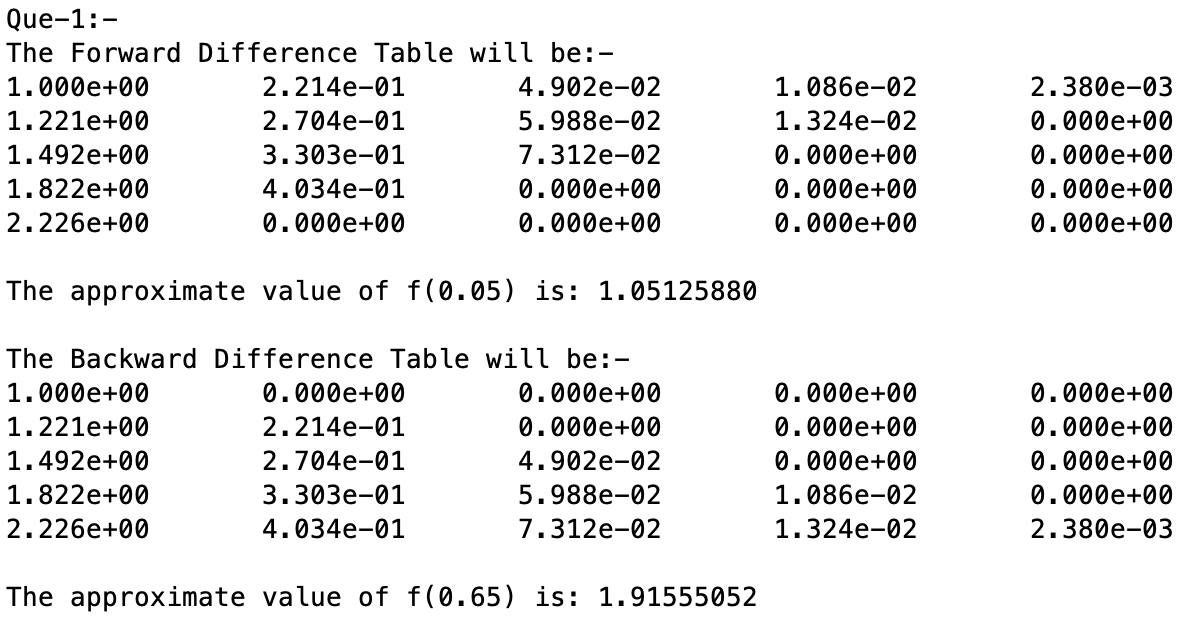
MA322 – Scientific Computing Lab

Lab – 05

Dipanshu Goyal 210123083

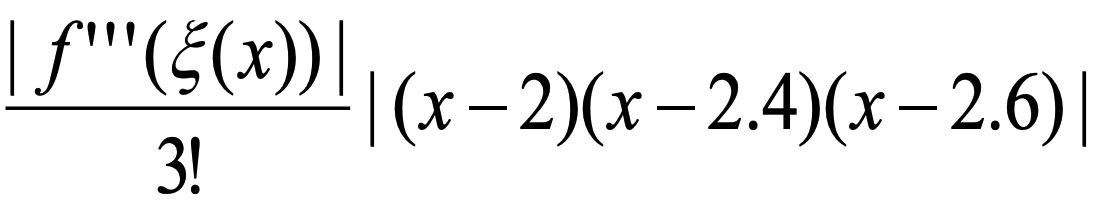
# Run output\_file.m to run the code.

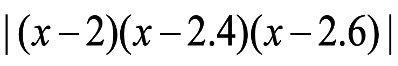
# Ques – 1



# Ques – 2

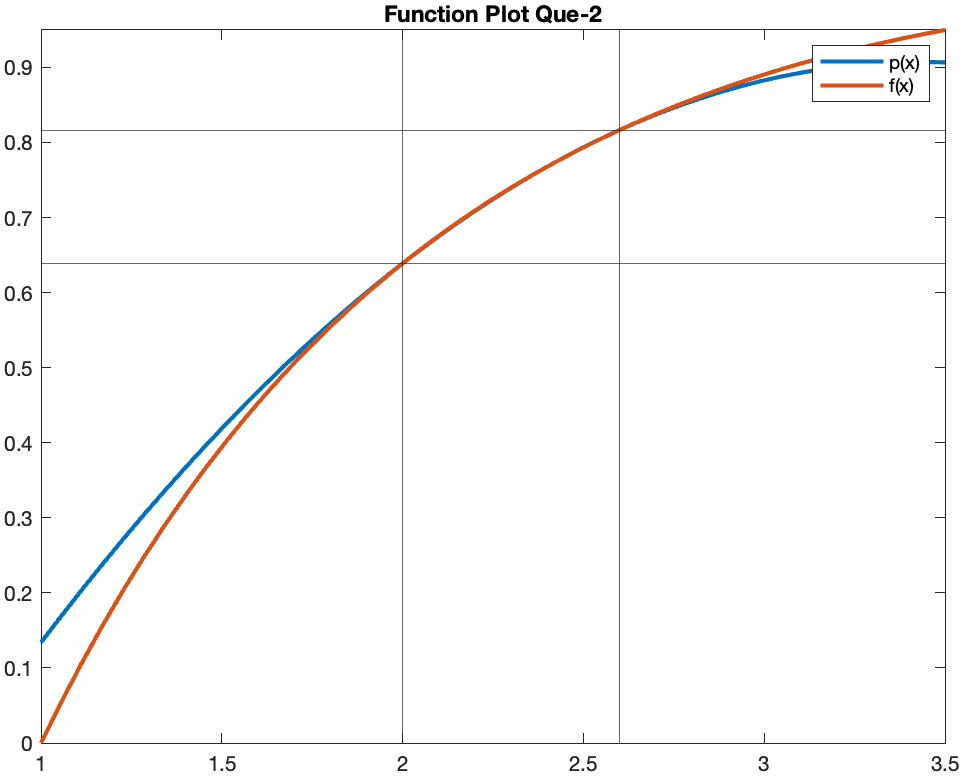
Here, for calculating the theoretical error bound, I found the maximum value of the following expression: -

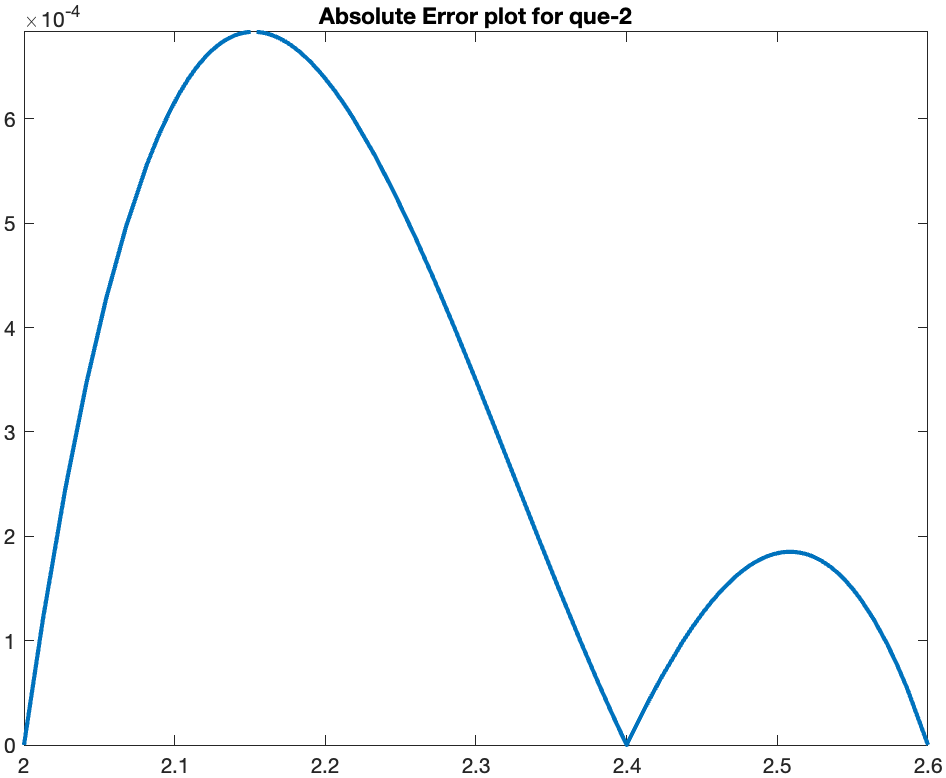


in the interval [2, 2.6] The maximum value of is attained at 2 only as f’’’(x) is decreasing, and max value of is attained around x = 2.157 which thus gives the theoretical Error bound as: -9.457890 \* 10-4.

Now, to calculate this error bound in MATLAB, I took an array of x values from 2 to 2.6 and xi+1 – xi ­= 0.001, and calculated the values of both terms at all x, took max m1 and m2 and gave error bound as m1\*m2/6. This error bound turned out to be close to the theoretical error bound.

Also, the error plot is given by plotting |p(x) – f(x)| and the function plots are given by simply plotting f(x) and p(x). These show that p(x) successfully interpolates f(x) in interval [2, 2.6].



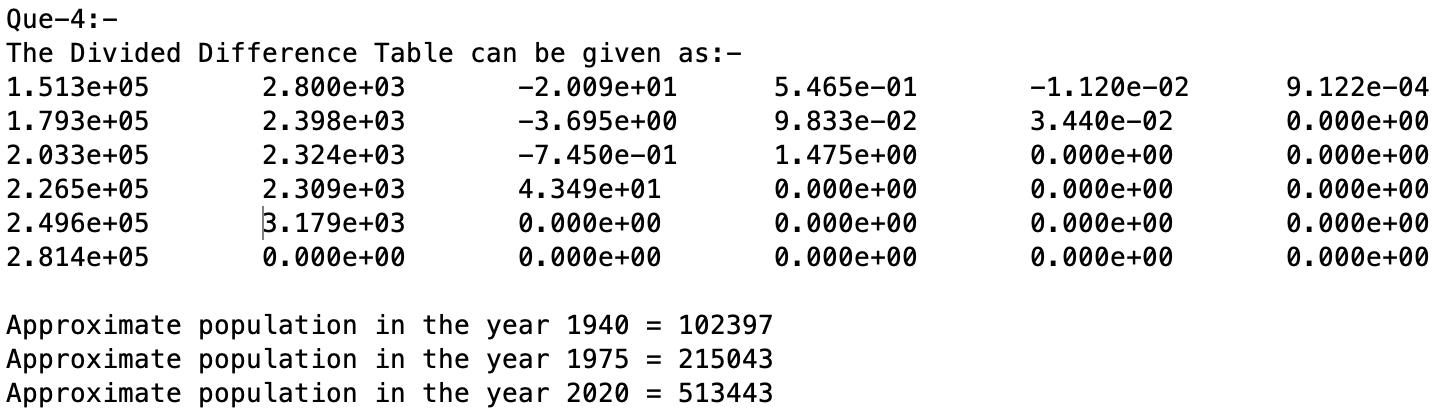


# Ques – 3

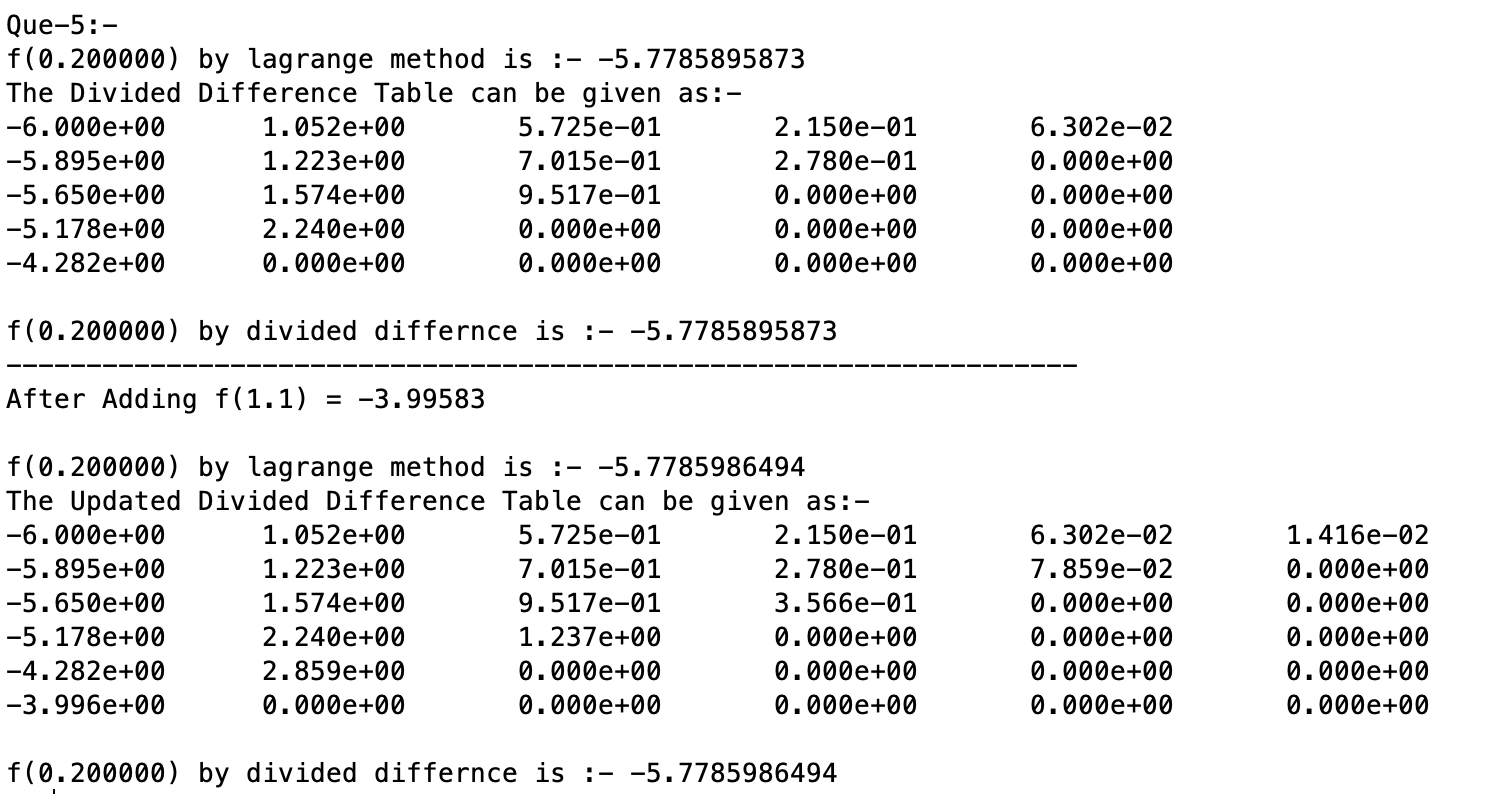
Part-(a): - f (0.430000) by Lagrange method is: - 2.3606047341

Part-(b): - f (0.900000) by LaGrange method is: - 0.4419850025

# Ques – 4



# Ques – 5



Here, after adding f (1.1), instead of calculating the whole table again for Newton’s Divided Difference method, I resized and updated the existing table and re-calculated the interpolating polynomial p(x) by adding the correction term to it.

We can see both the methods are giving same answer for both the tables, this is because the interpolating polynomial (p(x)) formed by both the methods is completely same, just the method of calculating that is different, both methods are just different representations of each other.

Also, after adding f (1.1) = −3.99583 to the table, the solution is changed but the change is of order 10-5 i.e., negligible.