# Department of Computing

# MATH 333: Numerical Analysis

# Class: BSCS-9ABC

# Lab 2: Matlab Basics (Function, Files, Decision/Loop Structures)

# Date: February 9, 2022

# Time: 10:00 am-1:00 pm

# Lab Engineer: Anum Asif

# Lab 2: Matlab Basics (Function, Files, Decision/Loop Structures)

**Introduction**

MATLAB, which stands for MATrix LABoratory, is a state-of-the-art mathematical software package, which is used extensively in both academia and industry.

**Objectives**

The purpose of this lab is to get familiar with the type of matlab files and matlab computation/simulation using loop/decision structures.

**Tools/Software Requirement**

Matlab R2016a

**Lab Tasks**

**Task 1**

Use MATLAB to calculate the expression a(b + c(c + d))a,

where a = 2, b = 3, c = −4 and d = −3.

**CODE:**

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

**OUTPUT:**

|  |
| --- |
| Graphical user interface, application  Description automatically generated |

**Task 2**

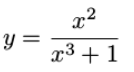
1. Construct the polynomial y = (x + 2)2 (x3 + 1) for values of x from minus one to one in steps of 0.1.

**CODE:**

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated |

**OUTPUT:**

|  |
| --- |
| Chart, line chart  Description automatically generated |

1. Construct the function for values of x from one to two in steps of 0.01. Also plot the result.

**CODE:**

|  |
| --- |
| **A picture containing text  Description automatically generated** |

**OUTPUT:**

|  |
| --- |
| **Chart  Description automatically generated** |

**Task 3**

**M-Files**

To take advantage of MATLAB’s full capabilities, we need to know how to construct long (and sometimes complex) sequences of statements. This can be done by writing the commands in a file and calling it from within MATLAB. Such files are called “m-files” because they must have the

filename extension “.m”. This extension is required in order for these files to be interpreted by MATLAB.

**Types of M-Files**

There are two types of m-files:

* Script files.
* Function files.

Script files contain a sequence of usual MATLAB commands, that are executed (in order) once the script is called within MATLAB. For example, if such a file has the name compute.m , then

typing the command compute at the MATLAB prompt will cause the statements in that file to be executed.

1. **Create m-file (which would be a script) and save it as sine.m. It should plot the sine of x which ranges from -1 to 1 with the step of 0.1.**

**Function files**, on the other hand, play the role of user defined commands that often have input and output. You can create your own commands for specific problems this way, which will have the same status as other MATLAB commands.

Every MATLAB function begins with a header, which consists of the following :

* The word function.
* The output(s) in brackets, (the variable a in the above example)
* The equal sign.
* The name of the function, which must match the function filename
* The input(s) (the variable x in the above example).

Any statement that appears after a “%” sign on a line is ignored by MATLAB

and plays the role of comments.

function [output] = function\_name(input)

**CODE:**

|  |
| --- |
| **A picture containing background pattern  Description automatically generated** |

**OUTPUT:**

|  |
| --- |
| **Chart, line chart  Description automatically generated** |

1. **Create m file (a function file) and named it as average.m. It should compute the average of the vector passed as an input.**

**CODE:**

|  |
| --- |
| Graphical user interface, application  Description automatically generated |

**OUTPUT**:

|  |
| --- |
| Background pattern  Description automatically generated with low confidence |

1. **Run the below commands and see the result.**

a = input(‘First number ’);

b = input(’Second number ’);

disp([’ Their sum is ’ num2str(a+b)])

disp([’ Their product is ’ num2str(a\*b)])

**CODE:**

|  |
| --- |
| **Text  Description automatically generated** |

**OUTPUT:**

|  |
| --- |
| **A picture containing graphical user interface  Description automatically generated** |

**Task 4**

The “for” loop allows us to repeat certain commands. If you want to repeat some action in a predetermined way, you can use the “for” loop.

for *index* = *values*

*statements*

end

**Write a loop statement which iterates 5 times and displays the values with the increment of 3.**

**CODE:**

|  |
| --- |
| **Graphical user interface, application  Description automatically generated with medium confidence** |

**OUTPUT:**

|  |
| --- |
| Graphical user interface, text  Description automatically generated with medium confidence |

**Task 5**

There are times when you would like your algorithm/code to make a decision, and the “if” statement is the way to do it. The general syntax in MATLAB is as follows:

if relation

statement(s)

elseif relation %if applicable

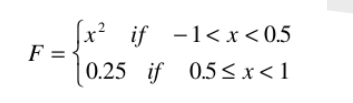
statement(s) %if applicable

else %if applicable

statement(s) %if applicable

end

**Code the below equation using decision structure and loop.**



**CODE:**

|  |
| --- |
| %% TASK 5  clc  for x = -1:0.1:1  if x<0.5 && x>=-1  disp(x.^2)  elseif x>=0.5 && x<=1  disp(0.25)  end    end |

**OUTPUT:**

|  |
| --- |
| Table  Description automatically generated |

**Deliverables**

Submit single word file with matlab code and screen shot of Output.