week-2-submission-2

January 29, 2024

[2]:
```
# Imports
import matplotlib.pyplot as plt
import numpy as np
```

0.0.1 Calculate Slope

[34]:
```
# Define the functions
def loch_ness(x, b):
    return 2 + np.exp(-b * x) * np.sin(x)

def bell(x, a):
    return np.exp(-(x - a)**2)

# Numerical Differentiation Forumla
    # f(x)  f(x+Δx)-f(x-Δx)/2Δx
```

[42]:
```
# X = 10 for both Δx
x_value = 10  # x value
a_value = 15  # Center for the Bell function
b_value = 0.05  # Parameter for the Loch Ness function
delta_x = 0.1  # Δx value
delta_x_2 = 0.01  # 2nd Δx value


# Calculate derivatives
loch_ness_derivative = (loch_ness(x_value + delta_x, b_value) -␣
 ↪loch_ness(x_value - delta_x, b_value)) / (2 * delta_x)
bell_derivative = (bell(x_value + delta_x, a_value) - bell(x_value - delta_x,␣
 ↪a_value)) / (2 * delta_x)
loch_ness_derivative_2 = (loch_ness(x_value + delta_x_2, b_value) -␣
 ↪loch_ness(x_value - delta_x_2, b_value)) / (2 * delta_x_2)
bell_derivative_2 = (bell(x_value + delta_x_2, a_value) - bell(x_value -␣
 ↪delta_x_2, a_value)) / (2 * delta_x_2)


print("Derivative of Loch Ness Monster function Δx = 0.1:",␣
 ↪loch_ness_derivative)
```

```
print("Derivative of Bell function Δx = 0.1:", bell_derivative)
print('')
print("Derivative of Loch Ness Monster function Δx = 0.01:",␣
 ↪loch_ness_derivative_2)
print("Derivative of Bell functionΔx = 0.01:", bell_derivative_2)
```

```
Derivative of Loch Ness Monster function Δx = 0.1: -0.49166525876701006
Derivative of Bell function Δx = 0.1: 1.6158730268299312e-10

Derivative of Loch Ness Monster function Δx = 0.01: -0.49241673974520506
Derivative of Bell functionΔx = 0.01: 1.3909710973466255e-10
```

[43]:
```
# X = 15 for both Δx
x_value = 15  # x value
a_value = 15  # Center for the Bell function
b_value = 0.05  # Parameter for the Loch Ness function
delta_x = 0.1  # Δx value
delta_x_2 = 0.01  # 2nd Δx value


# Calculate derivatives
loch_ness_derivative = (loch_ness(x_value + delta_x, b_value) -␣
 ↪loch_ness(x_value - delta_x, b_value)) / (2 * delta_x)
bell_derivative = (bell(x_value + delta_x, a_value) - bell(x_value - delta_x,␣
 ↪a_value)) / (2 * delta_x)
loch_ness_derivative_2 = (loch_ness(x_value + delta_x_2, b_value) -␣
 ↪loch_ness(x_value - delta_x_2, b_value)) / (2 * delta_x_2)
bell_derivative_2 = (bell(x_value + delta_x_2, a_value) - bell(x_value -␣
 ↪delta_x_2, a_value)) / (2 * delta_x_2)


print("Derivative of Loch Ness Monster function Δx = 0.1:",␣
 ↪loch_ness_derivative)
print("Derivative of Bell function Δx = 0.1:", bell_derivative)
print('')
print("Derivative of Loch Ness Monster function Δx = 0.01:",␣
 ↪loch_ness_derivative_2)
print("Derivative of Bell functionΔx = 0.01:", bell_derivative_2)
```

```
Derivative of Loch Ness Monster function Δx = 0.1: -0.37353989779568275
Derivative of Bell function Δx = 0.1: 0.0

Derivative of Loch Ness Monster function Δx = 0.01: -0.3742031685669778
Derivative of Bell functionΔx = 0.01: 0.0
```

[ ]:
```
For Lochness, the negative value indicates a decreasing function at (x = 10), a␣
 ↪downward slope.
```

```
For (x = 15), it is still negative but less steep which might mean it is at the␣
 ↪point where it may ascend again.

For bell, for (x = 10), the values are very close to zero, indicating a very␣
 ↪flat part of the curve.
Since 10 is much farther than 15, it means it is far away from the center which␣
 ↪makes sense.
The farther away from the center, the more flat it becomes.
For (x = 15), since it is at the center of the bell curve, the slope would be␣
 ↪exactly zero as well.
```

0.0.2 Calculate Slope and Plot

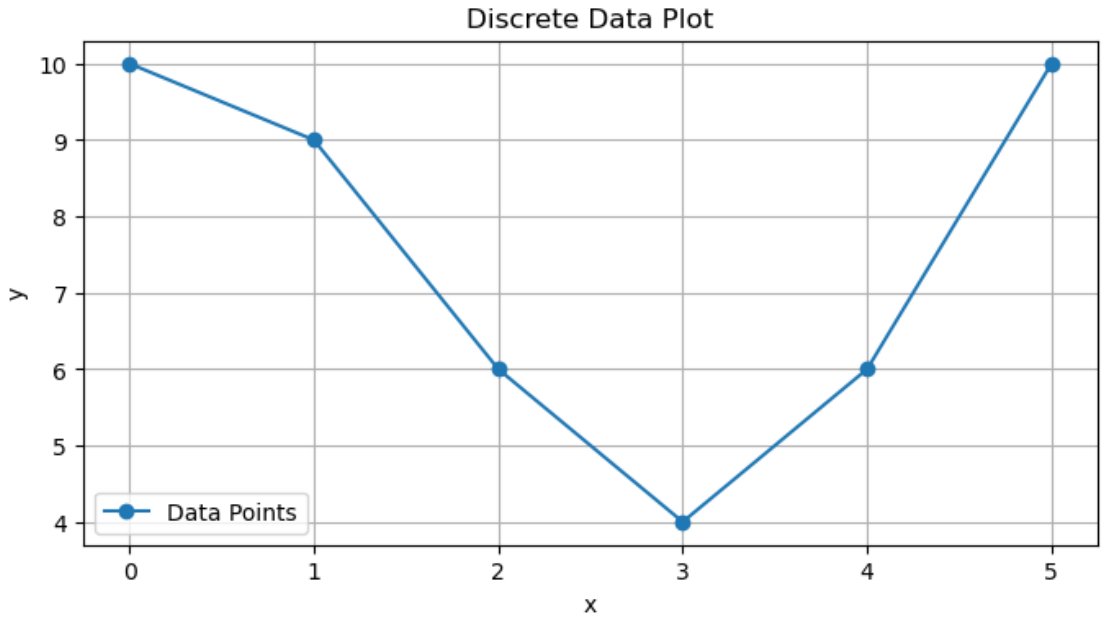
[51]:
```
# Define data points
x_discrete = [0, 1, 2, 3, 4, 5]
y_discrete = [10, 9, 6, 4, 6, 10]

# Plot the discrete data
plt.figure(figsize=(8, 4))
    # o- makes it so it is line plot with circle dots
plt.plot(x_discrete, y_discrete, 'o-', label='Data Points')
plt.title('Discrete Data Plot')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()
plt.show()

# Numerical Differentiation (Forward Difference) between x=1 and x=2
    # f(x)  f(x+Δx)-f(x)/Δx
    # Identify f(x) and f(x + Δx):
        # f(x) at x = 1 is y at x = 1, which is 9
    # Δx is 1 since 2-1 = 1
        # f(x + Δx) at 1 + 1 = 2
        # f(x) = 2 is = 6
    # Therefore it would be 6-9/1, hence equalling -3
slope_numerical_diff = (y_discrete[2] - y_discrete[1]) / (x_discrete[2] -␣
 ↪x_discrete[1])

# Simple Slope Calculation between x=4 and x=5
slope_simple = (y_discrete[5] - y_discrete[4]) / (x_discrete[5] - x_discrete[4])

print("Slope using Numerical Differentiation (x=1 to x=2):",␣
 ↪slope_numerical_diff)
print("Simple Slope Calculation (x=4 to x=5):", slope_simple)
```




```
Slope using Numerical Differentiation (x=1 to x=2): -3.0
Simple Slope Calculation (x=4 to x=5): 4.0
```

[ ]: