

Prática 2 - Introdução ao MATLAB, parte 2

Déric Augusto F. de Sales - 96718
Departamento de Engenharia Elétrica,
Universidade Federal de Viçosa, Viçosa - MG.
Email: deric.sales@ufv.br

1. INTRODUÇÃO

Como já apresentado na prática anterior [1], o MATLAB [2] é uma ferramenta poderosa ao engenheiro e cientista, e como também já dito, se comporta como uma linguagem de programação e instrumento para cálculos matemáticos. Assim, a prática em questão trabalha com o uso de testes lógicos, definição de funções e mais operações matemáticas dentro do software MATLAB, em acréscimo às já apresentadas na prática anterior.

Os testes lógicos são muito importantes dentro das linguagens de programação, já que são eles que permitem a escolha de qual decisão será tomada, usando estruturas de *if*, *else* ou mesmo para definir a condição de saída de um *loop*. Outra estrutura fundamental de linguagem de programação são as funções que podem ser definidas pelo usuário, que permitem um aumento da complexidade das operações realizadas sem que isso diminua a organização e consequentemente a compreensão do que está sendo feito. A estrutura de funções também evita a repetição dentro do código e também cria possibilidades extras como o uso da recursão.

Em relação às operações matemáticas abordadas, a concatenação e seleção de seções de matrizes são muito utilizados para juntar e separar banco de dados ou intervalos de um sinal, por exemplo.

2. OBJETIVOS

Assim como na Prática 1, os objetivos desta prática estão relacionados ao estudo do funcionamento do software MATLAB. Dessa vez, deseja-se testar no software a concatenação de matrizes, a criação de funções e o uso de testes lógicos.

3. MATERIAIS E MÉTODOS

Para a realização da prática, foi utilizada a versão R2020a do MATLAB e as operações realizadas foram guiadas pelo roteiro de aula prática [3].

No roteiro da prática, após serem apresentadas algumas funções úteis do programa, é requisitado a entrada de alguns vetores definidos. Após isso, é demandado um teste de concatenação de vetores e a criação de uma função que retorne o valor de um polinômio cúbico. Então, por fim, é demandada a execução de alguns testes lógicos e a criação de mais três funções: uma que calcula o fatorial de um número, outra que calcula a soma de suas matrizes e uma terceira que calcula a multiplicação de duas matrizes.

No MATLAB é possível concatenar matrizes definindo uma matriz através do uso de outras. Por exemplo, se A , B e C são matrizes com o mesmo número de linhas, é possível a criação de uma matriz D que seja definida da seguinte maneira: $D = [A \ B \ C]$ fazendo com que D seja o resultado da concatenação horizontal das matrizes A , B e C . O mesmo é válido para uma concatenação vertical.

Para a definição de funções dentro do MATLAB é necessário o uso do prefixo “*function*”, seguido da estrutura: $[resultado] = NomeDaFuncao(entradas)$ e após a designação das tarefas da função deve ser digitado “*end*”, assim como para o uso de *loops* dentro do MATLAB.

Para os testes lógicos, por sua vez, pode ser usado “ $==$ ”, para comparar duas expressões lógicas. Por exemplo, a expressão $A == B$ testa se A é igual à B . Se caso a expressão for verdadeira, o valor 1 será retornado, caso contrário, o valor 0. Também é possível usar os operadores $<$, $>$ ou mesmo “ $=$ ”, que é o oposto do “ $==$ ”.

4. RESULTADOS

Dentre os exercícios propostos no roteiro, está a definição do vetor A através de $A = [123;456;789]$ e $r = [13325]$, e ao executar $A = [A;r]$, o valor de A será atualizado para $[123;456;789;13325]$, ou seja:

$$A_{velho} = \begin{pmatrix} 123 \\ 456 \\ 789 \end{pmatrix}, r = (13325), A_{novo} = \begin{pmatrix} 123 \\ 456 \\ 789 \\ 13325 \end{pmatrix}.$$

Após a realização dos exercícios, foi requisitada a definição do vetor x com valores crescentes de 1 à 10 com passo um, da maneira $x = 1 : 10$, pela sintaxe do MATLAB. Assim, foram realizados os seguintes testes lógicos:

| | | |
|---------------------------------------|----------------------------------|-------------|
| $(x^2) == (x * x)$ | $\xrightarrow{\text{resultado}}$ | $ones(10)$ |
| $A^2 == A.^2$ | $\xrightarrow{\text{resultado}}$ | $zeros(10)$ |
| $any((x^3) > (3 * x))$ | $\xrightarrow{\text{resultado}}$ | 1 |
| $all((x./(x+1))) > ((x+1)./(x+1.5)))$ | $\xrightarrow{\text{resultado}}$ | 0 |

Após a realização dos testes lógicos foram geradas as funções requisitadas. As funções de somas e multiplicação de matrizes apenas executaram essas operações dentro de seu escopo, ou seja: para a soma de matrizes foi escrito que $resultado = A + B$ e para a multiplicação de

matrizes termo a termo foi feito $resultado = A.*B$. Para a função que calcula o resultado de um polinômio foi feito:

```
function y = resPolinomio(x,a,b,c,d)
% Calcula o resultado do polinômio de 3ª ordem
y = a*(x^3) + b*(x^2) + c*x + d;
end
```

E para a função que calcula o fatorial de um número foi escrito:

```
function res = fatorial(numero)
% Retorna o fatorial de um número inteiro
if floor(numero) ~= ceil(numero)
    disp('O valor de entrada dado não é um
    número inteiro!')
end
n = numero;
x = n;
while n > 1
    x = x*(n-1);
    n = n-1;
end
res = x;
end
```

Todas as funções foram testadas com várias entradas visando abordar todos os possíveis casos e seu funcionamento foi comprovado.

5. DISCUSSÕES

No primeiro teste lógico realizado apresentado, ao se comparar 2 vetores, o MATLAB gera um terceiro vetor de respostas como saída, onde cada valor deste último vetor corresponde ao valor de resposta do teste lógico realizado entre 2 elementos em posições correspondentes nos 2 primeiros vetores. Para simplificar a resposta, esta foi apresentada como a função *ones(10)*, que gera um vetor de tamanho 10 com valores iguais à um. Assim, o teste lógico afirma que os 2 vetores comparados são iguais. O segundo teste lógico possui o mesmo princípio, tendo como saída um vetor de valores nulos, representando que os vetores comparados possuem todos os seus valores diferentes entre si.

No terceiro teste de expressão lógica foi utilizado a função *any*, que funciona como um análogo à porta lógica *OR* para um vetor de valores. Ou seja, se qualquer valor do vetor for diferente de zero, sua saída será igual à 1, e caso contrário, 0. No último teste lógico realizado há uma noção similar, visto que a função *all* é análoga à porta lógica *AND*, seguindo a mesma lógica de saída da função *any*. Assim, o resultado obtido é que no vetor do terceiro teste lógico alguns pontos dos vetores de expressão coincidem, e no quarto teste lógico não há pontos coincidentes.

Agora, a respeito das funções criadas, na função *fatorial*, são utilizadas funções as funções nativas do MATLAB *floor* e *ceil* para verificar se o valor de entrada

é um número inteiro ou não, já que o teste verdade só será verdadeiro se o número for do tipo *int*. Para o cálculo do fatorial é utilizado a estrutura de *loop while*, que é interrompido quando a expressão $n \sim 1$ for falsa, ou seja, quando n for igual à 1, significando que o cálculo fatorial chegou ao seu fim e o ciclo pode ser interrompido

6. CONSIDERAÇÕES FINAIS

Mais uma vez é reforçada a evidência de que o MATLAB é uma ferramenta muito útil para a realização de operações aritméticas aplicadas a vetores e matrizes, e desta vez é destacada a habilidade do MATLAB de trabalhar com *scripts*, funcionando como uma linguagem de programação.

7. REFERÊNCIAS BIBLIOGRÁFICAS

REFERÊNCIAS

- [1] Dr. Leonardo Bonato Felix. *Aula Prática 1 – Matlab. CCE, DEL, Universidade Federal de Viçosa.*
- [2] MathWorks. *Math. Graphics. Programming. Disponível em: https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab.*
- [3] Dr. Leonardo Bonato Felix. *Aula Prática 2 – Matlab. CCE, DEL, Universidade Federal de Viçosa.*