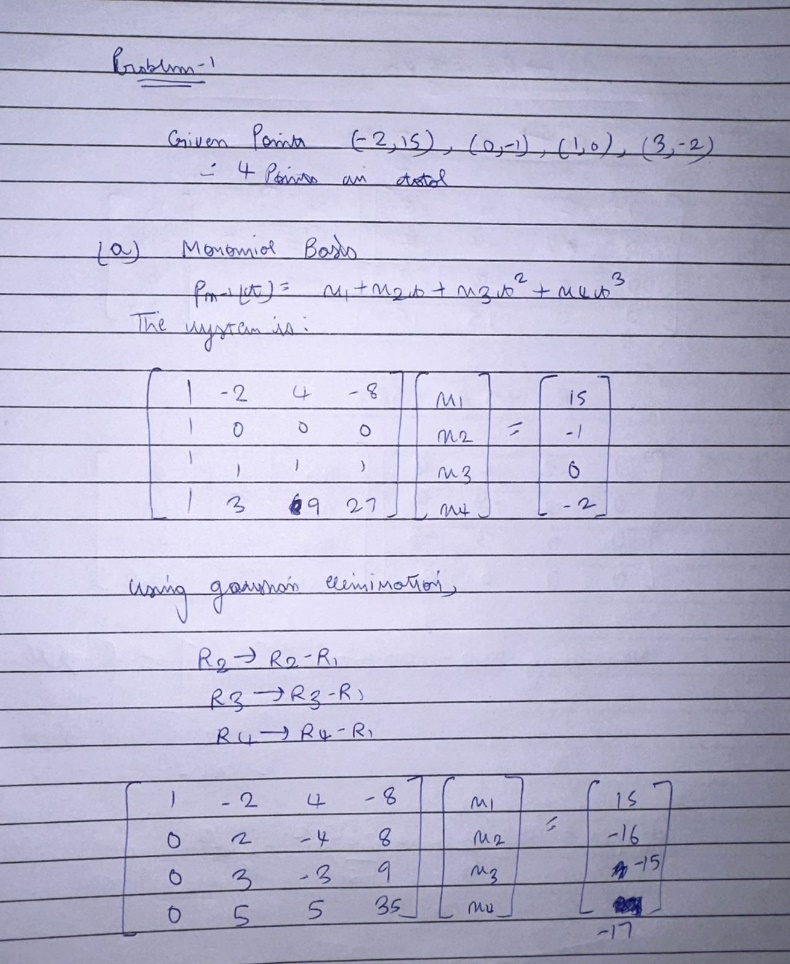
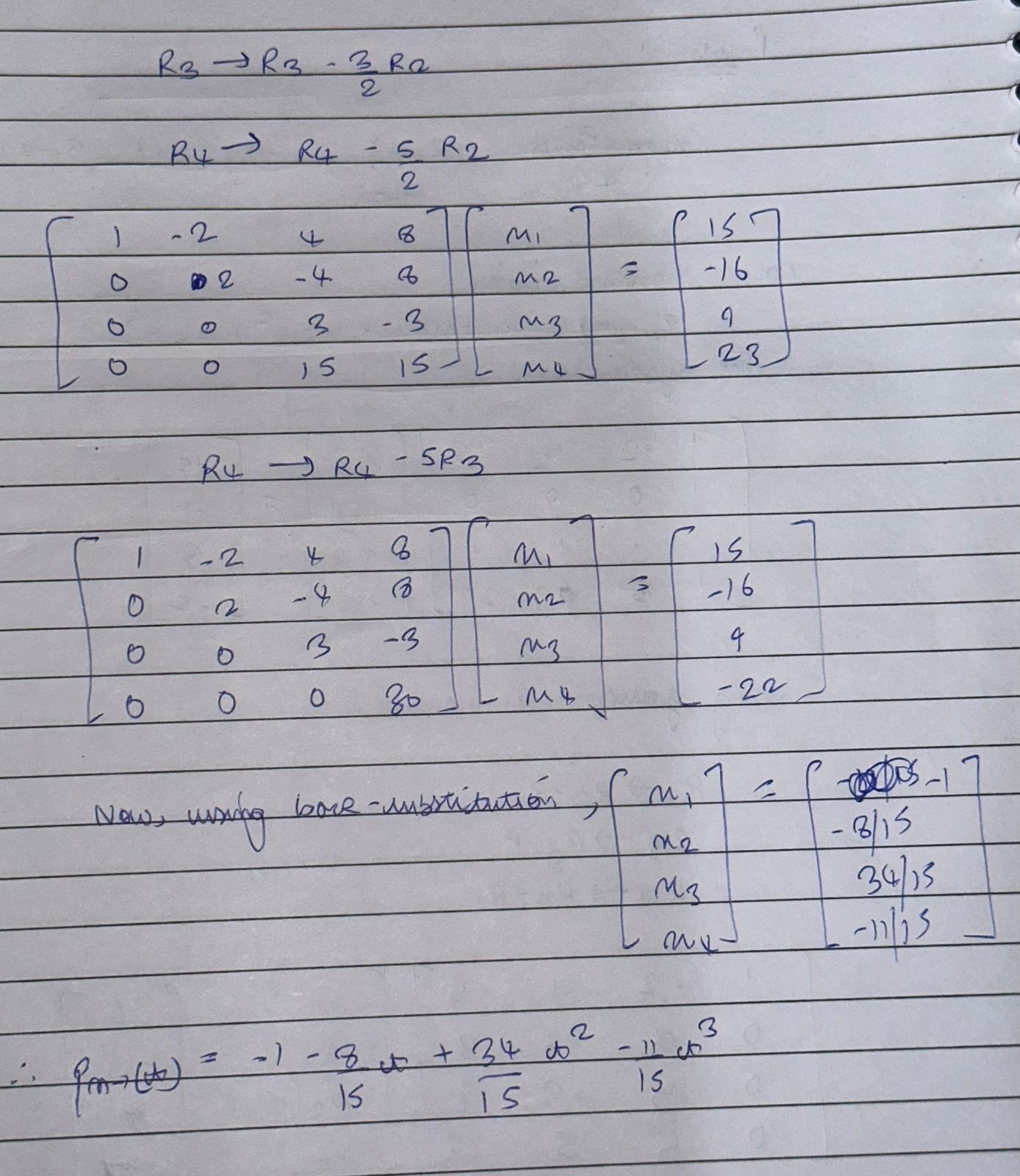
**Name: PARMESH YADAV**

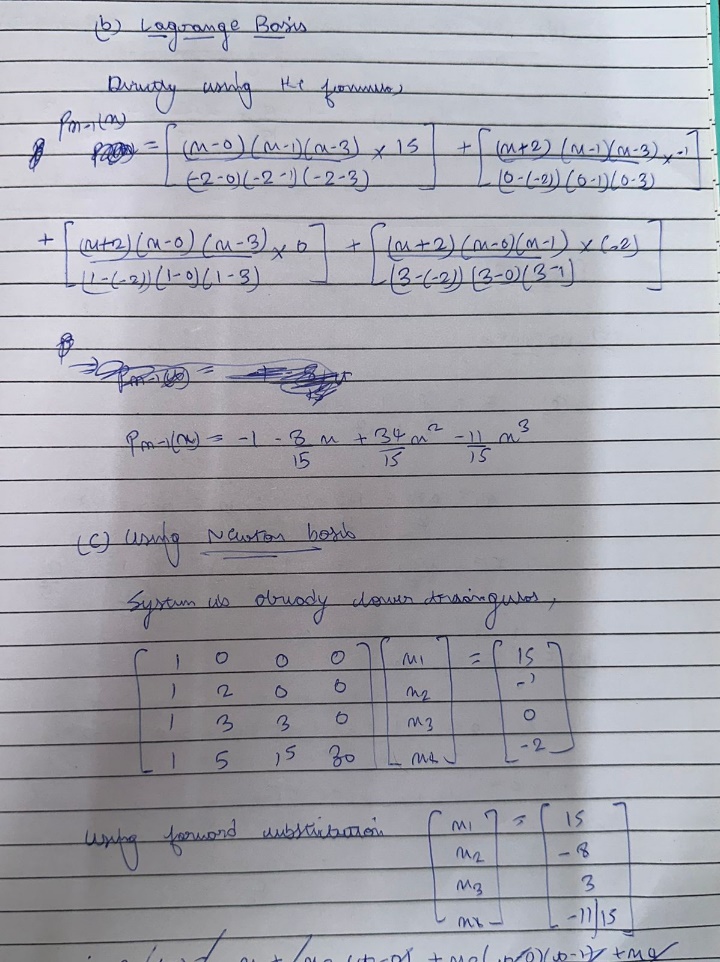
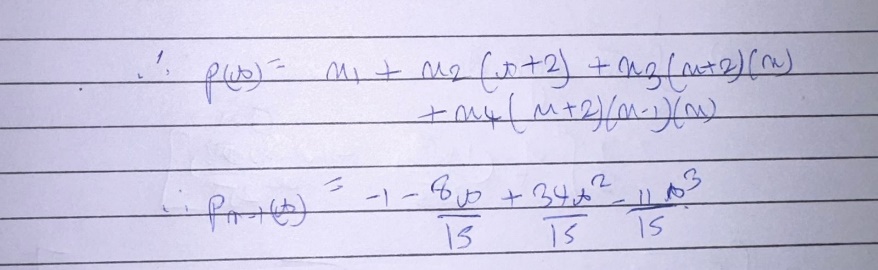
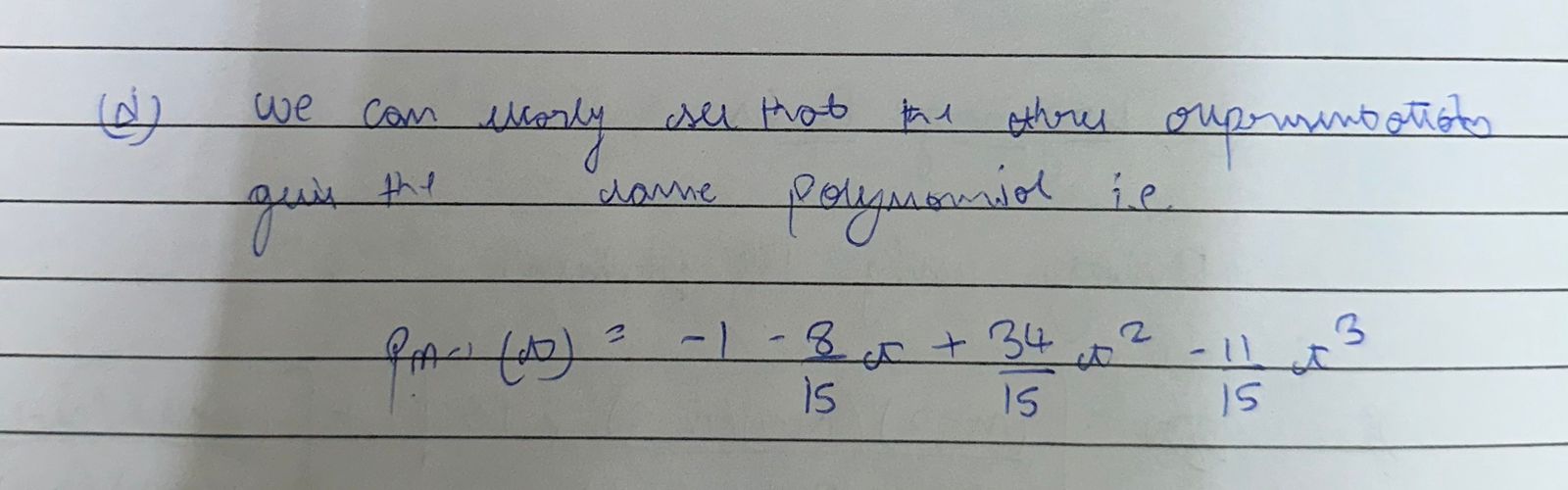
**Roll No: 2020319**

**SCIENTIFIC COMPUTING (MTH373)**

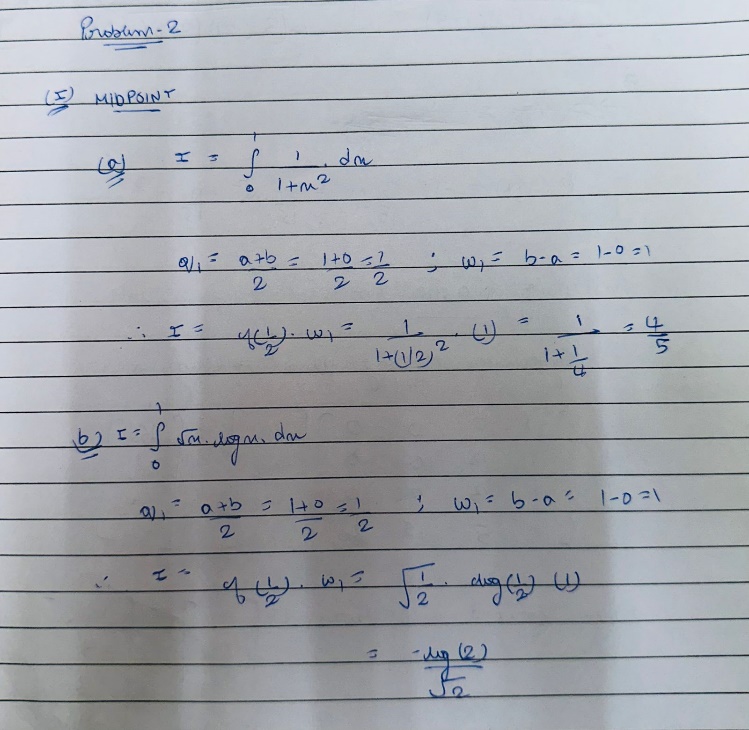
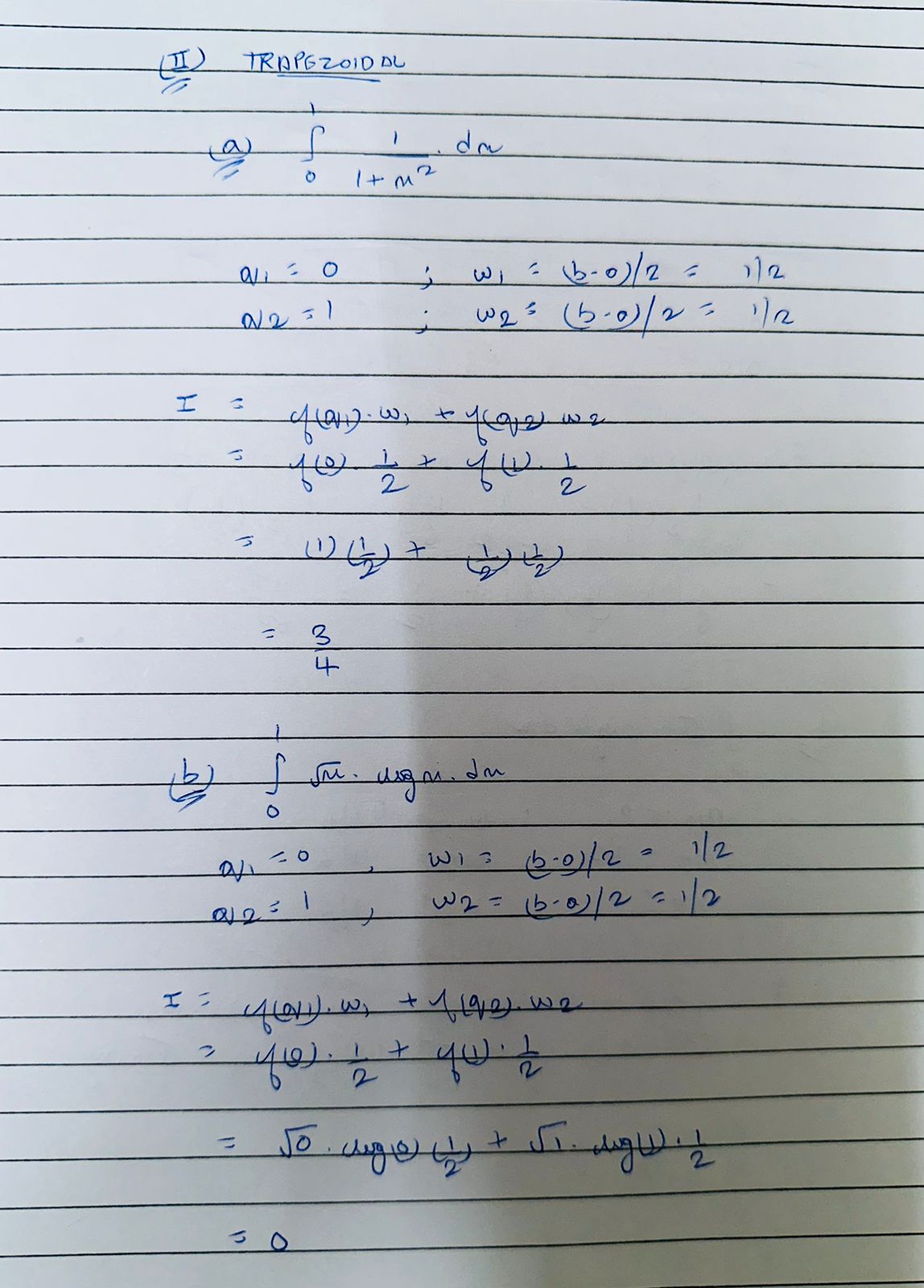
**HOMEWORK – 4**

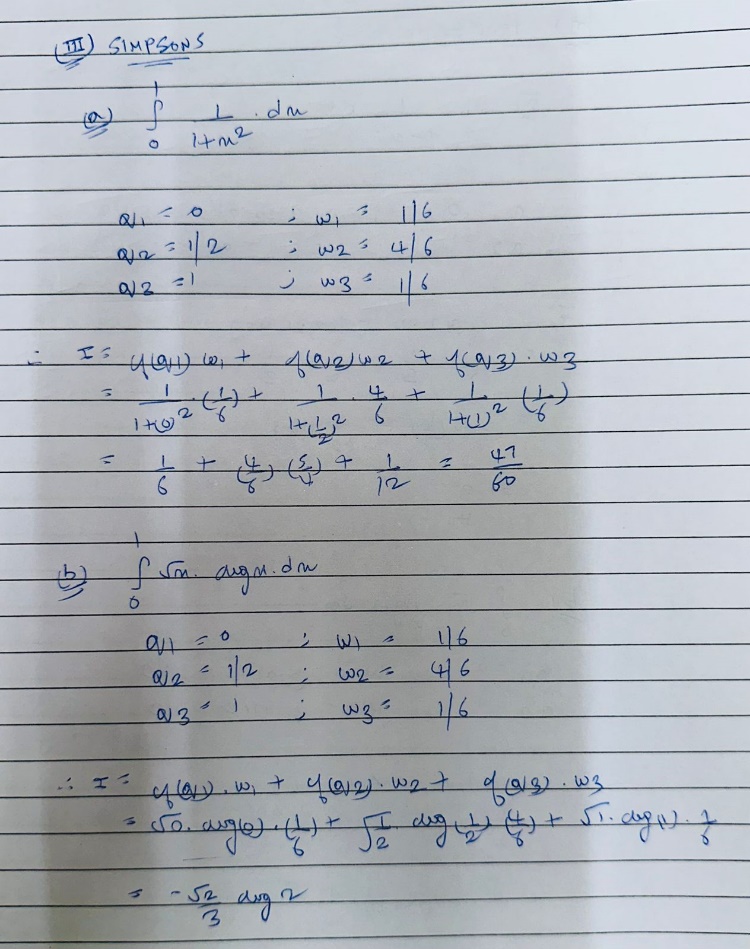
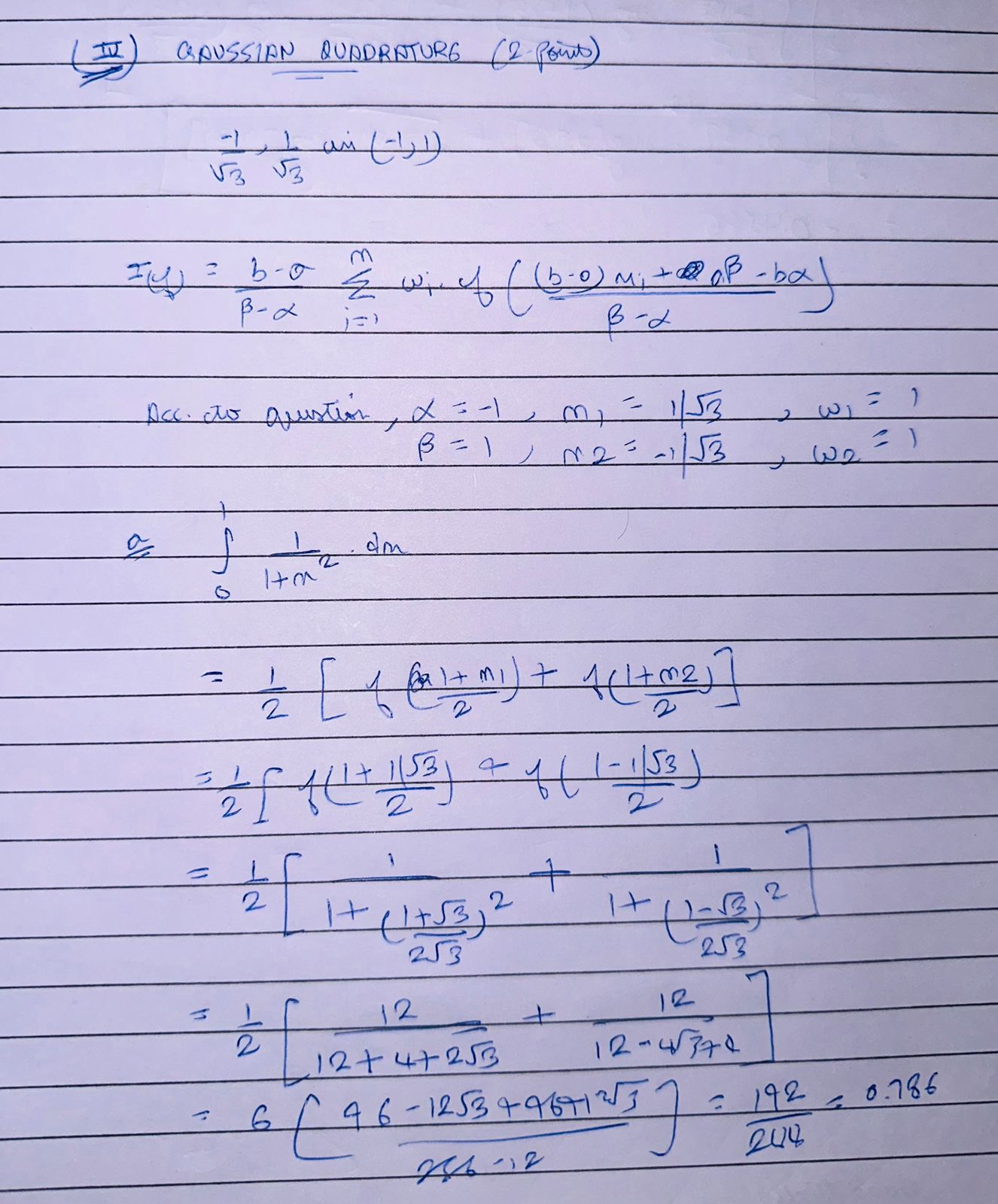
Problem – 1

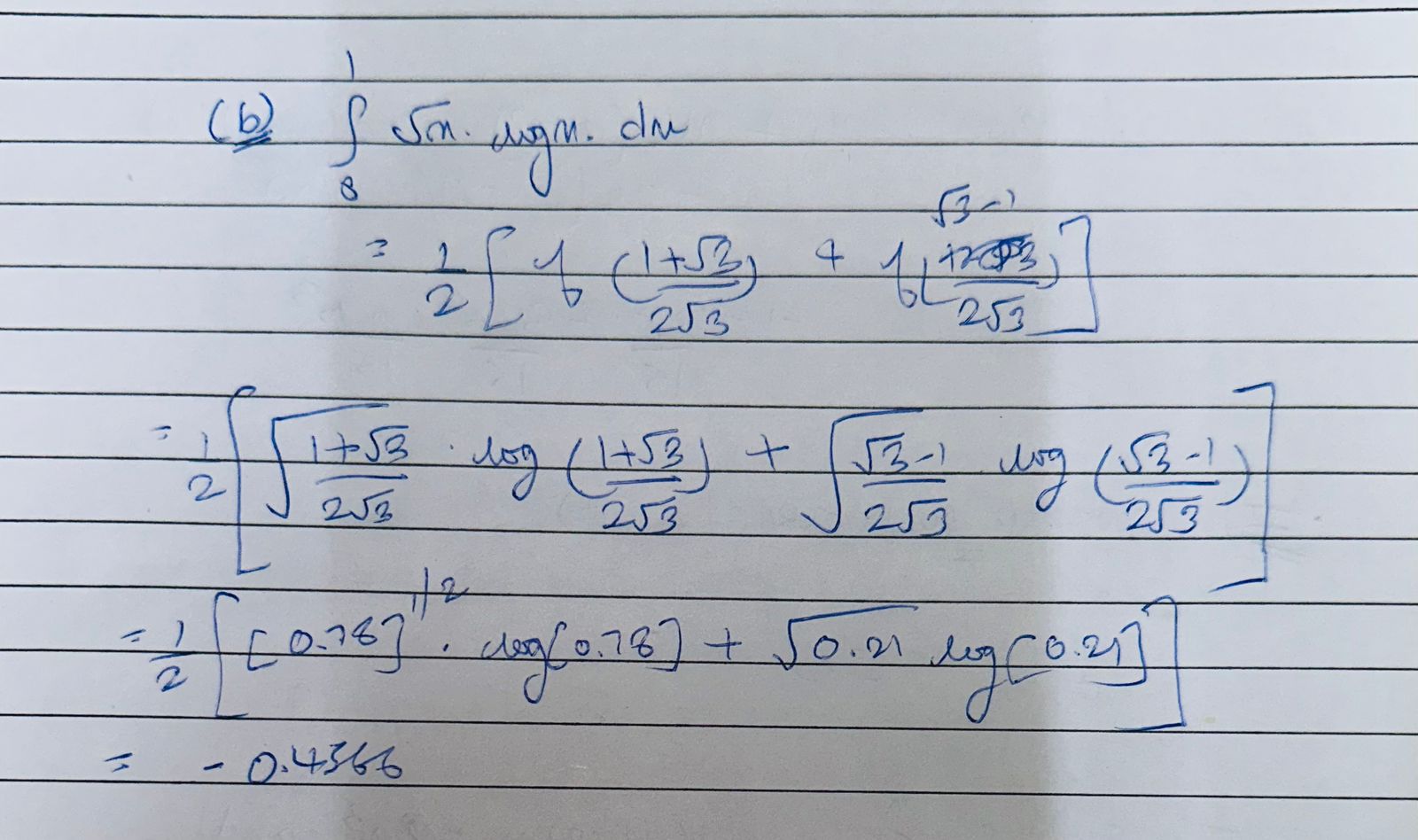
 

Problem – 2

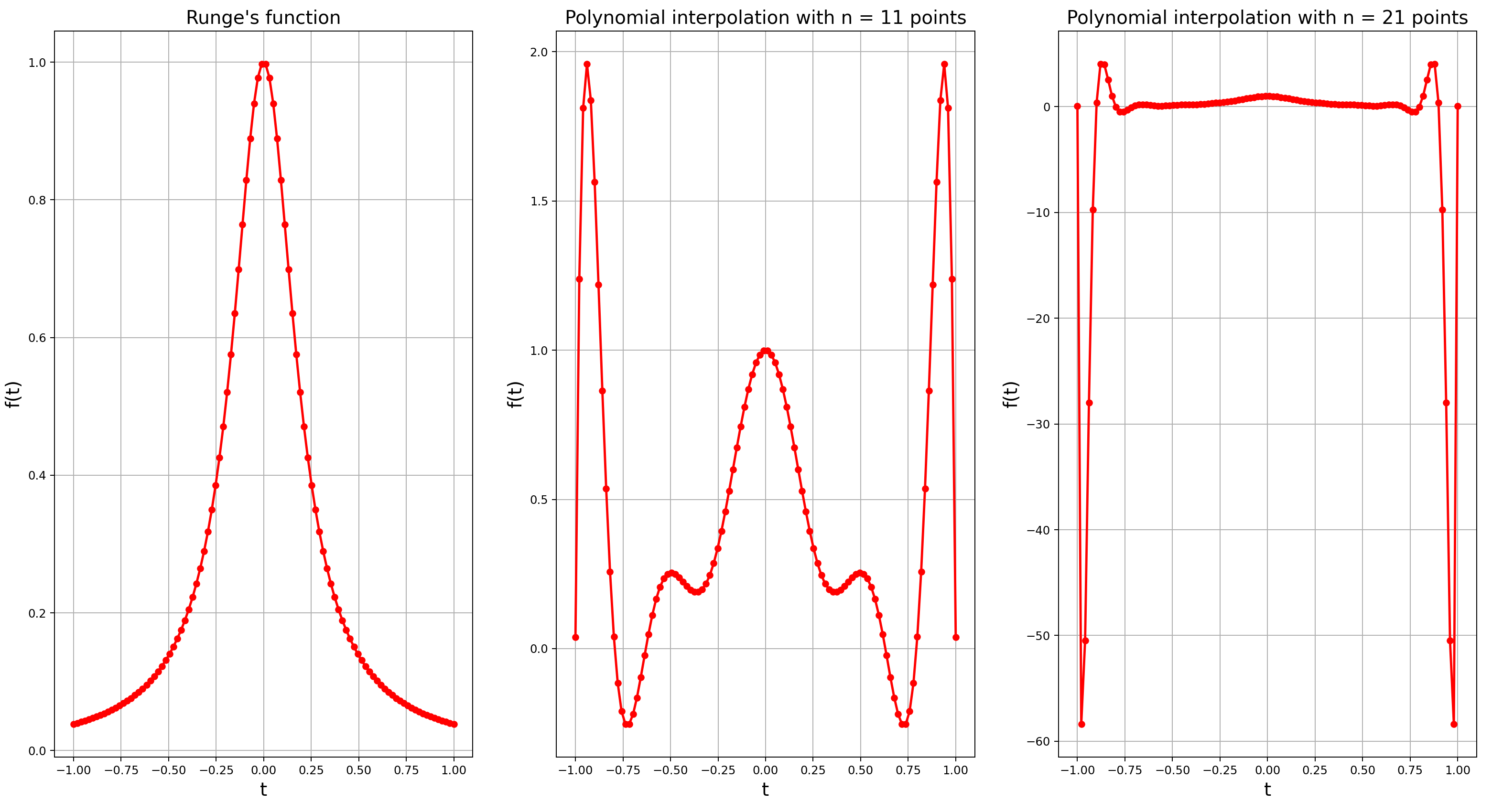
 



Problem – 3

1. Using polynomial interpolation

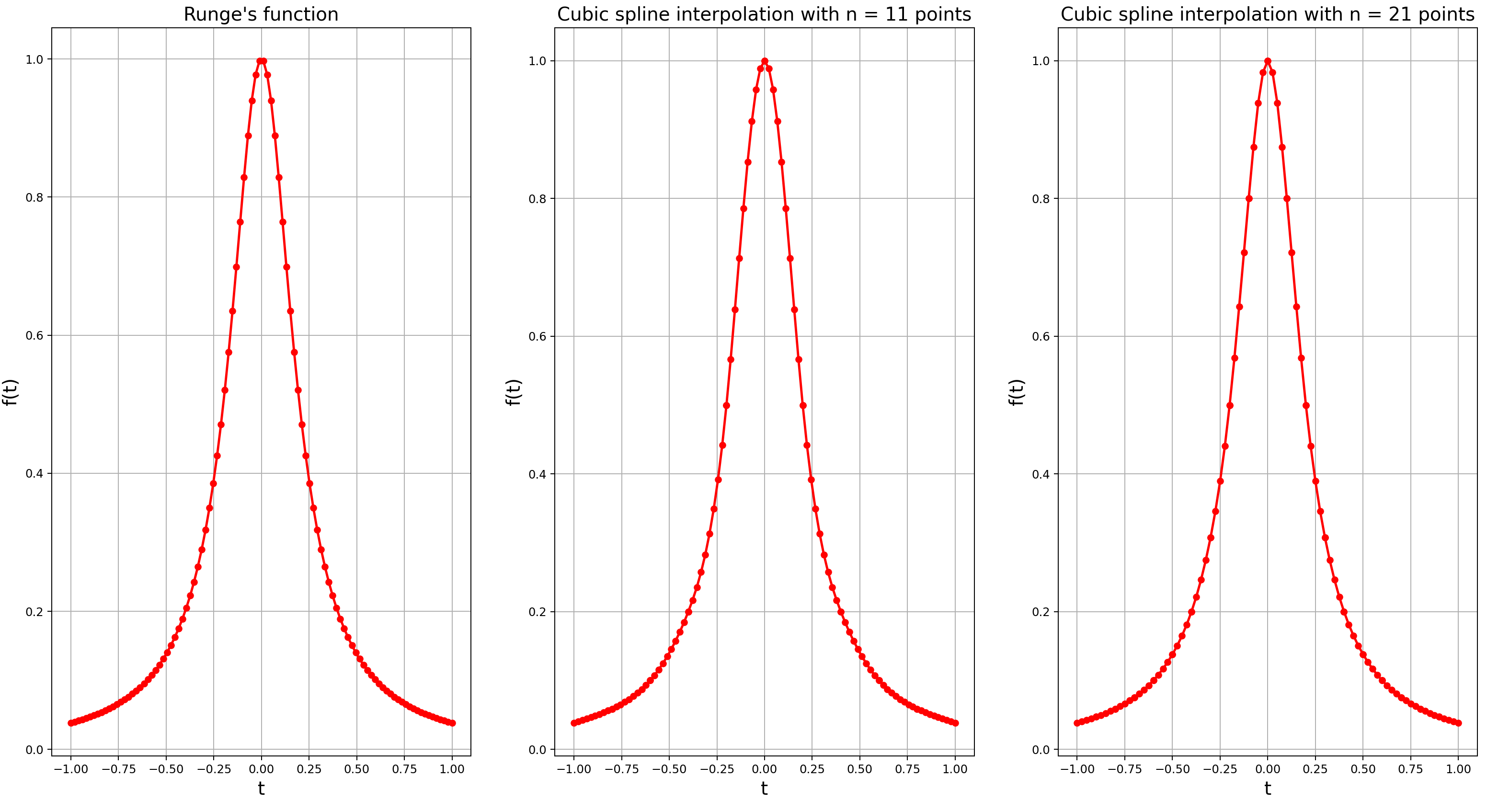
Plotting result using 100 points on interval [-1,1]



Using polynomial interpolation, we can observe that output of interpolation matches the original function up to a certain point in the provided domain and becomes less accurate when reaching the endpoints of the domain. Hence, we can say that polynomial interpolation is not very effective at the corners of any given domain. This is further verified by that fact that we studied in class that the vandermond matrix used in polynomial interpolation is ill conditioned at high values of n i.e., in this case n = 21.

1. Using Cubic Spline Interpolation

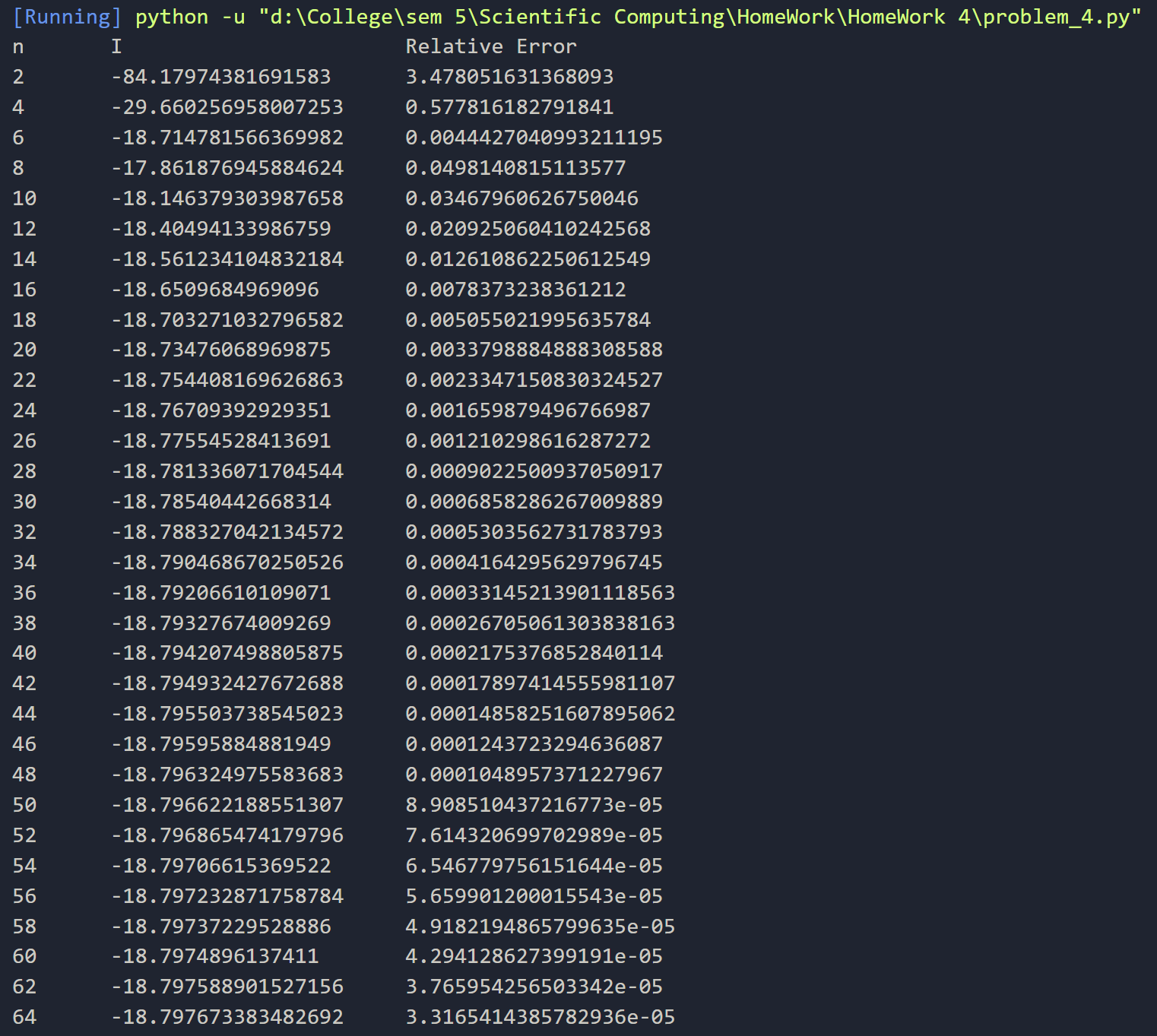
Plotting result using 100 points on interval [-1,1]



For n = 11, there will be 10 cubic, so 10 points for each polynomial and for n =21, there will be 20 cubic, so 5 points in each polynomial, in total 100 points plotted in each case. As we can see in the above graph, cubic spline interpolation given a way better result than polynomial interpolation mainly because it used piecewise polynomial approach rather than considering the whole domain as a single polynomial.

Problem – 4

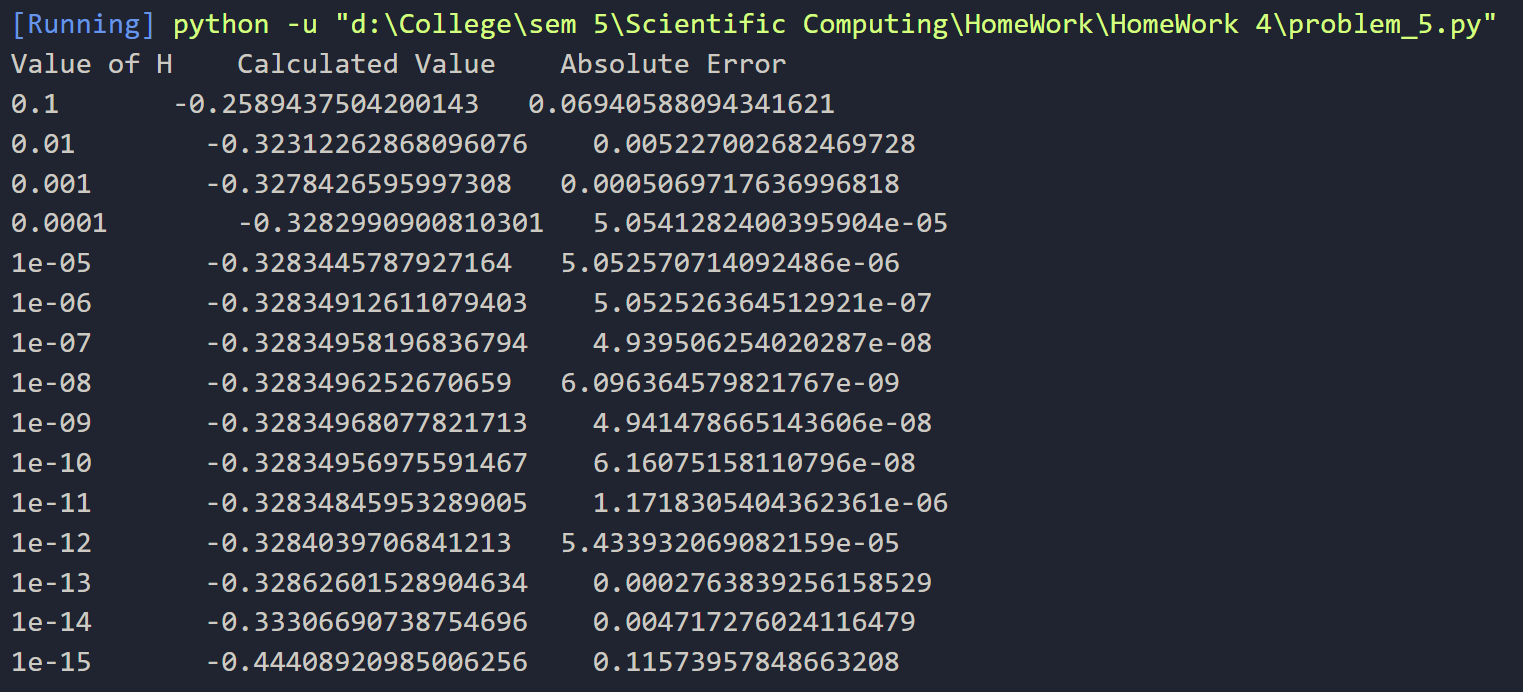
Output of calculated values with relative error for each n.



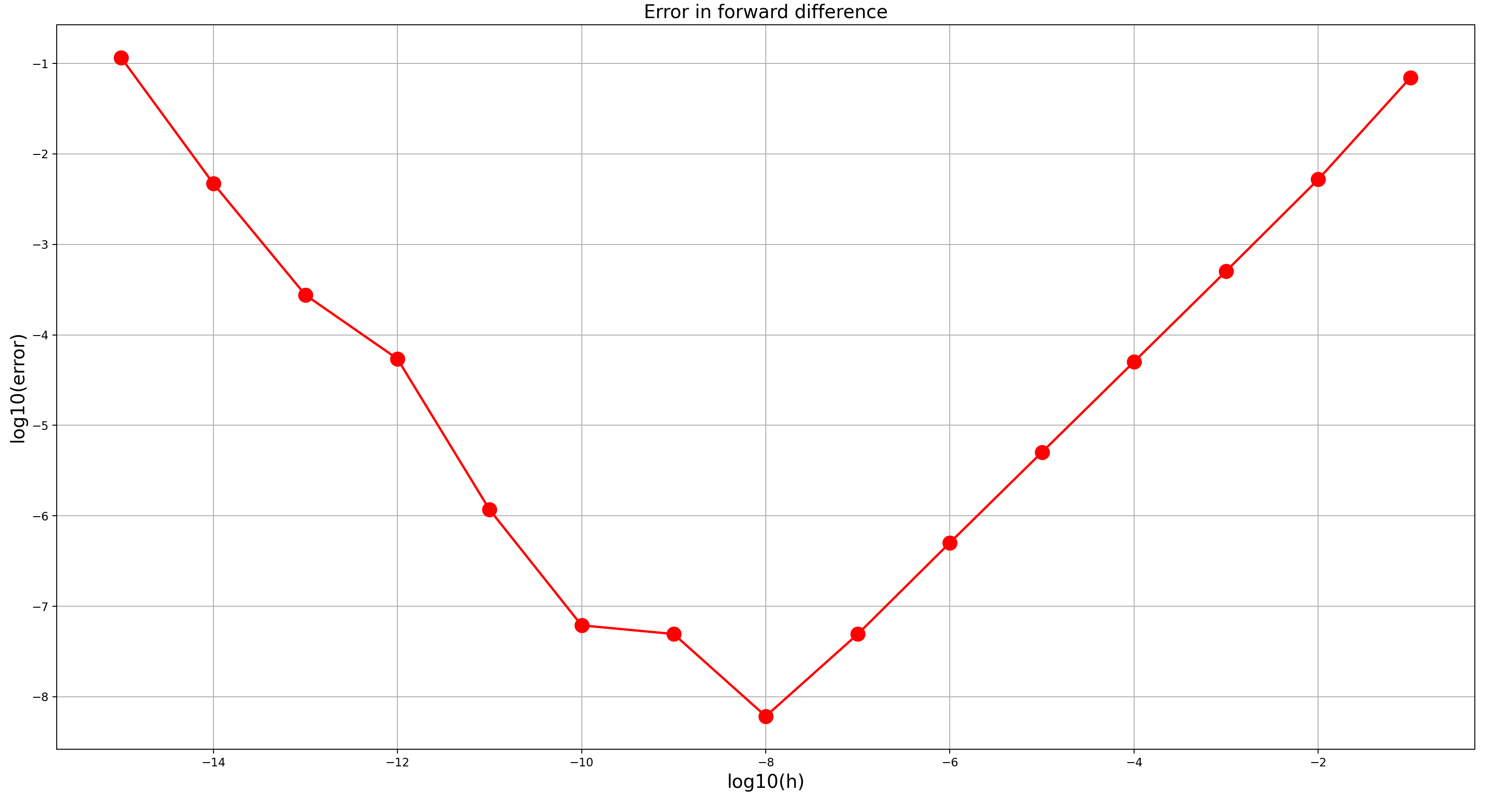
We can clearly see that using composite quadrate we have approximated the function given to us very accurately with very small relative error.

Problem – 5

Using forward difference to find approximation of the derivative of the function given at x = 1, we get the following values using different values of h,



According to the question, here is the log-log plot of this absolute error,



We can clearly see from the above graph that the error in the approximation starts to decrease as the value of h increases but then starts to increase again because of accumulating floating point round off errors cause by rounding of (1+h) in the approximation of the derivative, resulting in the above graph.