MATLAB and Matlab Grader interface

1 Signals Labs

Part of the signals course are labs that should help you get familiar with the software program MATLAB.

2 MATLAB

MATLAB is a software program used to make calculations, perform simulations and to analyse data. Instead of performing calculations by hand, you will now write a program (a script/function) that will perform these calculations for you. This is very useful for difficult problems, which can not be solved exactly, but require a calculation that approaches the correct answer.

For the labs it is required to have MATLAB installed on your (TU/e-) laptop. TU/e-laptops should come with a pre-installed version of MATLAB. If you do not (yet) own a TU/e-laptop, then you should download it from the following website: `https://intranet.tue.nl/universiteit/diensten/information-management-services/ict-services/`

If you have any problems with the installation you should go the notebook service center. The MATLAB program should be on your laptop **prior to** the first lab, since you cannot participate otherwise. The software takes too long to install in the first lab, so make sure it is pre-installed.

2.1 MATLAB Interface

After you have installed MATLAB an icon will appear in the start menu, and perhaps a shortcut has been added to the computer's desktop as well. After starting up MATLAB, the screen shown in figure 1 should appear. The central window A is the command window, and commands can be inserted after the MATLAB prompt:

```
>>
```

After a command as been input (possibly after the output), another MATLAB prompt will appear.

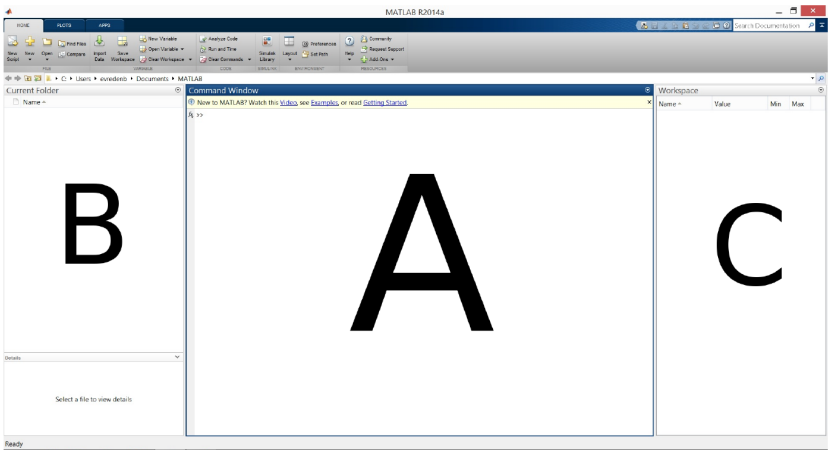

Figure 1: The start-up screen of MATLAB. A is the 'Command Window', used to input and execute commands. B is the 'Current Folder', the contents of the current folder, described by the file address bar. C is the 'Workspace' that displays all currently defined variables and their corresponding values.

After entering the following calculation in the command prompt:

```
>> a = 3/4
```

The following will appear in the command window:

```
a =

    0.7500
```

Also, the variable `a` has now appeared in the workspace (window C) its corresponding value. The workspace will keep tract of the defined variables and their corresponding values. Redefining a variable will overwrite its old value. The history of the workspace and the command window can be cleared by right clicking the corresponding window and selecting the `clear` option or by pressing `Shift + Q` (for the command window) or `Ctrl + Shift + Q` (for the workspace).

2.2 Scripts and Functions

Besides the command window, where commands can be executed, there is also an option to create a file of *multiple* commands (a script), which can be executed after each other, just like a programming file. A script can be created by selecting under the tab `Home` the `New Script` button. This should add another window above the command window. This is the script. Here you can write your code containing multiple commands (and/or loops). Before the untitled script can be executed it should be saved with a file name and be added to the path. The file can easily be saved, using the `Save` button under the `Editor` tab, which appears after selecting the script.

For MATLAB to be able to execute your script it has to know where it can find the script that you want to execute in your folders. To select the folder where your script is in, you can select it by using the address bar under the expanded tabs. This will update your `Current Folder`, which is displayed in window B in figure 1. Once your MATLAB file appears here you should add it to the path by right clicking it and selecting the corresponding option.

The script can now be executed by entering the script file name in the command prompt or by pressing the `Run` button under the `Editor` tab.

A special kind of script is a **function**. This is a script that can accept input arguments and that can also output arguments. To create a function you have to go to the `Home` tab and select the `New -> Function` button. This will create a function. A function template will look like:

```
function [ Output_Arguments ] = Function_Name ( Input_Arguments )
% Here your commands will go
% '%' Are used to comment text
end
```

A function can not be executed like scripts, since it requires input arguments. It has to be executed in the command prompt like:

```
>> Function_Name ( 100 )
```

Multiple arguments can be separated using commas.
Do always use the function name and variables provided in the template!!!

3 Matlab Grader

In the labs you will have to write script and functions to solve/visualise problems. These scripts/functions should be handed in on Matlab Grader. You will all (have) receive(d) an email with an invitation link to the website. If not, contact your teacher(s). In Matlab Grader, on the left you can see the lab(s). Expanding these labs will show the exercises per lab. These exercises are also available in the pdf-document of the corresponding lab. Opening an exercise will show you the problem description and the solution. In the solution text box, there is a template that you have to use for the exercise.

This template has to be copied into a script in MATLAB that you can modify to fulfill the requirements of the exercise. **Do not change the variables given!!!** If you do so, your code might not be corrected properly by Matlab Grader. Once you are satisfied with your solution, you can submit it. It will show that your code has passed the test. However, this is just a confirmation of you handing it in. After the submission deadline has passed, multiple other tests are run to grade your script/function. For a walkthrough of Matlab Grader, check out the following link: `https://nl.mathworks.com/videos/matlab-grader-overview-1532608334621.html` The results of these tests will appear a couple of days after the submission deadline. If one of the x tests has failed, then it will appear you have failed the whole exercise. Fortunately, your grade will be based on the amount of tests passed. It can hence occur that you failed all exercises but still passed most of the teste and have an acceptable grade. Handing in or correcting scripts after the submission deadline will result in zero points for the corresponding exercise! **Make sure your script/function runs in MATLAB! If it gives errors, then your solution will automatically be rejected after the submission deadline and this will result in 0 tests passed for the corresponding exercise.** Warning messages in MATLAB are okay, depending on the situation (imaginary part warning can mostly be ignored).

Starting Exercise

In Matlab Grader is a special lab added called "Lab 0 - Student ID". Open this assignment and open the only problem "Student Number". Click on solve and remove the line of text in the template and replace it by **only** your student number (**not your s-number!; the student number is a seven-digit number that for most of you will start with 09, 10, 11, 12, or 13**). Click on submit to submit your student number. **If not done properly, your grade will not become associated with your student number and you will not receive a grade for the labs!**