

Mini-Curso: Introdução ao Matlab como Ferramenta para Uso de Engenheiros - Parte I 2012.1



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Objetivos

MATLAB?? Alguém conhece?

O que já fez no Matlab?

E quanto a outras linguagens de programação?

Necessidade?



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

O mini-curso

- **Parte 1:** Apresentação e Introdução ao Ambiente do Matlab
 - Introdução ao Matlab; Descrição do ambiente gráfico e formas de acesso e manipulação de arquivos do Matlab; Comandos básicos; Comandos de acesso rápido aos recursos do Matlab; Números complexos, forma exponencial e cartesiana, módulo e fase
- **Parte 2:** Operação com Matrizes e Vetores
 - Polinômios; Demonstração de técnicas de manipulação de vetores. Declaração, módulo, exponenciação, divisão e manipulação de elementos de um vetor; Demonstração de técnicas de manipulação de matrizes; Declaração de matrizes simples e complexas, adição, multiplicação e divisão; Matriz transposta, inversa, escalonamento, autovetores e autovalores, exponenciação e matriz de funções).

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
 vicente.sousa@ct.ufrn.br

O mini-curso

- **Parte 3:** Controle de fluxo no Matlab, Gráficos e GUI
 - Controle de fluxo no Matlab (Controle por laços; Controle de fluxo encadeado); Gráficos bidimensionais; gerenciamento dos elementos gráficos no Matlab; Gráficos bidimensionais muito úteis para Engenharia; Gráficos tridimensionais.; interface gráfica do usuário.; matlab compiler
 - Toolboxes
 - Otimização de códigos em Matlab

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
 vicente.sousa@ct.ufrn.br

Palestrante

Vicente Angelo de Sousa Junior

- Graduação em Engenharia Elétrica: 2000
 - Universidade Federal do Ceará - UFC, Ceará.
- Mestrado em Engenharia Elétrica: 2002
 - Universidade Federal do Ceará - UFC, Ceará;
 - Título da Dissertação: Análise de desempenho de sistemas multi-serviço WCDMA com antenas inteligentes (**simulador em C++, pós-processamento em Matlab**)
- Doutorado em Engenharia de Teleinformática: 2009
 - Universidade Federal do Ceará - UFC, Ceará;
 - Universidade Estadual de Campinas, UNICAMP, Campinas;
 - Título da Tese: Seleção de Acesso e Gerenciamento de Recursos de Rádio em Redes Multi-Acesso (**simulador em Matlab**)



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Palestrante

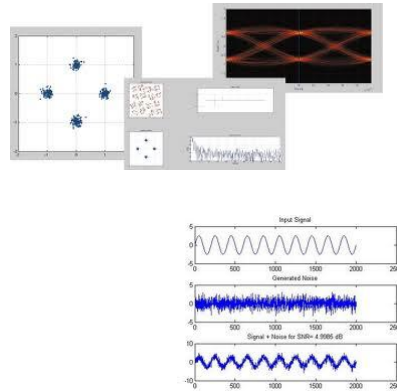
- 2000: Pesquisador Junior:
 - PCT-Motorola (Universidade Federal do Ceará - UFC, 2000).
 - **Matlab como auxílio a disciplinas.**
- 2001-2006: Pesquisador sênior:
 - Cooperação Técnico-Científica entre a Ericsson do Brasil e a Universidade Federal do Ceará (UFC, GTEL, Ceará, 2001-2006).
 - **Vários simuladores de prova de conceito em Matlab (WCDMA, Antenas inteligentes, Multi-acesso, otimização de sistemas sem fio)**
- 2006-2010: pesquisador, Lider de P&D e gerente de projetos
 - Instituto Nokia de Tecnologia - INdT, Manaus, 2006-2010.
 - Simuladores em Matlab para: (i) desenvolvimento de novos produtos; (ii) padronização; (iii) pré-venda e auxílio ao núcleo de negócios.
- 2010-futuro: Professor
 - Universidade Federal do Ceará, 2003-2004.
 - Universidade Federal do Rio Grande do Norte, 2010-futuro
 - **Auxílio em disciplinas, trabalhos científicos e projetos acadêmicos**
 - **Grupo de pesquisa: Grupo de Pesquisa em Prototipagem Rápida de Soluções para Comunicação Sem Fio**



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Palestrante

- 1 capítulo de livro internacional:
- 1 artigo em jornal internacional
- 1 artigo em jornal nacional
- 2 mini-cursos em congresso nacionais
- 19 artigos em congressos internacionais
- 8 artigos em congressos nacionais
- 2 softwares sem registro de patente
- 1 patente registrada internacionalmente
- Mais de 10 relatórios técnicos em projetos com companhias internacionais de telecomunicações



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Módulo 1: Apresentação e Introdução ao Ambiente do Matlab

- **Ao final desse módulo estaremos preparados a:**
 - Entender o que é e para que serve o Matlab
 - Entender o ambiente gráfico, formas de acesso e manipulação de arquivos do Matlab
 - Conhecer comandos básicos (adição, multiplicação, funções trigonométricas, logarítmicas e etc)
 - Declarar números complexos, forma exponencial e cartesiana, módulo e fase.

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução

- O **MATLAB** é um pacote de software para utilização em atividades científicas e de Engenharia.
 - Ele resolve problemas utilizando análise numérica, matricial, processamento de sinais, controle, identificação de sistemas, redes neurais, etc., podendo apresentar soluções gráficas com a mesma facilidade das apresentadas pelas soluções analíticas.
- O nome **MATLAB** significa **Matrix Laboratory**.
- A principal idéia do **MATLAB** é servir como uma ferramenta de prototipagem rápida por software.
- Possui uma vasta biblioteca de comandos.
- O **MATLAB** é também uma ferramenta fácil de aprender, possui um código otimizado para realizar operações matemáticas e sua **linguagem é interpretada**, o que torna a identificação de erros mais simples.

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução



Desenvolvedor: Mathworks (<http://www.mathworks.com>)

Distribuidor no Brasil: (<http://www.opencadd.com.br>)



A Opencadd disponibiliza seminários gratuitos na internet (em Inglês e Português):

<http://www.opencadd.com.br/site/content/news/webinars/>

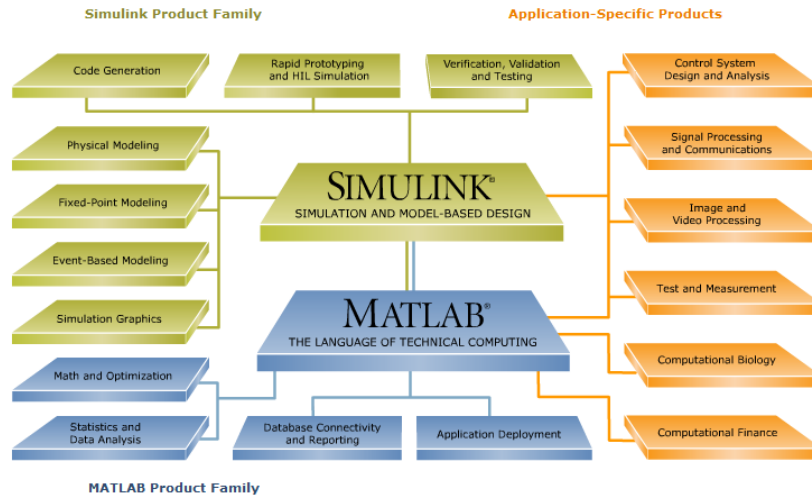


Existe uma infinidade de material gratuito na **Matlab Central**:

<http://www.mathworks.com/matlabcentral/>

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Famílias de produtos do Matlab



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

11

Preço (cotação: 15 de setembro 2010)

- Licença Acadêmica: Classroom License
 - **uso educacional** (não considerando Pesquisa para geração de software). Classroom License é uma modalidade exclusiva para sala de aula.
Cada toolbox: 144,00
Matlab: 411,00 e Matlab compiler: 411,00
- A Licença Acadêmica: Individual License
 - **uso restrito à Universidade**, podendo gerar relatórios, publicações e apresentações acadêmicas (Trabalho de Conclusão de Curso, Teses e Dissertações). **Não pode ser utilizada para ministrar aula. Caso o resultado da pesquisa gere um produto que possa ser comercializado, deverá ser usada a Licença Comercial**
Cada toolbox: 625,00
Matlab: 1.791,00 e Matlab compiler: 1.791,00

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

12

Introdução

Programar em **MATLAB**

- Vantagens
 - Linguagem de alto nível
 - Código simples
 - Implementação rápida e fácil
 - Várias bibliotecas com comandos e funcionalidades disponíveis
- Desvantagens
 - Lentidão devido a ser uma linguagem interpretada
 - Má programação deixa o código muito lento (tempo de execução)
 - Laços de concatenados geralmente são muito lentos

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
 vicente.sousa@ct.ufrn.br

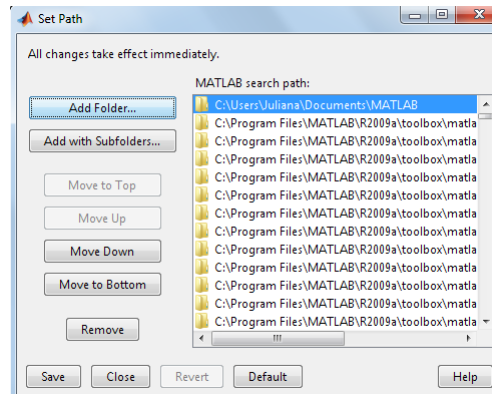
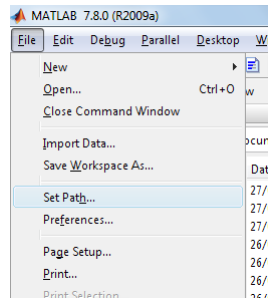
Introdução - Arquivos do **MATLAB**

- Arquivos **.m**
 - Tipo **script**: sequência de comandos MATLAB que manipulam as variáveis do *workspace*
 - sem argumentos
 - **nome**: qualquer_nome.m
 - Tipo **função**: definição de uma função para tratamento de variáveis locais
 - **Nome**: nome_da_função.m (primeira linha é a definição da função)
- Arquivos **.MAT**
 - Arquivos com conteúdo das variáveis
 - Economiza recursos de processamento e espaço de memória
 - Também é importante para guardar resultados que serão utilizados em cálculos futuros.
 - Para salvar o conteúdo use o comando **save** enquanto que para recuperar, o comando **load**.
- Arquivos **.fig**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
 vicente.sousa@ct.ufrn.br

Introdução - Arquivos do MATLAB

- **Ao executar um comando:** Matlab buscará na pasta local e em todas pastas definidas no PATH
- Como adicionar pastas no PATH
 - Via menu



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução - Arquivos do MATLAB

- **Ao executar um comando:** Matlab buscará na pasta local e em todas pastas definidas no PATH
- Como adicionar pastas no PATH
 - Via linha de comando

`path(path, 'C:\Documents\MATLAB')`

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução - Arquivos do MATLAB

- Exemplo de função


```
function y = soma(x, z)
% Realiza a soma x + z
y=x+z
```

Nome do arquivo:

soma.m

Executar

```
> r = soma( 9, 13 )
ans =
22
```

Ainda

```
> help soma
realiza a soma x + y
```

- Exemplo de script

```
x = 1;
s = 2;
Valor = soma(s , x)
```

Nome do arquivo:

proj1.m

Executar

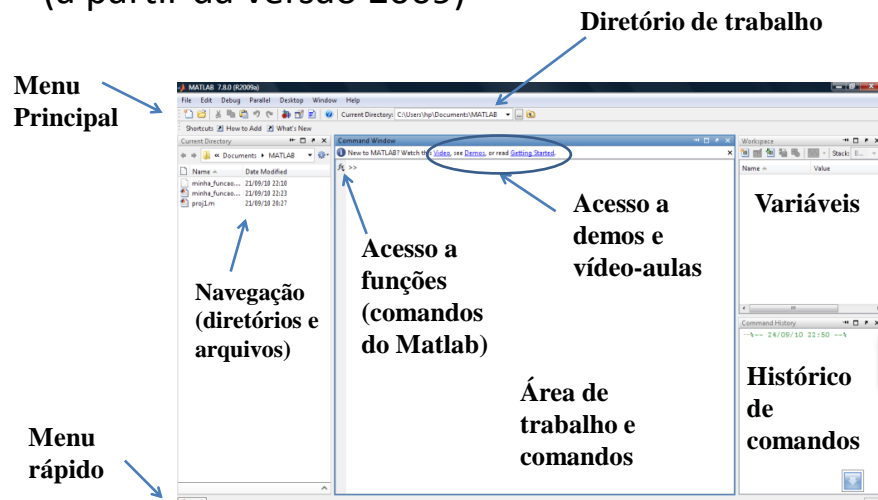
```
> proj1
ans =
3
```

Ainda

- Scripts podem chamar funções

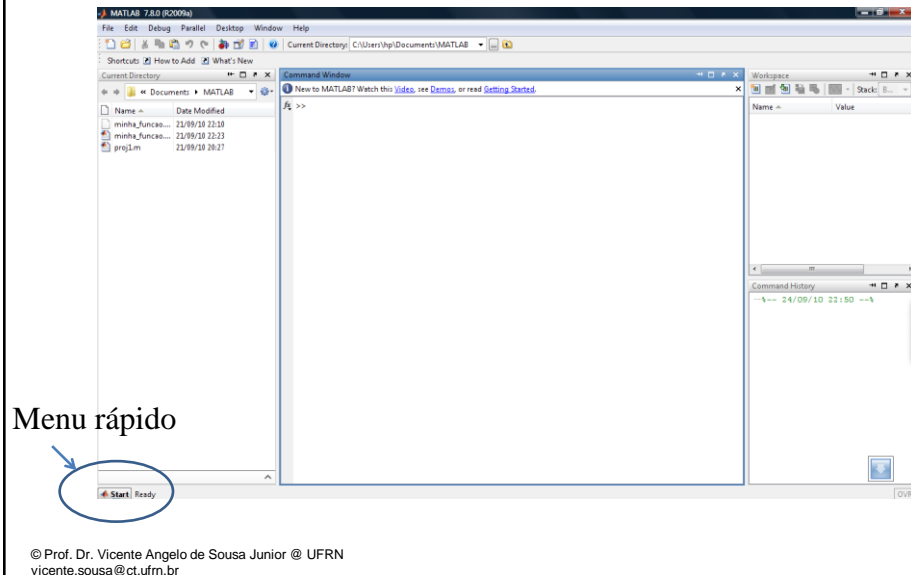
© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Ambiente de desenvolvimento (a partir da versão 2009)

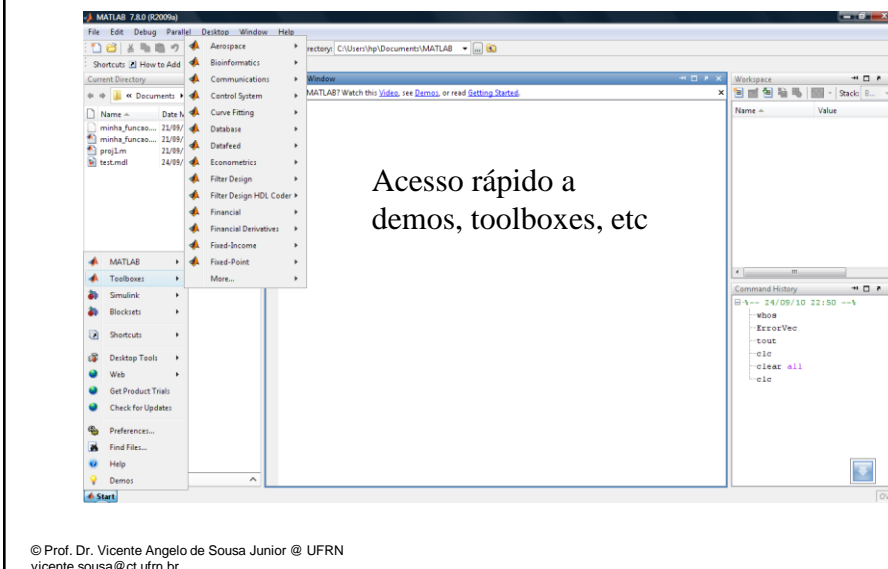


© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

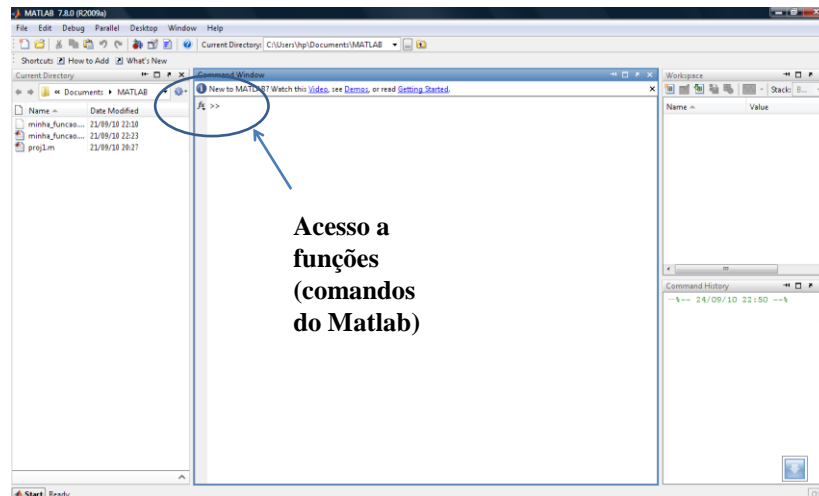
Ambiente de desenvolvimento (a partir da versão 2009)



Ambiente de desenvolvimento (a partir da versão 2009)

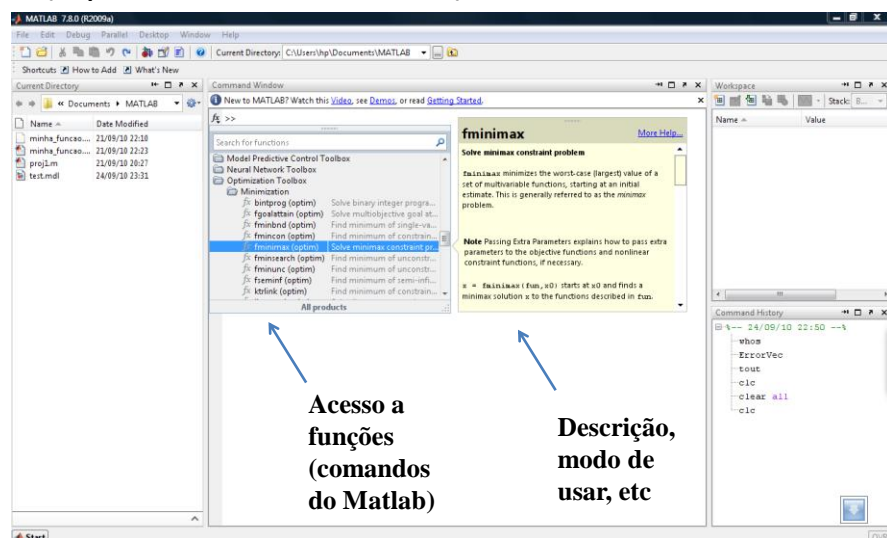


Ambiente de desenvolvimento (a partir da versão 2009)



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

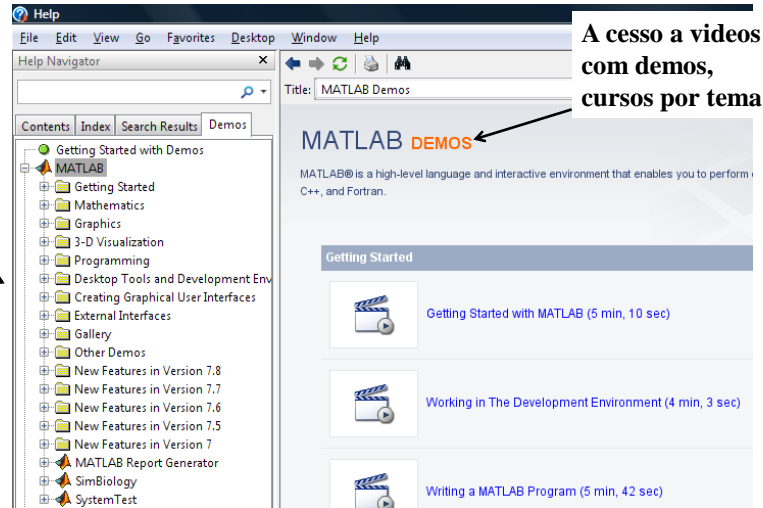
Ambiente de desenvolvimento (a partir da versão 2009)



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Ambiente de desenvolvimento (a partir da versão 2009)

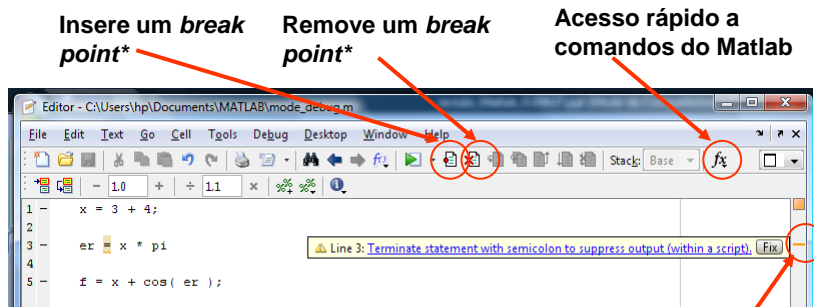
Videos
sobre
toolboxes,
atualizações
do Matlab,
entre outros



Acesso a videos
com demos,
cursos por tema

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Ambiente de Edição – Modo Debug



Insere um *break point**

Remove um *break point**

Acesso rápido a
comandos do Matlab

M-Lint: dicas e warnings para melhoria de códigos são sugeridos inline através de faixas coloridas na barra direita do editor do Matlab

*Break points estipulam em quais linhas o programa irá parar sua execução e assim esperar o comando do usuário.

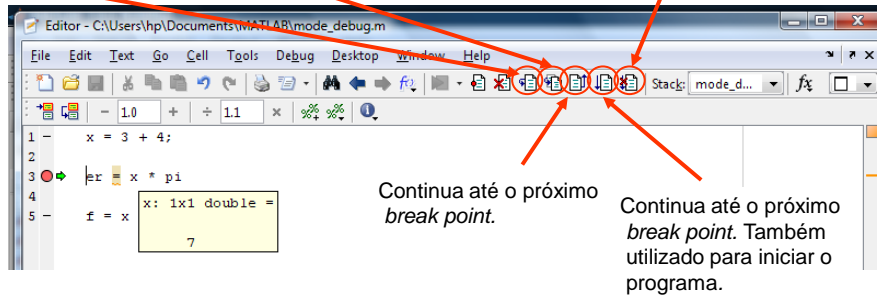
© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Ambiente de Edição – Modo Debug

Vai para a próxima
linha
sem entrar na função.

Vai para a próxima
linha
entrando na função.

Sai do programa



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Vamos praticar

A água congela a 32° e ferve a 212° na escala Fahrenheit. Se C e F são temperaturas nas escalas Celsius e Fahrenheit, respectivamente, a fórmula $F = 9C/5 + 32$ converte uma temperatura de escala Celsius para a Fahrenheit. Utilize linhas de comando do Matlab para converter a temperatura de 37°C para Fahrenheit. A seguir crie uma função na qual o usuário entre um valor em graus Celsius e seja retornado o valor em graus Fahrenheit.

Faça uma função que peça a temperatura em °C ao invés de entrar como argumento.

Nome do arquivo: converter.m e converter2.m

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução

Teclas de acesso:

- ↑ Retornar a linha anterior
↓ Retornar a linha posterior

```
clc           % limpa a área de trabalho
dir           % lista o diretório de trabalho
ls           % mesmo que o comand dir
edit <arquivo.m> % cria um arquivo para edição
cd           % muda de diretório
exit         % sai do MATLAB
ver          % mostra versão do Matlab e
              toolboxes do instalados
which        % Identifica e localiza arquivo
What        % exibe arquivos Matlab salvos no
              diretório
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Introdução

```
who          % mostra as variáveis de ambiente
whos         % mostra as variáveis de ambiente
              com detalhes
help <comando> % mostra a ajuda para o comando
              especificado
lookfor <p. chave> % Procura comando baseado na
              palavra chave especificada
clear all    % limpa as variáveis da memória
clear <variável> % limpa uma variável específica da
              memória memória
pack         % Comprime dados na memória
              (garbage collection)
save (load)  % salva (abre) variáveis para o
              disco (workspace)
saveas (open) % salva (abre) figuras (.fig)
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Operações Básicas

Operações Básicas:

% Declaração de Variáveis:

```
a = 1;
b=3; c=4.2;
```

%Operações Básicas:

% Adição

```
S = a + b
```

% Subtração

```
S = S - c;
```

% Multiplicação

```
S = a * b;
```

% Divisão

```
S = a / b;
```

% Exponenciação

```
S = a ^ b;
% Divisão pela esquerda
S = a \ b;
```

% Resto da Divisão

```
S = rem(a,b)
```

% AND lógico

```
S = a & b;
```

% OR lógico

```
S = a | b;
```

% Not lógico

```
S = ~ b;
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Operações Básicas

Operadores Relacionais:

```
<      % Menor que
<=     % Menor ou igual que
>      % Maior que
>=     % Maior ou igual que
==     % Igualdade
~=     % Desigualdade
```

% Observações:

```
% - Existe diferença entre variáveis
%   escritas em maiúsculas e minúsculas.
% - Ao colocar ";" ao final do comando,
%   o resultado não é apresentado na tela.
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Operações Básicas

Funções Trigonométricas

cos	%Cosseno
sin	%Seno
tan	%Tangente
csc	%Cossecante
sec	%Secante
cot	%Cotangente
sinh	%Seno hiperbólico
cosh	%Cosseno hiperbólico
tanh	%Tangente hiperbólica
asin	%Seno inverso

Observações:

- Os comandos acima recebem argumento em radiano
- O Sufixo "d" recebe os argumentos em graus

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Operações Básicas

Funções Trigonométricas:

acos	%Cosseno inverso
atan	%Tangente inversa

Exemplo:

```
% Observações:
% Para converter radianos para graus utilize o
% comando rad2deg e para o inverso, utilize deg2rad.
```

Exemplo:

```
S = rad2deg(pi) ou convert2deg(pi)
Z = deg2rad(180)
```

%Resultado
S = 180
Z = 3.1416

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Operações Básicas

Exponenciais, Logaritmos, Arredondamentos e Múltiplos Comuns:

```
exp      %Exponencial
log      %Logaritmo Natural
log2     %Logaritmo base 2
log10    %Logaritmo base 10
sqrt     %Raiz quadrada
```

Exemplo:

```
>> exp(3)
>> log2(5)
>> sqrt(2)
```

Arredondamento:

```
floor    % Arred. para baixo
ceil     % Arred. para cima
round    % Arred. para o inteiro mais próx.
```

Exemplo:

```
>> floor(3.4)
>> ans = 3
```

Múltiplo Comum:

```
gcd      % máx. múlt. comum
lcm      % mín. múlt. comum
```

Exemplo:

```
>> lcm(3,2)
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Números Complexos

Módulo, Fase, Conjugado, Forma Exponencial e Cartesiana

```
abs      % Módulo
angle    % Fase
conj     % Complexo conjug.
imag     % Parte imaginária
real     % Parte real
```

Exemplo:

```
>> z = 3+2*j
>> abs(z)
>> angle(z)
>> real(z)
```

```
% Outras formas de atribuição
% de números complexos:
```

```
>> z = complex(2,3);
```

```
>> z = 2+4*j
```

```
>> z = 2+4*i
```

```
>> z = 4*exp(j*pi/3)
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Notação científica e formatos

Use help format para mais informação.

Comando	Exemplo de saída
<code>format short</code>	11.3045 (4-casas decimais)
<code>format short e</code>	1.1304e+01
<code>format long e</code>	1.130452467450893+01
<code>format bank</code>	11.30 (2-casas decimais)
<code>format hex</code>	Formato hexadecimal
<code>format +</code>	São impressos os símbolos +, - e espaços em branco

Para definir números de maneira rápida

1e3 – kilo
1e6 – mega
1e-3 – mili
1e-6 - micro

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Vamos praticar

- Através do MATLAB, calcule o valor de **x** nas expressões abaixo.

$$a) x = 1 + 4^2$$

$$b) x = \left(1 - \frac{4}{3}\right)^3 - 1$$

$$c) x = (1 + j)^2$$

$$d) x = \frac{\ln(3)}{2} + e^{-j4}$$

$$e) x = \frac{3}{\sqrt{2}} - |1 + 4j|^2$$

$$f) x = \tan^{-1}\left(\frac{\sqrt{3}}{3}\right)$$

$$g) x = \angle(3 + 4j)$$

$$h) x = \sin^3(\pi/3) + \cos(\pi^2/4)$$

$$i) x = \sqrt[4]{3}$$

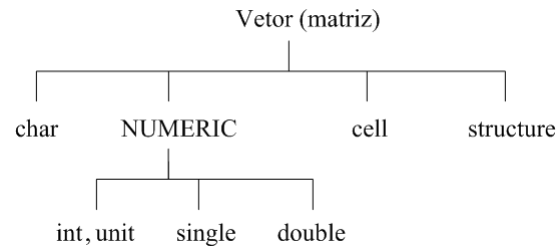
$$j) x = \log_2(7)$$

- Armazene os comandos em um arquivo chamado: **exercicio1.m** e use o comando **disp** para mostrar os resultados no formato **x = expressão = resultado**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Tipos de dados

- O MATLAB tem vários tipos de dados, alguns mais utilizados são:



- **cadeias de caracteres** (*strings*)
- **células** (cell arrays)
- **estruturas** (*structure*)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Tipos de dados

- **strings** são dados tipo texto, formados por cadeias de caracteres
 - ASCII (American Standard Code for Information Interchange). Cada caractere ocupa dois bytes de memória. No Matlab, uma string é definida entre ''. Exemplo.
`>> texto1 = 'Aluno'; c = '9';`
- **Células (cell arrays)** é um tipo de variável que permite armazenar matrizes de naturezas diferentes (escalares, vetores, matrizes, strings) em uma única estrutura.
 - Para criar uma célula, utiliza-se a função “cell”. Células utilizam {} ao invés de []. Exemplo:
`>> celula = cell(2);`
`>> celula{1,1} = eye(2);`
`>> celula{1,2} = 'matriz identidade 2x2';`
`>> celula{2,1} = 2;`
`>> celula{2,2} = [1 2 3]`

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Strings (declaração e manipulação)

String = cadeia de caracteres

```
disp      % mostra string na
          tela
num2str   % converte numero
          para string
int2str   % Arre. e converte
          para string
eval      % executa uma string
          como um comando
```

Exemplo:

```
>> a = 1;
>> out = ['a=' num2str(a)];
>> disp(out)
a=1
```

```
% Outras formas de atribuição
% de números complexos:
```

```
>> out(1) = 'b';

>> out(3) = num2str(4);

>> disp(out)
b=4

>> eval(out);

>> whos b
Name Size  Bytes  Class
b      1x1    8     double array
```

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Tipos de dados

- **Estruturas (*structures*)** é um tipo de dado que permite o armazenamento de matrizes de naturezas diferentes (escalares, vetores, matrizes, *strings*) *identificáveis por nomes de campos*.

Aluno	Matrícula	Nota
João	001	1,0
Maria	002	10,0
José	003	7,0

```
>> s = struct('Aluno',{'Joao'},'Matricula',{001},'Nota',{1.0});
>> s(2).Aluno = 'Maria';
>> s(2).Matricula = 002;
>> s(2).Nota = 10;
>> s(3).Aluno = 'Jose';
>> s(3).Matricula = 003;
>> s(3).Nota = 7;
```

s =
1x3 struct array with fields:
Aluno
Matricula
Nota

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Vamos praticar

- Através do MATLAB, utilize uma variável para armazenar as seguintes informações:

Item	Preço Unitário	Quantidade	Preço total
Geladeira	2500	1	2500
TV	2300	2	4600
Rack	1500	1	1500
Cadeira	500	2	1000
Computador	30	1	30

→ Criar uma variável chamada **compras**

- Nome do script para criar a estrutura **compras**:
`compras_str.m`
- Faça uma função chamada **addcompras** para inserir itens na variável `compras` (arquivo `addcompras.m`). Utilize o seguinte formato:

`compras_saida = addcompras(compras_entrada, nome, quantidade, preco)`

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Alguns atalhos úteis (vamos treinar?)

No editor:

Selecionar texto e apertar F9: executa somente o texto (linhas de comando) selecionado

F5: executa somente todo o script

F10: executa próxima linha (com breakpoint ativado)

F5: executa até o próximo breakpoint (com breakpoint ativado)

F12: Insere breakpoint na linha em que o cursor se encontra

Crtl+i: posiciona o texto selecionado de acordo com a tabulação padrão (*smart indent*)

No Workspace:

Crtl+c: interromper a execução de comandos (se rodando)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN
vicente.sousa@ct.ufrn.br

Preparação para vetores e Matrizes

- Encontrar o valor de x_1 , x_2 e x_3 :

$$\begin{bmatrix} -1 & 2 & 0 \\ 1/2 & 9 & 3 \\ 2 & 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 7 \end{bmatrix}$$

$$Ax = B \rightarrow A^{-1}Ax = A^{-1}B \rightarrow x = A^{-1}B$$

Criar um script chamado **AxB.m**