

# Minicurso de Octave/Matlab

Uma apresentação de ferramentas básicas para utilização

Prof. Fernando Gonçalves de Almeida Neto

2 de Dezembro de 2016



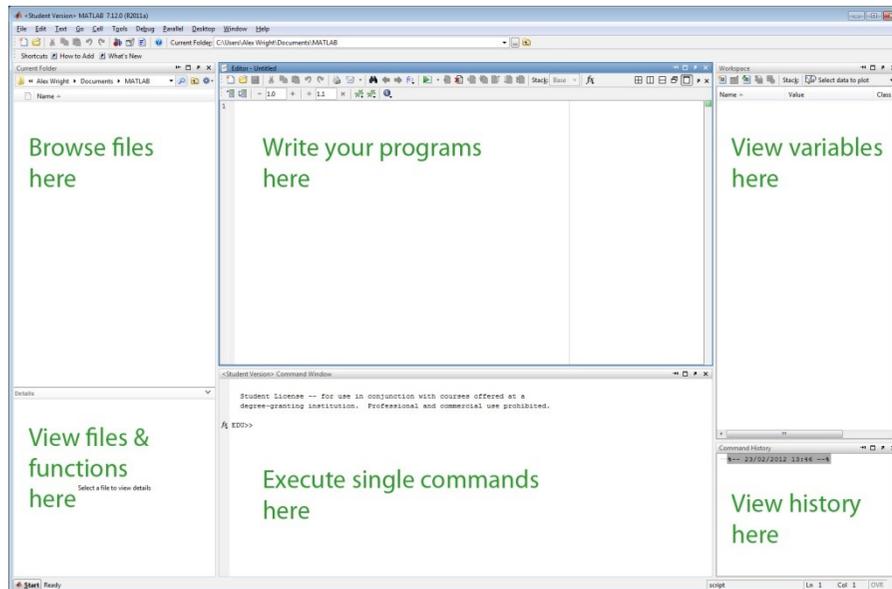


# Octave x Matlab

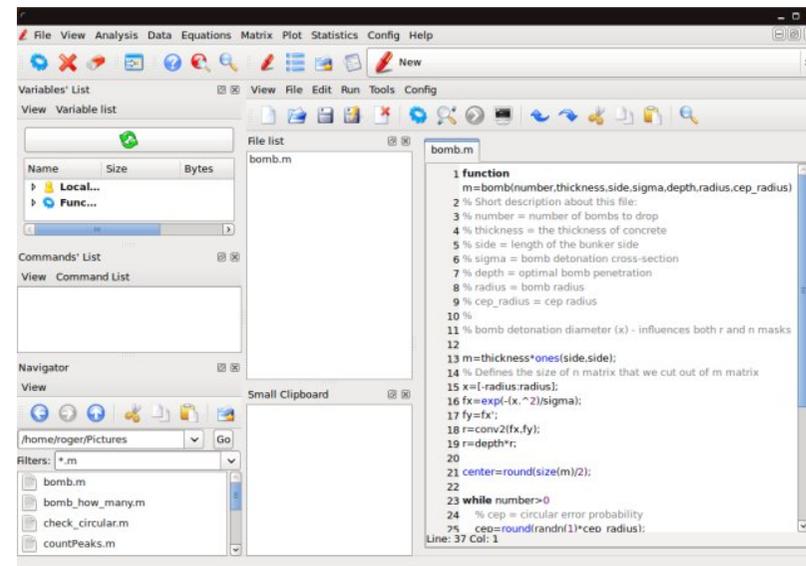


- *Softwares* para aplicação em simulação numérica intensiva
- **Octave**: *software* livre (pode ser obtido livremente online)
- **Matlab**: *software* proprietário (versão estudantil por \$29)
- Semelhanças: sintaxe de programação e funções semelhantes
- Diferenças: *Toolboxes* específicas e tempo de execução de código
- Ambos possuem distribuições Linux e Windows

# Interfaces



Matlab



Octave

# Janela de comando x *Scripts*

- Janela de comando: “calculadora científica” em que as operações ficam “empilhadas”
- *Scripts*: listagens de código que são realizadas sequencialmente
  - *Scripts* são salvos em arquivos *.m*

# Operações básicas com escalares e vetores (I)

- Não precisamos definir as variáveis como double, float, int ...
- Apenas atribuímos os valores às variáveis
- Sintaxe:
  - $a = 1$  (escalar)
  - $b = [1 \ 2.65 \ 3.333]$  (vetor linha)
  - $c = [4; 5; 6]$  (vetor coluna)

# Operações básicas com escalares e vetores (II)

## ➤ Escalares:

- pi

- +, -, \*, /, \, ^, exp(), log(), log10(), sqrt(), sin(), cos()...

- Complexos: 1i ou 1j

- Exemplos

## ➤ Operações com vetores:

- Operações vetor - escalar

- Operações vetor – vetor

- Exemplos

- Problema: média ponderada

## Operações básicas com escalares e vetores (III)

- Obtendo um elemento de um vetor

- Exemplo:  $a = [1 \ -2 \ 6 \ -4]$

- $a(2) = -2$

- $a(3) = 6$

- ***Diferentemente de linguagem C, C++, ..., em Octave/Matlab, o primeiro elemento do vetor é o  $a(1)$***

- Vetor com elementos sequenciais

- $b = 1:1:10$

- $C = -2: 2:10$

- $D = 10: -1:3$

## Operações básicas com escalares e vetores (IV)

➤ Exemplo:

➤  $f = [1 -2 3 0 0.3 4]$

➤ Obter o valor máximo e o valor mínimo de  $f$  →  $\max(f)$  e  $\min(f)$

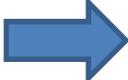
➤ Obter o número de elementos no vetor →  $\text{length}(f)$

➤ Obter o vetor  $g = f'$  (o transposto do vetor  $f$ )

➤ Obter o vetor  $h = \{\text{os dois primeiros elementos de } f\}$

$$h = f([1 2]) \text{ ou } h = f(1:1:2)$$

## Operações básicas com matrizes (I)

- $A = [1\ 2\ 3; 4\ 5\ 6]$   usamos o ponto-e-vírgula para mudar de linha
- $A(2, 3)$   o terceiro elemento da linha dois de A
- $A(1, 2:1:4)$  ou  $A(1, [2\ 3\ 4])$   os elementos 2, 3 e 4 da primeira linha
- Exemplo:
  - $A = [1\ 2\ 3; 4\ 5\ 6]$
  - $B = [2\ 3\ 4; 5\ 6\ 7]$
  - $C = [1\ 2; 3\ 4; 5\ 6]$ 
    - $G = 10*A - 5*B$
    - $H = \exp(A)$
    - $D = A + B$
    - $E = A*C$
    - $F = C*A$

## Operações básicas com matrizes (II)

➤  $A = [1 \ 0 \ 0; 4 \ 5 \ 6; 78 \ 9]$   matriz quadrada

➤  $\det(A)$ : determinante de A

➤  $\text{rank}(A)$ : posto de A

➤  $\text{size}(A)$ : dimensões de A

## Operações básicas com matrizes (III)

### Operações elemento-a-elemento

➤  $A = [100 \ 200 \ 300; \ 400 \ 500 \ 600]$

➤  $B = [0.01 \ 0.02 \ 0.03; \ 0.09 \ 0.10 \ 0.08]$

➤ Calcule a operação elemento-a-elemento:  $C = A .* B$

***Os operadores  $.*$ ,  $./$ ,  $.^$ , etc realizam operações elemento-a-elemento em uma matriz ou vetor***

## Operações básicas com matrizes (IV)

➤ Resolução de sistemas lineares

➤  $A*x = b$

➤  $A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 0]$  e  $b = [14; 32; 23]$  . Qual o valor de  $x$ ?

➤ Lembre de Álgebra Linear:

$$A*x = b$$

$$(A^{-1}*A) x = A^{-1}*b$$

$$x = A^{-1}*b$$

*Podemos usar  $\text{inv}(A)$  ou o operador  $\backslash$  para resolver*

## Alguns comandos úteis

- Variável ans
- Ponto-e-vírgula
- clc e clear
- % (para comentários)

- O mais importante comando do Octave/Matlab:

**help**

- **Exemplo: help cos**

# Scripts

- Listagem de códigos executados sequencialmente
- Exemplo:
  - Vá para a aba editor e crie um arquivo *teste.m*
  - Defina um o vetor  $A = [1 \ 2 \ 0; 0 \ 1 \ 2; 0 \ 0 \ 2]$
  - Defina o vetor coluna  $b = [1; 1; 1]$
  - Calcule a solução de  $Ax = b$
  - Defina  $c = [1; 2; 3]$
  - Calcule a solução de  $Ay = c$
  - Salve o arquivo e pressione *run*

## *Figuras em Octave/Matlab (I)*

- Comando mais básico: plot
- Usar: `graphics_toolkit('fltk')`
- Sintaxe básica:
  - `plot(a, b)`
    - a: vetor de dimensão N
    - b: vetor de dimensão N

## *Figuras em Octave/Matlab (II)*

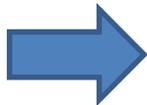
➤ Exemplo:

➤ Crie o arquivo plotagem.m

```
a = 1:1:100;           % vetor linha com valores de 1 até 100
```

```
b = sin(2*pi/100*a);  % amostras com os valores do seno
```

```
plot(a, b);
```



O vetor a contém os dados do eixo x e b contém os dados do eixo y

## *Figuras em Octave/Matlab (III)*

- Vamos adicionar título, identificar os eixos e colocar legenda:

```
a =1:1:100;           % vetor linha com valores de 1 até 100
```

```
b = sin(2*pi/100*a); % amostras com os valores do seno
```

```
plot(a, b);
```

```
xlabel('Eixo x');    % identificação do eixo x
```

```
ylabel('Eixo y');    % identificação do eixo y
```

```
title('Seno');        % título
```

```
legend('Figura do seno'); % legenda
```

## *Figuras em Octave/Matlab (IV)*

- Colocar duas figuras na mesma plotagem: comando hold

```
a = 1:1:100;           % vetor linha com valores de 1 até 100
b = sin(2*pi/100*a);  % amostras com os valores do seno
c = cos(2*pi/100*a);  % amostras com os valores do cosseno
```

```
plot(a, b);
hold on;
plot(a, c, 'r');
hold off;
```

```
xlabel('Eixo x');      % identificação do eixo x
ylabel('Eixo y');      % identificação do eixo y
title('Seno e Cosseno'); % título
legend('Figura do Seno ', 'Figura do Cosseno'); % legenda
```

## *Estruturas de controle de fluxo: loops*

### ➤ **for**

➤ Sintaxe (exemplo):

```
a = 0;  
for i = 1:1:4  
    a = a + 1;  
end
```



*Diferentemente de Python, não é a indentação que indica o final do loop: é a partícula end*

### ➤ **while**

➤ Sintaxe (exemplo):

```
a = 0;  
while a < 5  
    a = a + 1;  
end
```

## ***Estruturas de controle de fluxo: condicionais (I)***

➤ if, elseif e else

➤ Sintaxe:

```
If cond_1
    execucao_1
elseif cond_2
    execucao_2
elseif cond_3
    execucao_3
else
    execucao_4
end
```

➤ Operadores: <, >, >=, <=, ~=, ==, &&, ||

## ***Estruturas de controle de fluxo: condicionais (II)***

➤ Exemplo:

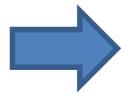
➤  $v = [1 \ -1 \ 4.1 \ 2 \ 5 \ 0 \ 3 \ 1.5 \ 10 \ \pi]$

➤ Fazer a contagem dos elementos de  $v$  que se incluem nos seguintes conjuntos:

➤  $N_b =$  Número de elementos de  $v$  menores do que 2

➤  $N_c =$  Número de elementos de  $v$  maiores do que 4

➤  $N_d =$  Número de elementos de  $v$  entre 2 e 4 (incluindo 2 e 4)}



Use uma estrutura condicional e for

## *Estruturas de controle de fluxo: condicionais (III)*

```
v = [1 -1 4.1 2 5 0 3 1.5 10 pi];
```

```
N= length(v);    % obtém o número de elementos de v
```

```
Nb = 0;          % Número de elementos menor do que 2
```

```
Nc = 0;          % Número de elementos maior do que 4
```

```
Nd = 0;          % Número de elementos entre 2 e 4
```

```
for i = 1:1:N
```

```
    if v(i) < 2
```

```
        Nb = Nb + 1;
```

```
    elseif v(i) > 4
```

```
        Nc = Nc + 1;
```

```
    else
```

```
        Nd= Ncd+ 1;
```

```
    end
```

```
end
```

## *Funções (I)*

**Sintaxe:**

```
function [saida1, saida2, ..., saidaN] = nome(entrada1, ..., entradaM)
```

**Importante:** *o arquivo em que a função é salva e o nome da função devem ser iguais!!!*

## Funções (II)



Generalizar este código para que calcule Nb, Nc e Nd para qualquer vetor v (com qualquer número de elementos)

```
v = [1 -1 4.1 2 5 0 3 1.5 10 pi];
```

```
N = length(v); % obtém o número de elementos de v
```

```
Nb = 0; % Número de elementos menor do que 2
```

```
Nc = 0; % Número de elementos maior do que 4
```

```
Nd = 0; % Número de elementos entre 2 e 4
```

```
for i = 1:1:N
```

```
    if v(i) < 2
```

```
        Nb = Nb + 1;
```

```
    elseif a(i) > 4
```

```
        Nc = Nc + 1;
```

```
    else
```

```
        Nd = Nd + 1;
```

```
    end
```

```
end
```

## *Funções (III)*

% Esta funcao recebe um vetor a de N elementos e conta o numero de entradas,  
% da seguinte forma:

%

% Nb = numero de entradas menores que 2

% Nc = numero de entradas maiores que 4

% Nd = numero de entradas entre 2 e 4, incluindo esses valores

function [Nb, Nc, Nd] = conta\_elementos(v)

N= length(v);    % obtém o número de elementos de v

Nb = 0;            % Número de elementos menor do que 2

Nc = 0;            % Número de elementos maior do que 4

Nd = 0;            % Número de elementos entre 2 e 4

for i = 1:1:N

    if v(i) < 2

        Nb = Nb + 1;

    elseif a(i) > 4

        Nc = Nc + 1;

    else

        Nd= Nd + 1;

    end

end

## *Funções (IV)*

Obter uma função que tem como entrada os coeficientes de um polinômio  $p(x)$  e tem como saída os coeficientes do polinômio derivado  $q(x)$ .

Exemplo:

$$p(x) = x^3 + 3x^2 + 5x + 4 \quad \xrightarrow{\text{derivada}} \quad q(x) = 3x^2 + 6x + 5$$

function [q] = derivada(p)

$p = [1 \ 3 \ 5 \ 4]$   entrada

$q = [3 \ 6 \ 5]$   saídas

## *Funções (V)*

% Esta funcao recebe os coeficientes de um polinomio no vetor linha p (na  
% ordem decrescente da potencia )e retorna o vetor q, com os coeficientes do  
% polinomio derivado.

```
function [q] = derivada(p)
```

```
N = length(p);
```

```
M = N -1;
```

```
if M == 0
```

```
    q = 0;
```

```
else
```

```
    for i = 1:1:M
```

```
        q(i) = p(i)*(N - i);
```

```
    end
```

```
end
```

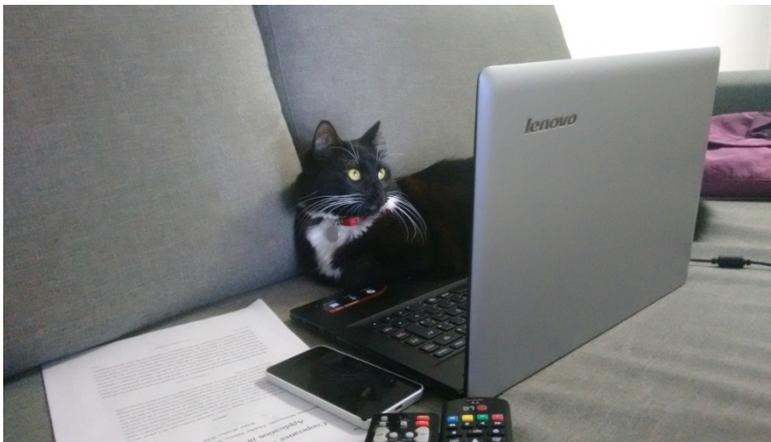
## ***Referências Úteis***

1. Apostila Introdutória desse curso:  
<http://www.eletronica.uacsa.ufrpe.br/documentos>
2. Matlab Guide - Desmond J. Higham e Nicholas J. Higham - Second edition, Siam
3. A Matlab Guide for Beginners and Experienced Users - Brian R. Hunt, Ronald L. Lipsman e Jontahan M. Rosenberg - Second edition, Cambridge
4. Matlab for Engineers - Adrian Biran - 2. ed. England: Addison Wesley (Disponível na biblioteca da UACSA)
5. <http://www.mathworks.com/>
6. <http://www.gnu.org/software/octave/> (Para *download* e informações sobre o Octave)
7. <http://www.castilho.prof.ufu.br/cn/Octave.pdf> (Mais uma apostila básica de Octave)



# *O segredo para aprender Octave/Matlab*

PRATICAR!



# Obrigado!

[fernando.galmeida@ufrpe.br](mailto:fernando.galmeida@ufrpe.br)



## *Tópicos adicionais: matrizes especiais*

Matriz de zeros:  $A = \text{zeros}(a, b)$    $a = \text{número de linhas}$   
 $b = \text{número de colunas}$

Matriz identidade :  $A = \text{eye}(a, b)$

Matriz aleatória:  $A = \text{rand}(a, b)$   Distribuição Uniforme entre 0 e 1

$A = \text{randn}(a, b)$   Distribuição Normal média 0 e  
variância 1

## ***Tópicos adicionais: Polinômios***

roots(p)



Encontra as raízes de um polinômio (colocar os coeficientes multiplicativos de todas as potências, da ordem mais elevada para a menor)

poly(q)



Obtém o polinômio cujas raízes estão listadas no vetor q

## ***Tópicos adicionais: autovalores e autovetores***

$\text{eig}(A)$   Retorna uma lista com os autovalores de A

$[V \ D] = \text{eig}(A)$   Retorna uma matriz V cujas colunas são os autovetores da matriz A e a matriz diagonal D