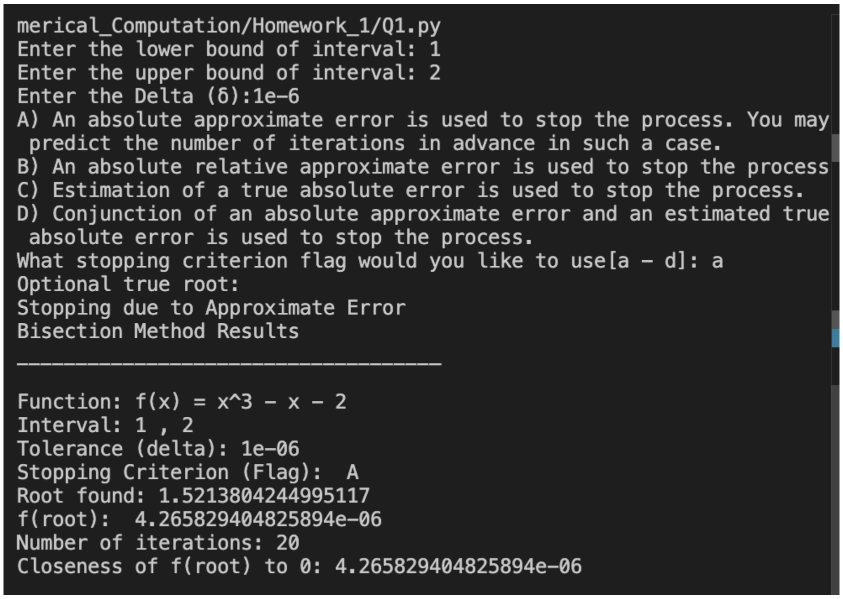
Natasha Piedrabuena
09/19/2024

5/100) Write a brief technical report presenting your results (**all roots found**) and **the corresponding number of iterations** in the **table**. Make a conclusion about the best stopping criterion or about their equivalence to each other and justify your conclusion. Make a conclusion about the quality of the AI-generated implementation of the bisection method.

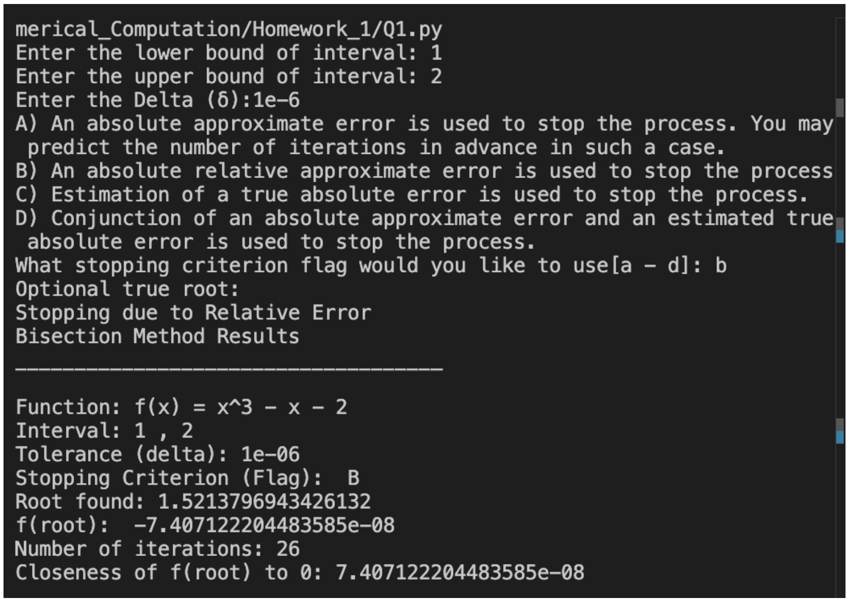
Q1)

In this problem I built a dissection function utilizing python. In this function the bisection method was manually implemented by collecting user input such as x1, x2, delta, flag and optional true root. I also implemented the stopping criteria in which the user can decide what flag either A(Absolute Approximate Error), B(Relative Approximate Error), C(True Absolute Error), or D(Both Approximate and True Absolute Error). The function manually computes midpoint iteratively and updates the interval to evaluate the function.
Test run (function used was f(x) = x^3−x−2 = 0):
Flag A)



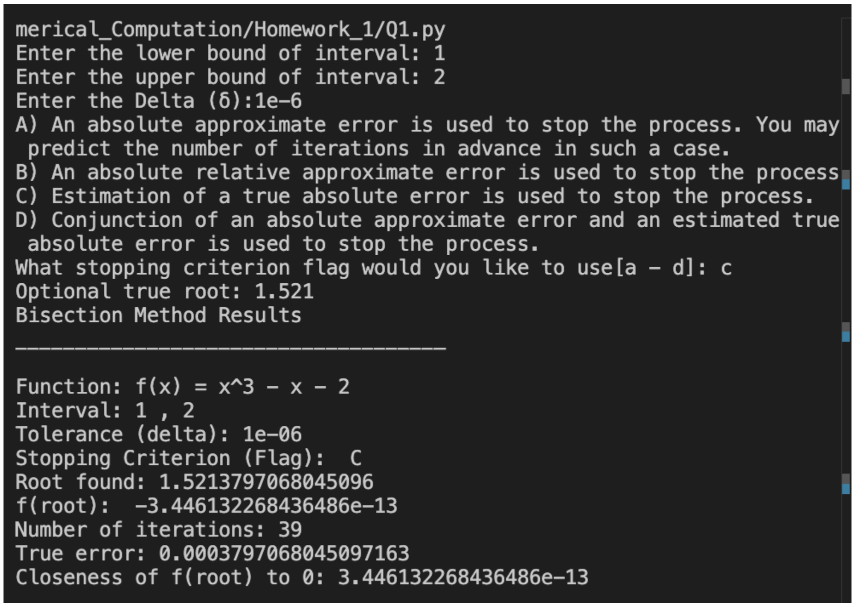
```
merical_Computation/Homework_1/Q1.py
Enter the lower bound of interval: 1
Enter the upper bound of interval: 2
Enter the Delta (δ):1e-6
A) An absolute approximate error is used to stop the process. You may
 predict the number of iterations in advance in such a case.
B) An absolute relative approximate error is used to stop the process
C) Estimation of a true absolute error is used to stop the process.
D) Conjunction of an absolute approximate error and an estimated true
 absolute error is used to stop the process.
What stopping criterion flag would you like to use[a - d]: a
Optional true root:
Stopping due to Approximate Error
Bisection Method Results

------------------------------------

Function: f(x) = x^3 - x - 2
Interval: 1 , 2
Tolerance (delta): 1e-06
Stopping Criterion (Flag):  A
Root found: 1.5213804244995117
f(root):  4.265829404825894e-06
Number of iterations: 20
Closeness of f(root) to 0: 4.265829404825894e-06
```

Flag B)



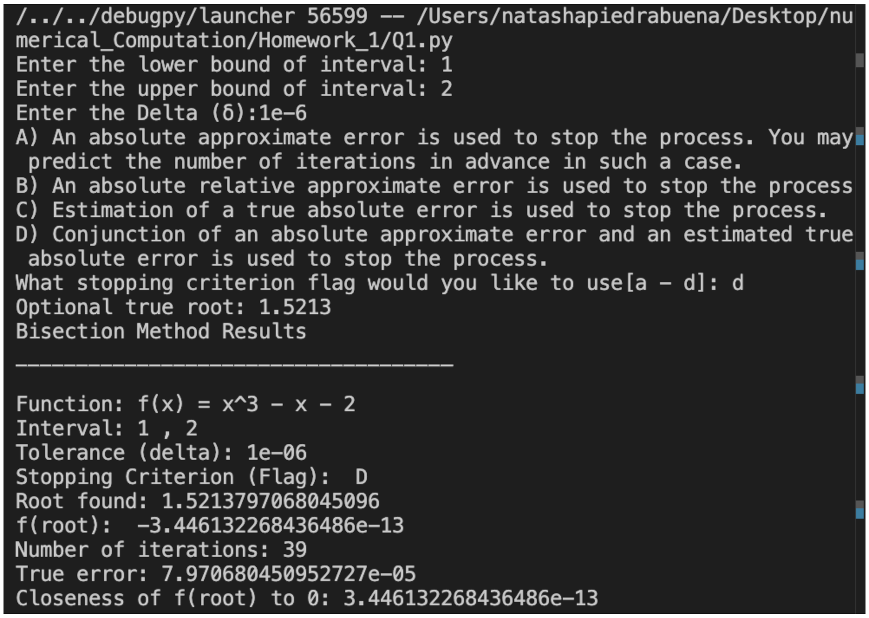
```
merical_Computation/Homework_1/Q1.py
Enter the lower bound of interval: 1
Enter the upper bound of interval: 2
Enter the Delta (δ):1e-6
A) An absolute approximate error is used to stop the process. You may
 predict the number of iterations in advance in such a case.
B) An absolute relative approximate error is used to stop the process
C) Estimation of a true absolute error is used to stop the process.
D) Conjunction of an absolute approximate error and an estimated true
 absolute error is used to stop the process.
What stopping criterion flag would you like to use[a - d]: b
Optional true root:
Stopping due to Relative Error
Bisection Method Results
------------------------------------

Function: f(x) = x^3 - x - 2
Interval: 1 , 2
Tolerance (delta): 1e-06
Stopping Criterion (Flag):  B
Root found: 1.5213796943426132
f(root):  -7.407122204483585e-08
Number of iterations: 26
Closeness of f(root) to 0: 7.407122204483585e-08
```

Flag C)



```
merical_Computation/Homework_1/Q1.py
Enter the lower bound of interval: 1
Enter the upper bound of interval: 2
Enter the Delta (δ):1e-6
A) An absolute approximate error is used to stop the process. You may
 predict the number of iterations in advance in such a case.
B) An absolute relative approximate error is used to stop the process
C) Estimation of a true absolute error is used to stop the process.
D) Conjunction of an absolute approximate error and an estimated true
 absolute error is used to stop the process.
What stopping criterion flag would you like to use[a - d]: c
Optional true root: 1.521
Bisection Method Results
------------------------------------

Function: f(x) = x^3 - x - 2
Interval: 1 , 2
Tolerance (delta): 1e-06
Stopping Criterion (Flag):  C
Root found: 1.5213797068045096
f(root):  -3.446132268436486e-13
Number of iterations: 39
True error: 0.0003797068045097163
Closeness of f(root) to 0: 3.446132268436486e-13
```

Flag D)



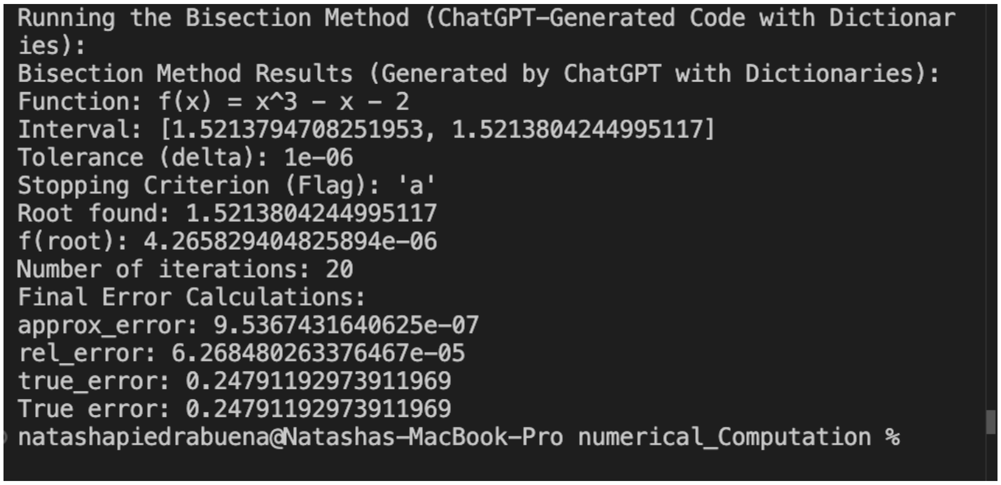
```
/../../debugpy/launcher 56599 -- /Users/natashapiedrabuena/Desktop/nu
merical_Computation/Homework_1/Q1.py
Enter the lower bound of interval: 1
Enter the upper bound of interval: 2
Enter the Delta (δ):1e-6
A) An absolute approximate error is used to stop the process. You may
 predict the number of iterations in advance in such a case.
B) An absolute relative approximate error is used to stop the process
C) Estimation of a true absolute error is used to stop the process.
D) Conjunction of an absolute approximate error and an estimated true
 absolute error is used to stop the process.
What stopping criterion flag would you like to use[a - d]: d
Optional true root: 1.5213
Bisection Method Results
------------------------------------

Function: f(x) = x^3 - x - 2
Interval: 1 , 2
Tolerance (delta): 1e-06
Stopping Criterion (Flag):  D
Root found: 1.5213797068045096
f(root):  -3.446132268436486e-13
Number of iterations: 39
True error: 7.970680450952727e-05
Closeness of f(root) to 0: 3.446132268436486e-13
```

Based on less iteration flag A is less than the others. However flag C and D got closer to closeness of f(root).

Q2)

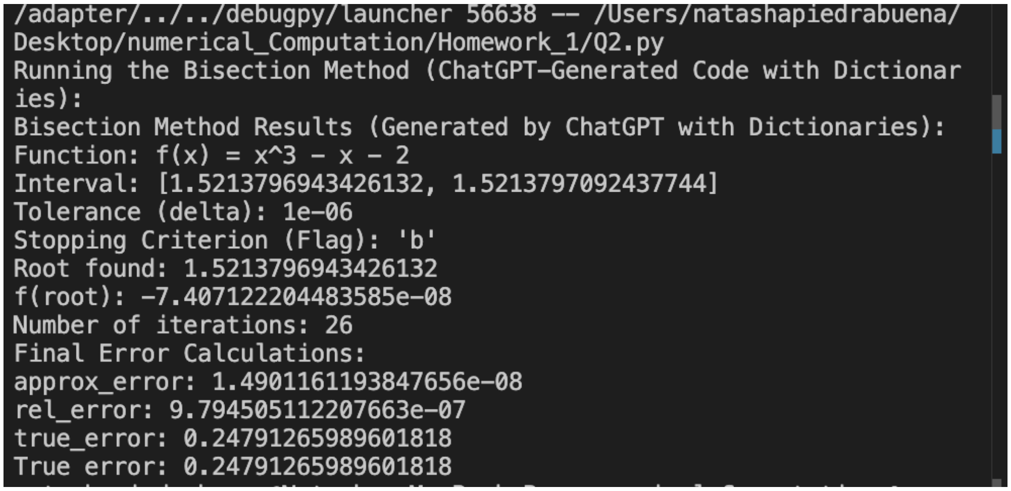
I asked chat gpt 4o to solve the problem, but in the beginning it didn’t understand the problem exactly. So I told it to create a solution based on some sections of my code so it will actually output something that can be compiled and run. The biggest issue was that its not flexible and it didnt know how to do user input or anything like that.

Flag A)



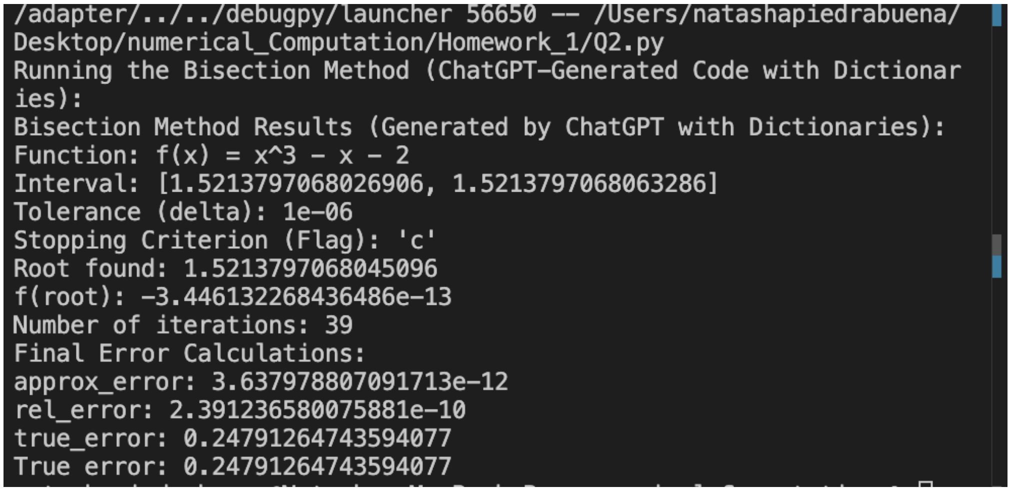
```
Running the Bisection Method (ChatGPT-Generated Code with Dictionar
ies):
Bisection Method Results (Generated by ChatGPT with Dictionaries):
Function: f(x) = x^3 - x - 2
Interval: [1.5213794708251953, 1.5213804244995117]
Tolerance (delta): 1e-06
Stopping Criterion (Flag): 'a'
Root found: 1.5213804244995117
f(root): 4.265829404825894e-06
Number of iterations: 20
Final Error Calculations:
approx_error: 9.5367431640625e-07
rel_error: 6.268480263376467e-05
true_error: 0.24791192973911969
True error: 0.24791192973911969
natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
```

Flag B)



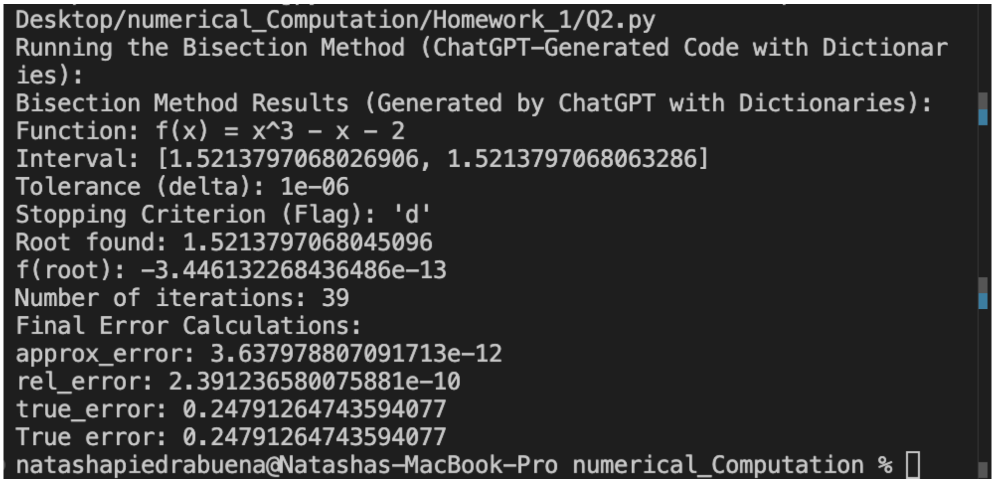
```
/adapter/../../debugpy/launcher 56638 -- /Users/natashapiedrabuena/
Desktop/numerical_Computation/Homework_1/Q2.py
Running the Bisection Method (ChatGPT-Generated Code with Dictionar
ies):
Bisection Method Results (Generated by ChatGPT with Dictionaries):
Function: f(x) = x^3 - x - 2
Interval: [1.5213796943426132, 1.5213797092437744]
Tolerance (delta): 1e-06
Stopping Criterion (Flag): 'b'
Root found: 1.5213796943426132
f(root): -7.407122204483585e-08
Number of iterations: 26
Final Error Calculations:
approx_error: 1.4901161193847656e-08
rel_error: 9.794505112207663e-07
true_error: 0.24791265989601818
True error: 0.24791265989601818
```

Flag C)



```
/adapter/../../debugpy/launcher 56650 -- /Users/natashapiedrabuena/
Desktop/numerical_Computation/Homework_1/Q2.py
Running the Bisection Method (ChatGPT-Generated Code with Dictionar
ies):
Bisection Method Results (Generated by ChatGPT with Dictionaries):
Function: f(x) = x^3 - x - 2
Interval: [1.5213797068026906, 1.5213797068063286]
Tolerance (delta): 1e-06
Stopping Criterion (Flag): 'c'
Root found: 1.5213797068045096
f(root): -3.446132268436486e-13
Number of iterations: 39
Final Error Calculations:
approx_error: 3.637978807091713e-12
rel_error: 2.391236580075881e-10
true_error: 0.24791264743594077
True error: 0.24791264743594077
```

Flag D)

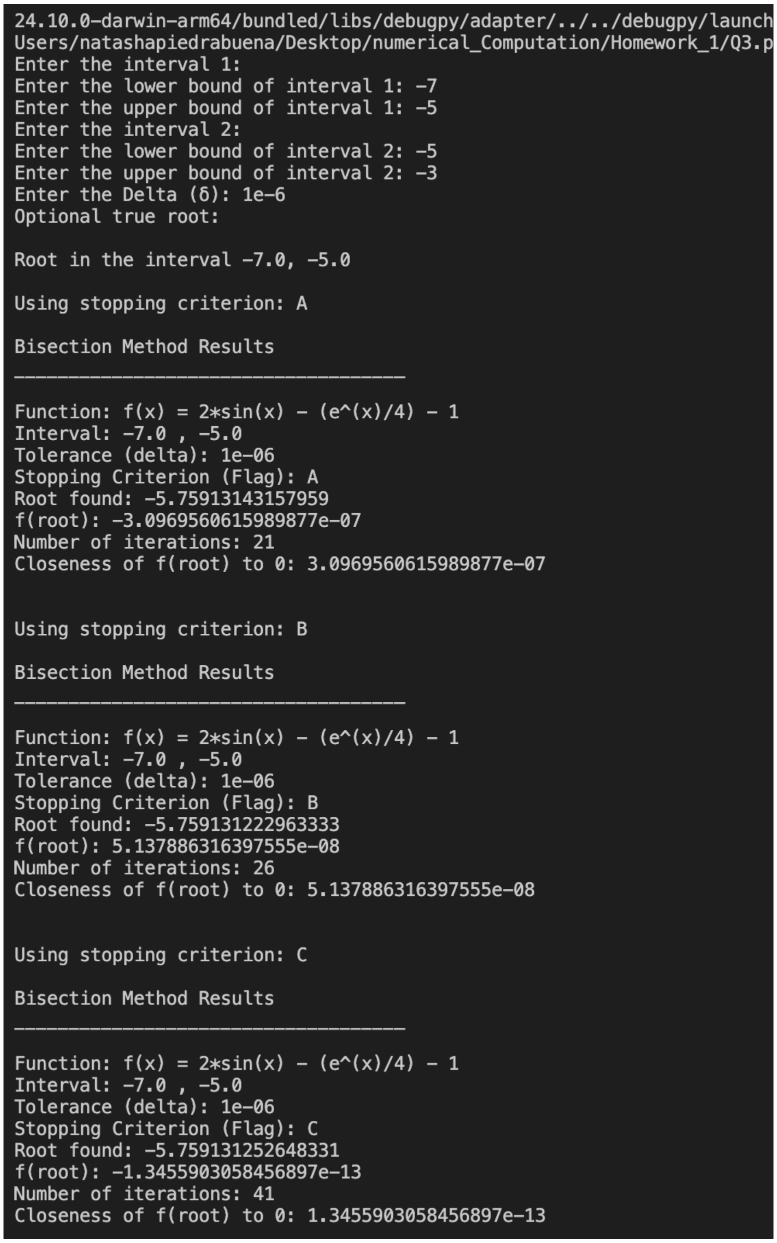


```
Desktop/numerical_Computation/Homework_1/Q2.py
Running the Bisection Method (ChatGPT-Generated Code with Dictionar
ies):
Bisection Method Results (Generated by ChatGPT with Dictionaries):
Function: f(x) = x^3 - x - 2
Interval: [1.5213797068026906, 1.5213797068063286]
Tolerance (delta): 1e-06
Stopping Criterion (Flag): 'd'
Root found: 1.5213797068045096
f(root): -3.446132268436486e-13
Number of iterations: 39
Final Error Calculations:
approx_error: 3.637978807091713e-12
rel_error: 2.391236580075881e-10
true_error: 0.24791264743594077
True error: 0.24791264743594077
natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
```

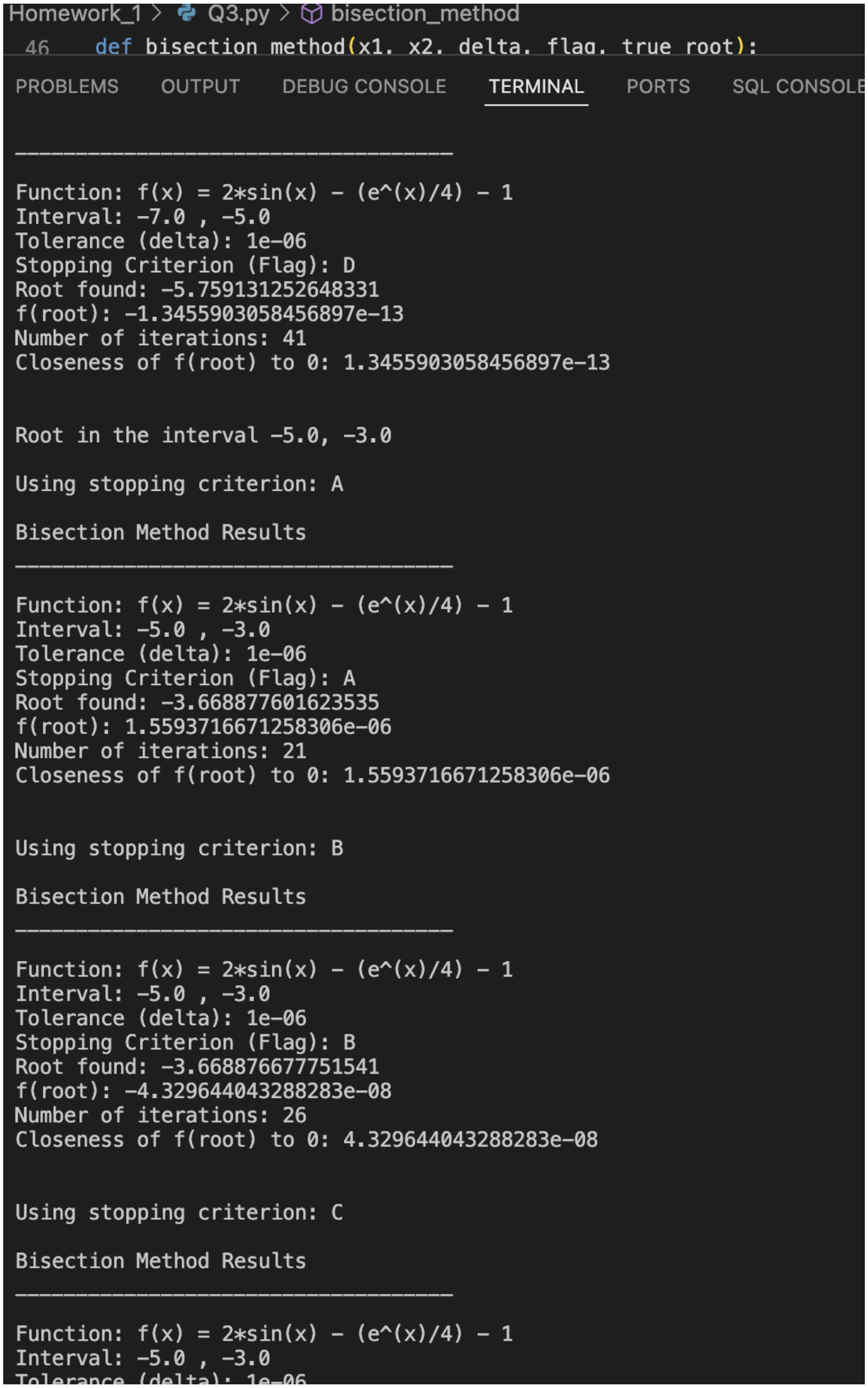
So far it seems it knows how to improve my code and the accuracy is similar to mine. I believe that using my code improved the way it organized the code. The iteration is the same amount as my own.

Q3)

For this problem, I applied the new equation, and I had to make adjustments where I made a list that added the interval 1 and 2 (like both intervals). It will output for all four flags as it iterates for each flag when its output.

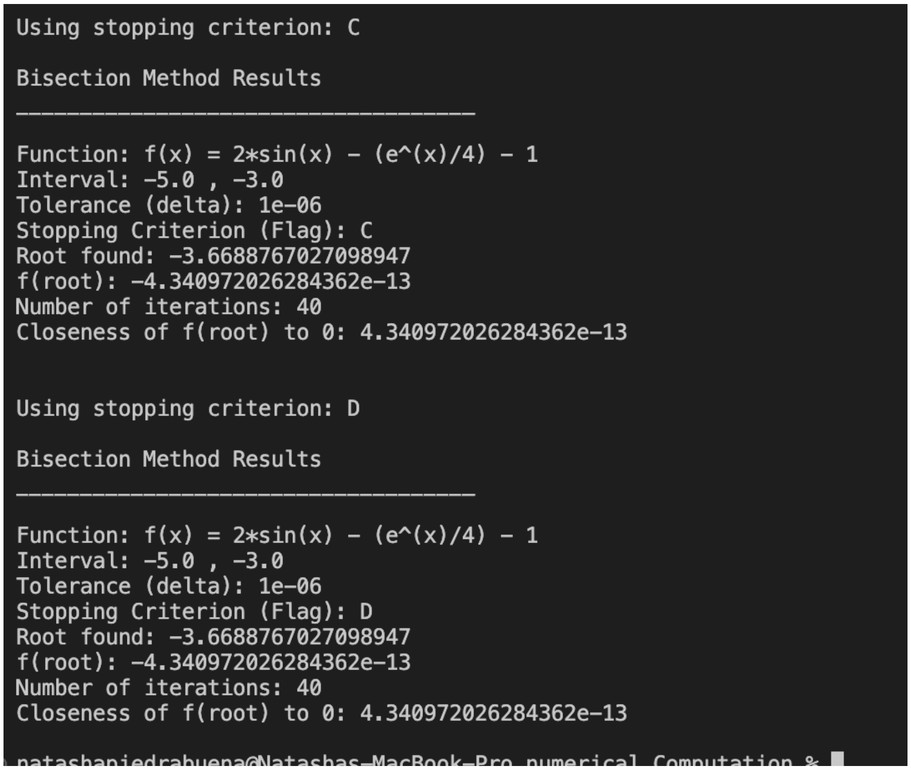


```
24.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launch
Users/natashapiedrabuena/Desktop/numerical_Computation/Homework_1/Q3.p
Enter the interval 1:
Enter the lower bound of interval 1: -7
Enter the upper bound of interval 1: -5
Enter the interval 2:
Enter the lower bound of interval 2: -5
Enter the upper bound of interval 2: -3
Enter the Delta (δ): 1e-6
Optional true root:

Root in the interval -7.0, -5.0

Using stopping criterion: A

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -7.0 , -5.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): A
Root found: -5.75913143157959
f(root): -3.0969560615989877e-07
Number of iterations: 21
Closeness of f(root) to 0: 3.0969560615989877e-07


Using stopping criterion: B

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -7.0 , -5.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): B
Root found: -5.759131222963333
f(root): 5.137886316397555e-08
Number of iterations: 26
Closeness of f(root) to 0: 5.137886316397555e-08


Using stopping criterion: C

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -7.0 , -5.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): C
Root found: -5.759131252648331
f(root): -1.3455903058456897e-13
Number of iterations: 41
Closeness of f(root) to 0: 1.3455903058456897e-13
```



```
def bisection_method(x1, x2, delta, flag, true_root):
```

```
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -7.0 , -5.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): D
Root found: -5.759131252648331
f(root): -1.3455903058456897e-13
Number of iterations: 41
Closeness of f(root) to 0: 1.3455903058456897e-13


Root in the interval -5.0, -3.0

Using stopping criterion: A

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -5.0 , -3.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): A
Root found: -3.668877601623535
f(root): 1.5593716671258306e-06
Number of iterations: 21
Closeness of f(root) to 0: 1.5593716671258306e-06


Using stopping criterion: B

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -5.0 , -3.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): B
Root found: -3.668876677751541
f(root): -4.329644043288283e-08
Number of iterations: 26
Closeness of f(root) to 0: 4.329644043288283e-08


Using stopping criterion: C

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -5.0 , -3.0
Tolerance (delta): 1e-06
```



```
Using stopping criterion: C

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -5.0 , -3.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): C
Root found: -3.6688767027098947
f(root): -4.340972026284362e-13
Number of iterations: 40
Closeness of f(root) to 0: 4.340972026284362e-13


Using stopping criterion: D

Bisection Method Results
------------------------------------

Function: f(x) = 2*sin(x) - (e^(x)/4) - 1
Interval: -5.0 , -3.0
Tolerance (delta): 1e-06
Stopping Criterion (Flag): D
Root found: -3.6688767027098947
f(root): -4.340972026284362e-13
Number of iterations: 40
Closeness of f(root) to 0: 4.340972026284362e-13

natashapiedrabuena@Natashas-MacBook-Pro numerical Computation %
```

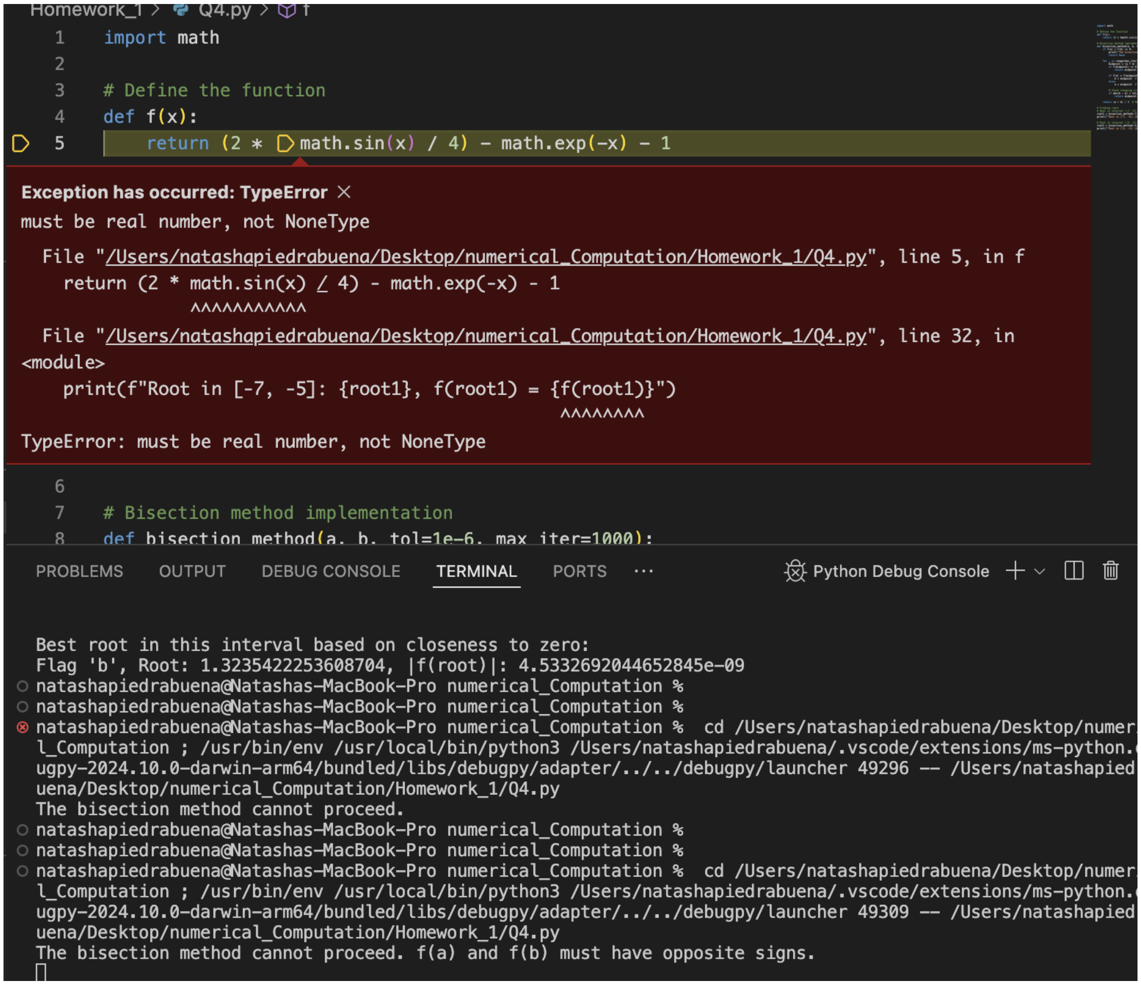
So far my code seems to run well, I didn’t input a true root, however it was still able to compute stop criteria c and d. I can see that it takes more iteration for c and d, This gives me an idea that it has to do with the lack of true root provided.

Q4)

For this question I utilized chat gpt 4o , in which I used the function that was provided for the homework plus the two intervals I had to implement in the code prior to this problem. I can see that the logic itself isn’t much different from my own. However I used the flowchart shown in class to help me implement my original code for question 1. There seems to be variations like the output and how the code was set up in the main function.

For some reason, it wouldn't run, I didn't want to fix it as it's kind of like we want to see how chatgpt behaves with these problems.
This screenshot shows some of the issues with the code. The code seems overly simplified with how it tried to solve this problem compared to question 2. The difference is that I used my code as a reference point for chatgpt. But for question 4 I didn’t because I wanted to see how capable

it was to do on its own.



```python
import math

# Define the function
def f(x):
    return (2 * math.sin(x) / 4) - math.exp(-x) - 1
```

```
Exception has occurred: TypeError ×
must be real number, not NoneType

  File "/Users/natashapiedrabuena/Desktop/numerical_Computation/Homework_1/Q4.py", line 5, in f
    return (2 * math.sin(x) / 4) - math.exp(-x) - 1
                ^^^^^^^^^^^
  File "/Users/natashapiedrabuena/Desktop/numerical_Computation/Homework_1/Q4.py", line 32, in
<module>
    print(f"Root in [-7, -5]: {root1}, f(root1) = {f(root1)}")
                                                   ^^^^^^^^
TypeError: must be real number, not NoneType
```

```python

# Bisection method implementation
def bisection_method(a, b, tol=1e-6, max_iter=1000):
```

```
Best root in this interval based on closeness to zero:
Flag 'b', Root: 1.3235422253608704, |f(root)|: 4.5332692044652845e-09
○ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
○ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
⊗ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %  cd /Users/natashapiedrabuena/Desktop/numer
l_Computation ; /usr/bin/env /usr/local/bin/python3 /Users/natashapiedrabuena/.vscode/extensions/ms-python.
ugpy-2024.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 49296 -- /Users/natashapied
uena/Desktop/numerical_Computation/Homework_1/Q4.py
The bisection method cannot proceed.
○ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
○ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %
○ natashapiedrabuena@Natashas-MacBook-Pro numerical_Computation %  cd /Users/natashapiedrabuena/Desktop/numer
l_Computation ; /usr/bin/env /usr/local/bin/python3 /Users/natashapiedrabuena/.vscode/extensions/ms-python.
ugpy-2024.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 49309 -- /Users/natashapied
uena/Desktop/numerical_Computation/Homework_1/Q4.py
The bisection method cannot proceed. f(a) and f(b) must have opposite signs.
```