DACC | Departamento Acadêmico de Ciência da Computação

FUNDAÇÃO UNIVERSIDADE FEDERAL DE RONDÔNIA

Álgebra Linear

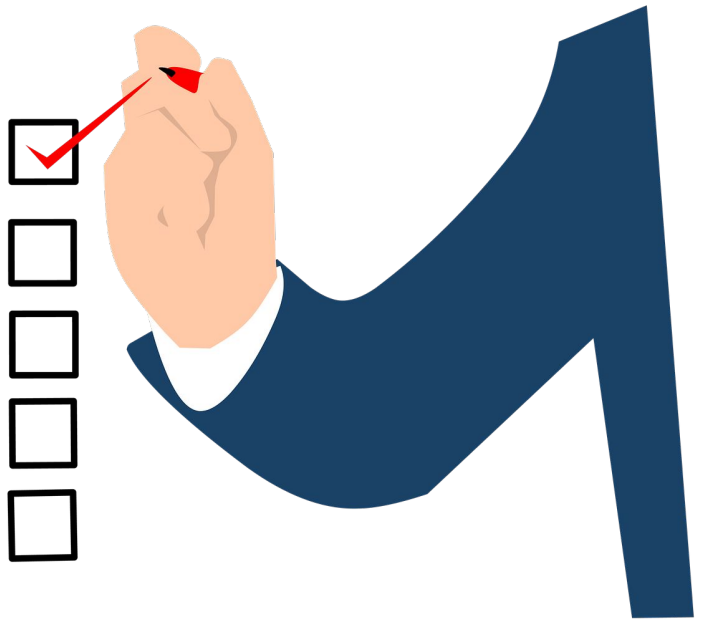
Professor:

Me. Lucas Marques da Cunha

lucas.marques@unir.br

Roteiro

1. Apresentação
2. Horário
3. Metodologia de Avaliação
4. Critérios para desenvolvimento do projeto
5. Cronograma de Aulas
6. Dúvidas e/ou sugestões
7. Introdução ao GNU Octave
8. Exercícios práticos



Apresentação

- Formação
  - Bacharel em Sistemas de Informação - UNIFIP (2012)
  - Especialista em Gestão da Educação Municipal - UFPB (2017)
  - Mestre em Informática - UFPB (2016)
  - Doutorando em Bioinformática - UFRN (Atual)
- Linhas de Pesquisa
  - Processamento de Imagens Digitais: Análise Forense
  - Inteligência Artificial e Aprendizagem de Máquina
  - Softwares Educacionais/Desenvolvimento *mobile*
  - Bioinformática: proteômica e *Big data*
  - Linguagens de programação
    - C, Perl, Java, Octave, Matlab, R e Python.
- Atuação
  - Unidade de Educação a Distância - UFPB (2014 - 2021)
  - Instituto de Educação Profissional - INEP (2014 - 2016)
  - UNINASSAU (2019 - 2021)

Apresentação

1. Você é aluno de qual curso (Licenciatura ou Bacharelado)?
2. Você já cursou Álgebra Linear anteriormente? Se sim, qual foi sua experiência?
3. Você consegue enxergar a relação e/ou aplicação da Álgebra Linear e a Computação?

Horário

COMPONENTE CURRICULAR: Álgebra Linear

PERÍODO: 3º

CARGA HORÁRIA: 80H

HORÁRIO:

Presencial:
Terça-feira (13h50min - 17h20min) - Tarde
Assíncrona:
Sábado (13h50min - 18h00min) - Tarde

Metodologia de avaliação

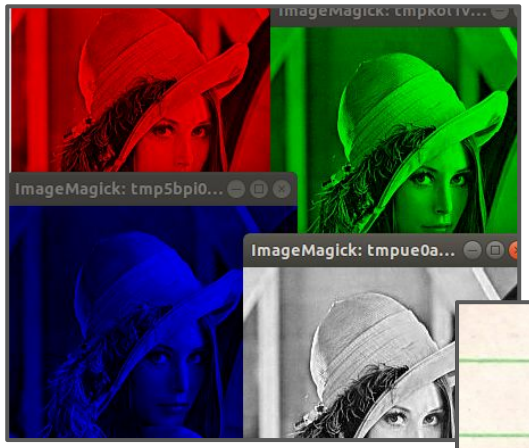
1. Durante o período o letivo serão realizadas 02 (duas) avaliações.
2. **A primeira a avaliação será composta de:**
   a. Prova escrita (peso 7,0)
   b. Lista de Exercícios (peso 3,0)
3. **A segunda avaliação será composta de:**
   a. Prova escrita (Peso 4,0)
   b. Projeto (Peso 3,0)
   c. Lista de Exercícios (Peso 3,0)
4. Se **média final < 60** o aluno fará avaliação repositiva nos termos regimentais da UNIR.
5. A **avaliação repositiva** irá substituir o menor das notas ($M1$ ou $M2$). Então se calculará novamente a **média final**.

Critérios para o desenvolvimento do projeto

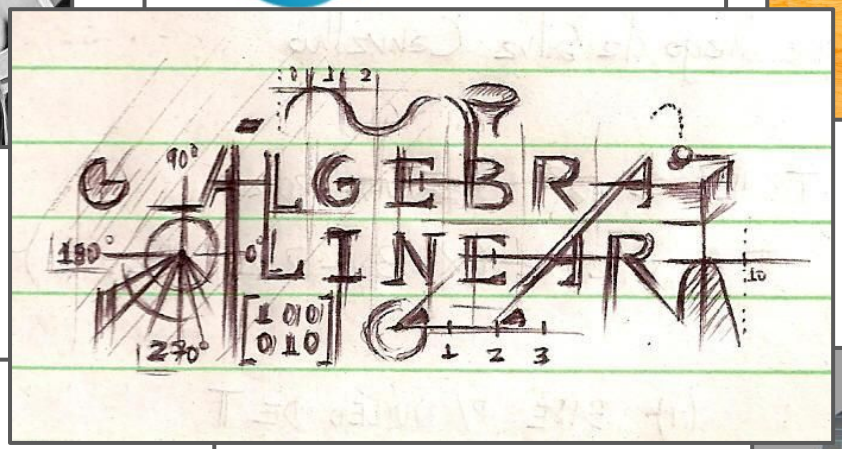
Critérios para o desenvolvimento do projeto

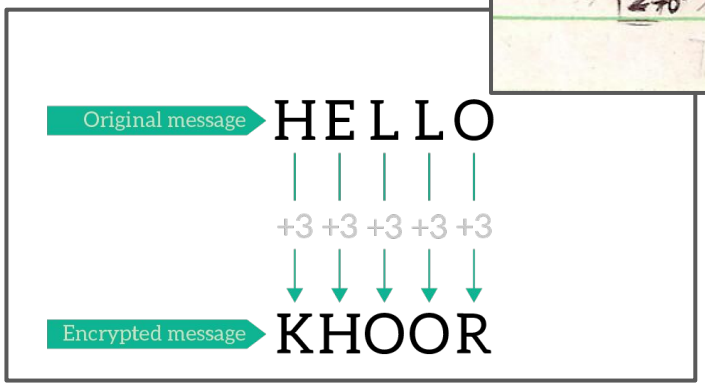
UNIR

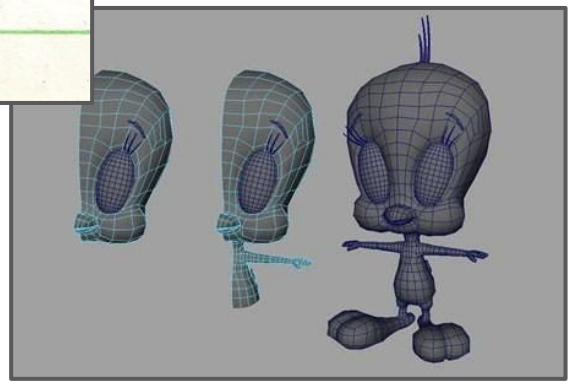
O que deverá ser desenvolvido?



GNU Octave







Critérios para o desenvolvimento do projeto

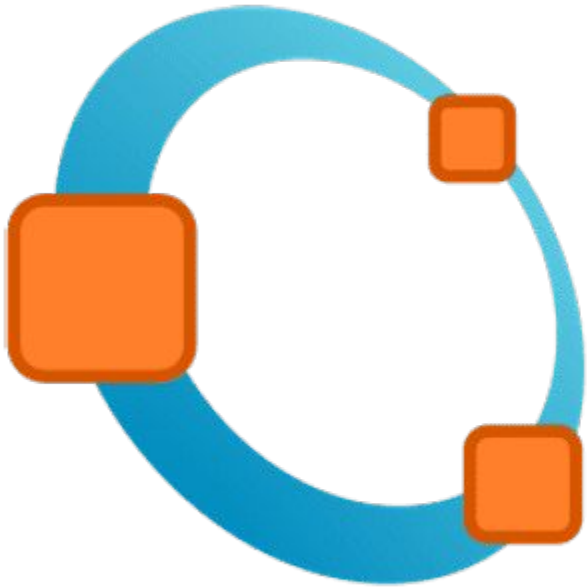
1. A equipe deve ser composta por, no máximo, **3 membros**.
2. A equipe deverá selecionar um **tópico** referente à aplicação da Álgebra Linear na Computação:
   a. Criptografia
   b. Inteligência Artificial
   c. Jogos de Estratégia
   d. Processamento de Imagens
   e. Computação Gráfica
3. Como deve ser feito:
   a. GNU Octave
   b. Apresentação (Seminário)
4. **Prazo para entrega: 27/12!**

Cronograma de aulas

Dúvidas, sugestões?

Introdução ao GNU Octave

UNIR



GNU Octave

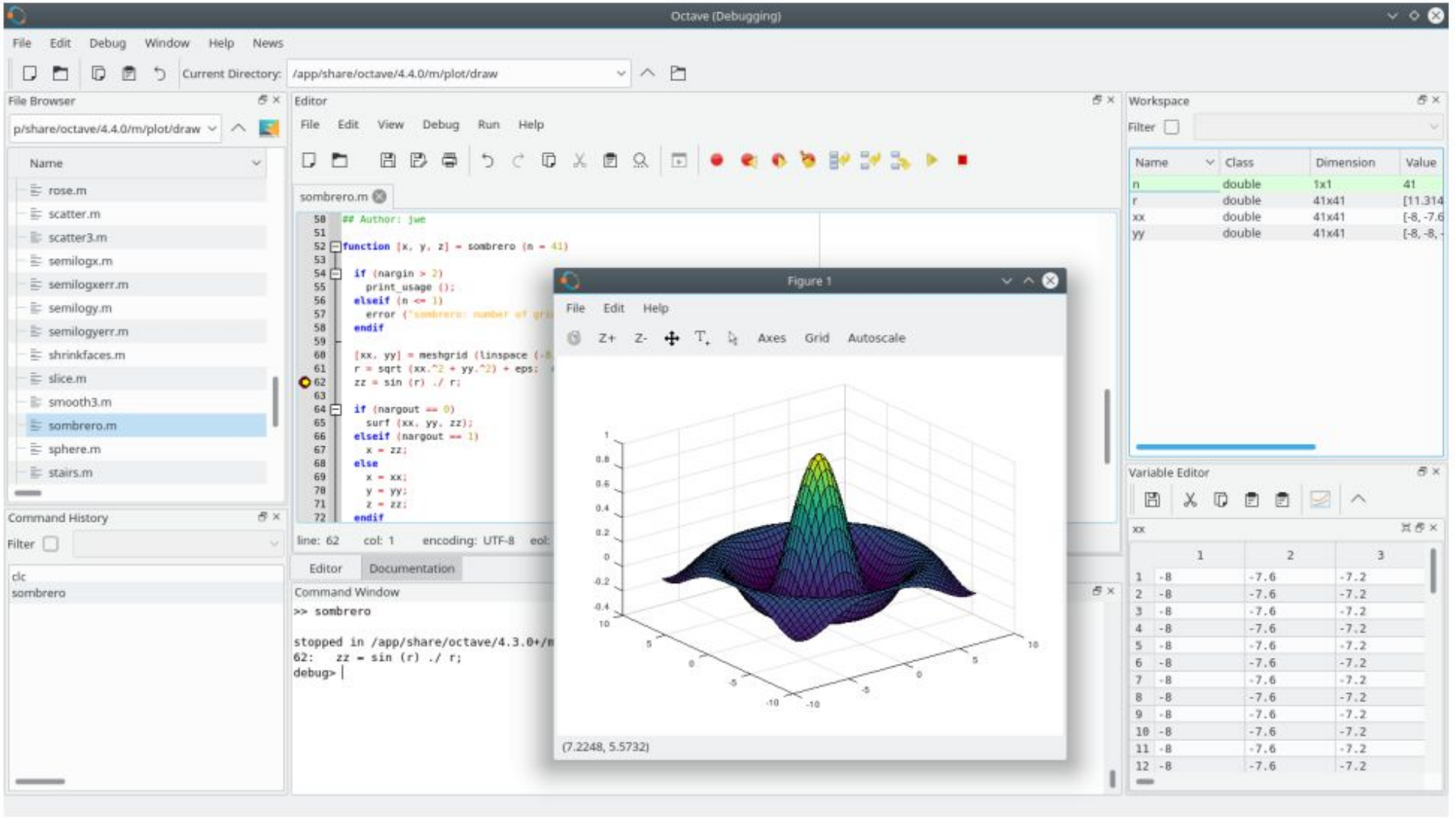
Objetivos de aprendizagem

- Aprender ferramentas e IDE’s para programação científica Octave;
- Estudar conceitos teóricos e práticos sobre variáveis e operadores aritméticos.
- Estudar conceitos teóricos e práticos sobre vetores e matrizes.

O que é Octave?

- O Octave (oficialmente GNU-Octave) é uma **linguagem de alto nível interpretada** que destina-se principalmente para cálculos numéricos.
- O Octave é um “clone” **open-source** do Matlab.
- O Octave pode ser usado como uma “calculadora com muitos recursos” ou como um ambiente de programação.
- Interativo e expansível;
- Disponível online: https://octave-online.net/

Qual a cara do Octave?



Comandos básicos

- **Símbolo >>**
  - Ao abrir o Octave, o símbolo >> aparece na Janela de Comandos.
  - Ele indica que o programa está aguardando comandos.
  - Se o símbolo >> não estiver aparecendo, significa que o Octave ainda está rodando algum programa.

Exercício

Digite: `2+3 <enter>`

Resposta: `ans = 5`

A resposta (ANSwer) do comando que você digitou é 5.

Agora digite: `2+3*5 <enter>`

Resposta: `ans = 17`

E agora: `(2+3)*5`

Resposta: `ans = 25`

GNU Octave

Comandos básicos

- **Janela de comandos**
  - A Janela de Comandos é onde você usa o Octave de forma interativa.

```
Janela de Comandos
>> 1/7
ans =  0.14286
>> format long
>> 1/7
ans =    1.42857142857142e-01
>>
```

Comentário, variáveis e o “;”

```
>> # O Octave ignora tudo que vier após o caracter #, assim
>> # isso é uma ótima forma de acrescentar comentários
>>
>> p1 = 8.6;   # note que o valor de p1 não é exibido
>> p2 = 7.3    # porém o valor de p2 sim
p2 =  7.3000
>> mt = 9.5;
>>
>> m = 0.3 * p1 + 0.4 * p2 + 0.3 * mt;
>> m
m =  8.3500
>>
```

O comando **diary**

- O comando **diary** permite gravar em um arquivo texto um diário (ou um registro) de tudo que for feito dentro do Octave.

```
>> y = sin(8*pi/9)
y =  0.34202
>> diary registro.txt     # Salvando a partir daqui
>> x = 28;
>> v = sin(x)^2 + cos(x)^2
v =  1
>> diary off              # Encerra o salvamento em disco
>> x+y
ans =  28.342
```

Operadores aritméticos

Operação	Símbolo	Exemplo
Adição	+	5 + 3
Subtração	–	5 – 3
Multiplicação	*	5 * 3
Divisão	/	5 / 3
Exponenciação	^	5 ^ 3 (significa $5^3 = 125$)

Operações matemáticas

- Ordem em que o Octave faz as operações:

Ordem	Operação Matemática
Primeiro	Parênteses. Para vários parênteses, o que estiver por dentro é executado primeiro
Segundo	Exponenciação
Terceiro	Multiplicação, divisão (mesma ordem)
Quarto	Adição e subtração

- Se duas ou mais operações tiverem a mesma ordem de precedência, a expressão mais à esquerda será executada primeiro.

Operadores lógicos

- No GNU Octave, o valor lógico verdadeiro é escrito como **true** e o valor lógico falso como **false**.
- Temos os seguintes operadores lógicos disponíveis:
  - & e lógico
  - | ou lógico
  - ! negação
  - == igualdade
  - != diferente
  - < menor que
  - > maior que
  - <= menor ou igual que
  - >= maior ou igual que

Funções matemáticas elementares

UNIR

- `exp(x) - e^x`
- `abs(x) - valor absoluto`
- `log(x) - logaritmo natural (base e)`
- `log10(x) - logaritmo na base 10`
- `sqrt(x) - raiz quadrada`
- `factorial(x) - x!`
- `sin(x) - seno (x em radianos)`
- `sind(x) - seno (x em graus)`

GNU Octave

Funções matemáticas elementares

UNIR

- `cos(x) - coseno (x em radianos)`
- `cosd(c) - coseno (x em graus)`
- `tan(x) - tangente (x em radianos)`
- `tand(x) - tangente (x em graus)`
- `cot(x) - cotangente (x em radianos)`
- `cotd(x) - cotangente (x em graus)`
- `mod(x,y) - resto da divisão inteira entre x e y.`

GNU Octave

Exercícios

1. **Experimente as operações abaixo no Octave. Entenda os resultados.**
   a. ~0
   b. ~1
   c. ~4
   d. [1 0 1] & [1 0 0]
   e. [1 3 4] & [1 2 1]
   f. 1 + (4 <=5)
   g. x = 4; y = x; x == y
   h. x = 3; y = 4; x == y
   i. 1 < 4; y = ans; z = y == 1

GNU Octave

Exercícios

2. **Informe o resultado das duas expressões abaixo:**
   a. [1 1 1 1] & [0 0 1 0];
   b. [1 1 1 1] | [0 0 1 0];
3. **Informe o resultado das expressões:**
   a. ~4
   b. !0
   c. mod(4,4)
   d. mod(4.5,5)
   e. x = 1:10; y = x .^2;

Limpar memória e comandos

- **Comando `clc` ou ctrl+l:**
  - Limpa os comandos exibidos na Janela de Comandos.
- **Comando clear:**
  - Limpa a memória (todas as operações realizadas).

Undefined variáveis

- **O Octave é case sensitive, ou seja, faz distinção entre letras maiúsculas e minúsculas.**
- **Erro comum:** declara-se uma variável com letras minúsculas e na hora de utilizá-la, usa-se letra maiúscula.
- Exemplo:

  >> a = 2;

  >> A

- Aparecerá a mensagem: `error: 'A' undefined near line 1 column 1`

Regras para nomes de variáveis

UNIR

- Podem conter até 63 caracteres;
- Podem conter letras, números e o caractere sublinhar;
- Devem iniciar com uma letra;
- Evite usar nomes de funções nativas do Octave para nomear variáveis (p.ex. cos, sin, exp, sqrt, etc.)
- **O Octave é case sensitive, ou seja, faz distinção entre letras maiúsculas e minúsculas.**

Exercícios

- Resolver a equação quadrática:

$$3x^2 - 7x + 4 = 0$$

$$\Delta = (-7)^2 - 4*3*4$$

$$\Delta = 49 - 48$$

$$\Delta = 1$$

$$x = \frac{-(-7) \pm \sqrt{1}}{2*3}$$

$$x = \frac{7 \pm 1}{6}$$

$$x' = \frac{7+1}{6} = \frac{8}{6} = \frac{4}{3}$$

$$x'' = \frac{7-1}{6} = \frac{6}{6} = 1$$

Utilize `format long`.

Vetores e Matrizes

- Seja a seguinte matriz A 3x4:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 0 & 8 & 7 \end{bmatrix}$$

- Neste exemplo a matriz A é uma matriz de três linhas e quatro colunas.
- Para criar a matriz A no octave, executar:
  - >> A = [1 2 3 4; 5 6 7 8; 9 0 8 7]
- Observe que o ponto e vírgula ; serve para separar as linhas da matriz.

Vetores e Matrizes

- Seja a seguinte matriz A 3x4:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 0 & 8 & 7 \end{bmatrix}$$

- Neste exemplo a matriz A é uma matriz de três linhas e quatro colunas.
- Para criar a matriz A no octave, executar:
  - >> A = [1 2 3 4; 5 6 7 8; 9 0 8 7]
- Observe que o ponto e vírgula ; serve para separar as linhas da matriz.

Componentes de uma matriz

- Na linguagem matemática, a matriz definida anteriormente tem os seguintes componentes:

$$a_{11} = 1, a_{12} = 2, a_{13} = 3, a_{14} = 4$$
$$a_{21} = 5, a_{22} = 6, a_{23} = 7, a_{24} = 8$$
$$a_{31} = 9, a_{32} = 0, a_{33} = 8, a_{34} = 7$$

- No octave é possível ter acesso ao componente `aij` pondo na linha de comando A(i, j).
- Assim, por exemplo:

```
>> A(1,1)
ans = 1
>> A(2,4)
ans = 8
```

Linhas e colunas de uma matriz

- No octave é possível selecionar linha ou coluna de uma matriz.
- Por exemplo:

```
>> A(1, :)
ans = 1 2 3 4
>> A(3, :)
>> ans = 9 0 8 7
```

Por sua vez, a coluna 2 é obtida com o comando:

```
>> A(:, 2)
2
6
0
```

Definição de um vetor coluna

- Um vetor coluna é uma matriz com uma coluna e uma ou mais linhas. Por exemplo:

$$v = \begin{bmatrix} 4 \\ 3 \\ 10 \end{bmatrix}$$

- executar na linha de comandos:

```
>> v = [4; 3; 10]
```

Definição de um vetor linha

- Um vetor linha (ou vetor deitado) é uma matriz com uma linha e uma ou mais colunas. Por exemplo:

```
w = [1 -2 3 -4]
```

- executar na linha de comandos:

```
>> w = [1 -2 3 -4]

w =

1 -2 3 -4
```

Componentes de um vetor

- Na linguagem matemática, o vetor coluna v tem os seguintes componentes:

$$v = \begin{bmatrix} 4 \\ 3 \\ 10 \end{bmatrix}$$

```
v1 = 4, v2 = 3, v3=10
```

- No octave, o componente v(*i*) pode ser obtido com o comando v(i). Assim, por exemplo:

```
>> v(1)

ans = 4
```

Transposta de uma matriz

- Seja a seguinte matriz definida no Octave:

```
>> A = [1 2 3 4; 5 6 7 8; 9 0 8 7];
```

- A transposta de A e obtida com o comando:

```
>> A'
ans =

   1   5   9
   2   6   0
   3   7   8
   4   8   7
```

GNU Octave

Transposta de uma matriz

- A transposta de um vetor coluna é um vetor linha. A transposta de um vetor linha é um vetor coluna.
- Experimente executar a seguinte sequência de comandos:

```
>> v = [1; 2; 3]

>> w = [1 2 3]

>> v'

>> w'
```

Combinação linear de matrizes e vetores

- Sejam as seguintes matrizes, vetores definidos no Octave:

```
>> A = [1 2; 3 4; 5 6; 7 8];
>> B = [9 0; 1 -2; 3 -4; 5 -6];
>> a = [1; 2; 3; 4];
>> b = [5; -6; 7; -8];
```

- A combinação linear de vetores 3a - 2b calcula-se em octave da seguinte forma:

```
>> 3 * a - 2 * b
```

GNU Octave

Combinação linear de matrizes e vetores

- A combinação linear de matrizes 3A - 2B é:

```
>> 3*A - 2*B
ans =

  -15    6
    7   16
    9   26
   11   36
```

Produto de matrizes

- Sejam as seguintes matrizes e vetores definidos no Octave:

```
>> A = [1 2 3 4; 5 6 7 8; 9 0 8 7];
>> B = [1 2; 3 4; 5 6; 7 8];
>> w = [1; 2; 3; 4];
>> v = [1 2 3];
```

Produto de matrizes

- O produto AB (matriz x matriz) pode ser calculado com o comando:

```
>> A * B
```

- cujo resultado é:

```
ans =
```

$$\begin{matrix} 50 & 60 \\ 114 & 140 \\ 98 & 122 \end{matrix}$$

Produto de matrizes

- O produto Aw (matriz x vetor) se obtém fazendo:

```
>> A   * w
```

- cujo resultado é:

```
ans =
30
70
61
```

- O produto vA (vetor x matriz) pode-se obter com:

```
>> v * A
```

cujo resultado é:

```
ans =
38 14 41 41
```

Funções

- **Exemplo 1:** Seja $f(x) = x^3 - 9x + 3$. Vamos analisar o sinal desta função.
- Construindo uma tabela de valores para f(x) e considerando apenas os sinais, temos:

x	$-\infty$	-100	-10	-5	-3	-1	0	1	2	3
$f(x)$	-	-	-	-	+	+	+	-	-	+

Funções

- **No editor:**

```
function [y] = f(x)
   y=x.^3-9*x+3
endfunction
```

- Para podermos chamar essa função na janela de comandos, o arquivo com a função deve ter o mesmo nome dado à função (ex. f) e a extensão .m.
- Além disso, é necessário que estejamos com a pasta na qual a função foi salva aberta.

Funções

- **Na janela de comandos:**

```
>> f(0)
ans=3
>> [y]=f(0)
y=3
```

Funções

- Podemos calcular o valor de f(x) para vários valores de x.

```
>> x=-5:5
```

```
>> y = f(x)
```

- Qual resultado é apresentado após a execução dos comando acima?
- Para introduzir mais pontos entre -5 e 5, podemos fazer:

```
>> x=-5:0.5:5
```

A estrutura SE

- Uma das estruturas que podemos utilizar no Octave é a se, dada pela instrução if seguida de uma expressão:

```
> if expressao
>   bloco_de_comandos;
> end
```

As estruturas SE, SENÃO, SENÃO SE

- Uma das estruturas que podemos utilizar no Octave é a se, senão e senão se, dadas pelas instruções if, else e elsif seguida de uma expressão:

```
> if expressao
>   bloco_de_comandos;
> end
```

```
> if expressao
>   bloco_de_comandos;
> else
>   outro_bloco;
> end
```

```
> if expressao
>   bloco_de_comandos;
> elseif outra_expressao
>   outro_bloco;
> else
>   outro_bloco_ainda;
> end
```

As estruturas SE, SENÃO, SENÃO SE

UNIR

- Exemplo:

```
> limite_baixo = 18;
> limite_alto = 60;
> idade_usuario = input('Digite sua idade: ');
> if idade_usuario < limite_baixo
>   disp('Usuário fora do grupo (limite menor). Desculpe!');

> elseif idade_usuario > limite_alto
>   disp('Usuário fora do grupo (limite maior). Desculpe!')
> else
>   disp('Usuário entra no grupo!')
> end
```

GNU Octave

A estrutura SWITCH

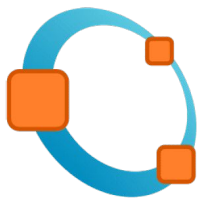
- Imagine que precisamos classificar os elementos em cinco grupos entre 18 e 60 anos.
- Para evitar o uso excessivo de ifs e elses, utilizaremos a estrutura SWITCH.

```
> switch expressao
>   case cond_1
>      bloco_comandos;
>   case cond_2
>      outro_bloco;
>   case cond_3
>      mais_um_bloco;
    ...
>   otherwise
>      outro_bloco_ainda;
> end
```

A estrutura SWITCH

- Exemplo 01:

```
letra_base = input('Indique a letra da base [ACGT]: ', 's');
switch letra
    case {'a' 'A'}
        disp('Adenina');
    case {'c' 'C'}
        disp('Citosina');
    case {'g' 'G'}
        disp('Guanina');
    case {'t' 'T'}
        disp('Timina');
    otherwise
        disp('Não existe nenhuma base correspondente!');
    end
end
```



GNU Octave

A estrutura SWITCH

- Exemplo 02:

```
> idade_usuario = input('Digite sua idade: ');
> switch (idade_usuario)
>   case num2cell([18:21])
>      disp('Usuário no Grupo 1');
>   case num2cell([22:30])
>      disp('Usuário no Grupo 2');
>   case num2cell([31:40])
>      disp('Usuário no Grupo 3');
>   case num2cell([41:50])
>      disp('Usuário no Grupo 4');
>   case num2cell([51:60])
>      disp('Usuário no Grupo 5');
>   otherwise
>      disp('Usuário nao cumpre requerimentos. Desculpe!');
> end
```

GNU Octave

A estrutura PARA

- Uma das estruturas de repetição é a declaração para, dada pela instrução **for** :

```
> for iter = intervalo
>   bloco_comandos;
> end
```

- Exemplo:

```
> for iter = 1:10
>   disp(sin(iter))
> end
```

GNU Octave

A estrutura PARA

- Exemplo 01: Execute o código abaixo e indique sua saída:

```
> tam_grupo = 8;
> idade_grupo = zeros(1, tam_grupo);
> for elem = 1:tam_grupo
>   idade_grupo(elem) = input('Digite sua idade: ');
> end
```

GNU Octave

A estrutura PARA

```
function classgruposid()
    %{
    classgruposid()

    CLASSGRUPOSID() recebe as idades de um grupo de pessoas com
    tamanho predefinido (tam_grupo) e as classifica em cinco
    grupos distintos utilizando a estrutura switch.
    %}

    tam_grupo = 8;
    idade_grupo = zeros(1, tam_grupo);
    for elem = 1:tam_grupo
        idade_grupo(elem) = input('Digite sua idade: ');
        switch (idade_grupo(elem))
            case num2cell([18:21])
                disp('Usuário no Grupo 1');
            case num2cell([22:30])
                disp('Usuário no Grupo 2');
            case num2cell([31:40])
                disp('Usuário no Grupo 3');
            case num2cell([41:50])
                disp('Usuário no Grupo 4');
            case num2cell([51:60])
                disp('Usuário no Grupo 5');
            otherwise
                disp('Usuário nao cumpre requerimentos.
                    Desculpe!');
        end
    end
end
```

GNU Octave

A estrutura enquanto

- Outra estrutura de repetição é enquanto, representada pela instrução while :

```
> while condicao
>   bloco_de_comandos;
> end
```

A estrutura enquanto

- **Exemplo 01:** Uma série geométrica é uma soma cujos termos sucessivos possuem razão constante entre si.

```
function soma = seriegeomwhile(termos)
    %{
    soma = seriegeomwhile(termos)

    SERIEGEOMWHILE calcula a soma da série geométrica
    1/2 + 1/4 + 1/8 + 1/16 + ... utilizando do
    e recebe como argumento a quantidade de termos
    a serem somados, retornando o valor da soma.
    %}

    soma = 0;
    cont = 1;
    while (cont != termos)
        soma = soma + (1/2)^cont;
        cont++;
    end
end
```

GNU Octave

Atividade

1. Utilize o while para criar a função **classgruposid2()** e melhorar o sistema de classificação, tomando as idades de usuários sem estipular um tamanho fixo para o grupo.

Atividade

```
function idade_grupo = classgruposid2()
    idade_grupo = -1;
    while (idade_grupo(end) != 0)
        idade_grupo(end+1) = input('Digite sua idade: ');
        switch (idade_grupo(end))
            case num2cell([18:21])
                disp('Usuário no Grupo 1');
            case num2cell([22:30])
                disp('Usuário no Grupo 2');
            case num2cell([31:40])
                disp('Usuário no Grupo 3');
            case num2cell([41:50])
                disp('Usuário no Grupo 4');
            case num2cell([51:60])
                disp('Usuário no Grupo 5');
            otherwise
                disp('Usuário nao cumpre requerimentos.
                        Desculpe!');
        end
    end
    disp('Fim do processamento!')
end
```

GNU Octave

Dúvidas, sugestões?