# CS5016 : Computational Methods and Applications

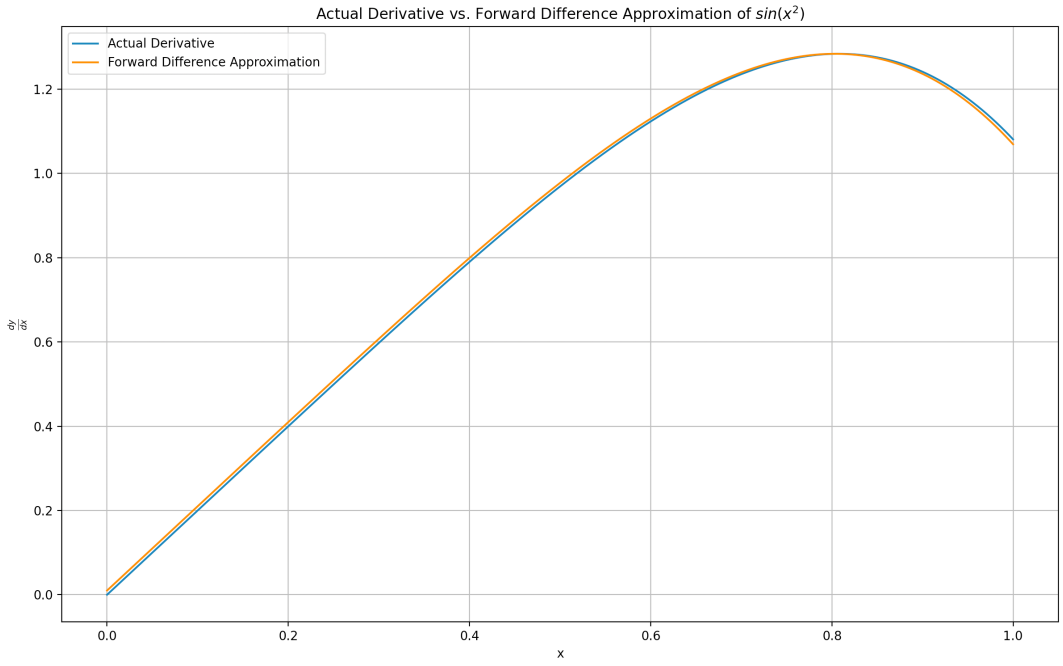
## Assignment 4 : Numerical Differentiation and Integration

112001051
VM Sreeram

---

**Question 1**

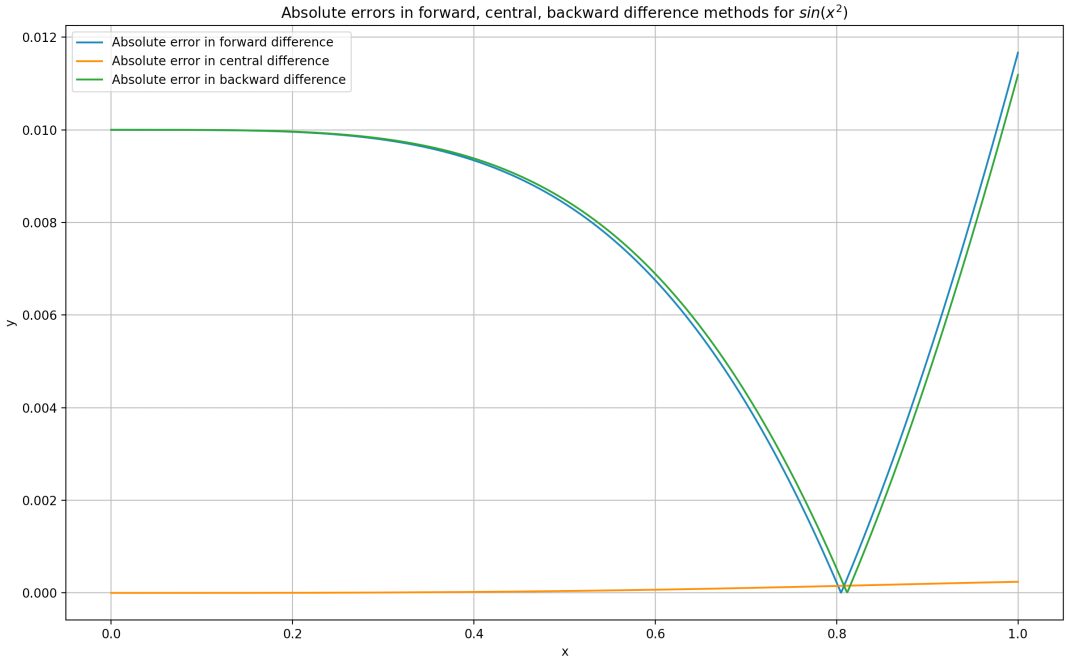
The actual derivative of $\sin(x^2)$ is $2x \cdot \cos(x^2)$. Using `np.linspace`, 1000 points in the interval [0,1] was used to plot in this problem. $\delta^{+}_{h=0.01}(x)$ and $2x \cdot \cos(x^2)$ was potted for all these 1000 points.




The output is pasted above for convenience.

**Question 2**

The actual derivative of $\sin(x^2)$ is $2x \cdot \cos(x^2)$. Using `np.linspace`, 1000 points in the interval [0,1] was used to plot in this problem. $\delta^{+}, \delta^{-}, \delta^{c}$ were computed for $h = 0.01$ for each of the 1000 points. This was used to compute the absolute error, which is $|\delta - f'(x)|$ for each $\delta$.



The output is pasted above for convenience.

## Question 3

The maximum absolute error over each $x$ in [0,1] was computed for each h and was plotted wrt h.

The maximum absolute error in forward and central finite difference is computed using $\delta^+$, $\delta^c$, and f" over all $x$ in [0,1].

The maximum theoretical absolute error in forward finite difference is

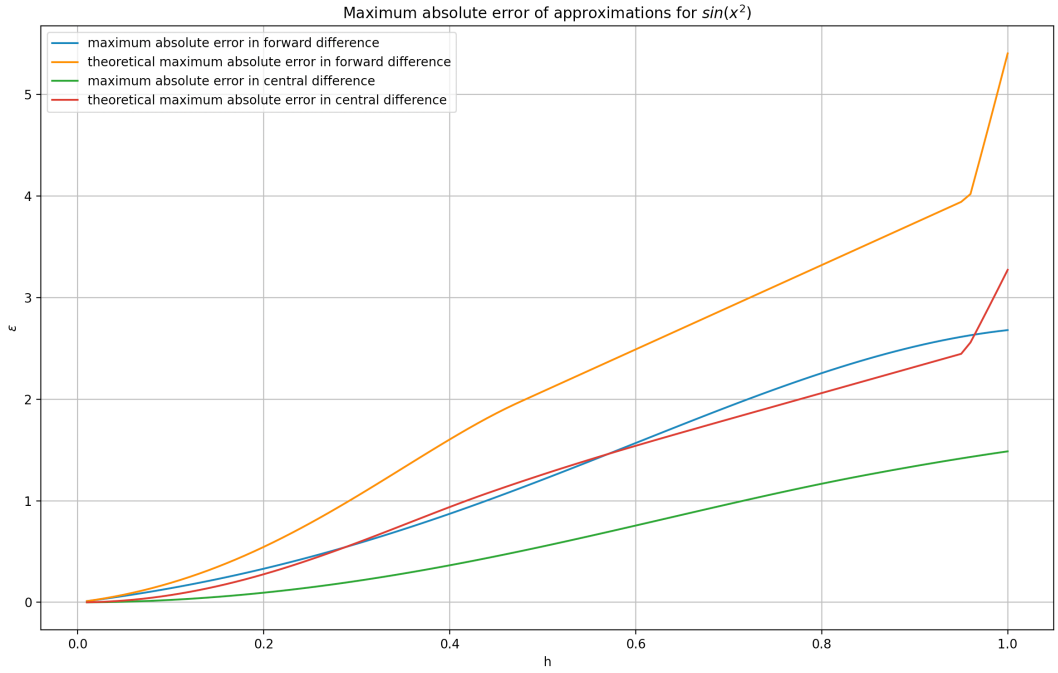
$$\max_{\zeta \in [x, x+h]} \frac{h}{2} \cdot |f''(\zeta)|$$

The maximum theoretical absolute error in central finite difference is

$$\max_{\zeta_1 \in [x, x+h], \zeta_2 \in [x-h, x]} \frac{h}{4} \cdot |f''(\zeta_1) - f''(\zeta_2)|$$

Both of these were computed using maximum error in f" and minimum error in f" over all possible $\zeta$ over and the h (constant for each vertical line in plot).

The max of these quantities are asked over [0,1], which is computed and plotted as theoretical errors against h.



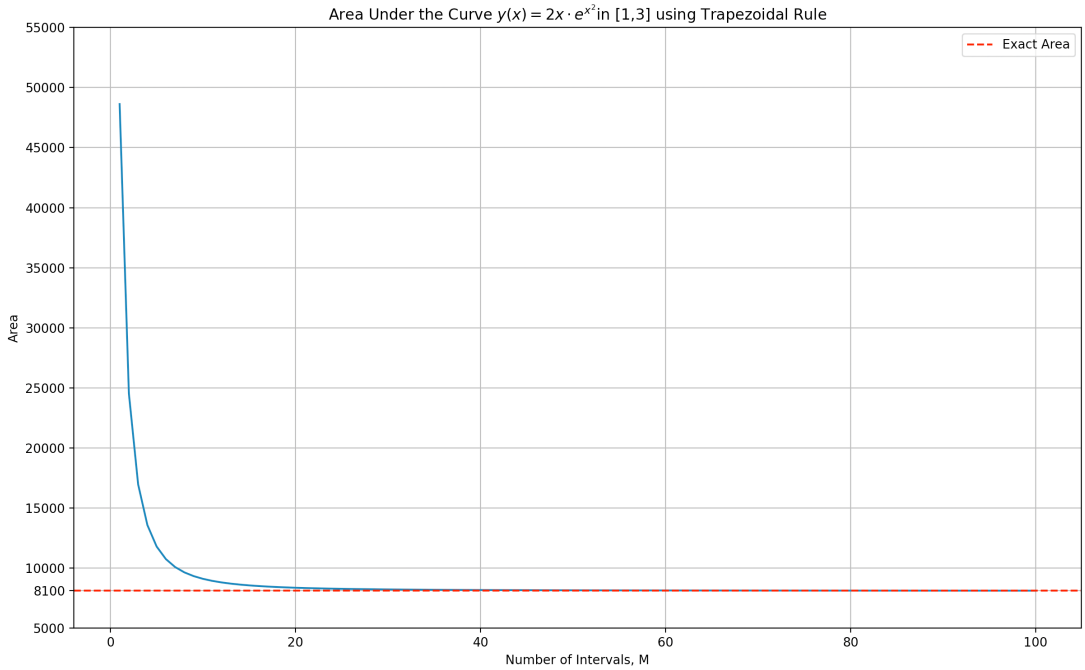

The output is pasted above for convenience. It takes about 7s to produce output for this problem.

## Question 4

The area according to trapezoidal approximation was computed for each M (number of intervals) from 1 to 100. The approximation is used to compute area under $2x \cdot e^{x^2}$ between x=1 to 3. A function `trapezoidal_rule(f, a, b, M)` was written that returns the trapezoidal approximated area (with M intervals) between a and b of the curve of function f.

The exact area was computed manually and was found out to be $e^9 - e^1$. This is plotted as dashed line.

The output is pasted below for convenience.



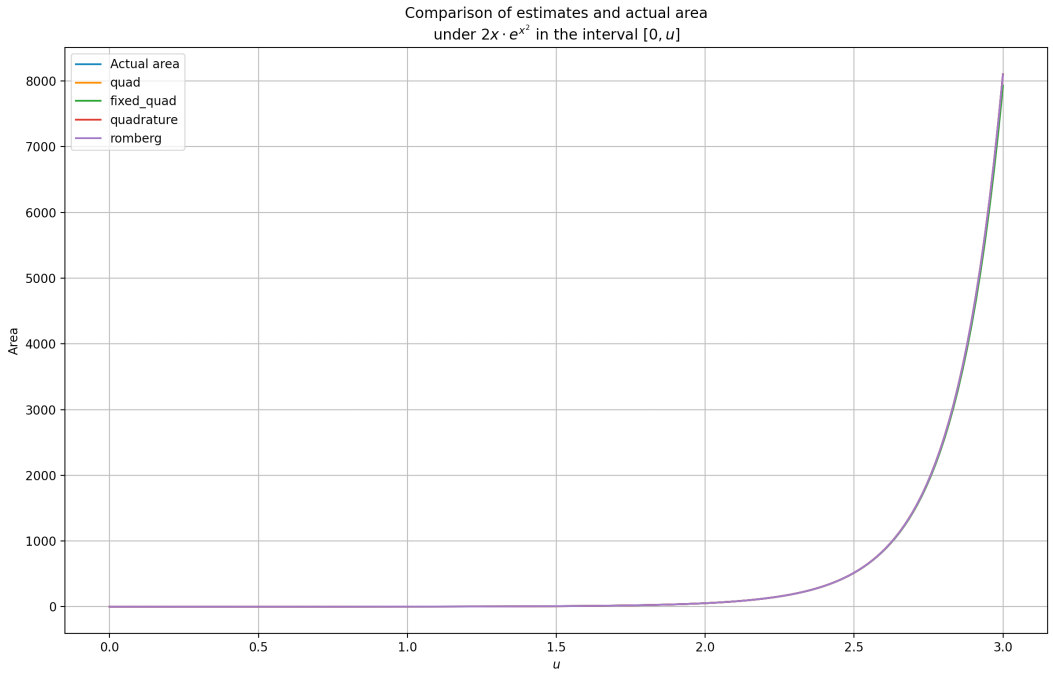

It was observed that the approximation converges to actual value very fast wrt x, and trapezoidal approximation is a very good approximation.

**Question 5**

quad, fixed_quad, quadrature, and romberg modules from `scipy.integrate` were used to compute the area under the curve $2x \cdot e^{x^2}$ between x=0 to u. The actual area between x=0 to u is $e^{u^2} - 1$. All of these were plotted wrt u. I took u to be from 0 to 3.

All the curves appear pretty close by, and image needs to be zoomed to see them separately. In particular, for this case, fixed_quad seems to be the worst function based on accuracy at u=3.

The output is pasted below for convenience.



**Question 6**

The derivative was computed based on the formula

$$\frac{d}{dx}k \cdot x^n = k \cdot n \cdot x^{n-1}$$

The area/integral was computed using the formula

$$\int k \cdot x^n = \frac{k}{n+1} \cdot x^{n+1} + C$$

Loop was used to traverse the polynomial and compute derivative/integral of it. The constant of integration gets subtracted out as area uses definite integral.

A sample output is given below for convenience.

```
Coefficients of the polynomial are:
0 0 3
Area in the interval [2, 3] is: 16.25
```

**Question 7**

Using `np.linspace`, 10 xs in the interval [0,1] was used to get $(x, e^x \cdot \sin x)$ points. A list was made of these points and was used to `fitViaMatrixMethod()` already implemented in `Polynomial` class. Then using `area(a,b)` method, we compute area between 0 and 0.5. pythonArea was computed by performing actual integration and plugging in the limits.

The output is given below for convenience

```
Area in the interval [0, 0.5] is: 0.17177502332324648
pythonArea = 0.17177502331472283
error = 8.523654004832792e-12
```