# Department of Computing

# MATH 333: Numerical Analysis

# Class: BSCS-6AB

# Lab 2: Graphical Method and Taylor Series

# Date: February 01, 2019

# Time: 10:00-12:50hrs & 14:00-16:50hrs

# Instructor: Atifa Kanwal

# Lab Engineer: Azaz Farooq

# Lab 2: Graphical Method and Taylor Series

**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 Graphical Method and Taylor series.

**Tools/Software Requirement**

Matlab R2016a

**Description**

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

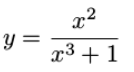
where a = 2, b = 3, c = −4 and d = −3.

Matlab code:

>>a = 2; b = 3; c = -4; d = -3;

>> a\*(b+c\*(c+d))\*a

Construct the polynomial y = (x + 2) 2 (x 3+ 1) for values of x from minus one to one in steps of 0.1.

Construct the function for values of x from one to two in steps of 0.01.

x = -1:0.1:1;

f = x+2;

g = x.ˆ3+1;

y = (f.ˆ2).\*(g);

x = 1:0.01:2;

f = x.ˆ2;

g = x.ˆ3+1;

y = f./g;

**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.

Create m-file and named it as sine and save it as sine.m. The contents of the file should be

format long

x = [0.1, 0.01, 0.001];

y = sin(x)./x

Go to console window and run sine command.

a = input(‘First number ’);

b = input(’Second number ’);

disp([’ Their sum is ’ num2str(a+b)])

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

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 j=1:4

j+2

end

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)

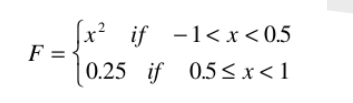
elseif relation %if applicable

statement(s) %if applicable

else %if applicable

statement(s) %if applicable

end



» x=-1:0.01:1;

» for i=1:length(x)

if x(i) < 0.5

F(i) = x(i)^2;

else

F(i) = 0.25;

end

end

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.

create m file and named it as log3.m. The contents of file should be:

function [a] = log3(x)

%[a] = log3(x) - Calculates the base 3 logarithm of x.

a = log(abs(x))./log(3);

% End of function

Go to console window and run log3(5)

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 (log3 in

the above example)

* 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] = xsq(input)

output = input.ˆ2;

**Lab Task**

1. Given

Determine the root of equation using Graphing technique correct to two decimal places in and .

### Solution:

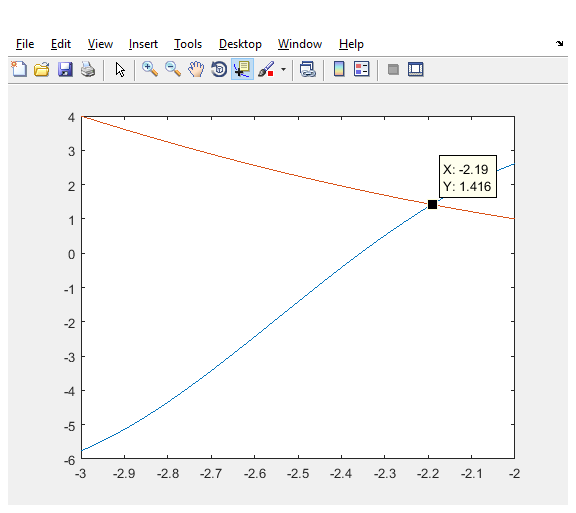
>> x=-3:0.01:-2;

>> f1 = (2.\*x).\*cos(2.\*x);

>> f2 = (x + 1).^2;

>> plot(x,f1)

>> hold on

>> plot(x,f2)  


>> x = -1:0.01:0;

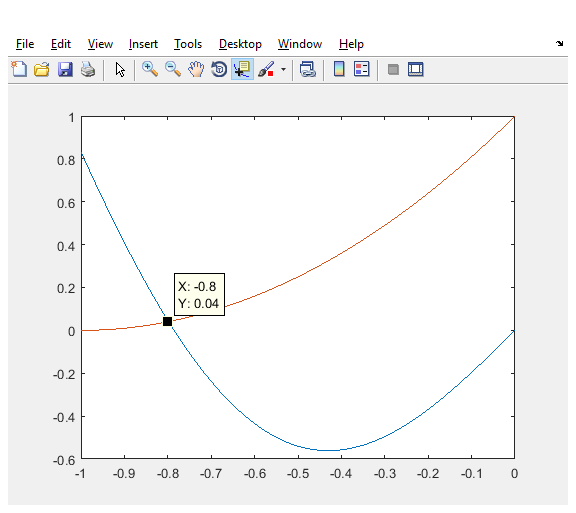
>> f1 = (2.\*x).\*cos(2.\*x);

>> f2 = (x + 1).^2;

>> plot(x,f1)

>> hold on

>> plot(x,f2)



**2.** Write Taylor polynomial of about x=0 upto

a) P1=second order derivative.

b) P2= forth order derivative.

c) P3= sixth order derivative.

d) Plot f(x), P1, P2 and P3 on the same plane by classifying them using different colors.

syms x

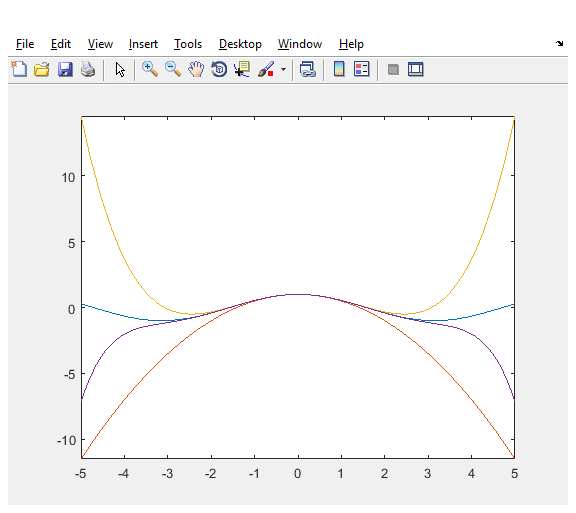
f = cos(x);

t1 = (taylor(f, x, 'Order', 2+1));

t2 = (taylor(f, x, 'Order', 4+1));

t3 = (taylor(f, x, 'Order', 6+1));

fplot([f t1 t2 t3])



**Deliverables**

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