Name: Phan Tran Thanh Huy

ID: ITCSIU22056

Lab 1

Part 1:

Code:

```python
import math

def solve(x, terms):
    result = 0
    for i in range(terms):
        temp = (x**i)/math.factorial(i)
        print(f'Terms {i}: {temp}')
        result += temp
    return result

print(f'The first 4 terms of the Taylor Series: ')
result = solve(1,4)
print('part 1')
print(f'Taylor approximate = {result}')
print(f'Taylor real = {math.exp(result)}')
```

Result:

```
PS E:\Homework\TMC\Lab1> & C:/Python312/python.exe e:/Homework/TMC/Lab1/ex1.py
The first 4 terms of the Taylor Series:
Terms 0: 1.0
Terms 1: 1.0
Terms 2: 0.5
Terms 3: 0.16666666666666666
part 1
Taylor approximate = 2.6666666666666665
Taylor real = 14.391916095149892
```

Part 2:

Code:

```python
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

x = sp.symbols('x')

def taylor_sin(x, terms):
    taylor_series = 0
    for i in range(terms):
        term = ((-1)**i * x**(2*i+1)) / sp.factorial(2*i+1)
        taylor_series += term
    return taylor_series

approximation = taylor_sin(x, 4)
print(approximation)

x_vals = np.linspace(-2*np.pi, 2*np.pi, 400)
y_actual = np.sin(x_vals)

y_approx_1 = [taylor_sin(x, 1) for x in x_vals]
y_approx_3 = [taylor_sin(x, 3) for x in x_vals]
y_approx_5 = [taylor_sin(x, 5) for x in x_vals]

plt.figure(figsize=(10, 5))
plt.plot(x_vals, y_actual, label='sin(x)', color='black')
plt.plot(x_vals, y_approx_1, label='1-term Taylor', linestyle='dashed')
plt.plot(x_vals, y_approx_3, label='3-term Taylor', linestyle='dotted')
plt.plot(x_vals, y_approx_5, label='5-term Taylor', linestyle='dashdot')

plt.legend()
plt.xlabel('x')
plt.ylabel('sin(x) and Approximations')
plt.title('Taylor Series Approximation of sin(x)')
plt.grid()
plt.show()
```

Result:

```
PS E:\Homework\TMC\Lab1> & C:/Python312/python.exe e:/Homework/TMC/Lab1/ex2.py
0.841468253968254
```

Plot the actual sin(x) against its Taylor approximations with 1, 3, and 5 terms:

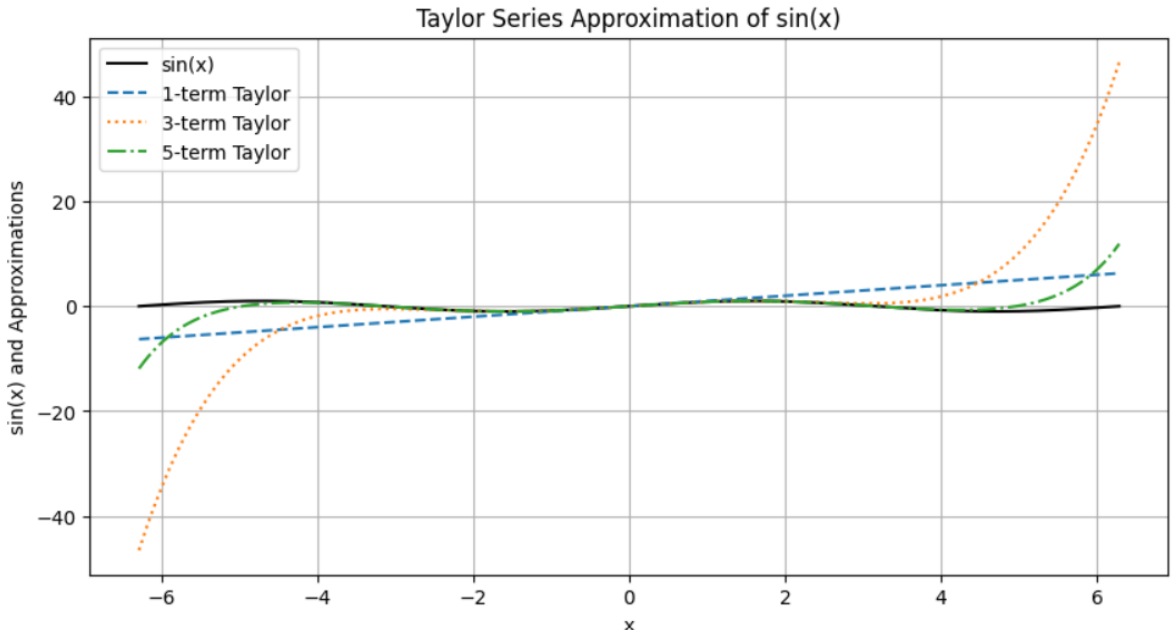
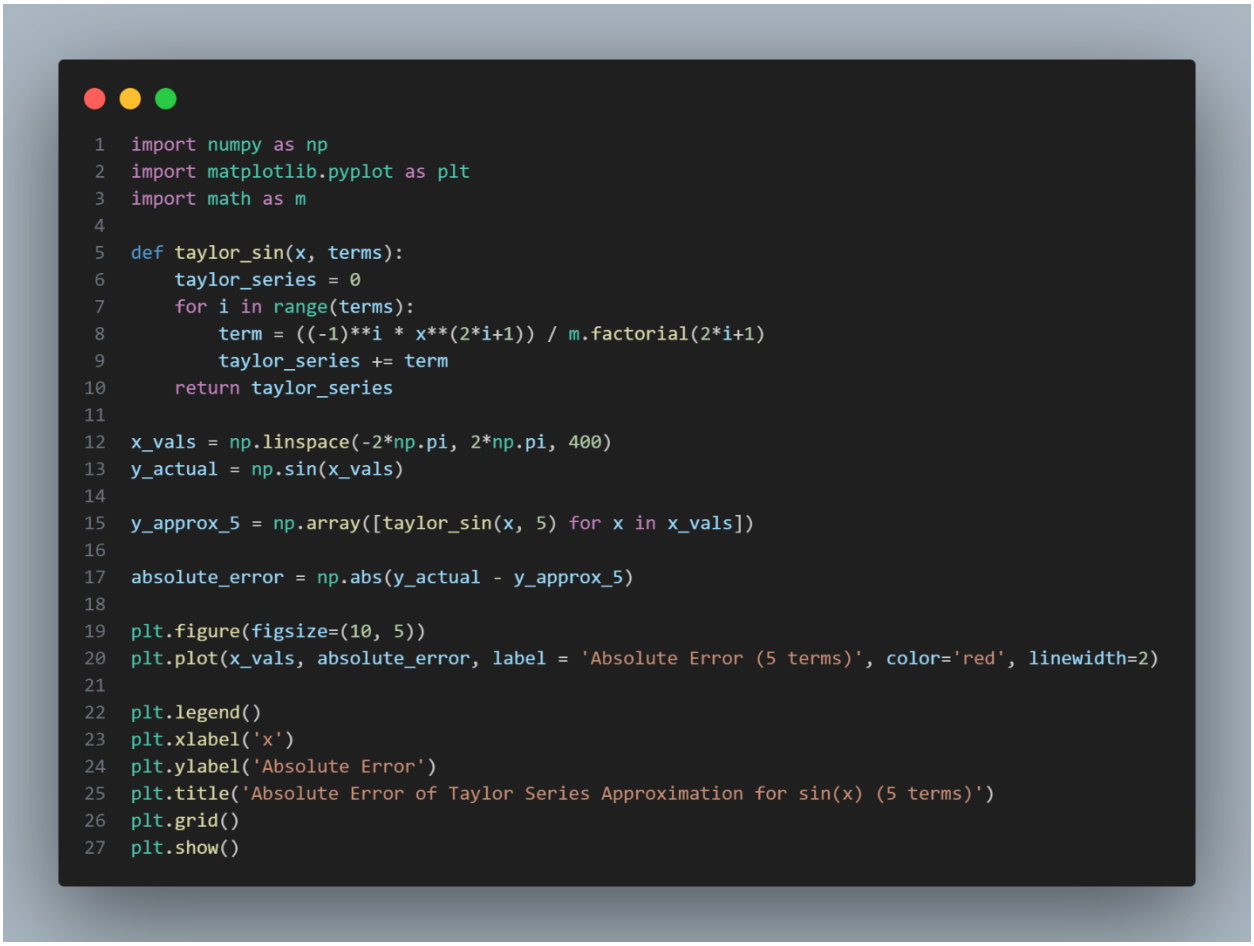


Figure 1. sin_taylor.png

Part 3:

Code:



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

def taylor_sin(x, terms):
    taylor_series = 0
    for i in range(terms):
        term = ((-1)**i * x**(2*i+1)) / m.factorial(2*i+1)
        taylor_series += term
    return taylor_series

x_vals = np.linspace(-2*np.pi, 2*np.pi, 400)
y_actual = np.sin(x_vals)

y_approx_5 = np.array([taylor_sin(x, 5) for x in x_vals])

absolute_error = np.abs(y_actual - y_approx_5)

plt.figure(figsize=(10, 5))
plt.plot(x_vals, absolute_error, label = 'Absolute Error (5 terms)', color='red', linewidth=2)

plt.legend()
plt.xlabel('x')
plt.ylabel('Absolute Error')
plt.title('Absolute Error of Taylor Series Approximation for sin(x) (5 terms)')
plt.grid()
plt.show()
```

Result:

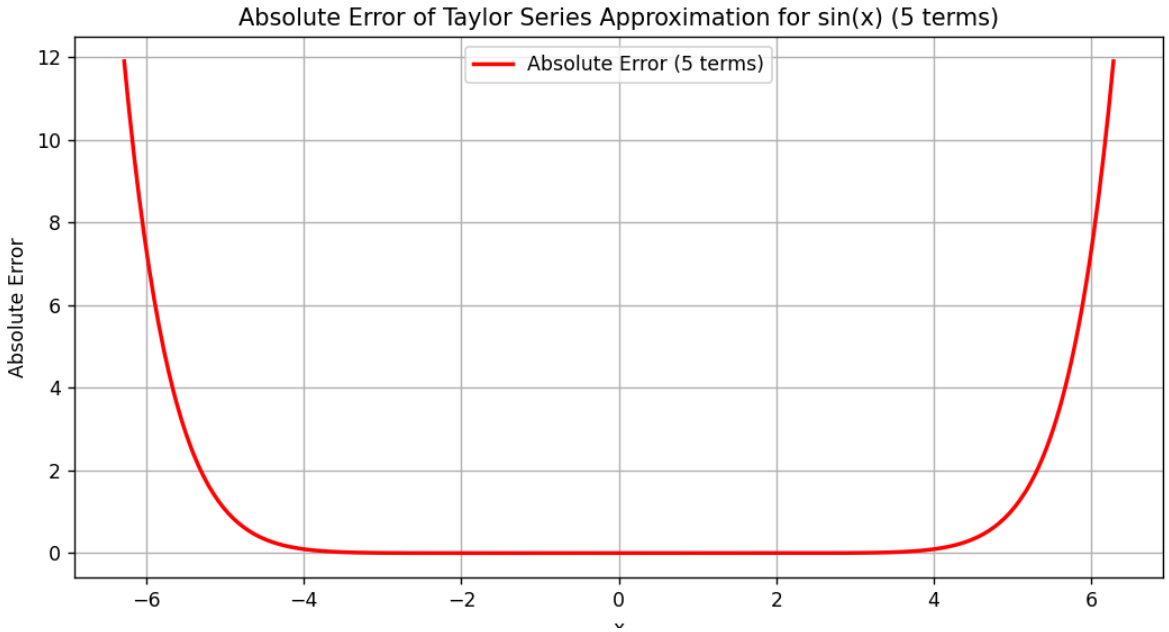


Figure 2. sin_error.png

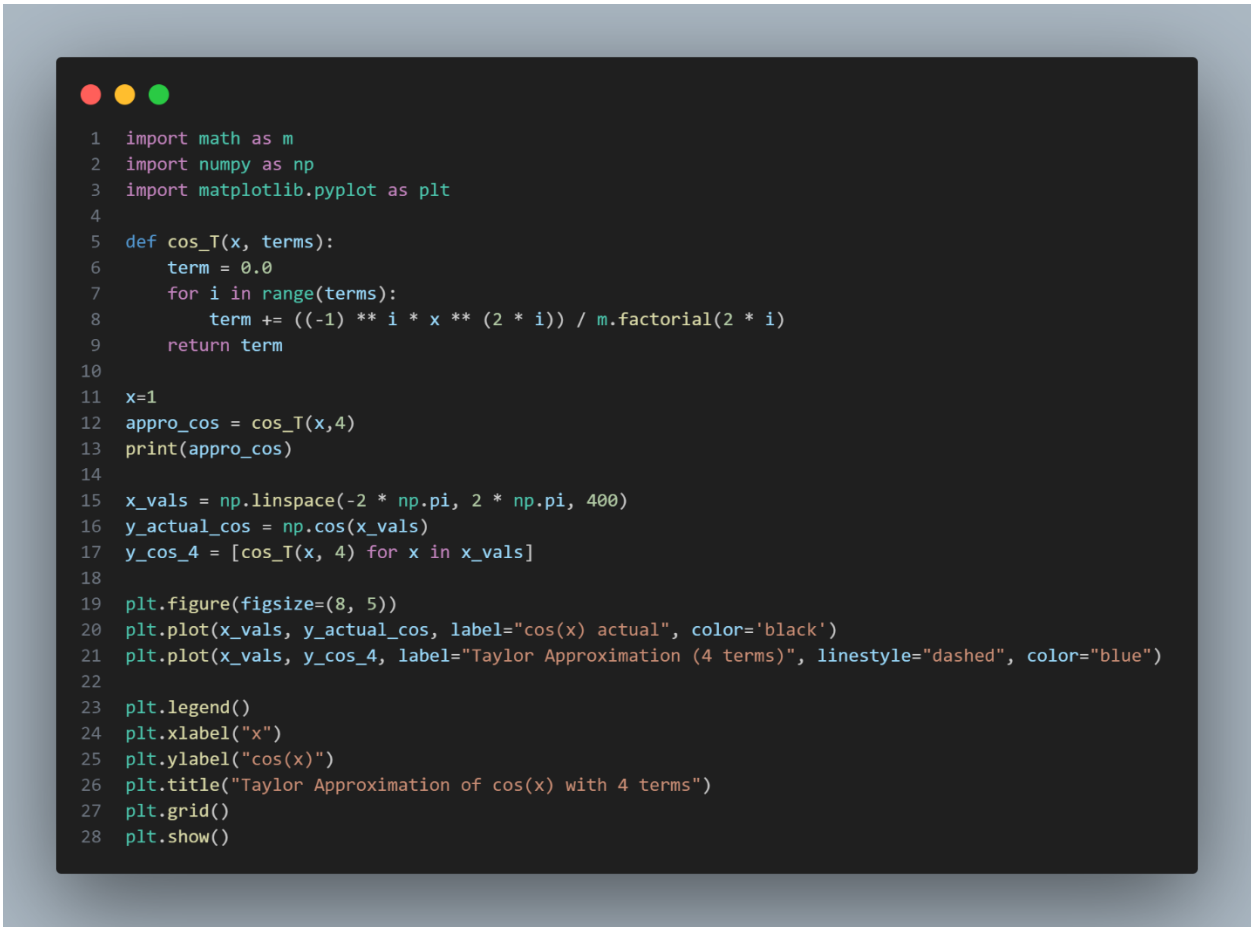
Where is the error largest? Why?

The error is largest at the endpoints of the interval, around $x=\pm 2\pi$.

This happens because Taylor series approximations are most accurate near the expansion point (which is implicitly $x=0x = 0$ in this case) and become less accurate as it moves farther away.

Implement the Taylor Series for cos(x) with 4 terms around x=0.

Code:



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

def cos_T(x, terms):
    term = 0.0
    for i in range(terms):
        term += ((-1) ** i * x ** (2 * i)) / m.factorial(2 * i)
    return term

x=1
appro_cos = cos_T(x,4)
print(appro_cos)

x_vals = np.linspace(-2 * np.pi, 2 * np.pi, 400)
y_actual_cos = np.cos(x_vals)
y_cos_4 = [cos_T(x, 4) for x in x_vals]

plt.figure(figsize=(8, 5))
plt.plot(x_vals, y_actual_cos, label="cos(x) actual", color='black')
plt.plot(x_vals, y_cos_4, label="Taylor Approximation (4 terms)", linestyle="dashed", color="blue")

plt.legend()
plt.xlabel("x")
plt.ylabel("cos(x)")
plt.title("Taylor Approximation of cos(x) with 4 terms")
plt.grid()
plt.show()
```

Result:

```
PS E:\Homework\TMC\Lab1> & C:/Python312/python.exe e:/Homework/TMC/Lab1/ex3b.py
0.5402777777777777
```

Plot:

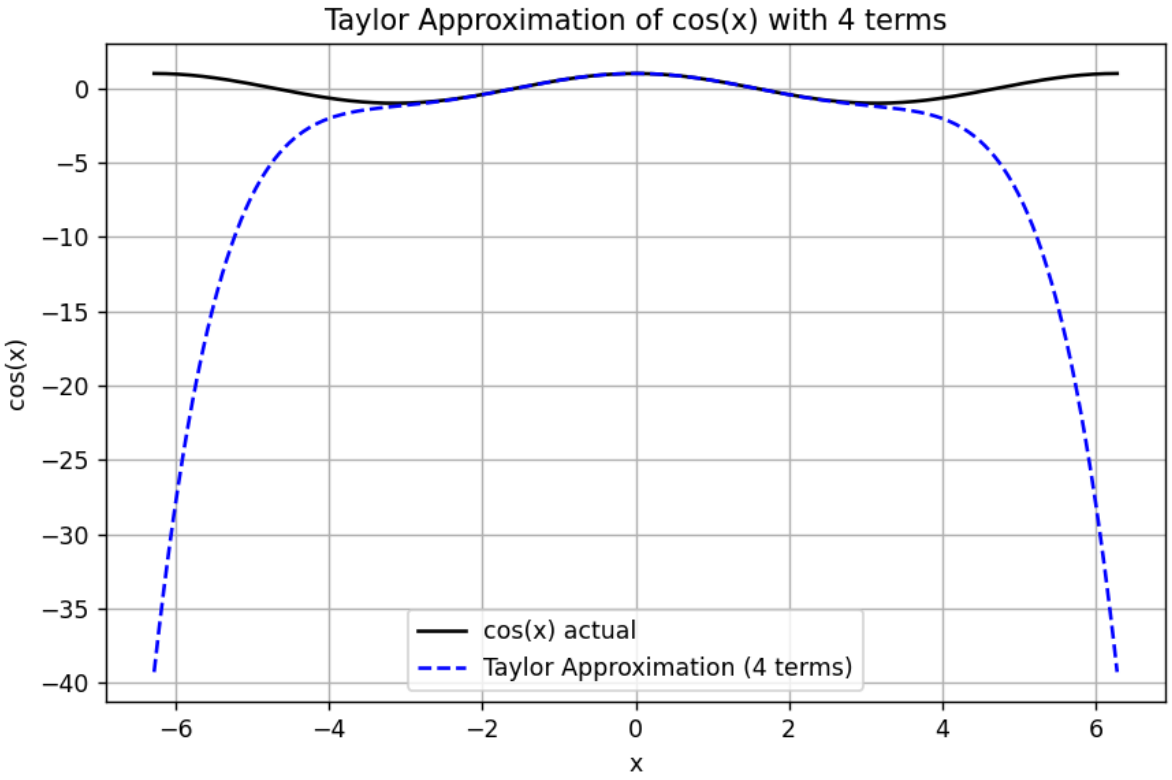


Figure 3. cos_taylor.png

Compare the convergence of the Taylor Series for sin(x) and cos(x)

The Taylor series for sin(x) and cos(x) have similar structures but distinct properties. The sin(x) series consists of only odd-powered terms, beginning with x, whereas the cos(x) series includes only even-powered terms, starting with 1. Both series provide accurate approximations when x is small, but their precision diminishes for larger |x| unless additional terms are included. Since cos(x) starts at 1, it generally offers better accuracy near x=0, whereas sin(x), which starts with x, has slightly higher initial error. Moreover, sin(x) is an odd function, meaning $\sin(-x)=-\sin(x)$, while cos(x) is even, satisfying $\cos(-x)=\cos(x)$. In general, both series perform well for small x, but increasing the number of terms is necessary to maintain accuracy for larger values.