Natasha Piedrabuena
Numerical Computation
Golden Section Homework

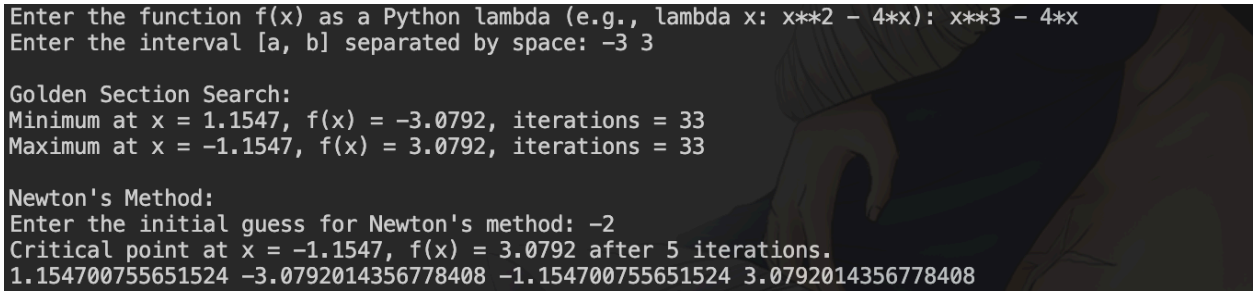
This report addresses the design and implementation of two optimization methods: the Golden Section method and Newton's method for finding both the minimum and maximum of a given function. Both algorithms were implemented in Python. Their performance was evaluated based on the accuracy of results and the number of iterations required for convergence.

The Golden Section method is a bracketing optimization technique that locates the extremum of a unimodal function within a specified interval in this case [-3,3]. For the minimization task, the method exploits the property of the golden ratio to iteratively reduce the interval size until the desired precision is achieved. For maximization, the algorithm was adapted to consider the reverse criterion.

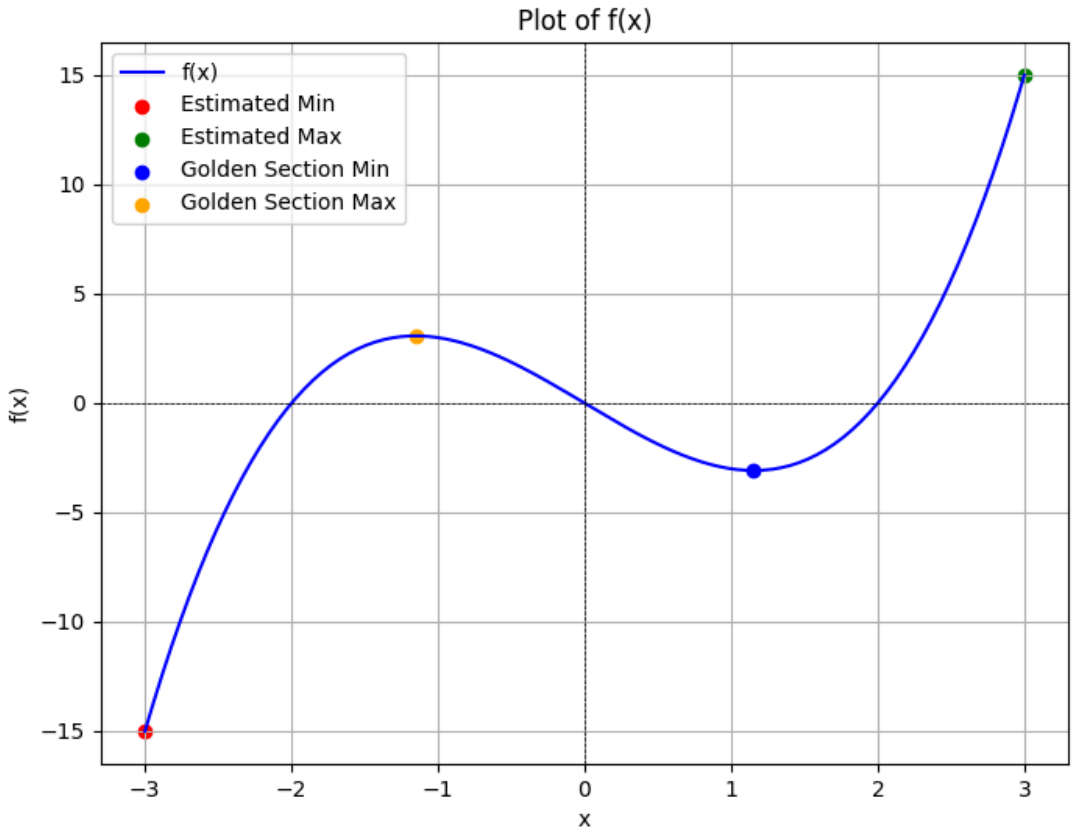
While on the other hand, Newton's method uses the derivative and second derivative of a function to locate extremal points. This method does not inherently distinguish between minima and maxima, requiring additional verification of the second derivative f''(x) at the extremum to classify the result.

Testing:

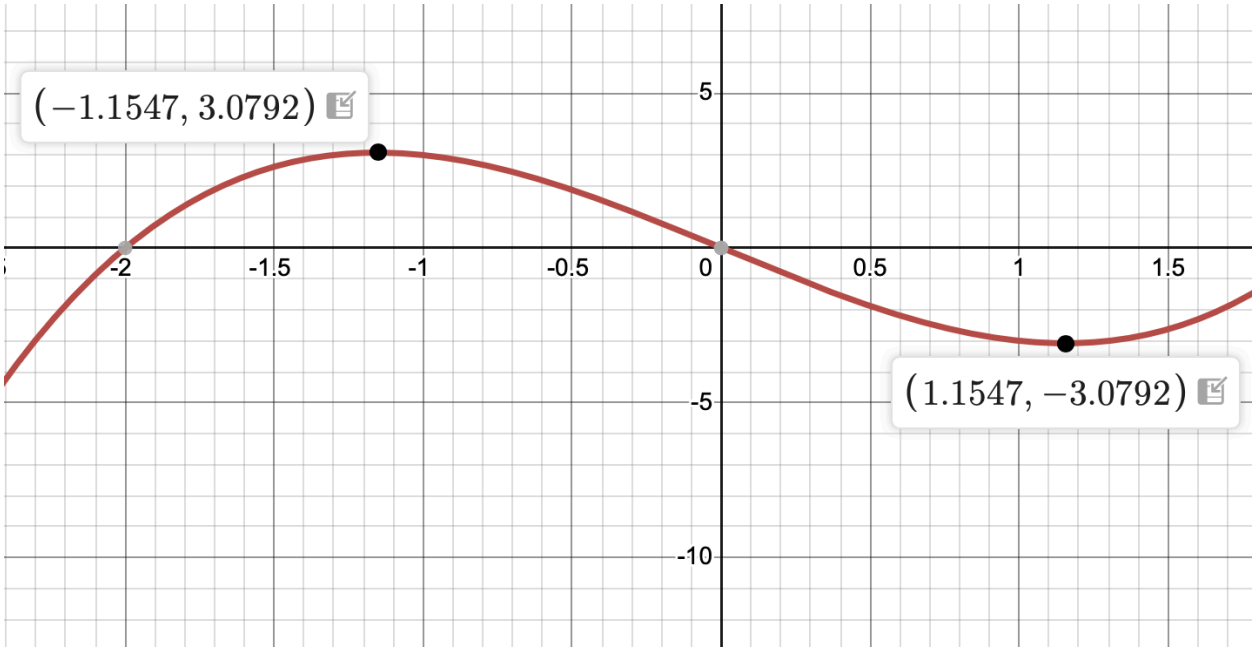
f(x) = x^3 - 4x



Matplotlib graph

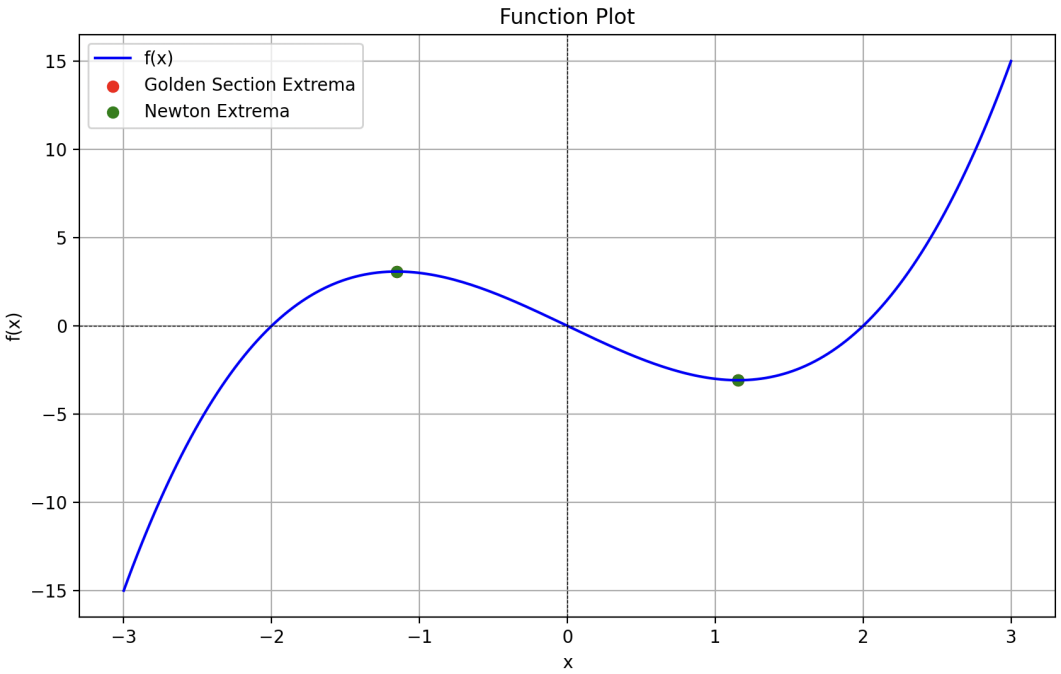


Desmos Graph for testing (min and max for function x^3 - 4x in interval [-3, 3]



The AI-generated Golden Section and Newton's methods were tested using the same functions. Results from the AI implementation were consistent with those of the manually implemented methods, suggesting that the generated code was both correct and usable.

```
/Fall\ 2024/numerical_Computation ; /usr/bin/env /Users/natashapiedrabuena/Desktop/Fall\ 2024/nur
ation/Homework_11/venv/bin/python /Users/natashapiedrabuena/.vscode/extensions/ms-python.debugpy-
win-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 52640 -- /Users/natashapiedrabuena/
 2024/numerical_Computation/Homework_10/AI_main.py
2024-12-06 18:03:42.138 Python[28997:500024] +[IMKClient subclass]: chose IMKClient_Modern
2024-12-06 18:03:42.138 Python[28997:500024] +[IMKInputSession subclass]: chose IMKInputSession_M
Results:
Golden Section Method (Min): x = 1.1547009010479887, f(x) = -3.0792014356775486, Iterations = 28
Golden Section Method (Max): x = -1.1547009010479887, f(x) = 3.0792014356775486, Iterations = 28
Newton's Method (Min): x = -1.1547005400098185, f(x) = 3.079201435678004, Iterations = 4
Newton's Method (Max): x = 1.1547005400098185, f(x) = -3.079201435678004, Iterations = 4
```



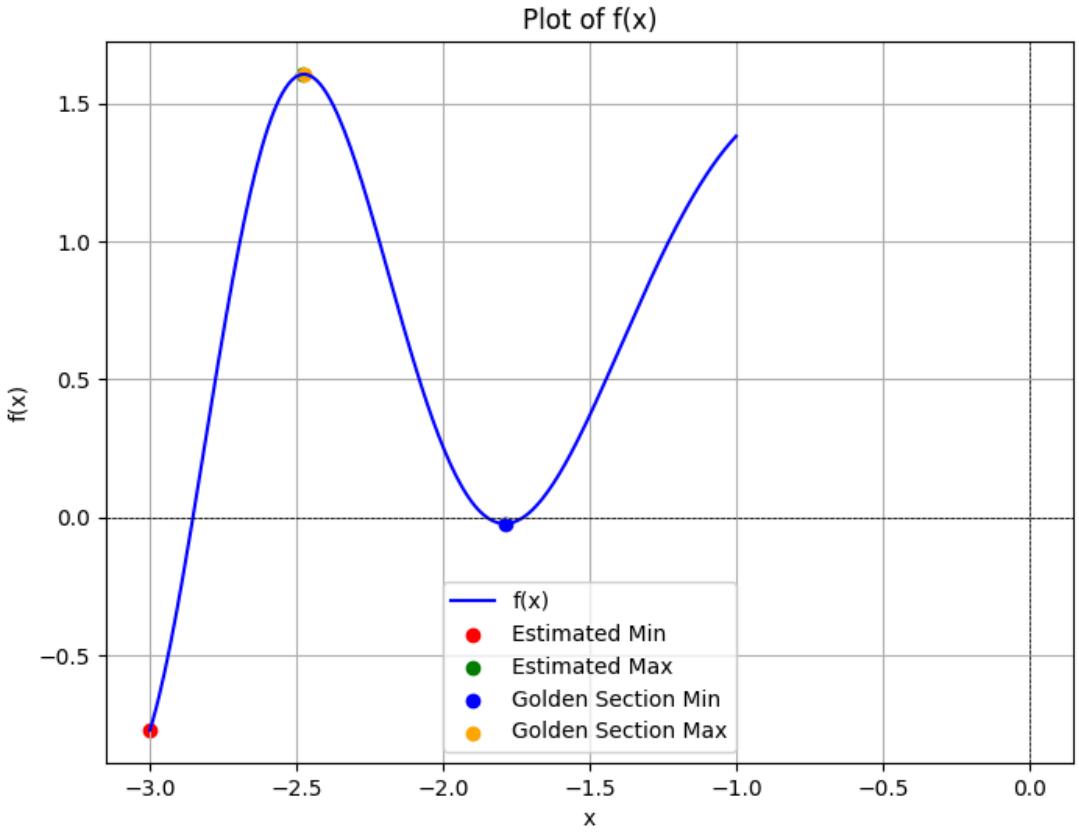
Both the Golden Section and Newton's methods were successfully implemented and tested on two functions with distinct characteristics. While Newton's method proved to be more computationally efficient, the Golden Section method offered greater flexibility. The code itself worked the same as mine, however the structure is not like regular python code. I can tell it's a chat gpt generated.

Extra Credit

```
Enter the function f(x) as a Python lambda (e.g., lambda x: x**2 - 4*x): -1*sin(x) + cos(x**2)
Enter the interval [a, b] separated by space: -3 -1

Golden Section Search:
Minimum at x = -1.7895, f(x) = -0.0220, iterations = 31
Maximum at x = -2.4746, f(x) = 1.6059, iterations = 31

Newton's Method:
Enter the initial guess for Newton's method: -2
Critical point at x = -1.7895, f(x) = -0.0220 after 4 iterations.
-1.78948218199236 -0.02197772865005898 -2.4746228958029035 1.6059264076178699
```



For the extra credit for - sin(x) + cos(x^2)