

1. Submit your **report** as a **Microsoft Word** file on Canvas before the deadline.
2. **Annotate** all your answers in your submission using the following **format**:

Answer to Question X

3. Include your **Python code at the end** of your **report**.
4. Name your file in following format: **Lab2_SID_LastName_FirstName**
5. Type **your name** and **student ID** on the first page of your report.

10 marks

Q1) In this question, you will use Python to approximate the function $f(x) = e^{2x}$ using polynomials.

1. In Python, generate an array of 100 equally-spaced numbers between 0 and 1 and name it X. Plot $f(x)$ for every point in X. (see the hint at the end of the question)
2. Compute the forth order Taylor's approximation of $f(x)$ around $x_i = 0$. (Calculate on the paper)
3. For every point in X, plot $f(x)$ and its 0th order, 1st order, 2nd order, 3rd order and 4th order Taylor's approximations around $x_i = 0$. Include the plot in your report. (4 marks)
4. What will happen as the order of approximation increases? (1 mark)
5. For every point in X, plot the percentage of **the absolute value** of the true relative error of the 0th order, 1st order, 2nd order, 3rd order and 4th order Taylor's approximations of $f(x)$ around $x_i = 0$. Include the plot in your report. (4 marks)
6. What will happen to the error as x increases? (1 mark)

Hint:

1-you can compute and plot e^{2x} using the following commands:

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0.01, 0.5, 50)
y=np.exp(2*x) # multiplies each element of x by 2 and computes the exponential of the
result.
plt.plot(x, y, label = " f(x)") #plots y as a function of x and labels it as f(x)
plt.legend()#plots the label
plt.figure() #creates a new figure
```

2- you can compute $\frac{x^2}{2!}$ using the following command:

```
a) y2= np.power(x, 2)/ np.math.factorial(2)#raises each element of x to the power of 2 and
divides the result by 2!
```

10 marks

Q2: In this question, you will use Python to numerically differentiate the function $f(x) = e^x$.

- 1- Create an array of 100 equally spaced numbers between 0.001 and 1 and name it H . Each element in H will be a step-size for the numerical differentiation. See the hint at the end.
- 2- Compute the forward difference approximation of $f(x)$ at $x_i = 0$ for every step-size in H . Compute the percentage of **the absolute value of** relative error of forward difference approximation for every step-size in H .
Plot the percentage of relative error versus H . Include the plot in your report.
(3 marks)
- 3- Compute the backward difference approximation of $f(x)$ at $x_i = 0$ for every step-size in H .
Compute the percentage of **the absolute value of** relative error of backward difference approximation for every step-size in H .
Plot the percentage of relative error versus H . Include the plot in your report.
(3 marks)
- 4- Compute the centred difference approximation of $f(x)$ at $x_i = 0$ for every step-size in H . Compute the percentage of **the absolute value of** relative error of centred difference approximation for every step-size in H .
Plot the percentage of relative error versus H . Include the plot in your report.
(3 marks)
- 5- What will happen as the step-size decreases? (0.5 marks)
- 6- Which approximation is more accurate? (0.5 marks)

Hint:

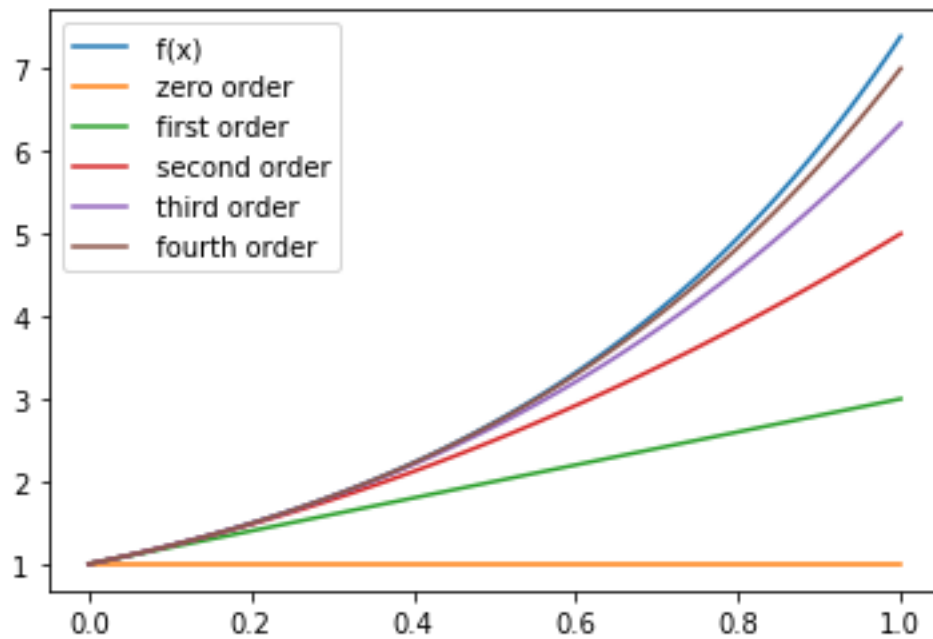
```
H=np.linspace(0.001, 1, 100);#creates an array of equally spaced numbers between 0.001 and 1
FM=np.zeros(100);#creates an array of 100 zeros.
BM=np.zeros(100);#creates an array of 100 zeros.
CM=np.zeros(100);#creates an array of 100 zeros.
```

```
for i in range(0,100):
```

```
    FM[i]= #write your code for computing the forward approximation for step-size H[i]
    BM[i]= #write your code for computing the backward approximation for step-size H[i]
    CM[i]= #write your code for computing the centred approximation for step-size H[i]
```

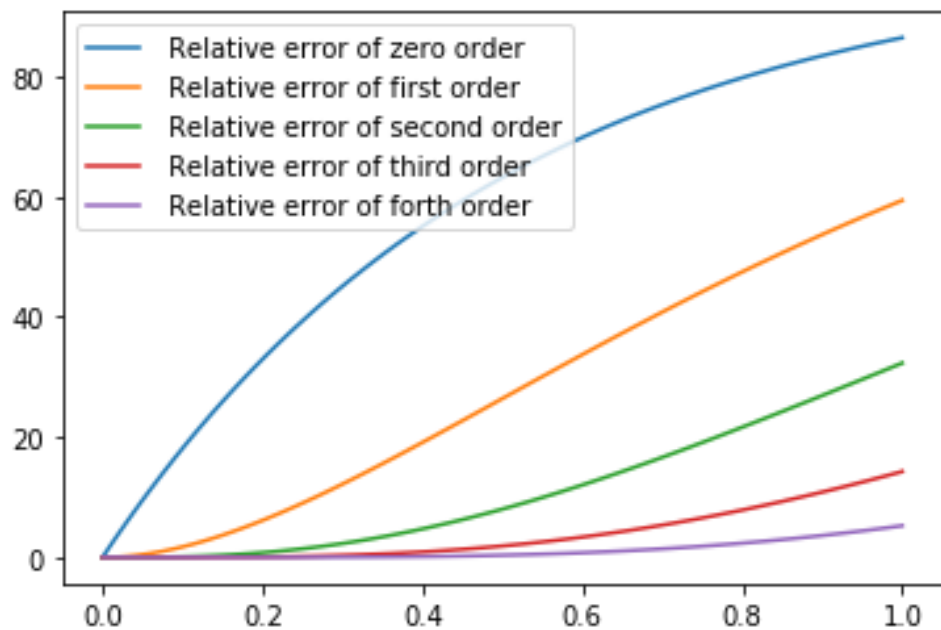
Q1)

Part 3.



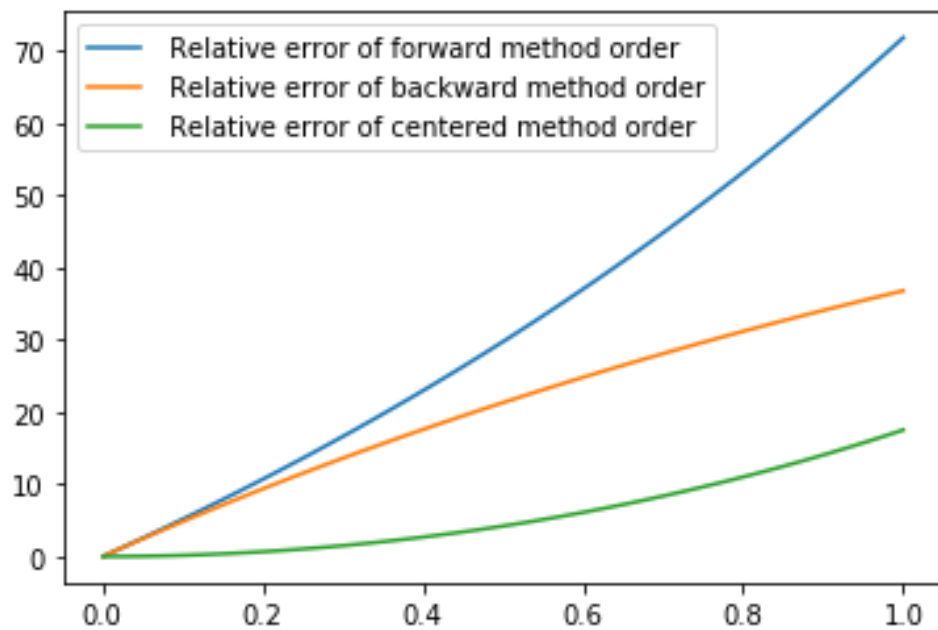
Part 4: The approximation becomes more accurate as the order increases.

Part 5



Part 4: The approximation becomes less accurate as x increases. Equivalently, the error of approximation increases as x becomes large.

Q2)



Part 5: The approximation becomes more accurate as the step-size decreases (or the error decreases with the step-size)

Part 6: Centred approximation