Condition Monitoring of Structures, Machines and Processes

Tutorial

1.2

INTRODUCTION TO MATLAB

Exercice 9

Array and Matrix Operations

- In contrast to element-wise array operations, marked with the period character (.) i.e. (A./B), (A.*B) and (A.^B), MatLab also provides matrix operations following the rules of linear algebra i.e. (A/B), (A*B)

- In MatLab you can apply a variety of functions to an array. I.e. max(), min(), sqrt().

- Some functions can be applied to produce multiple outputs. I.e. [xMax,idx] = max(x) provides the maximum Value xMax and the corresponding index idx. Note, by replacing the first output variable with ~ you can supress the output.

Task:
- Create a Matrix A by adding 2 to each element of [1 2 3]
- Transponde A, multiply each element with 2 and store the result in B
- Perform the operation A/B. Explain the result.
- Repeat the steps with the initial elements of the matrix [2 4; 6 8]
- Additionally, extract the minimum, maximum and the corresponding indices of the result by using min() and max() functions.

Exercice 10

Plotting Data

- To plot data x vs y of the same length use the command plot(x,y)

- The command plot() allows for further input elements to specify i.e. color, line style and marker style. As a excercise look up the specfics online.

Tasks:
- Plot a sin curve, visualizing two periods. The command linspace() may be useful.
- Try different line styles by using i.e. plot(x,y,'-o')
- Plot a cos-curve of two periods on the same figure. Notice that MatLab delets the previous plot. Use the hold on command to keep previous plots. To reverse this setting, use hold off.
- Add labels and titles to the plot by using the commands described at https://www.mathworks.com/help/matlab/titles-and-labels.html?s_tid=CRUX_lftnav

Exercice 11

For and While Loops

- For-Loop: Run the following script and explain the result:

```
for v = 10:-2:0

    disp(v)

end
```

- Write a script representing the formular $\sum_{i=1}^{20} i$

- While-Loops: Explain the following Script:

```
n = 10;

sum_while = 0;

while i <= n

    sum_while = sum_while + i;

    i = i+ 1;

end
```

- Using the while-loop, write a Script loop to create a sum of odd numbers

Exercice 12

Functions

- Within your script, if you repeatedly use the same lines of codes over and over again, it is convenient to define functions. Functions allow you to package these lines of code together and apply it it to variable input data. You already made use of this concept by working with the bulit in functions like [M,I] = max(Input), returning the output maximum value M and the corresponding Indice I of a given Input Data.

- One way to implement functions is to append them at the end of your script in following form:

```
Function [output] = myfunction(input)

script

end
```

- Consider and explain the following function and apply it to the array A = 1:1:100 and B = randi(100,10)

```
function [mx, mn, avg] = stats(data)

mx = max(max(data); mn = min(min(data)); avg = mean(data);

End
```

- Write a function to calculate the faculty of a positive integer n

Exercice 12 (optionally)

Logical Indexing

- Relational operators like <,>, ==, ,<= etc. are use to compare values with logical outputs 0 (false) and 1 (true).

- The operators can be applied to scalars and matrices and can be used to index the elements (logical indexing)

Tasks:
- Use relational operators to test if $\sqrt{2}$ is less or greater than 1.5
- Create a random 1x10 Matrix M with values ranging between 0 and 10.
  - Hint: You can use the command randi()
- Identify the elements of M greater than 5.
  - Hint: Writing M([logical array]) extracts all the values of M corresponding to true values 1 of the logical array
- Modify M, so that all values greater than 5 are replaced by 1
- Combinations: With the command & (AND) and | (OR) you can combine relational operators.
  - Extract all values of M ranging between 1 and 3.

UNA