CECC421: Numerical Analysis
Lecture 3: Taylor Series-02

Eng. Aya Kherbek
Eng. Baher Kherbek
Faculty of Engineering
Department of Informatics
Manara University

Purpose of the Session:

- Plot the Taylor series for the exponential function, sine function, and cosine function using the **pyplot.matplotlib** library in Python for different numbers of series terms.

- To install it use this command on your computer's terminal :
'pip install matplotlib'

Exercise 1:
Using pyplot.matplotlib in Python:

1.To plot the exponential function e^x represented by its Taylor series, implemented using a for loop for the first order (n=0), second order (n=1), third order (n=2), and fourth order (n=3).

2.To plot the actual exponential function e^x , implemented using the actual exp function from the math library.

Remember that the Taylor series for the exponential function is given by the following form:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + .. = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

(n=2)

$$e^x = 1 + x + \frac{x^2}{2!}$$

(n=1)

$$e^x = 1 + x$$

(n=0)

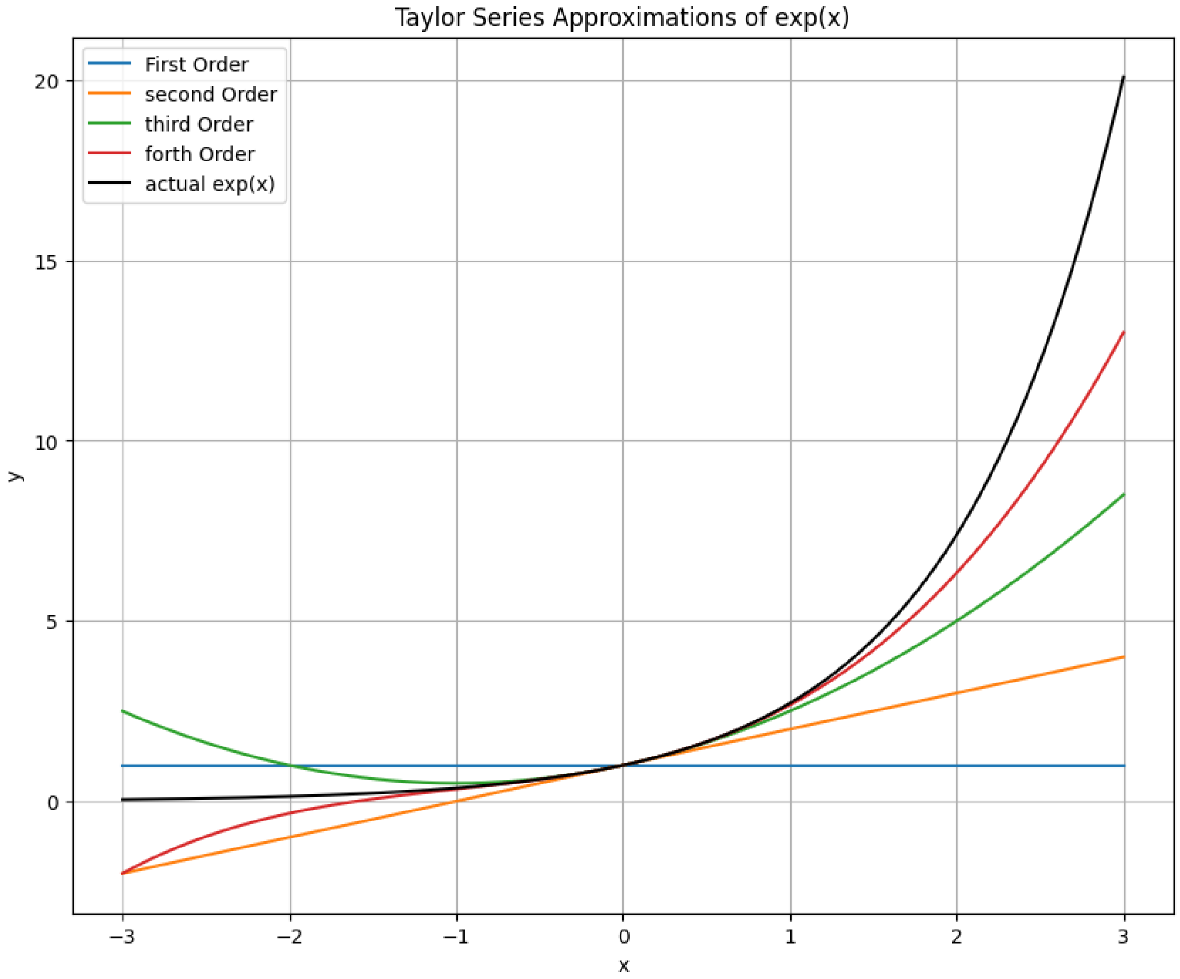
$$e^x = 1$$

```python
import numpy as np
import matplotlib.pyplot as plt
import math

num= np.linspace(-3, +3, 200)
e_to_num = np.zeros(len(num))
labels = ['First Order', 'second Order', 'third Order',
 'forth Order']
plt.figure(figsize = (10,8))

for i in range(4):
    label = labels[i]
    e_to_num = e_to_num + (num**i) / math.factorial(i)
    plt.plot(num, e_to_num, label=label)
```

```python
plt.plot(num, np.exp(num), 'k', lab
'actual exp(x)')
plt.grid()
plt.title('Taylor Series
Approximations of exp(x)')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```


Taylor Series Approximations of exp(x)
First Order
second Order
third Order
forth Order
actual exp(x)
20
15
10
5
0
y
−3
−2
−1
0
1
2
3
x

Exercise 2:

Use matplotlib.pyplot in Python:

1.To plot the exponential function e^x represented by the Taylor series, which is implemented using a function for the first-order approximation (n=0), second-order (n=1), third-order (n=2), and fourth-order (n=3).

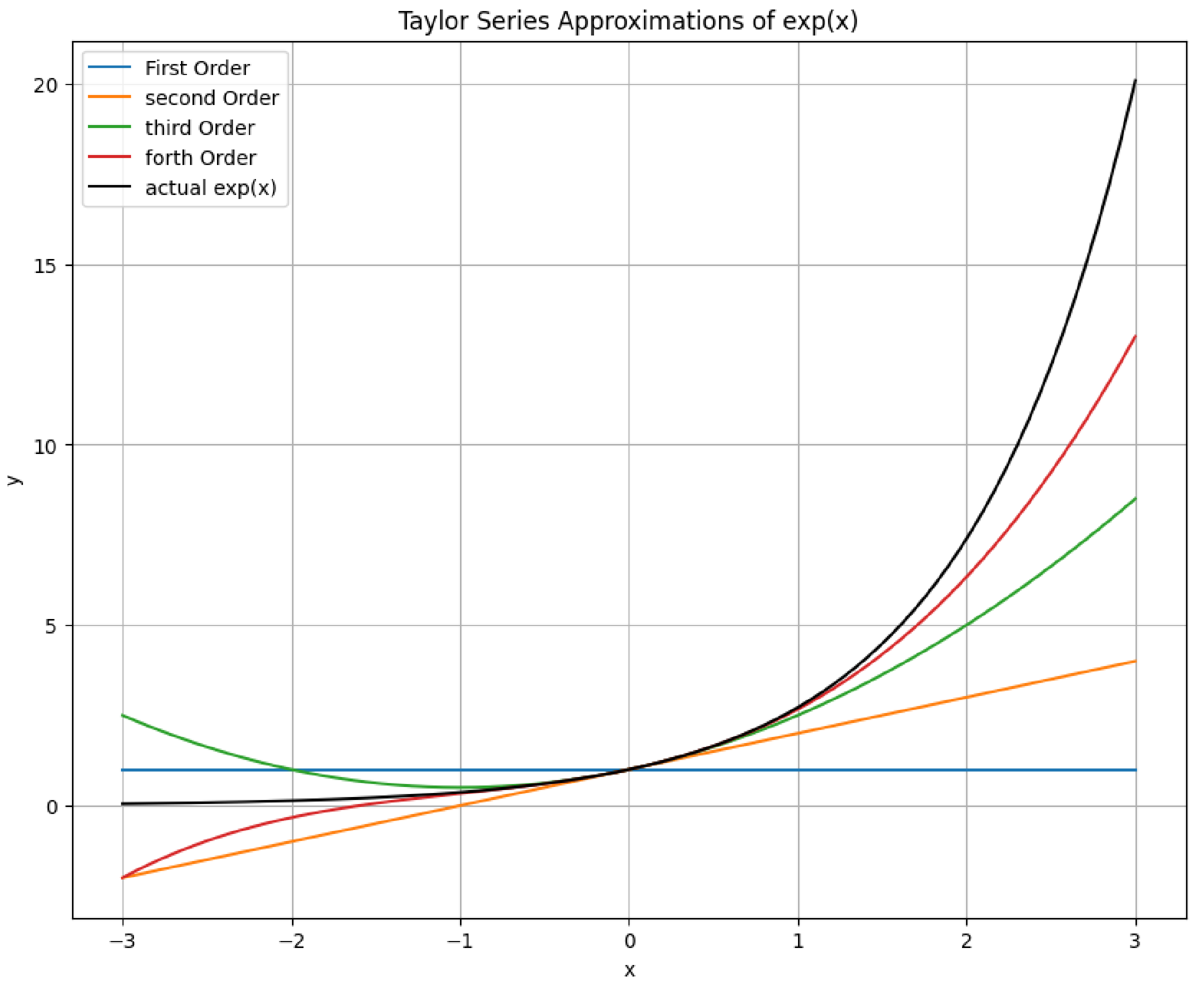
2.To plot the actual exponential function e^x, which is implemented using the actual exp function from the math library.

```python
import numpy as np
import matplotlib.pyplot as plt
import math

def Tylor_Series_Function_with_return(num,range_limit):
    exp_num=np.zeros(len(num))
    for j in range(range_limit):
      exp_num = exp_num + num**j/math.factorial(j)
    return (exp_num)
num= np.linspace(-3, +3, 200)
labels = ['First Order', 'second Order', 'third Order', 'forth Order']
plt.figure(figsize = (10,8))

for i in range(4):
    label = labels[i]
    exp_num_1 = Tylor_Series_Function_with_return(num,i)
    plt.plot(num, exp_num_1, label=label)
#Matplotlib will color each line  by default
```

```python
plt.plot(num, np.exp(num), 'k', label =
'actual exp(x)')
plt.grid()
plt.title('Taylor Series
Approximations of exp(x)')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```


Taylor Series Approximations of exp(x)
First Order
second Order
third Order
forth Order
actual exp(x)
20
15
10
5
0
y
−3
−2
−1
0
1
2
3
x

Exercise 3:

Use matplotlib.pyplot in Python:

1.To plot the trigonometric function $\sin x$ represented by the Taylor series, which is implemented using a for loop for the first-order approximation (n=0), third-order (n=1), fifth-order (n=2), and seventh-order (n=3).

2.To plot the actual trigonometric function $\sin x$ which is implemented using the actual sin function from the numpy

The Taylor series expansion for the $\sin x$ function is given by:

$$\sin(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{(x)^{2n+1}}{(2n+1)!}$$
$$\approx 1 - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \cdots.$$

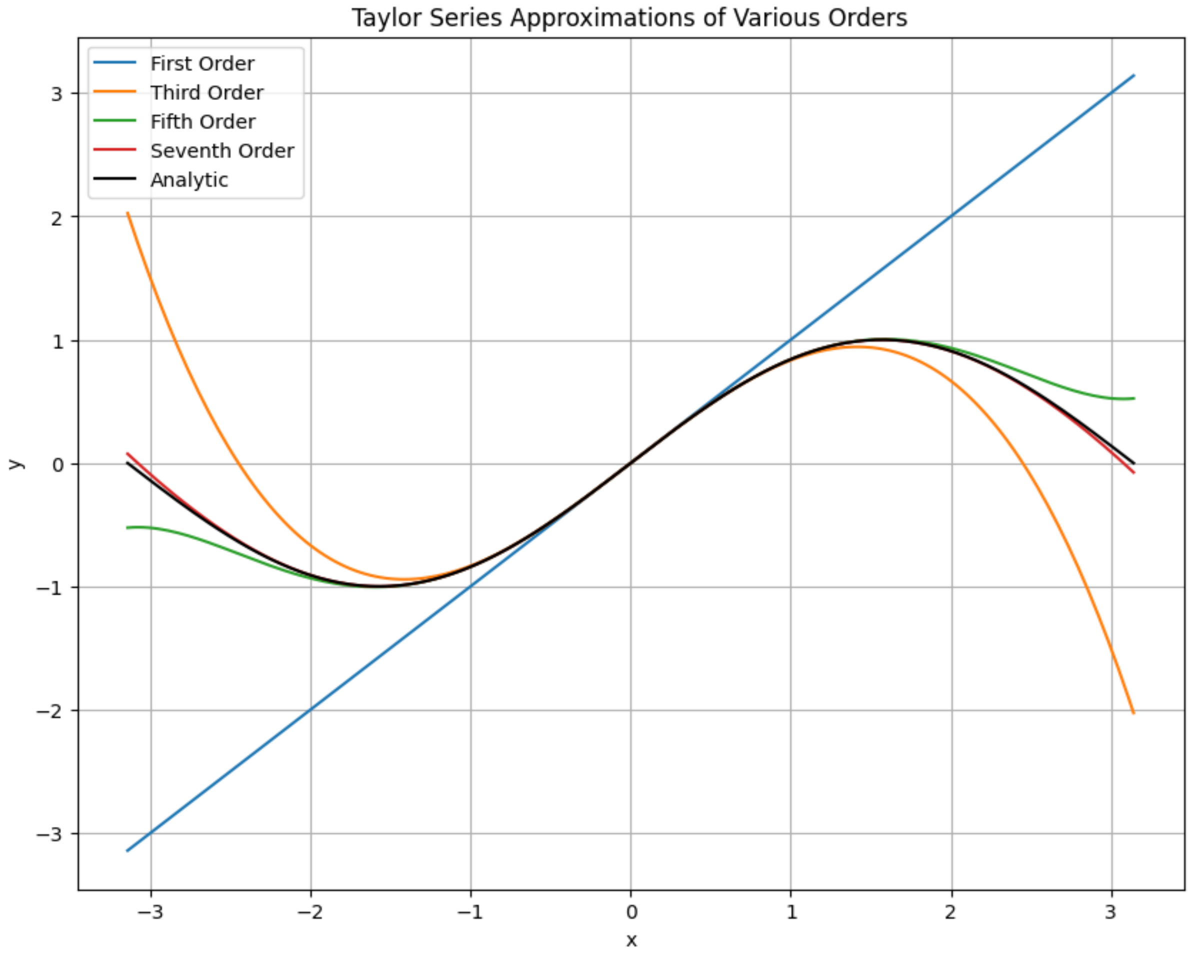
This formula consists of a sequence of odd-power terms of x, divided by the factorial of the exponent, alternating in sign. It provides an increasingly accurate approximation of $\sin x$ as more terms are included.

```python
import numpy as np
import matplotlib.pyplot as plt
import math

x = np.linspace(-np.pi, np.pi, 200)
y = np.zeros(len(x))

labels = ['First Order', 'Third Order', 'Fifth Order', 
'Seventh Order']
plt.figure(figsize = (10,8))
for n in range(4):
    label = labels[n]
    y = y + ((-1)**n * (x)**(2*n+1)) /math.factorial(2*n+1)
    plt.plot(x,y, label = label)
plt.plot(x, np.sin(x), 'k', label = 'Analytic')
```

```python
plt.grid()
plt.title('Taylor Series
Approximations of Various
Orders')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```


Taylor Series Approximations of Various Orders
First Order
Third Order
Fifth Order
Seventh Order
Analytic
y
x
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Exercise 4:

Use matplotlib.pyplot in Python:

1.To plot the trigonometric function $\cos x$ represented by the Taylor series, which is implemented using a for loop for the first-order approximation (n=0), second-order (n=1), fourth-order (n=2), and sixth-order (n=3).

2.To plot the actual trigonometric function $\cos x$, which is implemented using the actual cos function from the numpy library.

Remember that the Taylor series for the $cos\,x$ function is given as follows:

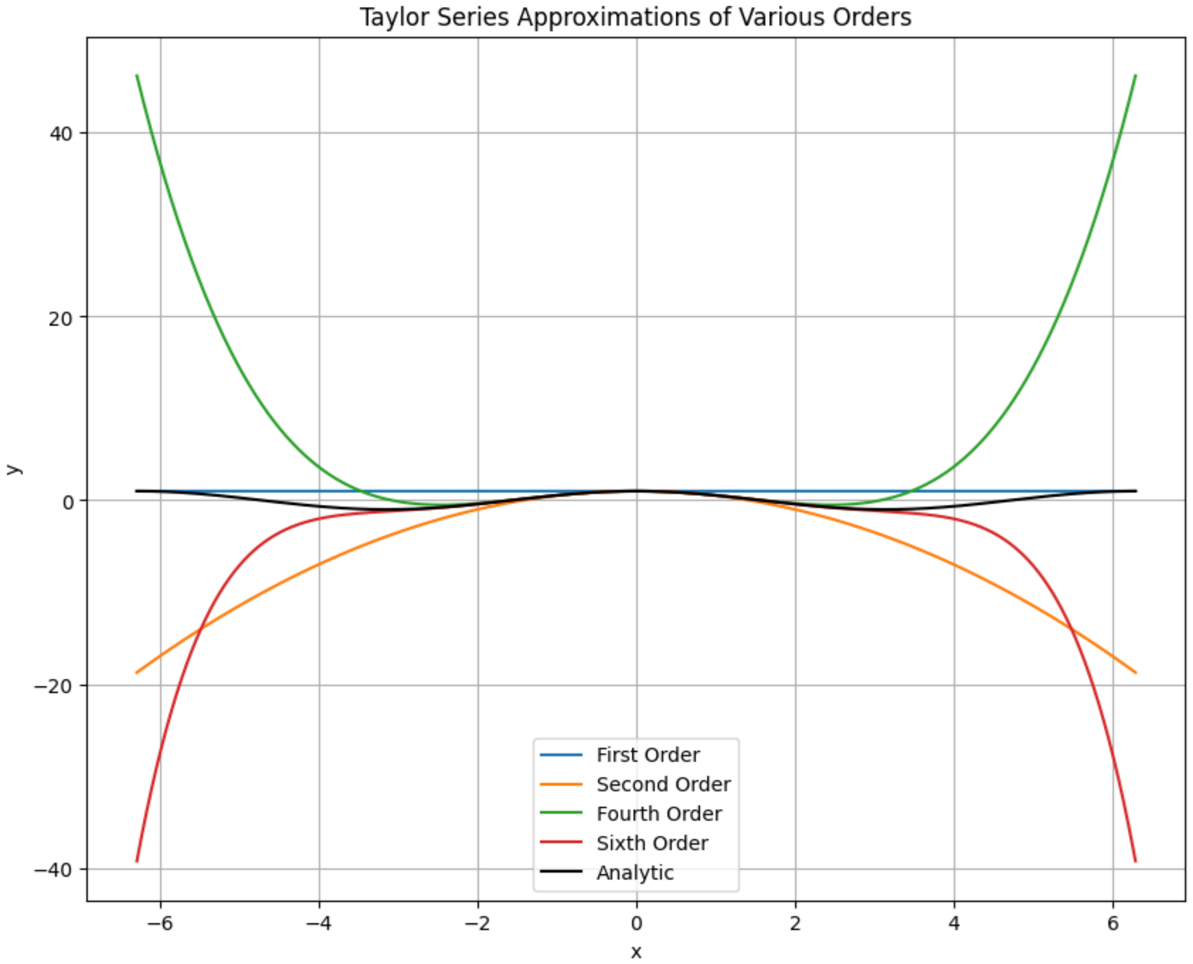
$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{(x)^{2n}}{(2n)!}$$

$$\approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \cdots.$$

```python
x = np.linspace(-2*np.pi, 2*np.pi, 20000)
y = np.zeros(len(x))

labels = ['First Order', 'Second Order', 'Fourth Order',
'Sixth Order']
plt.figure(figsize = (10,8))
for n  in range(4):
    label = labels[n]
    y = y + ((-1)**n * (x)**(2*n)) / np.math.factorial(2*n)
    plt.plot(x,y, label = label)
plt.plot(x, np.cos(x), 'k', label = 'Analytic')
```

```python
plt.grid()
plt.title('Taylor Series
Approximations of Various
Orders')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```


Taylor Series Approximations of Various Orders
40
20
0
−20
−40
y
−6
−4
−2
0
2
4
6
x
First Order
Second Order
Fourth Order
Sixth Order
Analytic

Thanks for Listening