

PROJECT 2 - MATLAB AS A CALCULATOR

Objectives: To learn how to use MATLAB as a calculator, in particular:

- How to enter numbers
- How to use variables
- How to do arithmetic and use scientific functions
- How to use scripts and basic input/output

Outcome:

- Write basic scripts that take input, make a calculation and display the results.

1. Back to MATLAB

1. In Windows, go to the **Start** menu and find MATLAB. Select it and wait for MATLAB to start.
2. Arrange the windows so that you can both see this document and the MATLAB window. We'll be working primarily in the Command Window in MATLAB, so if you need space, you can close the other panes and resize the window.
3. Command Window:
 - (a) To get MATLAB to do things you type commands into the command window and then when you hit return, MATLAB executes the command, and displays the result. Try typing $7 + 9$ and then hit Enter.
 - (b) If you make a mistake MATLAB will give an error message and try to point to where the mistake was. Try typing $7 + (9 - 6$ and hit Enter.
 - (c) You can use the left-right arrow keys to edit a command before you hit enter. You can use the up-down arrow keys to go review previous commands. Try using the up arrow to return to the statement $7 + (9 - 6$, add the closing parenthesis, and hit enter.
 - (d) If you want MATLAB to stop executing a command or you want it to just ignore a command you're typing, use control-C (ctrl-C).

2. Calculation Basics

1. Numbers use the digits 0 – 9, decimal point $.$, sign $-$, like -3.14159 . For scientific notation, use e , like $1e - 4$ for 0.0001. For complex numbers use i , like $4.7 - 3.2i$. Be careful to not use spaces, because, as we'll see later, MATLAB also works with lists of numbers and uses the space as a separator.
2. Variable names use letters, upper and lower case, numbers and the underscore. Variables are case sensitive so 'A' is different from 'a'. There are a few reserved words, but not many and MATLAB will give you an error message if you try to use one. Unfortunately, you can create variables that override some built-in functions.
3. Variable names should be descriptive so that they are meaningful, but they shouldn't be too long.
4. Arithmetic uses $+$, $-$, $*$, $/$, and \wedge for addition, subtraction (and negation), multiplication, division and exponentiation, respectively. Use parenthesis: (and), to group terms, otherwise normal precedence holds.
5. Basic Scientific functions are **sin**, **cos**, **tan**, **log** (natural log), **exp** (for e^x) and **log10**. There are many other built-in functions, but these are the major ones. Also, **pi** is π .

3. Putting it Together (Scripts and Input/Output)

1. MATLAB is both interactive, i.e. you type expressions in the command window and MATLAB evaluates it; and it is a programming environment where you can save your command(s) in a file and execute them.
2. All programs in MATLAB are called m-files as their filenames are always of the form *filename.m*. There is also a built-in M-file editor in MATLAB that helps format your commands.

3. For your first program, we'll create a sample script. From the File menu in MATLAB, select New: M-File. You should get a new blank window. In the window, type the following exactly:

```
% Script Example by Chuck Collins
% Takes two numbers and adds them together.

% Enter 2 numbers
n1 = input('Enter a number: ');
n2 = input('Enter another number: ');
result = n1 + n2;           % sum the two numbers
% display the results
disp('The sum of the two numbers you entered is')
disp(result)
```

4. Notes: Anything after a % is a comment. As always a program should be labeled with the author's name and the programs purpose at the beginning. The different parts of the program are also commented as to their function or purpose.
5. Two new commands: `input` and `disp`. I think their function is obvious for accepting input from the user and displaying text and results. Note that `disp` accepts only one argument, i.e you can't do `disp(x,y)`.
6. Save the script in the folder 'My Documents' and give it the name **proj2samp.m**.
7. We have to tell MATLAB where to find your script. So, at the top of the command window where it says 'Current Directory' browse and find the folder 'My Documents' and click on it.
8. Now, in the command window, type **proj2samp** (this runs the program you typed in). If you don't get any error messages then you should see the computer asking you to Enter a number. Do so and hit return, enter another number and see if it prints out the right result.
9. If MATLAB says it is an undefined function, go back and make sure that you named it correctly (case matters) and that 'My Documents' is your Current Directory.
10. If you have any other problems, MATLAB should give you a message trying to point out where the problem is. Go back to the M-file, make the correction, **save it**, and run it again.

4. To Do and Turn In

1. Create a script called **proj2eval.m** that evaluates the following expressions. Put your name in the script. Label each evaluation with a comment and use appropriate variables.

- (a) Find the volume of the cylinder of diameter 3.4 and height 4.6. ($V = \pi r^2 h$).
- (b) With $x = 0.2$ and with $x = 1.2$, evaluate

$$f = \frac{x^3 - 2x^2 + x - 6.3}{x^2 + 0.0005x - 3.14}.$$

- (c) With $r_1 = 100$, $r_2 = 200$ and $r_3 = 300$, evaluate

$$R = \frac{1}{\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3}}.$$

This is the formula for the total resistance of three resistors in parallel.

- (d) For $r = -0.3$ and $\theta = \frac{3\pi}{4}$, evaluate

$$y = e^r \cos(\theta) + e^{2r} \sin(2\theta).$$

In the command window, before you run the script, type `diary proj2eval.out`. This will put everything that you type and MATLAB displays in a file. Now, run your script and then type `diary off`.

Submit your script **proj2eval.m** and the diary file **proj2eval.out**. They should be located in your 'My Documents' folder.

2. Write a script like the example script given above so that it accepts a temperature in Celsius and displays the temperature converted to Fahrenheit. ($F = 9/5C + 32$). Call the script **proj2temp.m**.
Run it and save all the results in a diary file **proj2temp.out**. Try converting the temperatures, -40, 0, 30 and 100.
Submit your script and the diary file.
3. You should submit 4 files: two scripts and two output files.
4. Finally, go back to the Assignment page and fill out the survey.