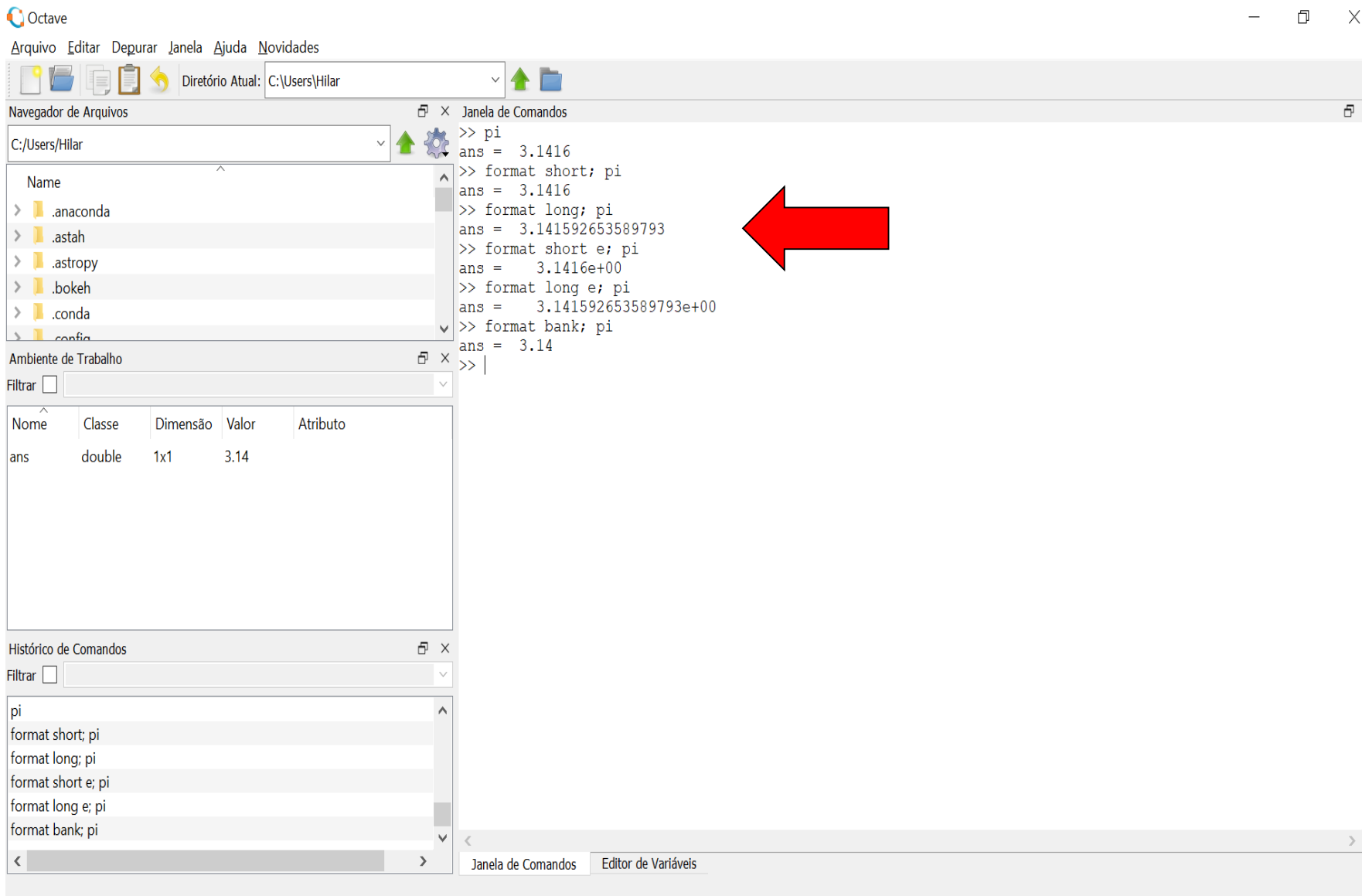
- Vetor:

```
>> vetor = [1 2 3 4 5 6]
```

Octave – Formato das Variáveis

- É possível alterar a forma com que o **Octave** apresenta os valores da variáveis na Janela de Comandos.
 - **format short** → formato de ponto fixo com 5 dígitos.
 - **format long** → formato de ponto fixo com 15 dígitos.
 - **format short e** → formato de ponto flutuante com 5 dígitos.
 - **format long e** → formato de ponto flutuante com 15 dígitos.
 - **format bank** → formato com duas casas decimais.
- Veja mais formatos usando comando **help format**.

Octave – Formato das Variáveis



The screenshot displays the Octave environment with the following components:

- Menu Bar:** Arquivo, Editar, Depurar, Janela, Ajuda, Novidades.
- Toolbar:** Includes icons for file operations and a dropdown menu showing the current directory: C:\Users\Hilar.
- Navegador de Arquivos:** A file explorer on the left showing the directory structure of C:\Users\Hilar.
- Janela de Comandos:** A command window on the right showing the execution of commands to format the variable 'ans':

```
>> pi
ans = 3.1416
>> format short; pi
ans = 3.1416
>> format long; pi
ans = 3.141592653589793
>> format short e; pi
ans = 3.1416e+00
>> format long e; pi
ans = 3.141592653589793e+00
>> format bank; pi
ans = 3.14
>> |
```
- Ambiente de Trabalho:** A workspace window at the bottom left showing the current state of the variable 'ans':

Nome	Classe	Dimensão	Valor	Atributo
ans	double	1x1	3.14	
- Histórico de Comandos:** A command history window at the bottom right showing the sequence of commands entered.

A large red arrow points from the command window towards the workspace window, highlighting the relationship between the command execution and the variable's state.

Octave – Comandos Básicos

Comandos	Descrição
clear	Apaga todas as variáveis da Área de Trabalho
clear <i>nomeVariavel</i>	Apaga a variável ' <i>nomeVariavel</i> ' da Área de Trabalho
clc	Apaga o que está impresso na Janela de Comandos.
help <i>nomefunção</i>	Mostra como utilizar a função ou comando ' <i>nomefunção</i> ', bem como o que ele faz.

Octave – Funções Básicas

Função	Descrição
<i>round()</i>	Arredonda os números para baixo se a parte fracionário for menor 5, e arredonda para cima se a parte fracionária for maior ou igual a 5
<i>ceil()</i>	Arredonda todos os números para cima, independentemente da sua parte fracionária.
<i>floor()</i>	Arredonda todos os números para baixo, independentemente da sua parte fracionária.
<i>sqrt()</i>	Calcula raiz quadrada.
<i>abs()</i>	calcula o valor absoluto de um número (real ou imaginário).

Octave – Funções Básicas

Função	Descrição
<i>sin()</i>, <i>cos()</i>, <i>tan()</i>	Calcula o seno, cosseno e tangente de um número.
<i>asin()</i>, <i>acos()</i>, <i>atan()</i>	Calcula o arco seno, arco cosseno e arco tangente de um número.
<i>log()</i>	Calcula o log natural de um número.
<i>log2()</i>	Calcula o logaritmo na base 2 de um número.
<i>log10()</i>	Calcula o logaritmo na base 10 de um número.

Exercício 4

4. Qual o resultado das seguintes operações:

a) `round(4.6)`

b) `round(3.4)`

c) `sqrt(16)`

d) `sqrt(-49)`

e) `abs(-99)`

f) `sin(45)`

g) `log2(64)`

h) `mod(45, 2)` → O que essa função faz?



Octave – Vetores

- Criação de um vetor

```
Janela de Comandos
>> vetor = [1 2 3 4 5]
vetor =

    1.00    2.00    3.00    4.00    5.00

>> vetor_2 = [1, 2, 3, 4, 5]
vetor_2 =

    1.00    2.00    3.00    4.00    5.00

>> |
```

Octave – Vetores

- Acessar um elemento do vetor
 - Sintaxe: `nome_vetor(indice)`
 - **OBS:** o índice começa pelo número 1.

```
>> vetor = [1 2 3 4 5]
vetor =

    1.00    2.00    3.00    4.00    5.00

>> vetor_2 = [1, 2, 3, 4, 5]
vetor_2 =

    1.00    2.00    3.00    4.00    5.00

>> vetor(1)
ans = 1.00
>> vetor(3)
ans = 3.00
>> vetor_2(5)
ans = 5.00
>> vetor(6)
error: vetor(6): out of bound 5
>> |
```

Octave – Funções Básicas

Função	Descrição
sum()	Efetua a soma de todos os elementos do vetor.
min()	Retorna o menor número do vetor.
max()	Retorna o maior número do vetor.
mean()	Retorna o valor médio dos elementos do vetor.
prod()	Retorna o produto dos elementos do vetor.
sort()	Retorna o vetor ordenado do menor elemento para o maior.

Exercício 5

5. Dados os vetores $v1 = [5, 8, 9, 4]$ e $v2 = [12, 15, 19, 22]$, responda as seguintes questões:
- a) Qual o resultado da operação $v1 + v2$?
 - b) Qual o resultado da operação $v1 * 4$?
 - c) Qual o resultado da operação $prod(v1)$?
 - d) Qual o valor máximo e mínimo do vetor $v1$?
 - e) Qual a média aritmética dos valores do $v2$?



Octave – Matrizes

- Criação de uma Matriz

```
>> matriz_1 = [1 2 3; 4 5 6; 7 8 9]
matriz_1 =
```

1.00	2.00	3.00
4.00	5.00	6.00
7.00	8.00	9.00

```
>> matriz_2 = [1, 2, 3; 4, 5, 6; 7, 8, 9]
matriz_2 =
```

1.00	2.00	3.00
4.00	5.00	6.00
7.00	8.00	9.00

Octave – Matrizes

- Acessar um elemento de uma matriz.
- Sintaxe: `nome_matriz(linha, coluna)`
- **OBS:** o índice começa pelo número 1.

```
>> matriz_1(2,3)
ans = 6.00
>> matriz_1(1,1)
ans = 1.00
>> matriz_2(1,:)
ans =
    1.00    2.00    3.00

>> matriz_2(:,2)
ans =
    2.00
    5.00
    8.00

>> matriz_1(1,1:2)
ans =
    1.00    2.00
```


Octave – Vetores e Matrizes

- As funções `zeros()` e `ones()` ajudam na criação de matrizes ou vetores de zeros ou uns, respectivamente.

```
>> vetor_0 = zeros(1,8)
vetor_0 =
    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
```

```
>> vetor_1 = ones(1, 3)
vetor_1 =
```

```
    1.00    1.00    1.00
```

```
>> matriz_0 = zeros(3,4)
matriz_0 =
```

```
    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00
```

```
>> matriz_1 = ones(2,2)
matriz_1 =
```

```
    1.00    1.00
    1.00    1.00
```

Octave – Vetores e Matrizes

- Outras formas de criação de vetores e matrizes:

```
>> vetor = 1:15
vetor =

Columns 1 through 13:

    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00    9.00   10.00   11.00   12.00   13.00

Columns 14 and 15:

    14.00   15.00

>> vetor_2 = 1:0.2:3
vetor_2 =

    1.00    1.20    1.40    1.60    1.80    2.00    2.20    2.40    2.60    2.80    3.00

>> vetor_3 = 10:-1:0
vetor_3 =

    10.00    9.00    8.00    7.00    6.00    5.00    4.00    3.00    2.00    1.00    0.00

>> matriz = [vetor_2;vetor_3]
matriz =

    1.00    1.20    1.40    1.60    1.80    2.00    2.20    2.40    2.60    2.80    3.00
   10.00    9.00    8.00    7.00    6.00    5.00    4.00    3.00    2.00    1.00    0.00
```

Octave – Operações com Vetores

```
>> vetor_1 = [1 2 3 4]  
vetor_1 =
```

```
1 2 3 4
```

```
>> vetor_2 = [4 5 6 7]  
vetor_2 =
```

```
4 5 6 7
```

```
>> vetor_2 * 2  
ans =
```

```
8 10 12 14
```

```
>> vetor_2 + 3  
ans =
```

```
7 8 9 10
```

```
>> vetor_2 - 4  
ans =
```

```
0 1 2 3
```

```
>> vetor_2 / 2  
ans =
```

```
2.0000 2.5000 3.0000 3.5000
```

```
>> vetor_1 + vetor_2  
ans =
```

```
5 7 9 11
```

```
>> vetor_2 - vetor_1  
ans =
```

```
3 3 3 3
```

Octave – Operações com Matrizes

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

```
1 3 5  
2 7 6  
9 3 8
```

```
>> matriz_b = [7 1 4; 5 0 3; 3 6 7]  
matriz_b =
```

```
7 1 4  
5 0 3  
3 6 7
```

```
>> matriz_a^2  
ans =
```

```
52 39 63  
70 73 100  
87 72 127
```

```
>> matriz_a*2  
ans =
```

```
2 6 10  
4 14 12  
18 6 16
```

```
>> matriz_a+matriz_b  
ans =
```

```
8 4 9  
7 7 9  
12 9 15
```

```
>> matriz_a-matriz_b  
ans =
```

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

```
>> matriz_a*matriz_b  
ans =
```

```
37 31 48  
67 38 71  
102 57 101
```

```
>> matriz_a/matriz_b  
ans =
```

```
-1.6875e+00 2.0937e+00 7.8125e-01  
1.1875e+00 -1.8438e+00 9.6875e-01  
6.3441e-16 1.5000e+00 5.0000e-01
```

Exercício 6

6. Dadas as duas matrizes ao lado, responda as seguintes questões:

- a) Qual a matriz resultante da operação $A + B$?
- b) Qual a matriz resultante da operação $B - A$?
- c) Qual a matriz resultante da operação $A * B$?
- d) Qual a matriz resultante da operação A / B ?

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ e } B = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix}$$

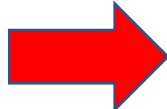


Octave – Scripts e Funções

Octave – Scripts

- Um script trata-se de uma **série de comandos** que são salvos em um arquivo (**com extensão .m**) e podem ser **executados posteriormente**.
- Ao executar um script, o Octave **executa linha a linha** do que está programado no script.

Octave – Scripts



Octave

Arquivo Editar Depurar Janela Ajuda Novidades

Diretório Atual: \IFES\Disciplinas\Cálculo Numérico\Octave

Novo Script

Nome

Teste.m

Ambiente de Trabalho

Filtrar

Nome	Classe	Dimensão	Valor	Atributo
x	double	1x1	25	
y	double	1x1	4.6000	

Janela de Comandos

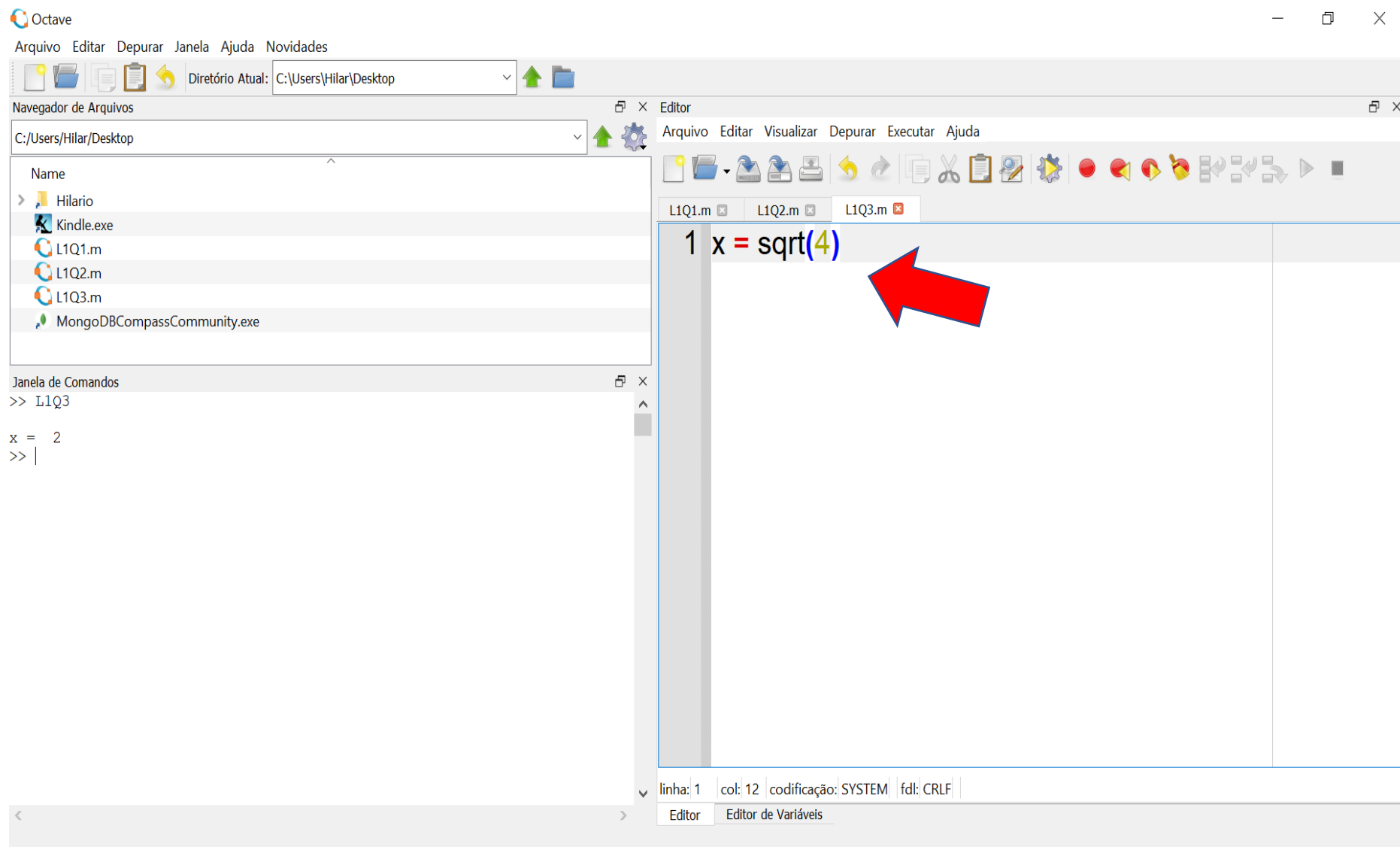
```
X = 25Y = 25.000000>> Teste
X = 25Y = 25.000000>> Teste
x = 25 e y = 4.600000>> Teste
x = 25 e y = 4.600000>> Teste
x = 25 e y = 4.600000>> Teste
x = 25 e y = 4.600000>> Teste
x = 25 e y = 4.600000>> Teste
x = 25 e y = 4.600000>> |
```

Editor

Arquivo Editar Visualizar Depurar Executar Ajuda

Editor de Variáveis

Octave – Scripts



Octave – Funções

- Uma **função** é um **conjunto de instruções** que realiza **uma tarefa específica**.
 - Funções são criadas **para serem reutilizadas** em outros programas.
 - A função é chamada por um outro programa através de **um nome atribuído a ela**.
- Funções podem receber **parâmetros** e **retornar resultados**.
 - As **variáveis** criadas e manipuladas dentro de uma função só são visíveis dentro dessa função (**Escopo da função**).

Octave – Funções

The screenshot displays the Octave IDE interface. The 'Arquivo' (File) menu is open, and the 'Nova Função...' (New Function...) option is highlighted. A red arrow points to this option. The interface includes a menu bar, a toolbar, a workspace table, a command window, and an editor area.

Arquivo Editar Depurar Janela Ajuda Novidades

- Novo
 - Novo Script Ctrl+N
 - Nova Função...**
 - Nova Figura
- Abrir... Ctrl+O
- Arquivos Recentes do Editor
- Carregar Ambiente de Trabalho...
- Salvar Ambiente de Trabalho Como...
- Sair

Ambiente de Trabalho

Filtrar ☐

Nome	Classe	Dimensão	Valor	Atributo
x	double	1x1	45	
y	double	1x1	4.6000	

Janela de Comandos

```
>> quadrado(3)
error: 'quadrado' undefined near line 1 column 1
>> |
```

Editor

Arquivo Editar Visualizar Depurar Executar Ajuda

Editor Editor de Variáveis

Octave – Funções

- Sintaxe (Função):

```
function [saida(s)] = nomeFuncao(parametron(s)_entrada)
```

```
% comentários para ajudar
```

```
comandos a serem executados
```

```
[saida(s)] = valor(es)
```

```
end
```

- **OBS:** Funções não são executadas como Scripts.
 - Precisam ser chamadas na **Janela de Comandos**.

Funções – Exemplo 1

The screenshot displays the Octave IDE interface. On the left, the 'Navegador de Arquivos' (File Navigator) shows the current directory as 'D:/Hilario/Ensino/IFES/Disciplinas/Cálculo Numérico/Octave' and lists a file named 'quadrado.m'. Below it, the 'Ambiente de Trabalho' (Workspace) table shows two variables: 'x' (double, 1x1, value 45) and 'y' (double, 1x1, value 4.6000). The 'Janela de Comandos' (Command Window) shows the execution of the function: `>> quadrado(4)` resulting in `ans = 16`. A red arrow points to this output. The main 'Editor' window on the right shows the code for 'quadrado.m':

```
1 function resultado = quadrado (x)
2     resultado = x * x;
3 endfunction
4
```

A green arrow points to the function definition. The status bar at the bottom indicates 'linha: 2 col: 22 codificação: SYSTEM fdl: CRLF'.

Nome	Classe	Dimensão	Valor	Atributo
x	double	1x1	45	
y	double	1x1	4.6000	

```
>> quadrado(4)
ans = 16
>>
```

Funções – Exemplo 2

The screenshot displays the Octave environment with the following components:

- Arquivo** (File) menu: Arquivo, Editar, Depurar, Janela, Ajuda, Novidades.
- Navegador de Arquivos** (File Navigator): Shows the current directory as `D:\Hilario\Ensino\IFES\Disciplinas\Cálculo Numérico\Octave`. It lists two files: `quadrado.m` and `soma_sub.m`.
- Ambiente de Trabalho** (Workspace): A table showing the current workspace variables.

Nome	Classe	Dimensão	Valor	Atributo
soma	double	1x1	14	
sub	double	1x1	2	
- Janela de Comandos** (Command Window): Shows the execution of the function `soma_sub` with arguments 8 and 6.

```
>> [soma, sub] = soma_sub(8, 6)
soma = 14
sub = 2
>> soma
soma = 14
>> sub
sub = 2
>> |
```
- Editor**: The main window for editing code. It shows the function definition in `soma_sub.m`:

```
1 function [soma, subtracao] = soma_sub(x, y)
2   soma = x + y;
3   subtracao = x - y;
4 endfunction
5
```

Two arrows highlight specific elements: a red arrow points to the Command Window output, and a green arrow points to the function definition in the Editor.

Octave – Funções Anônimas

- Funções mais simples podem ser declaradas de forma compacta.
- Sintaxe:
 - nome_da_função = @(lista_arg) expressão
- Exemplo

```
f = @(x, y) x^2 + y - 2; # Declaração
```

```
resultado = f(2, 4); # Utilização
```

```
printf("Resultado = %d", resultado);
```

Octave – Programação Estruturada

Octave – Comando de Saída

- O comando **disp** pode ser usado para exibir o valor de uma variável ou uma mensagem.
- Exemplo:
 - **disp**("Hello World");
 - **x = 45;**
disp(x);

Octave – Comando de Saída Formatada

- O outro comando utilizado para saída de dados é o *printf*.
- Esse comando **permite imprimir informações** de uma maneira mais completa e informativa.
- **printf**("Texto desejado %tipo_variavel", variaveis);

Código de Formato	Descrição
%d	Será impresso o valor de uma variável inteira
%f	Será impresso o valor de uma variável de ponto flutuante
Código de Controle	Descrição
\n	Indica que deve iniciar uma nova linha (enter)
\t	Indica que deve realizar uma tabulação horizontal

Octave – Comando de Saída Formatada

- Exemplos:
 - **printf**("Hello World!");
 - **x = 25;**
y = 4.6;
printf("\n x = %d e y = %f", x, y);

Octave – Comandos de Entrada de dados

- A forma mais simples de ler dados do usuário, é através da **instruções input**.
- Exemplos:
 - `numero = input ("Digite um número entre 1 e 100: ");`
 - `vetor = input("Introduza as temperaturas desta semana: ");`
 - Os valores devem ser digitados entre **[]**.
 - `nome = input("Digite seu nome: ", "s");`

Octave – Comandos de Entrada de dados

- O Octave permite de uma forma muito simples apresentar um menu.
- Exemplos:
 - `unidade = menu("Indique a unidade para ângulos", "Graus", "Radianos");`
 - `dia = menu("Indique o dia da semana", "Domingo", "Segunda-feira", "Terça-feira", "Quarta-feira", "Quinta-feira", "Sexta-feira", "Sábado");`

Exercício 7

7. Leia um número e um vetor, e multiplique o número pelo vetor.



Octave – Estrutura de Decisão (If)

- Permite **tomar decisões** baseadas em **condições**.
- As decisões são baseadas nos resultados de **operações lógicas** e **relacionais**.
- Sintaxe básica:
 - **OBS: elseif e else** são opcionais.

if (condicao1)

Operações

elseif (condicao2)

Operações

else

Operações

end

Octave – Estrutura de Decisão (If)

- Operadores Relacionais

Exemplo	Operador	Relação
<code>x == 0</code>	<code>==</code>	Igual
<code>x != y</code> <code>x ~= y</code>	<code>~=</code>	Diferente
<code>a < 0</code>	<code><</code>	Menor que
<code>a > b</code>	<code>></code>	Maior que
<code>a <= b</code>	<code><=</code>	Menor ou igual que
<code>val >= despesas</code>	<code>>=</code>	Maior ou igual que

Octave – Estrutura de Decisão (If)

- Representação
 - `true = 1`
 - `false = 0`
- Operadores Lógicos
 - `~x` (`not`) ou `!x` (`not`):
 - `x & y` (`and`):
 - `x && y` (`and`)
 - `x | y` (`or`)
 - `x || y` (`or`)

A	B	!A	A & B	A B
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False

Estrutura de Decisão (If) - Exemplo

```
numero = 3;  
minimo = 5;  
maximo = 20;  
  
if( numero >= minimo && numero <= maximo )  
    printf("\nO número %d está no intervalo", numero);  
elseif( numero > maximo )  
    printf("\nO número %d está com um valor muito alto!", numero);  
else  
    printf("\nO número %d está com um valor muito baixo!", numero);  
end
```

Exercício 8

8. Faça um Script para ler um número inteiro e determine se ele é par ou ímpar.
- Dica: utilize a função mod (resto da divisão).



Octave – Comando de Seleção(**switch-case**)

- A estrutura **switch-case** testa diferentes valores de uma mesma variável.
- Dependendo do valor da variável testada, diferentes ações podem ser tomadas.

```
switch variavel  
    case valor1  
        Operações 1  
  
    case valor2  
        Operações 2  
  
    . . .  
  
    otherwise  
        Operações  
  
end
```

Comando de Seleção(**switch-case**) - Exemplo

- Exemplo:

```
numero = 10;  
  
switch numero  
  case 1  
    x = 1 + 2;  
    disp("1");  
  case 2  
    disp("2");  
  otherwise  
    disp("Outro");  
end
```

Exercício 9

9. Faça um Script para ler dois números inteiros e depois exiba um Menu para o usuário escolher dentre as quatro operações básicas (+, -, *, /) e exiba o resultado da operação selecionada.



Octave – Comandos de Repetição

- São usados para **repetir a execução** de **um** ou **mais comandos**.
- Comandos:
 - **for**
 - Usado quando você sabe a quantidade de vezes que as operações serão executadas.
 - **while**
 - Executam um bloco de comando até que uma condição seja verdadeira.

Octave – Comandos de Repetição (for)

- A sintaxe básica do comando **for** é:

```
for contador = valor_minimo:valor_maximo
```

Operações

```
end
```

- Exemplo:

```
for i = 1:20  
    printf("%d ", i);  
endfor
```


Octave – Comandos de Repetição (while)

- A sintaxe básica do comando **while** é:

while condicao_satisfeita

Operações

end

- Exemplo:

```
i = 1;

while ( i <= 20 )
    printf ("%d ", i);
    i+=1;
endwhile
```

Exercício 10

10. Faça um Script para exibir todos os **números ímpares** de **1 a 100**.



Octave – Comandos de Repetição(do-until)

- Existe uma outra estrutura de repetição que é `do-until` que executa um ou mais comando até que uma condição seja VERDADEIRA.
- A sintaxe básica do comando `do-until` é:

`do`

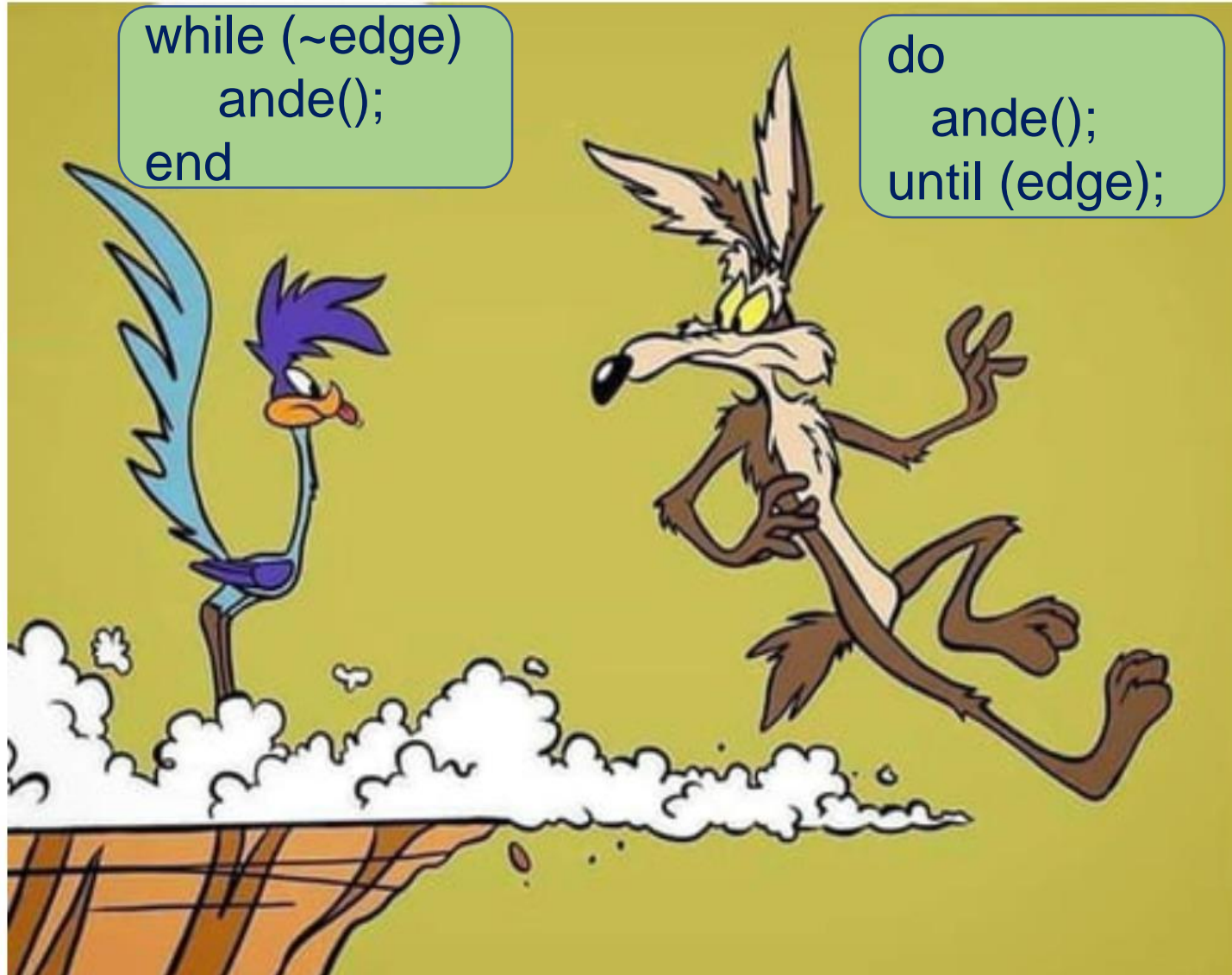
Operações

`until (condição)`

Comandos de Repetição (do-until) - Exemplo

```
contador = 1;  
  
do  
  
    printf("%d ", contador);  
  
    contador++;  
  
until( contador > 10 )
```

do-until VS while



Exercício 11

11. Faça um Script para ler um número do usuário e determine se ele é primo ou não.

- **OBS:** O Script deve repetir o processo acima lendo números do usuário e verificando se o número digitado é primo até que o mesmo digite o número -1 para sair.



Dúvidas



Referências

- Matlab para Cálculo Numérico. Prof. Alan Afif Helal.
- Curso Online Grátis (**em inglês**)
 - <https://www.udemy.com/learn-matlab-using-octave-online/>

Próxima aula ...

- **Aula 3** - Noções Básicas sobre Erro.