Lab 6 Prelab

Matthew Menendez-Aponte

February 27, 2024

Problem 1

Estimate the derivative of $f(x) = cos(x)$ at $x = \pi/2$ using the forward and centered difference methods.

Forward Difference

$$f'(s) = \frac{f(s+h) - f(s)}{h}$$

Centered Difference

$$f'(s) = \frac{f(s+h) - f(s-h)}{2h}$$

Approximate the derivative with multiple h values: $h = 0.02^{-np.arange(0,10)}$. Finally, determine the order of the approximation techniques.

Solution

The following numerical code written.

```
# 2/26/2024 - APPM4600 Prelab6
# Matthew Menendez-Aponte

# imports
import numpy as np
import matplotlib.pyplot as plt

# make functions for forward and centered difference
def forwardDifference(f,s,h):
    fprime = (f(s+h) - f(s))/h
    return fprime
def centeredDifference(f,s,h):
    fprime = (f(s + h) - f(s-h)) / (2*h)
    return fprime
def prelab6_driver():
    hVec = 0.01 * 2. ** (-np.arange(0, 10))
    print('h vector: ', hVec)
```

```
    def f(x): return np.cos(x)
    s = np.pi/2

    f_prime_forward = forwardDifference(f, s, hVec)
    f_prime_centered = centeredDifference(f, s, hVec)

    print('Approximate derivative using forward difference')
    print(f_prime_forward)
    print('Approximate derivative using centered difference')
    print(f_prime_centered)

    f_prime_true = -np.sin(s)

    log_forward_error = np.log10(abs(f_prime_forward - f_prime_true))
    log_centered_error = np.log10(abs(f_prime_centered - f_prime_true))

    log_h_vec = np.log10(hVec)

    m_forward, b_forward = np.polyfit(log_h_vec, log_forward_error, 1)
    m_centered, b_centered = np.polyfit(log_h_vec, log_centered_error, 1)
    print(f'forward slope: {m_forward:.2f}')
    print(f'centered slope: {m_centered: .2f}')

    # Plot differences
    # plot -log(h) vs log(error)
    plt.plot(log_h_vec, log_forward_error, 'xk--', markersize=10)
    plt.plot(log_h_vec, log_centered_error, '.b--', markersize=10)
    plt.legend([f'Forward Difference with Order: {m_forward:.4f}', f'Centered
        Difference with Order: {m_centered: .4f}'])
    plt.xlabel('-log(h)')
    plt.ylabel('log(error)')
    plt.title(f'Approximating the Derivative of cos(x) using \nForward and Centered
        Difference\n s = {s:.3f}')
    plt.grid(True)
    plt.gca().invert_xaxis()
#    plt.show()
    plt.savefig('Approx.png',bbox_inches='tight')


if __name__ == '__main__':
    prelab6_driver()
```

This code outputs:

```
/usr/bin/python3.10 /home/mcma/APPM4600/APPM4600/Labs/Lab6/PreLab6.py
h vector:  [1.000000e-02 5.000000e-03 2.500000e-03 1.250000e-03 6.250000e-04
 3.125000e-04 1.562500e-04 7.812500e-05 3.906250e-05 1.953125e-05]
Approximate derivative using forward difference
```

```
[-0.99998333 -0.99999583 -0.99999896 -0.99999974 -0.99999993 -0.99999998
 -1.         -1.         -1.         -1.        ]
Approximate derivative using centered difference
[-0.99998333 -0.99999583 -0.99999896 -0.99999974 -0.99999993 -0.99999998
 -1.         -1.         -1.         -1.        ]
forward slope: 2.00
centered slope:  2.00

Process finished with exit code 0
```

For $x = \pi/2$ the difference between the methods is negligible.

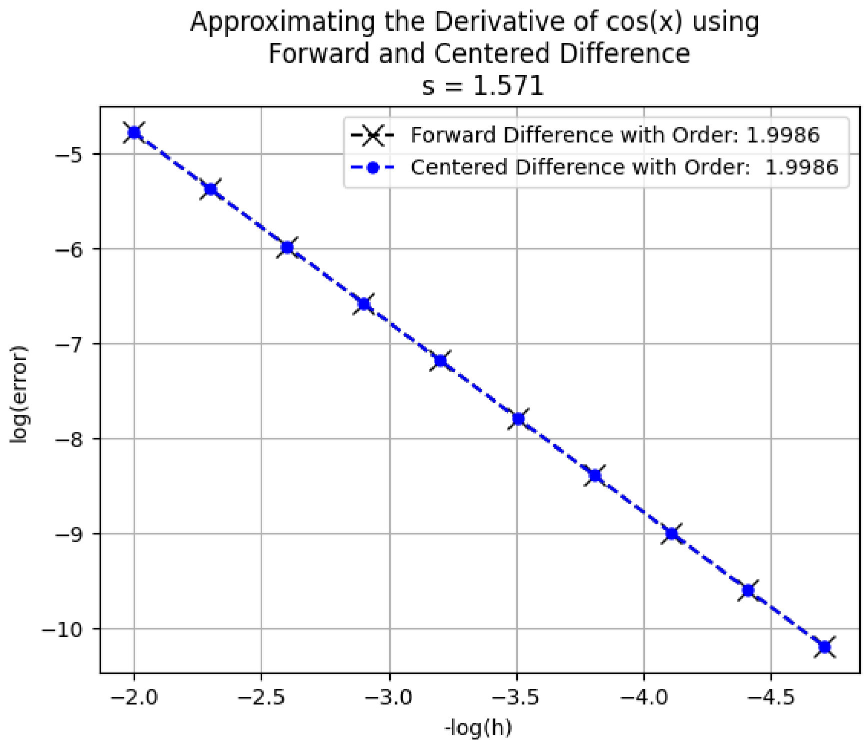

Figure 1: s = $\frac{\pi}{2}$

For $x = \pi/4$ the results are more interesting.

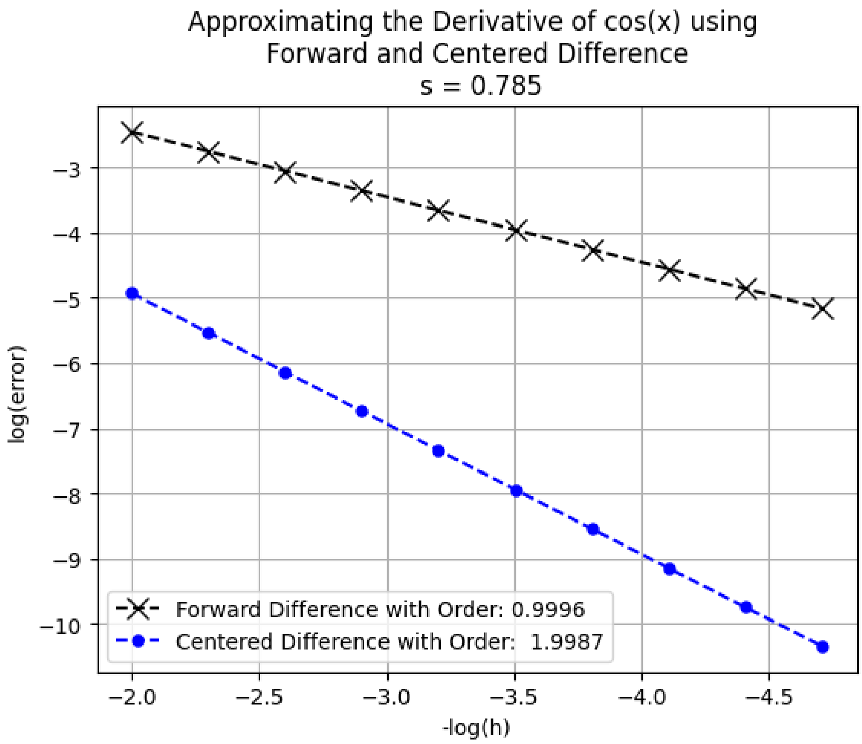

Figure 2: s = $\frac{\pi}{4}$

This trend continues for $x = \pi/8$.

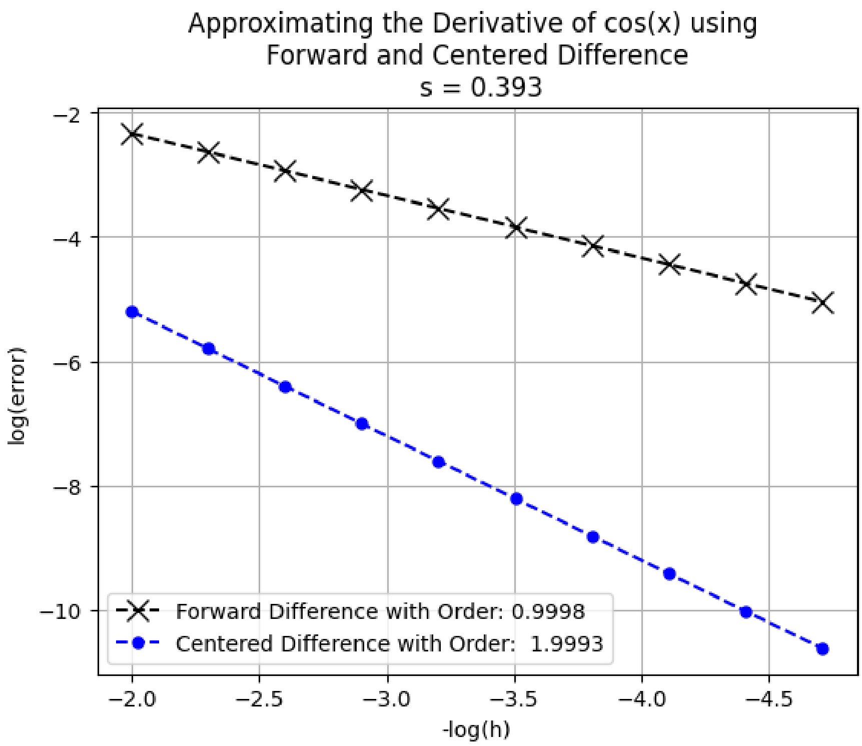

Figure 3: s = $\frac{\pi}{8}$

From these other x values we are able to determine that the forward and centered difference methods are of orders 1 and 2 respectively.