Assignment #1

This assignment is due on March 26th one hour BEFORE class starts via email to `christian.wallraven+EMS2019@gmail.com`.

Important: You need to name your file properly. If you do not adhere to this naming convention, I may not be able to properly grade you!!!

If you are done with the assignment, make one zip-file of the `assignment1` directory and call this zip-file `STUDENTID1_STUDENTID2_STUDENTID3_A1.zip` (e.g. `2016010000_2017010001_2017010002_A1.zip` for a team). The order of the IDs does not matter, but the correctness of the IDs does! **Please double- check that the name of the file is correct!!**

Also: Please make sure to comment the code, so that I can understand what it does. Uncommented code will BE RETURNED!

Finally: please read the assignment text carefully and make sure to implement EVERYTHING that is written here – if you forget to address something I wrote, this will also reduce your points! Precision is key☺!

Part1 Scripts, Plotting, Symbolic Toolbox (30 points):

Make a **script [not a function]** called `sinx.m`. The first few lines should clear the workspace, close all the figures, and clear the output window of Matlab.

Create a first, numbered figure using `figure(1);`. Set the position property of the figure to be at the bottom-left corner, with a width of 600 pixels and a height of 400 pixels [Hint: look for setting figure properties in the Matlab help!]. Using the standard plot command [not `fplot`!], plot the function $f(x) = \left|\frac{\sin(x)}{x}\right|$ [note the use of the absolute value!] in the interval $-100 \leq x \leq 100$. Make sure to supply the x-array with enough points to get a smooth plotting behavior. Insert code to save the figure into a PNG-file called `sinc.png`.

In the comments below the necessary code lines, tell me:

1a) Based on the plot, what seems to be the value of f(0)?

1b) Why am I asking about this value?

2) Based on the plot, what seems to be the value of $\lim_{x \to \pm\infty} f(x)$?

Insert your responses as comments into the script.

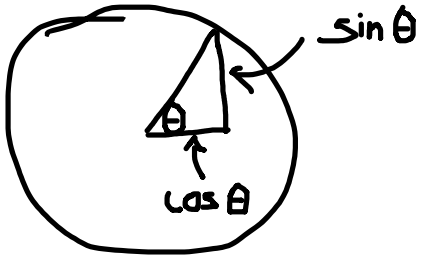
Next, in `sinx.m,` use the power of the symbolic toolbox to show that the limits are conforming to your expectations. Define the variable `t` to be a symbolic variable using `syms t;`. Then use the command `limit` with a suitable function definition [as a function of t, of course] to check the limits for the three cases in 1a) and 2) above. Write the results of the three limit computations into the command window using `fprintf` and some nice explanation text.

Part2 Scripts, Plotting (10 points):

Make a **script [not a function]** called `circleFigure.m`. The first few lines should clear the workspace, close all the figures, and clear the output window of Matlab.

In a numbered figure created with `figure(2)`, plot a **circle of radius 10** using TWO suitable functions f(x). They **CANNOT** use **ANGLES** explicitly! [The reason why we cannot use ONE function f(x) will become clear soon.] The circle needs to be centered on 0,0.

Next, insert code to plot a line of length 10 from the origin at an angle of 50deg with the x-axis. For this, you are allowed to use angles explicitly. Next, insert code that completes the right-angle triangle from this line. Finally, insert code that annotates the triangle with the angle (call it θ) and with annotated arrows pointing towards sin (θ) and cos(θ) [Hint: look at the Matlab help file for a command called annotate or use the command text]. The final figure should look like a professional version of my hand-drawn figure below [yeah, you try drawing on a laptop trackpad in Word ☺].



Finally, insert code to save the figure into a JPG-file called `circle.jpg`.

Part3 Statistics, function, plotting (20 points):

In this assignment you will use a script and a function to explore some data, which you do not know where it comes from or what it means. The data is called `dataForTesting.mat` and comes included in the zip-file.

Create a file called `processData.m`, and write its preamble to clear all variables, close all figures and clear the command window. **This is a script.**

Insert a command to load `dataForTesting.mat` from the file. You will see that it creates a variable `data` with 206 lines, each with 14 values. All you know is that the last column of that data is the variable of interest and that all values should be **positive or 0**.

You will need to clean this data, as you will see that it contains some problems. Figure out the problems and fix them in a separate **FUNCTION** called `fixData.m` after you've loaded the data. It should be easy to figure out what the function needs as input and what the function will deliver as output. Also, this function needs to make sure that the input data conforms to expectations on the **size of the input** data – insert proper error handling to deal with this using the command `error`. **Explain in the function using comments exactly what the problems are with the input, how you fixed them, and WHY you fixed them the way you did.**

With your nice, cleaned data, and using only exploratory `scatter` plots that you create within one figure of size 1000x1000 with an appropriate number of subplots in the script `processData.m`, tell me **the three variables** you think have the MOST explanatory power for determining that final column of data [other than, trivially and obviously, the final column itself!]. **Use red color for plotting the data of those variables**. Do NOT calculate anything, simply try to use your brain's pattern recognition algorithms to decide this question! **Insert all analyses as comments into `processData.`**