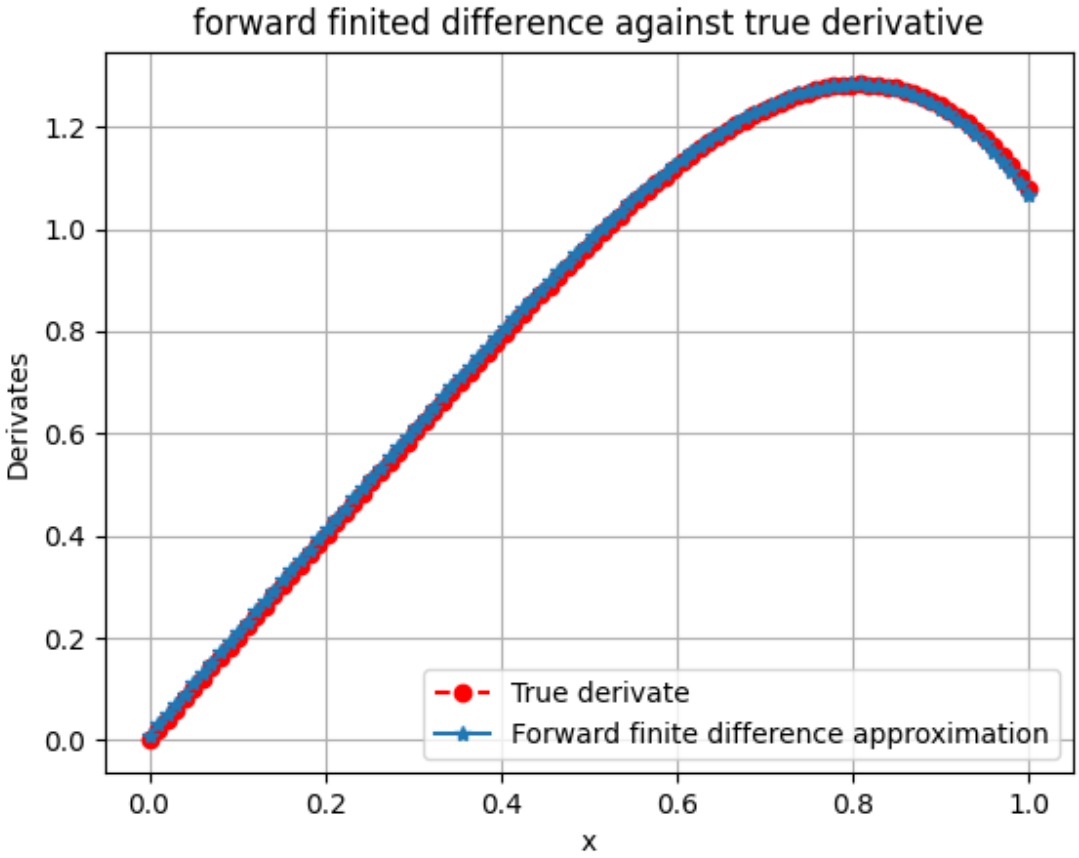
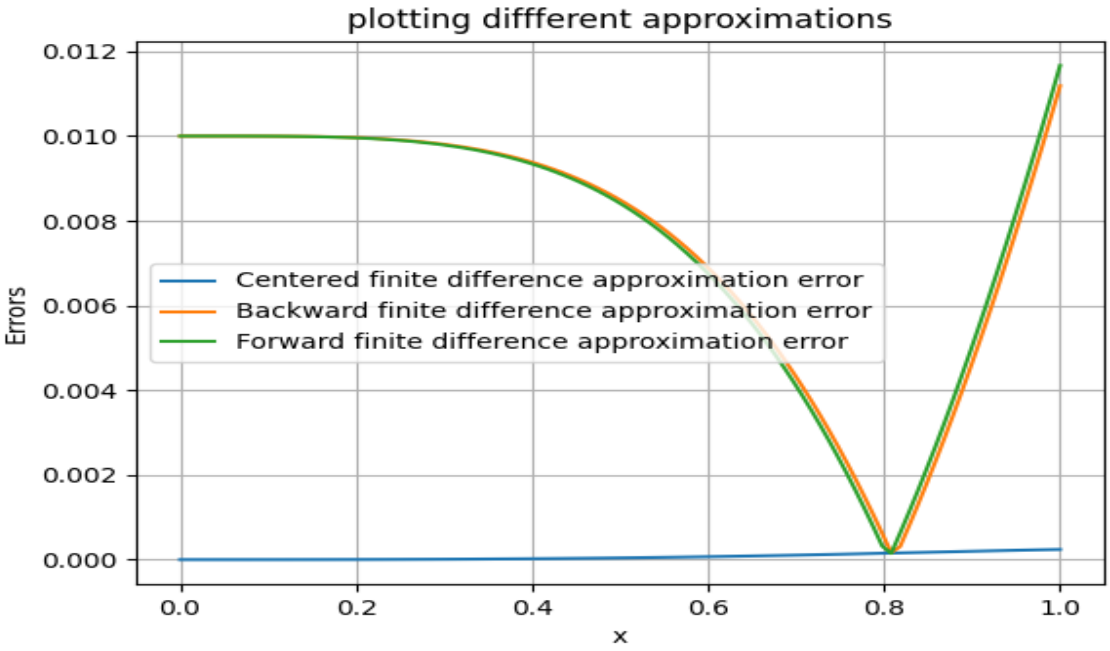
SHUBH PAREEK ,112001039

Code explanations are done in comments of code .
Q1.
Took 100 equal spaced points from the interval 0 to 1 , to plot finite difference against ,true derivative(which is 2x*cos(x^2)).



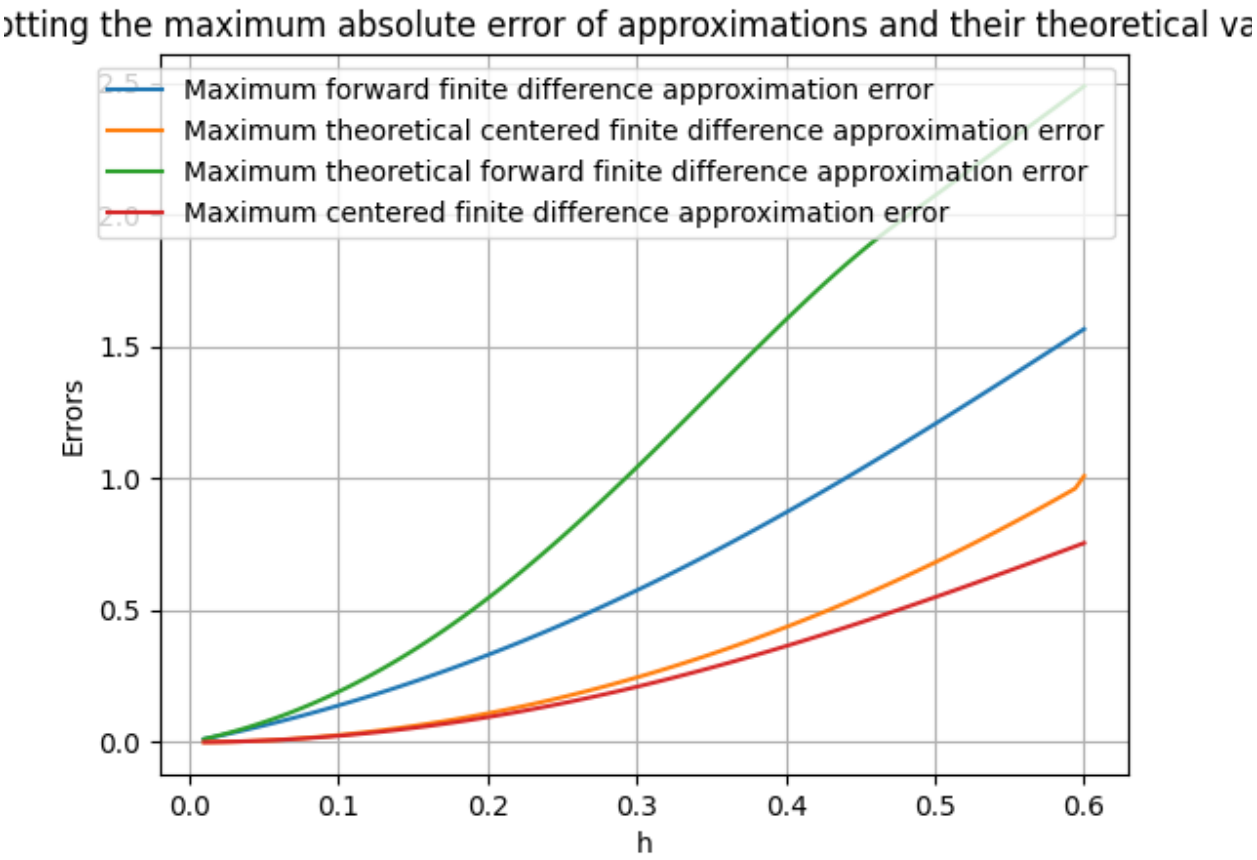
Output : both values came out to be the same.

Q2.Just did as it was asked in question .
Output: we can see that centered finite difference method was better and finally they meet at 0.8



Q3.

Output : as h increases,absolute error values increase and theoretical values represent upper bound above computed errors.

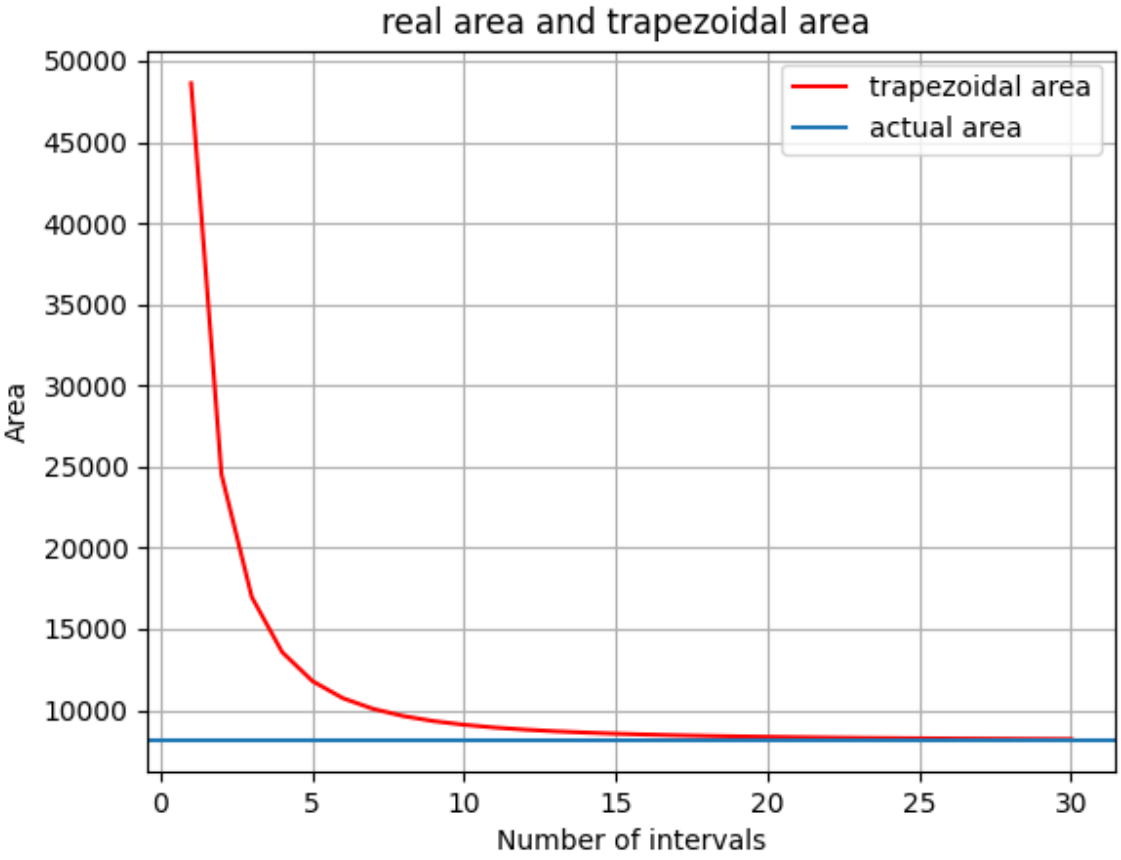


Q4.

We had to compare the real area under the curve with the trapezoidal version . so , I just implemented the trapezoidal formula in an algorithm. And did the plotting.

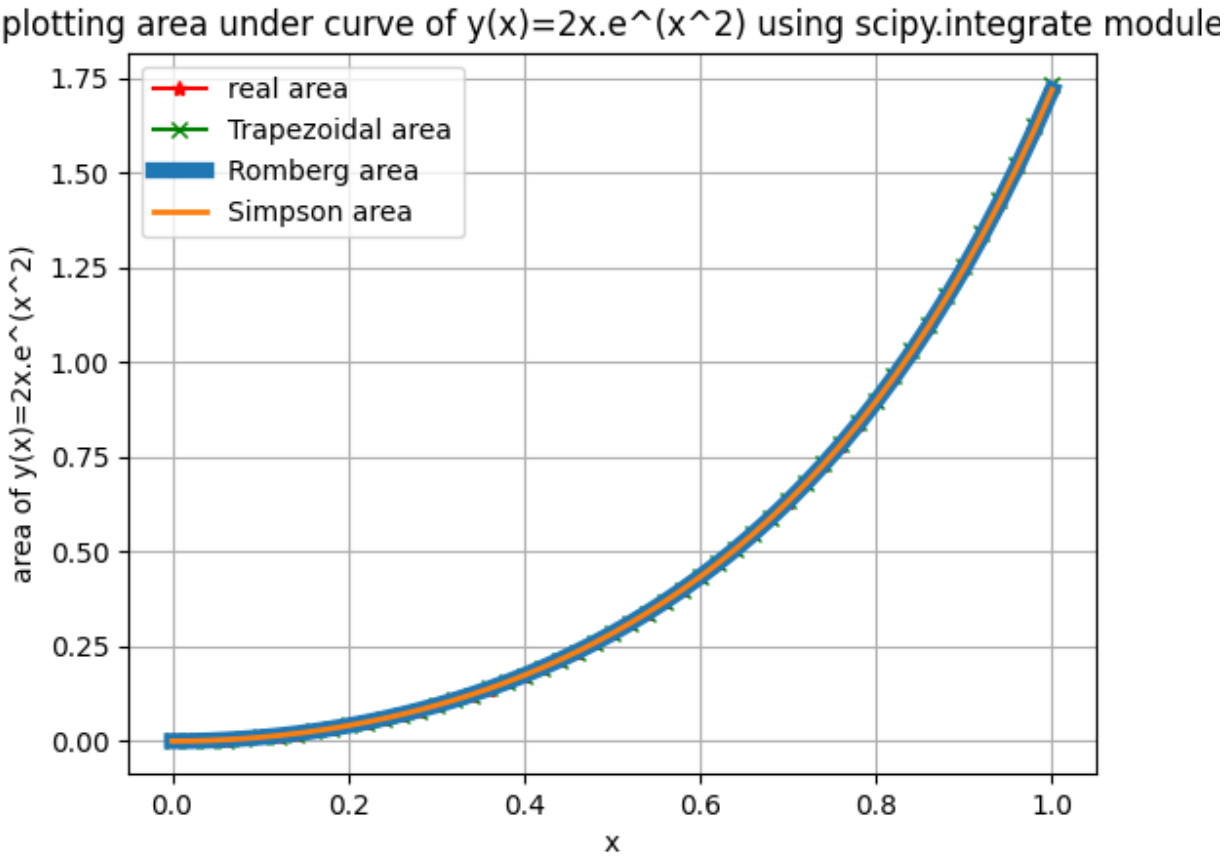
Output :

It can be clearly observed that as the number of intervals are increased both areas come closer.



Q5..
Just used the various methods present in scipy.integrate module like romberg technique ,simpson's method,trapezoidal method to calculate the area under curve between [0,x]

Output :



Q6.

Added the method derivative and area in the class polynomial created in the last assignment , for calculating the derivative and area i didn't use any library since it can be implemented just by multiplying some constants to the coefficients of the polynomial .

Output :

```
if __name__ == "__main__":

    p = Polynomial([1, 2, 3])
    pd = p.derivative()
    print(pd)

    p = Polynomial([1, 2, 3])
    print(p.area(1, 2))
```

```
ations/lab4$ python3 Q6.py
Coefficients of the polynomial are:
2 6
Area in the interval [1, 2] is: 11.0
```

Q7.
I created a polynomial with the taylor series coefficients of the function sin(x)*(e^x) to calculate the approximate area , using the area method created in last question .
And real area i calculate using the expression exp(x) * (sin(x) - cos(x)) / 2 which is the integration result of sin(x)*(e^x).
OUTPUT :

```python
if __name__ == "__main__":
    """
    getting coefficients of taylor series of sin(x)*e^x
    and creating the polynomial from these coefficients.
    """
    # Creating the polynomial
    fx = Polynomial([0, 1, 1, 1 / 3, 0, -1 / 30, -1 / 90, -1 / 630, 0, 1 / 22680, 1 / 113400])
    taylorarea = fx.area(0, 1 / 2)
    #integration of sin(x)*e^x is exp(x) * (sin(x) - cos(x)) / 2
    realarea = np.exp((1/2)) * (np.sin((1/2)) - np.cos((1/2))) / 2 - np.exp((0)) * (np.sin((0)) - np.cos((0))) / 2
    error = abs(realarea - taylorarea)

    # Printing the results
    print("The computed area under the curve is:",taylorarea)
    print("The actual area under the curve is:", realarea)
```

```
shubh@shubh-ROG-Strix-G712LU-G712LU:~/Documents/computational_methods_and_applic
ations/lab4$ python3 Q7.py
The computed area under the curve is: 0.17177502329846078
The actual area under the curve is: 0.17177502331472283
The error is: 1.6262047264348212e-11
```