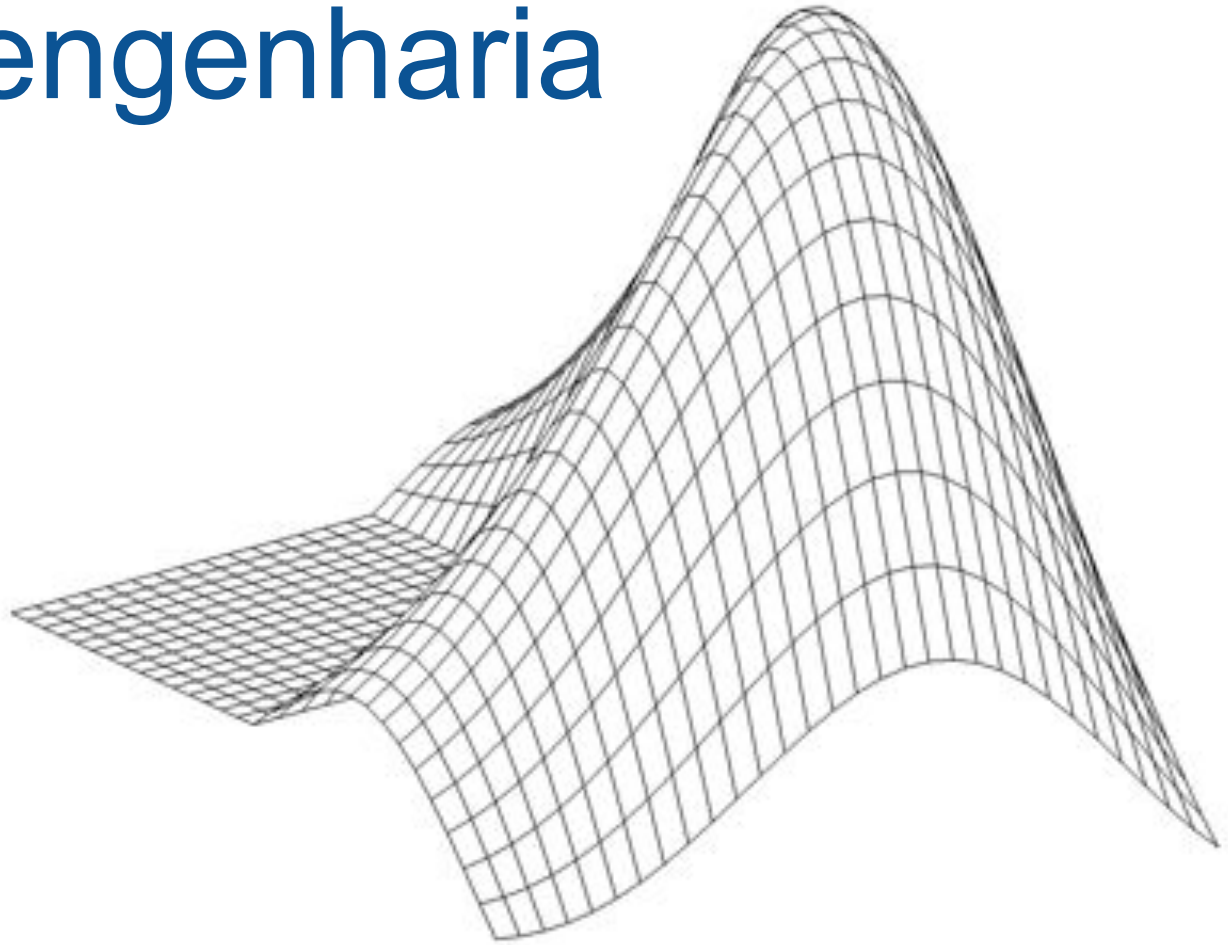
Programação em MATLAB para engenharia



Prof. Jackson Lago, Dr. Eng.
jackson.lago@ifsc.edu.br

Ementa

- Visão geral do MATLAB: janelas, comandos, variáveis, memória, arquivos, help;
- Arranjos numéricos e operações com matrizes;
- Scripts e funções;
- Algoritmos e estruturas de controle de fluxo;
- Vetorização como alternativa à loops;
- Plotagem de gráficos;
- Estruturas de dados;
- Importação e exportação de dados;
- Resolução de problemas de otimização utilizando MATLAB;
- Simulação de circuitos elétricos e sistemas físicos através do SIMULINK;
- Análise harmônica através do MATLAB.

Requisitos

- Álgebra Linear
- Programação de computadores I

Por onde começar?

- O que é o MATLAB (MATrix LABoratory)?

- Janelas

- Documentação

```
>> doc
```

Variáveis e memória (workspace)

- “weakly tiped” language
- Operador de atribuição =
- O tipo básico de variável em matlab é a matriz

```
>> a=5 % a é uma matriz 1x1
```

- Por padrão, cada elemento de uma matriz é representado em memória em ponto flutuante de precisão dupla (64 bits)

```
>> whos
```

Como nomear variáveis

- O nome de uma variável pode ser composto por letras (maiúsculas e minúsculas), por números e pelo caractere especial _ , contudo, deve iniciar sempre por uma letra

- Não pode conter caracteres especiais:
  , : / * - + ! = ( ) [ ] { } % & ' ~ . …

```
tensao = 10      % nome válido
tensão = 10      % inválido devido ao ã
tensao1 = 10     % nome válido
tensao_1 = 10    % nome válido
1tensao = 10     % nome inválido por iniciar com 1
tensao.um = 10 % . é reservado para estruturas
Tensao = 10      % nome válido (case sensitive)
```

Como nomear variáveis

- Alguns nomes são usados para variáveis pré-definidas e devemos tomar cuidado para não sobrescrevê-los

```
pi          % número π
i, j        % unidade imaginária
ans         % variável padrão para o resultado
Inf         % infinito (1/0)
NaN         % not a number (0/0)
realmin     % menor número representado
realmax     % maior número representado
eps         % precisão relativa
```

Como nomear variáveis

- Alguns nomes são usados para funções pré-definidas (por bibliotecas ou pelo usuário) e devemos tomar cuidado para não sobrescrevê-los:

```
sin       % seno
cos       % cosseno
sqrt      % raíz quadrada
```

- Caso alguma função seja renomeada e você deseje voltar atrás:

```
sin = 6      % sin passa a ser uma variável
clear sin    % sin volta a ser a função seno

clear all    % limpar todo o workspace
```

Operações básicas

- Utilizando o MATLAB como calculadora:

```
a+b   % soma
a-b   % subtração
a*b   % multiplicação
a/b   % divisão
a\b   % divisão à direita ( - b / a)
a^b   % elevado à potência
```

- Precedência dos operadores aritméticos:

```
()   % a começar pelo par interno
^    % da esquerda para a direita
*,/  % = precedência, da esquerda para a direita
+,-  % = precedência, da esquerda para a direita
```

Operações básicas

- Outras funções matemáticas:

```
sin(x)    % seno de x
cos(x)    % cosseno de x
tan(x)    % tangente de x
asin(x)   % arco cujo seno é x
log(x)    % logaritmo (base e)
log10(x)  % logaritmo na base 10
abs(x)    % módulo de x
exp(x)    % exponencial (base e)
rem(x,y)  % resto da divisão de x por y
sqrt(x)   % raiz quadrada
...
```

Matrizes

- Construção de matrizes:

```
A = [ 2, 5, 6; 1, 3, 6; 10, 8, 2]

A = [ 2 5 6; 1 3 6; 10 8 2]

A = [ 2 5 6
       1 3 6
      10 8 2]

ones(l,c)  % cria uma matriz lxc populado de 1s
zeros(l,c) % cria uma matriz lxc populado de 0s
eye(n)     % cria uma matriz identidade

[l,c] = size(A) % número de linhas e colunas
```

Matrizes

- Operações com matrizes:

```
det(A)   % determinante
inv(A)   % inversa
A'       % transposta
A+B   % soma
A-B   % subtração
A*B   % multiplicação (diferente de B*A)
A/B   % A*inv(B)
```

Matrizes

- Endereçamento:

```
A(l,c) % acessa o elemento contido em l,c
A(n)   % acessa o n-ésimo elemento de A
A(:,c) % acessa todas as linhas da col c
A(l,:) % acessa todas as colunas da linha l
A(2:3 , 1:2) % uma submatriz
```

- Sobrescrever um elemento/linha/coluna:

```
A(2,3) = 5  % sobrescreve o elemento (2,3) de A
A(:,1) = [] % apaga toda a primeira coluna de A
```

Salvando o workspace

- Salvando e carregando o workspace em um arquivo do sistema:

```
save('arquivo.mat') % salva workspace em arq.

load('arquivo.mat') % carrega workspace em arq.
```

ou

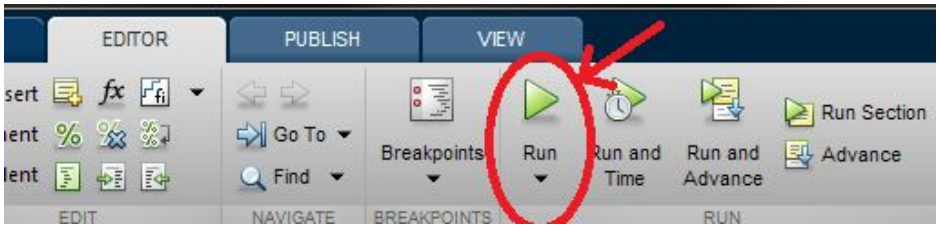
```
save arquivo.mat % salva workspace em arquivo

load arquivo.mat % carrega workspace de arquivo
```

Script

- Um **scrip** consiste em uma arquivo de texto com extensão **.m** contendo uma lista de comandos a serem invocados na sequência em que aparecem no arquivo, exatamente como se digitarmos estes comandos um a um no prompt de comando

```
edit nome_script.m  % cria ou abre para edição um scrip nomeado 'nome_script'
```



- Para executar um **scrip** podemos utilizar o botão RUN ou digitar o nome do **scrip** no prompt de comandos

```
nome_script % executa, na ordem em que aparecem, os comandos contidos no arquivo 'nome_arquivo.m'
```

Função

- Uma função (**function**) é um algoritmo que recebe uma lista de argumentos, executa uma lista de comandos sobre estes argumentos e retorna algum valor.
- Funções tem um espaço de memória próprio (variáveis locais/workspace próprio) e não acessam o workspace principal.
- Uma funções é também definida em um arquivo **.m** que inicia com a palavra reservada function:

```
function [y1,...,yN] = nome_funcao(x1,...,xM)

   % aqui vai o algoritmo
   y1 = % valor de retorno da função var 1
   ...
   yN = % valor de retorno da função var 1
end
```

Sintaxe Função