

# Introdução à Ciência da Computação

Programação de Computadores com  
MATLAB/Octave

Uso do Octave no Modo Interativo

Prof. Roberto M. de Faria/DSC/UFCG

# Ajuda para o Octave

- Para ter acesso à documentação *on line* do Octave, clique na aba “Documentação”, abaixo da Janela de Comandos, para ter acesso à janela de Documentação
- Digite na Área de Comandos o comando **help**, seguido de um nome de comando, nome de função ou de um “operador” (o operador deve estar entre aspas), seguido da tecla **<Enter>**, para obter ajuda sobre um item específico destes

# Janela da Documentação

The screenshot displays the Octave software interface. The main window is titled "Documentação" and shows the "GNU Octave Manual" for version 5.2.0. The interface includes a menu bar with "Arquivo", "Editar", "Depurar", "Janela", "Ajuda", and "Novidades". The "Navegador de Arquivos" (File Navigator) shows the current directory as "C:/Users/rober/Dropbox/\_Disciplinas/ICC/Periodo-2020.1". The "Ambiente de Trabalho" (Work Environment) section contains a table with columns for "Nome", "Classe", "Dimensão", "Valor", and "Atributo". The "Histórico de Comandos" (Command History) section is also visible. The documentation window displays the "GNU Octave (version 5.2.0)" title, copyright information, and a "Table of Contents" with various sections and sub-sections.

Octave

Arquivo Editar Depurar Janela Ajuda Novidades

Diretório Atual: ber\Dropbox\\_Disciplinas\ICC\Periodo-2020.1

Navegador de Arquivos

C:/Users/rober/Dropbox/\_Disciplinas/ICC/Periodo-2020.1

Nome

- execicios\_files
- exemplos - ICC - 2020.1
- exemplos\_files
- index\_arquivos
- listas
- listas\_t1\_arquivos
- listas\_t2\_arquivos

Ambiente de Trabalho

Filtrar

Nome	Classe	Dimensão	Valor	Atributo
------	--------	----------	-------	----------

Histórico de Comandos

Filtrar

Documentação

Conteúdo Índice de Funções Busca

GNU Octave Manual

## GNU Octave (version 5.2.0)

Copyright © 1996-2020 John W. Eaton.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

### Table of Contents

- [Preface](#)
  - [Acknowledgements](#)
  - [Citing Octave in Publications](#)
  - [How You Can Contribute to Octave](#)
  - [Distribution](#)
- [1 A Brief Introduction to Octave](#)
  - [1.1 Running Octave](#)
  - [1.2 Simple Examples](#)
    - [1.2.1 Elementary Calculations](#)
    - [1.2.2 Creating a Matrix](#)
    - [1.2.3 Matrix Arithmetic](#)
    - [1.2.4 Solving Systems of Linear Equations](#)
    - [1.2.5 Integrating Differential Equations](#)
    - [1.2.6 Producing Graphical Output](#)
    - [1.2.7 Help and Documentation](#)
    - [1.2.8 Editing What You Have Typed](#)
  - [1.3 Conventions](#)
    - [1.3.1 Fonts](#)
    - [1.3.2 Evaluation Notation](#)
    - [1.3.3 Printing Notation](#)
    - [1.3.4 Error Messages](#)
    - [1.3.5 Format of Descriptions](#)
      - [1.3.5.1 A Sample Function Description](#)
      - [1.3.5.2 A Sample Command Description](#)
- [2 Getting Started](#)
  - [2.1 Invoking Octave from the Command Line](#)
    - [2.1.1 Command Line Options](#)
    - [2.1.2 Startup Files](#)
  - [2.2 Quitting Octave](#)
  - [2.3 Commands for Getting Help](#)
  - [2.4 Command Line Editing](#)
    - [2.4.1 Cursor Motion](#)
    - [2.4.2 Killing and Yanking](#)
    - [2.4.3 Commands for Changing Text](#)

# Comandos Básicos do Octave

- Operadores Aritméticos (por precedência)
  - “^”           potenciação ou exponenciação
  - “\*” e “/”       multiplicação e divisão
  - “+” e “-”       soma e subtração
- A precedência dos operadores é a mesma da álgebra da matemática
- Numa expressão aritmética, usa-se parênteses para alterar a precedência dos operadores
- No Octave, usa-se ponto decimal “.” para separar as casas decimais de números reais

# Comandos Básicos do Octave

- Comandos para cálculos básicos
  - Encerra-se um comando com a tecla **<Enter>**
  - Ex:
    - >> 2 + 5
    - >> 7.8 + 3.5
    - >> 6 \* 5
    - >> 2.5 / 5
    - >> 2 ^ 7
    - >> (2 + 5) \* 7
  - O resultado da execução de um comando, não atribuído a uma variável, fica armazenado na variável “**ans**” (answer – resposta) do Octave

# Reedição dos Comandos já Submetidos

- Utilizando-se as teclas “**seta-para-cima**” e “**seta-para-baixo**”, pode-se acessar comandos já submetidos à execução anterior
- Um comando já executado pode ser reeditado com modificações e re-submetido à execução ou simplesmente ser reexecutado sem modificações
- Clicar duas vezes sobre um comando na “Área de Histórico”, faz com que este comando seja reexecutado

# Exercícios

- Calcule a energia cinética de uma partícula que tem a massa de 0,0023 kg e a velocidade de 3235 m/s
- Calcule o IMC de uma pessoa de peso 84,5 kg e altura de 1,72 m
- Calcule a área de um círculo e sua circunferência, sabendo que seu raio é 60 cm
- Calcule o valor final, após um mês, de um investimento de R\$ 1.000,00 numa caderneta de poupança, sabendo que a inflação do período foi de 2,37% e o juro mensal é de 0,5%

# Relembrando Variáveis

- Uma variável é simplesmente uma porção (parte) da memória do computador usada para armazenar algum dado.
- Uma variável possui um nome (um único caractere ou um conjunto de caracteres) que é por meio dele que se referencia aquela parte da memória. Um nome de variável pode conter letras, números e/ou um sublinhado, mas deve começar com uma letra. Além disso, os nomes de variáveis diferenciam maiúsculas de minúsculas. Ex.: a variável “**x**” é diferente da variável “**x**”.

# Relembrando Variáveis

- Em geral, uma variável pode armazenar:
  - um número (inteiro, real, complexo,...)
  - uma cadeia de caracteres (texto)
  - uma matriz de números, sequências de caracteres e/ou outras matrizes
  - um apontador para uma função anônima.

# Ainda sobre Variáveis

- Armazenamento numa Variável
  - Quando atribui-se um valor a uma variável, ela passa a existir e é mostrada na Área de Variáveis
    - Ex:

```
>> a = 2
>> b = 9
>> c = 7.5
>> delta = b ^ 2 - 4 * a * c
>> x1 = (-b + sqrt(delta)) / (2 * a)
>> x2 = (-b - sqrt(delta)) / (2 * a)
```
- Obs: “`sqrt()`” é uma função pré-definida do Octave

# Ainda sobre Variáveis

- Uma variável pode armazenar também um texto
  - Ex: `>> titulo = "Raízes de uma Equação do Segundo Grau"`
- O nome de uma variável, como comando, faz com que seu valor seja apresentado
  - Ex: `>> delta`
- O comando "**who**" mostra as variáveis já definidas
  - Ex: `>> who`
- O comando "**clear**" seguido do nome da variável remove a variável da memória
  - Ex: `>> clear delta`
- O comando "**clear all**" remove todas as variáveis definidas da memória
  - Ex: `>> clear all`

# Variáveis do Octave

- O Octave possui algumas variáveis com valores pré-definidos:
  - **nan** → not-a-number
  - **inf** → infinito
  - **pi** → 3.1416
  - **e** → 2.7183 – base do logaritmo natural (constante de Euler)
  - **i** → 0.0000 + 1.0000i → raiz quadrada de -1
  - **j** → 0.0000 + 1.0000i → raiz quadrada de -1
  - **eps** → 2.2204e-016 – epsilon – constante da máquina de dupla precisão em ponto flutuante
  - **true** → valor lógico verdadeiro
  - **false** → valor lógico falso

# Tipos de Variáveis

- Toda a variável tem um tipo a ela associado
- Os tipos para as variáveis do Octave são:
  - `"cell"` → arrays de células
  - `"struct"` → arrays de estruturas
  - `"logical"` → arrays lógicos
  - `"int8"` → inteiros com 1 byte
  - `"int16"` → inteiros com 2 bytes
  - `"int32"` → inteiros com 4 bytes
  - `"int64"` → inteiros com 8 bytes
  - `"float"` ou `"single"` → real de ponto flutuante com 4 bytes (4 decimais)
  - `"double"` → real de ponto flutuante com 8 bytes (14 decimais)

# Mais sobre Tipos de Variáveis

- Outros tipos para as variáveis do Octave são:
  - `"uint8"` → inteiros sem sinal com 1 byte
  - `"uint16"` → inteiros sem sinal com 2 bytes
  - `"uint32"` → inteiros sem sinal com 4 bytes
  - `"uint64"` → inteiros sem sinal com 8 bytes
  - `"char"` → para arrays de string
- Criação de variável usando tipo:
  - `>> x = int32(123)`
- Mudança de tipo com a função `cast()` :
  - `>> y = cast(x, "int64")`

# Múltiplos Comandos

- Octave permite que se coloque múltiplos comandos numa mesma linha de comando separados por “;”
  - Ex: `>> x = 7; y = x * 5; z = x + y`
- O “;” no final de um comando, faz com que o Octave não mostre a resposta do cálculo
  - Ex:  
`>> raio = 30;`  
`>> pi * raio ^ 2`

# Comentários e Limpeza da Área de Comandos

- Uma linha de comando que começa com “%” é um comentário
- Um comentário é colocado para acrescentar alguma informação sobre os comandos que estão sendo executados
- Um comentário não é executado, ou seja não interfere na execução de outros comandos nem no armazenamento de dados
- O comando “**clc**” executa a limpeza da “Área de Comandos”

# Funções Pré-definidas

- Uma função é um conjunto de comandos que possui um nome e que, quando executados, produzem um resultado
- O Octave possui muitas funções pré-definidas
  - Exemplos de algumas funções:
    - **abs ()** → valor absoluto
    - **max ()** → máximo
    - **min ()** → mínimo
    - **sum ()** → soma
    - **imag ()** → parte imaginária de um número complexo
    - **round ()** → arredonda para o inteiro mais próximo

# Mais Funções Pré-definidas

– Exemplos de algumas funções trigonométricas (usa radianos):

- **sin ()** → seno
- **cos ()** → cosseno
- **tan ()** → tangente
- **sec ()** → secante
- **csc ()** → cossecante
- **cot ()** → cotangente

# Mais Funções Pré-definidas

– Exemplos de mais algumas funções trigonométricas (retorna radianos):

- `asin()` → arco seno
- `acos()` → arco cosseno
- `atan()` → arco tangente
- `asec()` → arco secante
- `acsc()` → arco cossecante
- `acot()` → arco cotangente

# Mais Funções Pré-definidas

- Exemplos de mais algumas funções trigonométricas (usa graus):
  - `sind()` → seno
  - `cosd()` → cosseno
  - `tand()` → tangente
  - `secd()` → secante
  - `cscd()` → cossecante
  - `cotd()` → cotangente
- Existem outras funções tais como: logaritmo, exponencial, raiz quadrada, etc., que podem ser pesquisadas na documentação do Octave

# Formatos

- O Octave pode trabalhar com formatos de precisão (número de dígitos após o ponto decimal) diferentes
- A precisão pode ser alterada com os comandos “**format long**” e “**format short**”

- Ex:

```
>> format short
```

```
>> pi
```

```
ans =
```

```
3.1416
```

```
>> format long
```

```
>> pi
```

```
ans =
```

```
3.14159265358979
```

# Formato de Notação Científica

- O Octave pode utilizar o formato de notação científica com diferentes precisões

- Ex:

```
>> format short e
```

```
>> pi
```

```
ans =
```

```
3.1416e+000
```

```
>> format long e
```

```
>> pi
```

```
ans =
```

```
3.14159265358979e+000
```

- O Octave pode retornar à formatação padrão digitando-se:

```
>> format short
```

# Vetores e Matrizes

- Um vetor ou uma matriz é uma coleção de valores (elementos) de mesmo tipo que possui um nome e um conjunto de índices associados
- Os elementos de um vetor ou matriz são acessíveis por meio de um conjunto de índices

- Criação de um vetor:

```
>> vet = [-5, 20, -33, 15, 0]
```

- Criação de uma matriz:

```
>> mat = [-5, 7, -12, 0; 20, 100, -3,  
0; 88, -33, 15, 0]
```

# Mais sobre Vetores e Matrizes

- O acesso a um elemento de um vetor é realizado por meio do nome do vetor ou matriz seguido do conjunto de índices do elemento entre parênteses
- Acesso a um elemento de um vetor:  

```
>> vet(2) * 10
```
- Acesso a um elemento de uma matriz:  

```
>> mat(2, 3) = 10
```

# Mais sobre Vetores e Matrizes

- Criando vetores sequenciais:
  - Para criar vetores sequenciais, utiliza-se o operador “:”
    - Informando o valor inicial e o final:  
`>> vetor = 1 : 10`
    - Informando o valor inicial, o incremento e o final:  
`>> sequencia = 1 : 3 : 10`
    - Informando o valor inicial, o incremento e o final:  
`>> radianos = 0 : pi/4 : 2 * pi`

# Mais sobre Vetores e Matrizes

- Gerando valores randômicos para vetores e matrizes:
  - Para criar vetores e matrizes com valores reais randômicos, pode-se usar a função “**rand()**”
    - Matriz com uma linha e dez colunas (vetor):  

```
>> vetor = rand(1, 10)
```
    - Matriz 10x10:  

```
>> matriz_1 = rand(10, 10)
```
    - Matriz 10x10:  

```
>> matriz_2 = rand(10)
```
    - Matriz de 3 dimensões (2x3x2):  

```
>> matriz_3 = rand(2, 3, 2)
```

# Mais sobre Vetores e Matrizes

- Para criar vetores e matrizes com valores inteiros randômicos, utiliza-se a função `randi()` que retorna inteiros randômicos no intervalo `1:imax`, podendo-se usar um dos seguintes formados:
  - `randi(imax)`
  - `randi(imax, ordem_de_matriz_quadrada)`
  - `randi(imax, linhas, colunas, ...)`
- “[`imin, imax`]” pode ser usado no lugar de `imax` para definir um intervalo
- Ex.: Matriz com uma linha e dez colunas (vetor) com valores randômicos inteiros entre 1 e 15:
  - `>> vetor = randi(15, 1, 10)`
- Ex.: Matriz 10x10 com valores randômicos inteiros entre 1 e 7:
  - `>> matriz_1 = rand(7, 10)`
- Ex.: Matriz 3x4 com valores randômicos inteiros entre 21 e 77:
  - `>> matriz_2 = rand([21 77], 3, 4)`

# Algumas Operações com Matrizes

- Criação de matrizes:

```
>>> A = [10, 22, -3, 14; -1, 12, 37,  
4; 9, -8, 61, 4]
```

```
>>> B = rand(3, 4)
```

```
>>> C = [12, 27; -22, 14; -1, 0; 7, 4]
```

- Soma de matrizes:

```
>>> D = A + B
```

- Subtração de matrizes:

```
>>> E = A - B
```

# Algumas Operações com Matrizes

- Multiplicação de matrizes:

```
>> F = A * C
```

- Multiplicação de matriz por escalar:

```
>> G = A * 2
```

- Cálculo do determinante de uma matriz:

```
>> D = [22, -3, 14; -1, 37, 4; 9, -  
8, 61]
```

```
>> determinante = det(D)
```

# Algumas Operações com Matrizes

- Maior elemento de uma matriz:

```
>> maior = max(max(A))
```

- Menor elemento de uma matriz:

```
>> menor = min(min(A))
```

- Soma dos elementos de uma matriz:

```
>> soma = sum(sum(A))
```

- Soma acumulativa elementos das colunas de uma matriz:

```
>> somaAc = cumsum(D)
```

- Diferença entre elementos das colunas de uma matriz:

```
>> diff([2, 4, 8; 15, 31, 22])
```

# Mais Algumas Operações com Vetores

- Remover um elementos de um vetor:  

```
>> vetor = [22, -3, 14, -1, 37, 4, 9, -8, 61]
```

```
>> vetor(4) = []
```
- Remover vários elementos de um vetor:  

```
>> vetor(2:4) = []
```
- Acrescentar um elemento a um vetor:  

```
>> vetor(5) = 10
```
- Acrescentar vários elementos a um vetor:  

```
>> vetor(6:10) = 1:5
```

# Mais Algumas Operações com Matrizes

- Remover uma linha de uma matriz:

```
>> matriz = [6, 9, 0; 7, -4, 10; 22, -3, 12]
```

```
>> matriz(1, :) = []
```

- Acrescentar várias colunas a uma matriz:

```
>> matriz(3:5, 1:3) = 0
```

- Remover várias colunas de uma matriz:

```
>> matriz(:, 2:3) = []
```

- Acrescentar várias linhas a uma matriz:

```
>> matriz(6:10, 1) = -1
```

# Exercícios

1) A resistência combinada  $R_T$  de três Resistências  $R_1$ ,  $R_2$ , e  $R_3$  em paralelo é dada por

$$R_T = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Crie variáveis para os três resistores e armazene valores em cada um e, em seguida, calcule a resistência combinada.

2) Gere números aleatórios de acordo com o especificado, usando a função rand. Note que a função rand gera números pseudo-aleatórios no intervalo [0, 1):

- número real no intervalo (0, 20);
- número real no intervalo (20, 50);
- inteiro no intervalo inclusivo de 1 a 10;
- inteiro no intervalo inclusivo de 0 a 10;
- inteiro no intervalo inclusivo de 50 a 100.

# Exercícios

3) Uma planta química libera uma quantidade  $A$  de poluente em um córrego. A concentração máxima  $C$  do poluente em um ponto que é uma distância  $X$  da planta é:

$$C = \frac{A}{x} \sqrt{\frac{2}{\pi e}}$$

Criar variáveis para os valores de  $A$  e  $X$  e, em seguida, para  $C$ . Suponha que a distância  $X$  está em metros. Experimente com valores diferentes para  $X$ .  $e$  é a constante de Euler.

4) Usando o operador dois-pontos e também a função `linspace`, crie os seguintes vetores lineares:

- -5 -4 -3 -2 -1
- 5 7 9
- 8 6 4

# Exercícios

5) Encontre uma maneira eficiente de gerar a seguinte matriz:

**mat** =

```
 7  8  9 10
12 10  8  6
```

Então, dê expressões para matriz **mat** que:

- referencie o elemento na primeira linha, terceira coluna
- referencie a segunda linha inteira
- referencie as duas primeiras colunas.

6) Crie uma matriz de números reais aleatórios 3x5 de. Exclua a terceira linha.

# Exercícios

7) Crie uma matriz de inteiros aleatórios 4x6, cada um no intervalo inclusivo de -5 a 5; armazene-o em uma variável. Crie outra matriz que armazena para cada elemento o valor absoluto do elemento correspondente na matriz original.

8) Crie uma matriz 3x5. Execute cada uma das seguintes operações:

- localize o valor máximo em cada coluna.
- localize o valor máximo em cada linha.
- encontre o valor máximo em toda a matriz.

9) Suponha que a função `diff` não exista. Escreva sua própria expressão(ões) para realizar a mesma coisa para um vetor.

# Exercícios

10) Avalie a função  $f$  de duas variáveis  $x$  e  $y$ , onde  $x$  varia de 1 a 2 e  $y$  varia de 1 a 5.

$$f(x, y) = 3 * x - y$$