

Natasha Piedrabuena
Numerical Computation
Homework 8

In this homework I outline the implementation of a function that calculates numerical derivatives using three differentiation formulas: the 2-point forward difference, the 3-point forward difference, and the 3-point centered difference.

```
-----  
Enter the x values: 0.15 0.21 0.23 0.27 0.32 0.35  
Enter the corresponding f(x) values: 0.1761 0.3222 0.3617 0.4314 0.5051 0.5441  
Enter the step size (h): 0.01  
Enter the point at which to evaluate the derivative: 0.26  
  
Interpolating missing points...  
  
Interpolated value at x = 0.26:  $f(0.26) = 0.4150$   
Interpolated value at x = 0.28:  $f(0.28) = 0.4472$   
Interpolated value at x = 0.25:  $f(0.25) = 0.3979$   
  
-----  
  
Numerical differentiation Values:  
2-point forward difference:  $f'(0.26) = 1.6407$   
3-point forward difference:  $f'(0.26) = 1.6709$   
3-point centered difference:  $f'(0.26) = 1.6730$   
  
-----
```

It will take the values of x and corresponding f(x) and find the missing values and interpolate them to then compute the numerical differentiation of der at point 0.26.

The screenshot shows a VS Code editor with a file explorer on the left containing a directory named 'NUMERICAL_COMPUTAT...' with subfolders 'Homework_1' through 'Homework_11' and files like 'AI_main.py', 'main.py', 'Project-8.pdf', and 'README.md'. The main editor displays a Python script 'AI_main.py' with the following code:

```
22 evaluation_point = 0.26
23 step = 0.01
24
25 # Use interpolation to find missing values (manually interpolated)
26 interpolated_f_value = 0.3965 # This value was computed using Lagrange interpolation
27
28 # Adding interpolated value to the data
29 data_points_x.insert(4, evaluation_point)
30 data_points_f.insert(4, interpolated_f_value)
31
32 # Evaluate derivatives
33 result_2_point = compute_derivative(data_points_x, data_points_f, evaluation_point, step, "2-
34 result_3_point = compute_derivative(data_points_x, data_points_f, evaluation_point, step, "3-
35 result_4_point = compute_derivative(data_points_x, data_points_f, evaluation_point, step, "4-
36
37 print("2-Point Forward Formula Derivative:", result_2_point)
38 print("3-Point Forward Formula Derivative:", result_3_point)
39 print("4-Point Centered Formula Derivative:", result_4_point)
40
```

The terminal window at the bottom shows the command prompt history and a traceback error:

```
natashapiedrabuena@Natashas-MacBook-Pro-3 numerical_Computation % cd Homework_8
natashapiedrabuena@Natashas-MacBook-Pro-3 Homework_8 % source venv/bin/activate
(venv) natashapiedrabuena@Natashas-MacBook-Pro-3 Homework_8 % python3 AI_main.py
(venv) natashapiedrabuena@Natashas-MacBook-Pro-3 Homework_8 % python3 AI_main.py
Traceback (most recent call last):
  File "/Users/natashapiedrabuena/Desktop/Fall 2024/numerical_Computation/Homework_8/AI_main.py", line 34, in <module>
    >
      result_3_point = compute_derivative(data_points_x, data_points_f, evaluation_point, step, "3-point forward")
  File "/Users/natashapiedrabuena/Desktop/Fall 2024/numerical_Computation/Homework_8/AI_main.py", line 9, in compute_derivative
    - data_points_f[data_points_x.index(evaluation_point + 2 * step)] / (2 * step)
  File "/Users/natashapiedrabuena/Desktop/Fall 2024/numerical_Computation/Homework_8/AI_main.py", line 9, in compute_derivative
    - data_points_f[data_points_x.index(evaluation_point + 2 * step)] / (2 * step)
ValueError: 0.28 is not in list
(venv) natashapiedrabuena@Natashas-MacBook-Pro-3 Homework_8 %
```

The AI generated value computed the implementation incorrectly and I believe the reason it didn't run had to do with the function call and hallucination of certain functions that were never apart from the program. This happened before and I'm not surprised.

The screenshot shows a terminal window with the following output:

```
Enter the x values: 0.15 0.23 0.32 0.35
Enter the corresponding f(x) values: 0.1761 0.3617 0.5051 0.5441
Enter the step size (h): 0.01
Enter the point at which to evaluate the derivative: 0.26

Interpolating missing points...

Interpolated value at x = 0.26: f(0.26) = 0.4154
Interpolated value at x = 0.27: f(0.27) = 0.4318
Interpolated value at x = 0.28: f(0.28) = 0.4475
Interpolated value at x = 0.25: f(0.25) = 0.3983

Numerical differentiation Values:
2-point forward difference: f'(0.26) = 1.6404
3-point forward difference: f'(0.26) = 1.6731
3-point centered difference: f'(0.26) = 1.6758
```

For the extra credit these are the values I have gotten for the numerical differentiation and the for interpolating missing points lagrange from hw 7.