**Exercise #5**

Tutor in charge of this HW: Ayelet Sarel, e-mail for Questions: [matlab.wis@gmail.com](mailto:matlab.wis@gmail.com)

The mail subject should be: **'Matlab intro - exercise 5'**. Attach only the script file to your mail.

**HW instructions:**

* Don't forget to write both names in the header.
* **You should submit only 1 m file in this HW**
* Print to the screen only when required. All the questions should be answered in the proper location in the ‘m’ file (as comments).
* Printing should be done only with the *disp()* function. When the print-out contains both text and numbers use *disp(sprintf)*
* Before each question X.Y write the line: “%% Question X.Y”. Below there's an example of how the \*.m file should look like.

function [] = hw5\_XXX\_YYY ()

% HW5

% Avi Bitter XXX

% Eliko YYY

%% question 1.1

matlab code….

%% question 1.2

matlab code….

…..

The first row in the script should be (replace XX.. and YY.. of course ):

function [] = hw5\_XXX\_YYY ()

This function will be our primary function, it has the same name as the script, and executing the script is done by calling this function (from the Matlab command for instance).

**Question 1**

Create a nested function that imitate the function *subplot* in matlab, call it *mySubplot* (without using subplot of course). The function should open an axis in the correct size and position according to the inputs. The inputs should be the number of rows and columns of axes in the figure, and the number of the current axis (just as in subplot).

Comments:

1. Make sure to write errors to the user if there are invalid inputs (NANs for example).
2. Use the built-in function *axes*.
3. Note that the numbers in position vector in the *axes* function are in proportions of the total size of the figure.
4. You should have margins between the axes (two subplots should have space between them). The margins could be in proportion to the number of subplots or fixed number. But make sure it looks good in up to 10x10 subplots.
5. Example of numbers: If I want to have 3 subplots in a row I need to have 4 margins on the horizontal side (see picture below; 2 between the figures and 2 on the edges). To find the subplot width: I need to remove the margins from the total length (which is always 1) and then to divide by the number of plots:

Axis\_width=(1-4\*margin\_ size)/3.

1. In contrast to the example above - - - DON’T USE MAGIC NUMBERS
2. The function should be generic and work with any number. I will check the function with up to 10x10 to see if it looks good.
3. The index of the subplot can work just as in matrices (based on columns and not rows as in the matlab function subplot)

**Question 2**

1. Create a [nested function](https://www.mathworks.com/help/matlab/matlab_prog/nested-functions.html) named my*Factorial*() which calculates the factorial of the number n and prints its value. The factorial is defined as the product

You should program this function using recursion only - without any loops, matrices or other Matlab functions. The nested function should be inside the primary function.

1. Call the function with n=8, and print the result.

**Question 3**

1. Create a [nested function](https://www.mathworks.com/help/matlab/matlab_prog/nested-functions.html) named *calcTailorExp*() which gets as input a number x and the upper limit N (where n=0,…,N), and calculates the following series:

You should program this function using recursion without any loops, matrices or other Matlab functions. You should use the factorial function above.

1. Assume x=1,what is the minimal value of N that gives you a good approximation (with an error of less than 0.0005) for an exponent (This is a matlab course, not a math course.. we want to see the code, not the final result)

**\* NOTE-** we know not all of you know what Tailor expansion is. All you need to know is that in our case it is a way to approximate with sum of numbers. If you add more and more numbers from this series you get closer and closer to the real number (the error will decrease).

**Question 4**

Use the function you built in question 1 to plot the results you got in question 3.

1. You should have N subplots, where N is the number you got in question 3.2
2. In each subplot you should plot the real number as a horizontal line and the approximated numbers based on *calcTailorExp(n)*, where n are all numbers between 1 until the current number of the subplot.
3. Each approximation will be a different point in the graph.
4. The x-value of each point should be the number n (1 for the first point, 2 for the second and so on..), and the y-value of each point should be the number you will calculate ( or *calcTailorExp(n)*)
5. For example, if N is 4 (based on the result you got in q3.2) you will have 4 subplots, where:

* subplot 1 you will have the real number (horizontal line) and *calcTailorExp(1)* (in total 1 points in the graph: where x=1,y= *calcTailorExp(1)*)
* subplot 2 you will have the real number (horizontal line) , *calcTailorExp(1)* and *calcTailorExp(2)* (in total 2 points in the graph: where the first point is the same as in the previous graph and the second is: x=2,y= *calcTailorExp(2)*)
* )
* subplot 3 you will have the real number (horizontal line) ,*calcTailorExp(1)* , *calcTailorExp(2)*  and *calcTailorExp(3)* (in total 3 points in the graph: where the first two points are the same as in the previous graph and the third is: x=3,y= *calcTailorExp(3)*)
* subplot 4 you will have the real number (horizontal line) ,*calcTailorExp(1)* , *calcTailorExp(2), calcTailorExp(3)* and *calcTailorExp(4)* (in total 4 points in the graph: where the first three points are the same as in the previous graph and the fourth is: x=4,y= *calcTailorExp(4)*)

1. Add title to each subplot ‘approximating e with n=4’ in the fourth subplot for example.
2. The Y axis of all subplot should be the same and start with zero until ceil()