Week 1 Lab

EG1002 Computing for Engineers

Semester 1, 2013

1 Introduction

1.1 Aim

- To become familiar with MATLAB's interface
- To learn how to convert basic mathematical expressions into MATLAB code
- To observe the operation of variables within MATLAB scripts
- To modify a simple MATLAB script using the *if* statement

1.2 Assessment

This lab is assessable and will count towards your final grade, as specified in the subject outline. There are written answers that you must complete as you work through the lab. You may write your answers in a word processing document or on paper. **Your tutor will give you a mark at the end of your lab session.** Do not leave the lab room without getting a mark, or else you will receive zero marks.

2 Getting started with MATLAB

Start up MATLAB and work through the following steps. You are encouraged to experiment with MATLAB as you go.

1. Notice the *Command Window* in the centre of the screen. The MATLAB prompt >> indicates that you can type commands. Type in a mathematical expression, such as

   ```
   >> 12 * 12
   ```

   and press Enter. You should see the following:

   ```
   >> 12 * 12
   ans =
      144
   >>
   ```

   MATLAB has calculated the result of the expression that you typed. The prompt >> indicates that another command can be typed.

2. **Ensure that you understand how to translate mathematical expressions into MATLAB code.** Some key mathematical operations are shown below:

Mathematical operation	MATLAB operator	Example of use in MATLAB
$+$	`+`	`1 + 1`
$-$	`-`	`23 - 17`
$\times$	`*`	`123 * 8.2`
$\div$	`/`	`10 / 3`
$\sqrt{\cdot}$	`sqrt( )`	`sqrt(49)`
a^b (power)	`^`	`9^2`

Assessable Task 1

Translate each of the following mathematical expressions into MATLAB code. Write down the *MATLAB code* to calculate this expression, *not the numerical answer*. Your tutor will check your MATLAB code at the end of the lab.

(a)

$$\frac{2^3}{12+\sqrt{7}}$$

(b)

$$9\log_{10} 23$$

(c)

$$\left(\frac{2}{3}\right)^{2/3}$$

Hints:

- You might need to search the MATLAB help to find how to calculate logarithms. To open the MATLAB help, click the Help menu and choose Product Help.
- You should test your MATLAB code by typing it into the MATLAB window.

1. Clear the screen. Type the following:

```
clc
```

   This clears the command window.

2. **Experimenting with variables.** Type the following code into the MATLAB command window:

```
speed = 60
time = 0.25
distance = speed * time
```

Terminology

You just created three *variables*. A variable stores a value, such as the number 60. You can refer to variable by its name, and MATLAB will use the *value* it contains in calculations. You can check the value in a variable by typing its name:

```
>> speed
```

Variable names (such as *speed*, *time*, and *distance*) must satisfy certain criteria. A variable name cannot contain spaces or symbols and cannot begin with a number. Variable names are case sensitive, which means that "speed" and "Speed" are *different variables*.

The equals sign = is called the *assignment operator*. This operator *assigns* the value on the right hand side to the variable on the left hand side.

(a) What do you think will happen if you type

```
distance / time
```

into the MATLAB command window? Try it and see. Notice that Matlab has created a special variable called "*ans*" to hold the result of your computation. The variable *ans* will be overwritten the next time you write a computation without an explicit variable name for the result.

(b) Try creating your own variables. What happens if you assign a different value to an already existing variable?

(c) What happens if you create a variable called "log10", as in:

```
log10 = 14
```

Now try running your answer from Assessable Task 1, Part (b) again. What happened? (Check the MATLAB help on "clear" to undo the damage.)

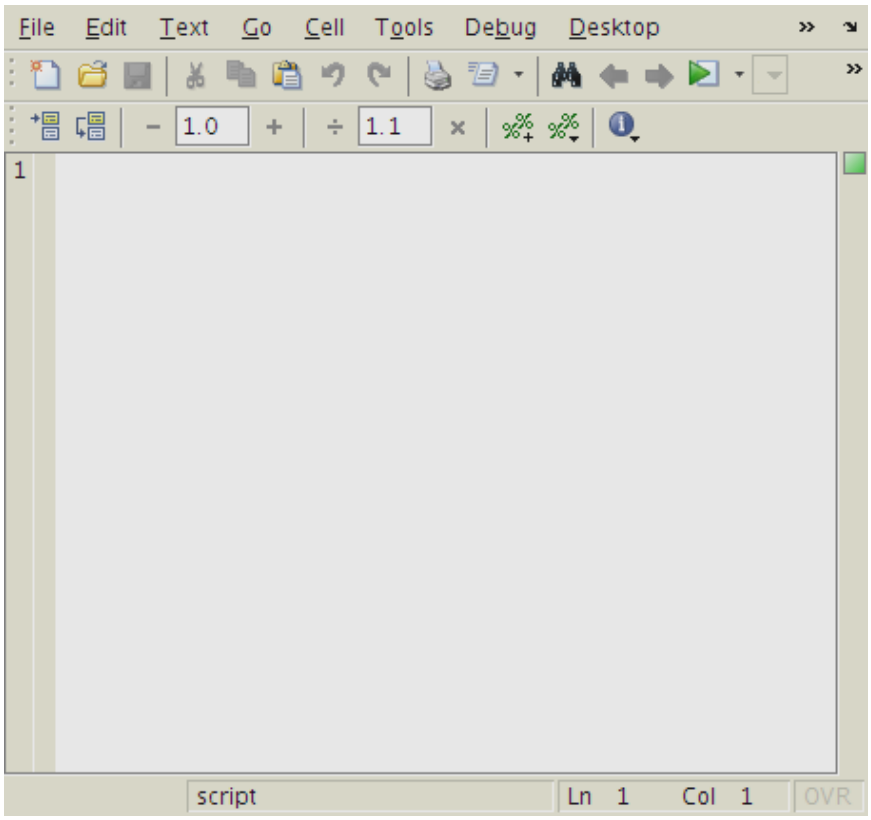
Checkpoint

In this lab - and in all other labs - you should ask your tutor if you are confused. You are being assessed on lab engagement. An excellent way to demonstrate engagement is through discussion with your tutor.
Part (c) above explores an important point: variable names *shadow* function names. If there is a variable and a function with the same name, MATLAB will always use the variable. This may not be what you intended.

3 Creating a MATLAB script

You can save MATLAB commands to a file so that you can run them again without re-typing them. This is called a *script*.

1. Open the MATLAB editor by choosing File -> New -> Script, or clicking the New icon on the toolbar, or by pressing Ctrl+N. You should see a window like the following:



2. Type the following text into the MATLAB window:

```
% EG1002 Lab 1
clc
g = 9.81
m = 85
F = m*g
```

3. Save the file. **Choose a filename without spaces.** All MATLAB script filenames cannot contain spaces or special symbols and cannot begin with a number. Notice that these are the same rules as for variable names; a valid variable name is also a valid filename.

4. Click the Run button (the green "play" button on the editor toolbar). If MATLAB prompts you, choose "**Change Folder**".

5. Look in the MATLAB command window. You should see the following:

```
g =
    9.8100
m =
    85
F =
  833.8500
```

Checkpoint

- The MATLAB commands in your script have been executed, one after the other.
- The *clc* command clears the MATLAB command window, so that your script output will be the only text visible.
- The line beginning with a percent symbol % is a *comment*. Comments are completely ignored by MATLAB. You can write comments to label your code, or to provide information to other programmers reading your code.
- What happens if you put a semicolon after a variable assignment, as in:

```
% EG1002 Lab 1
clc
g = 9.81;
m = 85
F = m*g
```

  Run your code again, this time with the semicolon. Notice that the value is not printed to the command window. The variable assignment occurs without printing any message.

4 Case study: Freight cost calculator

1. On LearnJCU, under Weekly Labs, Week 1, locate the file "freight_calc.m". Download this file and save it to your computer.
2. Open freight_calc.m in MATLAB. Read through the code. Run it and observe the output.

Checkpoint

- Make sure you understand the *if-else-end* block at the bottom of the file. You must understand this structure before you can complete the final assessable task.
- You may find it useful to set a breakpoint and step through the code line by line. Breakpoints were explained in the screencasts. Alternatively, you can read about them in the MATLAB help at `http://www.mathworks.com.au/help/techdoc/MATLAB_env/brqxeeu-175.html#brqxeeu-184`

Assessable Task 2

Modify freight_calc.m so that freight is calculated according to the following rule:

The cost of freight is the largest of:

(a) The weight cost, which is $4.10 / kg.

(b) The volume cost, which is $100.00 / cubic meter.

(c) The distance cost, which is $45.00 / 100 km.

Once you have modified this script in accordance with the rule above, raise your hand and your tutor will check your code.