

# SEGUNDA AULA PRÁTICA DE INTRODUÇÃO AO MATLAB

G. P. Calais, M. V. R. Campos

UFV, Viçosa, Brasil

E-mails: gabriel.calais@ufv.br, marciovonrondow96@gmail.com

**Resumo:** O presente artigo relata uma aula prática de introdução ao *software MATLAB*. O objetivo da aula foi que os alunos aprendessem comandos básicos e úteis do *software*. Para tanto, o professor exemplificou algumas aplicações de criação de funções, além de propor exercícios para os alunos. Com os exercícios os alunos foram capazes de aplicar os conhecimentos adquiridos para realizar operações de soma e multiplicação de matrizes, resolução de sistemas lineares e cálculo de fatorial.

**Palavras-chave:** *MATLAB*, funções, sistema, introdução, comando.

## Introdução

A segunda aula prática da disciplina de Processamento Digital de Sinais consistiu na apresentação de conceitos e operações básicas no *software MATLAB* para os alunos.

O *MATLAB* é um *software* com diversas aplicações em diferentes áreas da engenharia, especialmente na Engenharia Elétrica, podendo ser utilizado para processar sinais, fazer operações matemáticas e para programação e simulação de inteligências computacionais.

Na prática em questão, o *software* foi usado para ensinar aos alunos novos comandos básicos complementando os aprendidos na primeira aula e como criar funções para diversas finalidades, através de orientações do professor e utilizando o comando *help*.

## Materiais e métodos

Foi utilizado o *software MATLAB* com a finalidade de aprender sobre comandos básicos e como utilizá-los para criar funções e resolver problemas.

Através da janela *Editor* do programa, os alunos criaram códigos em programação de acordo com o que foram instruídos a fazer pelo roteiro da aula prática que foram salvos posteriormente em arquivo "PRATICA\_P2\_93506\_93529\_ELT410.m" no computador. Durante a implementação do código, os alunos selecionaram determinadas partes do mesmo e executaram com a tecla F9 analisando as respostas na *command window* para que pudessem testar e verificar se obtiveram êxito na realização das tarefas.

Alguns dos comandos aprendidos na aula prática constam na tabela 1.

Tabela 1: Comandos aprendidos na aula prática.

Comando	Função
<i>input</i>	Exibe uma mensagem na tela e pede ao usuário que digite uma entrada
<i>subplot</i>	Divide uma figura em vários gráficos
<i>save</i>	Salva as variáveis da <i>workspace</i> em arquivo
<i>load</i>	Carrega as variáveis de um arquivo para a <i>workspace</i>
<i>size</i>	Retorna a dimensão de uma matriz ou estrutura
<i>length</i>	Retorna o tamanho de um vetor
<i>abs</i>	Retorna o módulo (ou valor absoluto) de um valor algébrico ou complexo
<i>while, end</i>	Testa a expressão de controle para execução de outras sentenças
<i>for, end</i>	Cria, implementa e testa a expressão de controle para execução de outras sentenças
<i>if</i>	Expressa um condicional
<i>else</i>	Expressa um condicional contrário ao determinado anteriormente
<i>function</i>	Cria uma nova função

Inicialmente o professor pediu aos alunos que fizessem testes e aprendessem sobre os comandos apresentados na tabela anterior. Após tal feito, os alunos utilizaram os conhecimentos adquiridos para resolver alguns problemas propostos pelo roteiro e pelo professor.

Os primeiros exercícios propostos tiveram como objetivo introduzir alguns conceitos novos sobre matrizes e a implementação de uma função para encontrar soluções de equações de segundo grau. Assim, para essas equações, o coeficiente acompanhado do termo quadrático deveria pertencer a qualquer número real exceto o zero, pois o zero não geraria uma função quadrada. Sabendo-se disso, foi utilizado o comando *while, end* citado na Tabela 1 para exigir a inserção de

um termo diferente do valor nulo e então prosseguir com a resolução do exercício.

Dessa forma, na resolução do primeiro exercício, foi criado, através do comando *function* (Tabela 1), funções que calculavam o *Bhaskara* e as soluções do polinômio de segundo grau. As variáveis dessas funções foram salvas em formato *.m* com o nome “polinomio2grau” e carregadas, posteriormente, no *workspace* do arquivo “PRATICA\_P2\_93506\_93529\_ELT410.m” ao qual executou-se o exercício.

Além disso, foram apresentados comandos para comparação como *any*, *all* e “==”, usados para verificar algumas afirmações. O artifício *any* compara cada termo dos vetores analisados e retorna “1” se ao menos uma das comparações for verdadeira, caso contrário, retorna “0”. O *all* também compara os vetores analisados, porém retorna “1” somente se todas as comparações forem verdadeiras, caso contrário, retorna “0”. Já a sintaxe “==” verifica a igualdade entre vetores analisando os termos equivalentes e retorna um vetor com uns e zeros indicando verdadeiro ou falso respectivamente.

Posteriormente, os alunos tiveram a tarefa de criar quatro funções com os seguintes objetivos: calcular o fatorial de um determinado número, somar duas matrizes, multiplicar duas matrizes e solucionar um sistema linear de terceira ordem.

O comando *function* (Tabela 1) foi utilizado novamente para a execução dos quatro exercícios citados. Para a resolução do fatorial de um número, pediu-se a entrada de um termo “n”, o qual, pelo comando *while*, *end*, calculou-se o seu fatorial como observado a seguir:

```
x = n;
if x == 0
    x = 1;
else
    while n > 1
        x = x * (n - 1);
        n = n - 1;
    end
end
```

A função foi salva como “fatorial.m” e a variável “x” do cálculo foi retornada ao arquivo “PRATICA\_P2\_93506\_93529\_ELT410.m”, no qual foram executados os resultados finais.

Quanto a soma e a multiplicação de matrizes, para o cálculo feito pela função criada, pediu-se a entrada do número de linhas e de colunas de acordo com o que é exigido pela álgebra linear para a realização das respectivas operações entre as matrizes. Para a soma, o número de linhas e de colunas da matriz A deve ser igual ao da matriz B, e para a multiplicação, o número de colunas da matriz A deve ser igual ao número de linhas da matriz B. Em seguida ordenou-se a inserção dos valores dos termos das matrizes e utilizou-se o comando *reshape(A,m,n)*, o qual faz o rearranjo matricial onde “A” é o nome da matriz, “m” o número

de linhas e “n” o de colunas. Feito isso, foi executado a soma e a multiplicação das matrizes.

As funções foram salvas como “soma.m” e “multiplicacao.m” e as variáveis  $C = A + B$  (*Soma*) e  $C = A * B$  (*Multiplicação*) foram retornadas ao arquivo “PRATICA\_P2\_93506\_93529\_ELT410.m”.

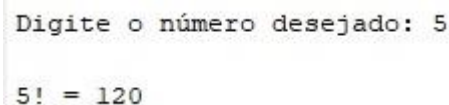
Finalmente, para o exercício solicitado pelo professor, o qual precisava construir uma função que calculava a solução de um sistema linear de terceira ordem. Pediu-se a entrada dos quatro coeficientes (“a”, “b”, “c” e “d”) das três equações necessárias para a formação de um sistema linear. A partir da inserção desses valores pelo comando *input*, foi calculado o determinante da matriz “A” 3x3 entre os coeficientes “a”, “b” e “c” e criada uma condição para ela. Se caso o determinante for zero, a solução é nula, porém se for diferente, a solução é a multiplicação do inverso da matriz “A” pela matriz B que era formada apenas por uma coluna pelos coeficientes “d” inseridos. Os comandos utilizados no cálculo são demonstrados a seguir:

```
if det(A) || 0
    X = (inv(A)) * B;
else
    X = [0; 0; 0];
end
```

A função foi salva como “sistema.m” e a solução “X” foi dada de forma matricial com apenas uma coluna e três linhas e carregada em “PRATICA\_P2\_93506\_93529\_ELT410.m” para sua execução final.

## Resultados

Após os testes feitos pelos alunos com relação aos comandos de comparação, estes criaram as funções com suas devidas finalidades. Os alunos conseguiram concluir a tarefa com êxito e as funções foram implementadas para operar da maneira mais eficiente que os mesmos conseguiram pensar. Nas figuras 1, 2, 3 e 4 constam os resultados da resolução dos problemas propostos.



```
Digite o número desejado: 5
5! = 120
```

Figura 1: Função que calcula o fatorial de um dado número.

```

Entre com os parâmetros pedidos:

Número de linhas da matriz A: 2
Número de colunas da matriz A: 1
Número de linhas da matriz B: 2
Número de colunas da matriz B: 1

Digite o termo a(1,1): 1
Digite o termo a(2,1): 1
Digite o termo b(1,1): 1
Digite o termo b(2,1): 1

A matriz resultante é:
C =
     2
     2

```

Figura 2: Função que calcula a soma de duas matrizes.

```

Entre com os parâmetros pedidos:

Número de linhas da matriz A: 2
Número de colunas da matriz A: 2
Número de linhas da matriz B: 2
Número de colunas da matriz B: 1

Digite o termo a(1,1): 1
Digite o termo a(1,2): 1
Digite o termo a(2,1): 1
Digite o termo a(2,2): 1
Digite o termo b(1,1): 1
Digite o termo b(2,1): 1

A matriz resultante é:
C =
     2
     2

```

Figura 3: Função que calcula a multiplicação de duas matrizes.

```

Entre com os coeficientes do sistema linear

Digite o termo a1: 1
Digite o termo b1: 1
Digite o termo c1: 1
Digite o termo resposta d1: 3

Digite o termo a2: 0
Digite o termo b2: 1
Digite o termo c2: 2
Digite o termo resposta d2: 3

Digite o termo a3: 2
Digite o termo b3: 0
Digite o termo c3: 1
Digite o termo resposta d3: 3

A solução do sistema linear é:
X =
     1
     1
     1

```

Figura 4: Função que calcula a solução de um sistema linear de terceira ordem.

## Discussão

A partir dos resultados obtidos dos exercícios, analisou-se o desempenho das funções que foram usadas nos cálculos.

Nos testes feitos com os comandos de comparação *any* e *all*, foi observado que estes retornam somente um valor lógico, sendo “1” para verdadeiro e “0” para falso. Já o comando “==” compara cada termo dos vetores analisados.

Com relação às funções implementadas pelos alunos, todas retornaram as respostas esperadas. A função “fatorial.m” aceita somente valores não negativos.

As funções de soma e multiplicação de matrizes verificaram corretamente as equivalências de linhas e colunas necessárias.

Por fim, a função “sistema.m” foi implementada para verificar se o sistema analisado tem solução única a partir do determinante da matriz de coeficientes.

## Conclusão

A prática em questão focou em familiarizar os alunos com a criação de funções e utilização de comandos para *loop*. Assim, de acordo com os resultados obtidos e discutidos, é possível afirmar que o *MATLAB* é um software completo e com várias aplicações, pois possui variedades de comandos que auxiliam na implementação de funções para resolução de problemas matemáticos diversos como a obtenção de soluções de sistemas, cálculos fatoriais, soma e multiplicação com matrizes, os quais são bases teóricas para o entendimento da disciplina.

## Referências

- [1] LATHI, B.P. Sinais e Sistemas Lineares. 2ª edição. Porto Alegre, Bookman, 2007.
- [2] Mathworks. Disponível em: <<https://www.mathworks.com/help/matlab/>>. Acessado em 24/08/2019