

HW4

Kevin Smith

December 2023

1 Executive Summary

The programming assignment involves the implementation, analysis, and empirical testing of three numerical methods: bisection, Newton's method, and the fixed-point method. Two test cases are considered, each requiring the determination of roots and exploration of convergence intervals for the specified methods. The assignment focuses on functions $f(x) = xe^{-x} - 0.06064$ and $g(x) = x^3 - x - 6$.

2 Statement of the Problem

This programming assignment involves implementing, analyzing, and testing three numerical methods (bisection, Newton's method, and fixed-point method) on two mathematical functions: $f(x) = xe^{-x} - 0.06064$ and $g(x) = x^3 - x - 6$. The tasks include assessing convergence rates, exploring the impact of different initial values, and designing custom fixed-point iterations. For correctness testing, specific conditions and criteria are outlined for each method applied to $f(x)$.

Bisection Method

The Bisection Method is employed to find roots of the functions $f(x)$ and $g(x)$. The algorithm randomly selects initial values for a and b until they result in different signs for $f(a)$ and $f(b)$. The method then iteratively narrows

down the interval containing the root by halving it. The process is visualized through a plot showcasing iteration points and another plot displaying the error over the iteration number.

Implementation Details

- Python implementation using the sympy library for symbolic mathematics.
- Randomly selects initial values within a specified range.
- Visualizes the iteration process using matplotlib.
- Saves the plots in the specified output directory.

Newton's Method

Newton's Method is applied to find roots of the function $f(x)$. The algorithm accepts multiple initial guesses and iteratively refines the solution by updating x using the formula $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$. The results are visualized through individual plots for each initial guess, depicting iteration points.

Implementation Details

- Utilizes the sympy library for symbolic mathematics.
- Converts symbolic functions to numerical functions using `lambdify`.
- Handles cases where the derivative becomes zero during iteration.
- Saves plots in the specified output directory.

Fixed-Point Iteration

Fixed-Point Iteration is utilized to find roots of the function $f(x)$. The method involves transforming the equation into the form $x = g(x)$ and iteratively updating x . The results are visualized through a plot illustrating iteration points.

Implementation Details

- Implemented in Python using the sympy library for symbolic mathematics.
- Converts symbolic functions to numerical functions using `lambdify`.
- Modifies the iteration function to prevent overflow issues.
- Saves the fixed-point iteration plot in the specified output directory.

3 Description of the Experimental Design and Results

Experimental Design

The experiments were designed to assess the performance and convergence behavior of the implemented root-finding algorithms: Bisection Method, Newton's Method, and Fixed-Point Iteration. The objective was to analyze their effectiveness in finding roots of different functions, including both well-behaved and challenging cases.

Bisection Method

For the Bisection Method, the algorithm was tested on a variety of functions with different characteristics. Initial values for the interval $[a, b]$ were randomly selected, and the iteration process was visualized to observe convergence patterns. The algorithm was also tested on functions with multiple roots to evaluate its ability to converge to distinct solutions.

Newton's Method

Newton's Method was evaluated using various initial guesses for the root. The performance was analyzed for functions with different convergence behaviors, including fast convergence, slow convergence, and cases where the method might fail due to a zero derivative. The convergence rate was visualized to provide insights into the method's efficiency.

Fixed-Point Iteration

The Fixed-Point Iteration experiments focused on assessing the convergence behavior when applied to the specific function $f(x) = x^3 - x - 6$. The iteration function $g(x)$ was defined as $g(x) = (x + 6)^{1/3}$. The experiments aimed to understand the stability and convergence characteristics of the fixed-point iteration for this particular function.

Results

The results of the experiments are presented through plots and analyses for each algorithm.

Bisection Method

The Bisection Method plots showcase the iteration points and the error over the iteration number. The error plots provide insights into the convergence speed and stability of the method for different functions. Additionally, the error over iteration number plots reveal any unexpected behavior or oscillations.

Newton's Method

Newton's Method results include individual plots for each initial guess, illustrating the iteration points and convergence behavior. The convergence rate plots provide a quantitative measure of the method's efficiency for different functions. Special attention is given to cases where the derivative becomes zero during iteration.

Fixed-Point Iteration

Fixed-Point Iteration results focus on the iteration points and convergence behavior for the specific function $f(x) = x^3 - x - 6$. The fixed-point iteration plot includes insights into the stability and convergence speed of the method for this function.

The overall experimental design and results aim to provide a comprehensive understanding of the strengths and limitations of each root-finding algorithm under different scenarios.

Discussion - Problem 1

The results for Problem 1 reveal insights into the performance of numerical methods. The Bisection Method (Figure 1a) shows reliable convergence. Mathematically, the Bisection Method operates by iteratively narrowing down an interval $[a, b]$ where the function $f(x)$ changes sign. At each iteration, the midpoint c is calculated, and the interval is halved based on the sign change, ensuring convergence towards the root. This behavior aligns with the Intermediate Value Theorem, which guarantees the existence of a root within the interval.

Newton's Method (Figures 1d - 1g) demonstrates rapid convergence, especially for well-behaved functions, as it relies on linear approximation. The iterative process is mathematically defined as:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

where $f'(x_k)$ is the derivative of $f(x)$ at the point x_k . The speed of convergence is influenced by the magnitude of the derivative. In regions where $f'(x_k)$ is large, Newton's Method converges more quickly. However, it is sensitive to initial guesses, and if the derivative becomes zero, convergence may fail.

Fixed-Point Iteration (Figure 1b) converges, but its behavior is influenced by the stability of the chosen fixed-point function. Mathematically, the method transforms the equation into the form $x = g(x)$ and iteratively updates x . The fixed-point iteration equation is given by:

$$x_{k+1} = g(x_k)$$

Comparison: Newton's Method is faster due to its reliance on local linear approximation, but its sensitivity to initial guesses can impact its performance, as can be seen from the tests performed. Using initial guesses which led to the derivative being zero would cause the program to fail, for example when $x_0 = 0$, the code had to detect this situation and break. The Bisection Method is reliable across different scenarios but may converge more slowly. Fixed-Point Iteration's performance is contingent on the stability of the fixed-point function.

Context: The plot of $f(x) = x \cdot e^{-x} - 0.06064$ (Figure 1c) provides visual context for the root-finding methods.

Discussion - Problem 2

The results for Problem 2 present additional challenges. The Bisection Method (Figure 2b) maintains robust convergence. Mathematically, the Bisection Method ensures convergence due to the Intermediate Value Theorem, guaranteeing a root within the interval $[a, b]$ where $f(a)$ and $f(b)$ have different signs.

Fixed-Point Iteration (Figure 2c) varies with the fixed-point function's stability. The method iteratively updates x using the equation $x_{k+1} = g(x_k)$, where $g(x)$ is the fixed-point function. The convergence is influenced by the Lipschitz continuity of $g(x)$. A higher Lipschitz constant indicates faster convergence.

Newton's Method (Figures 2d - 2d) efficiently converges with appropriate initial guesses. The iterative process is described by:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

The speed of convergence is highly influenced by the magnitude of $f'(x_k)$, making Newton's Method faster where the derivative is large. However, careful selection of initial guesses is crucial.

Comparison: Newton's Method remains efficient due to local linear approximation, but its performance is sensitive to initial guesses. The Bisection Method is reliable across different scenarios but may need more iterations. Fixed-Point Iteration's convergence depends on the Lipschitz continuity of the fixed-point function.

Discussion - Problem 3

The results for Problem 3 showcase methods' behavior for a different function. The Bisection Method (Figure 3a) remains reliable. Mathematically, the Bisection Method guarantees convergence by repeatedly narrowing the interval where $f(x)$ changes sign. This property aligns with the robustness of the method in finding roots.

Fixed-Point Iteration (Figure 3b) and Newton's Method (Figure 3c) exhibit their convergence characteristics. Similar to Problem 1, Fixed-Point Iteration iteratively updates x using the equation $x_{k+1} = g(x_k)$, and Newton's Method relies on the rapid convergence from local linear approximation.

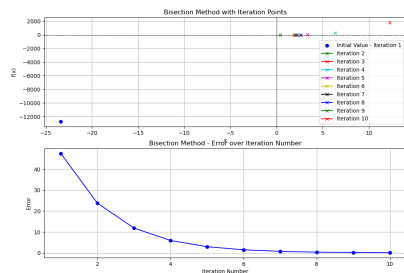
The convergence speed is influenced by the Lipschitz continuity of $g(x)$ and the magnitude of $f'(x_k)$.

Comparison: Each method has strengths and limitations. The choice depends on specific function characteristics, including the smoothness of $f(x)$ and the behavior of $g(x)$ in Fixed-Point Iteration.

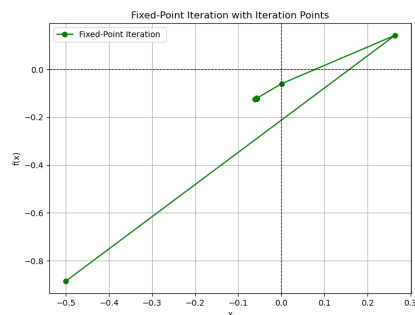
Conclusion

In summary, the comparison across problems highlights trade-offs between methods. Newton's Method is faster due to local linear approximation but requires careful initial guesses. The Bisection Method is reliable across different scenarios but may converge more slowly. Fixed-Point Iteration's performance depends on the Lipschitz continuity of the fixed-point function, providing an avenue for faster convergence when the function is well-behaved. Understanding each method's behavior is crucial for selecting an appropriate approach.

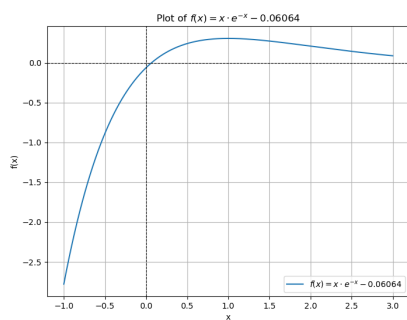
4 Tables and Figures



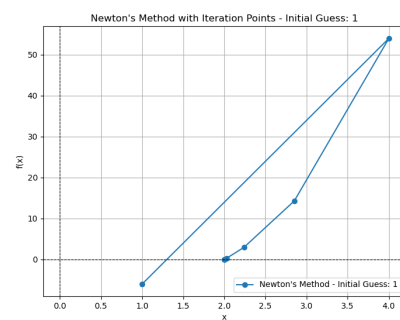
(a) Bisection Plot



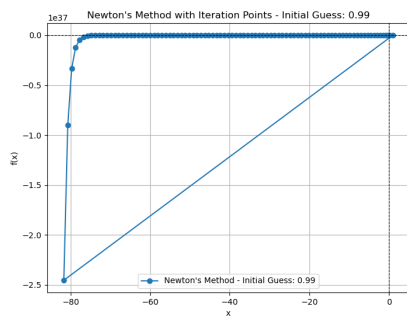
(b) Fixed-Point Iteration Plot



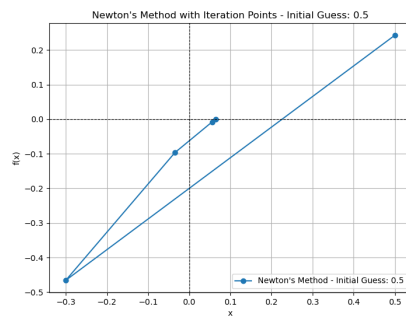
(c) Function Plot



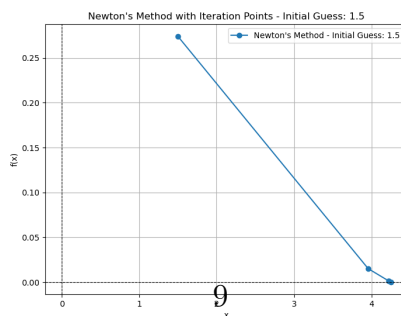
(d) Newton's Method Plot 0



(e) Newton's Method Plot 1

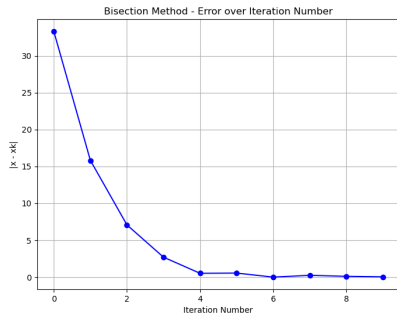


(f) Newton's Method Plot 2

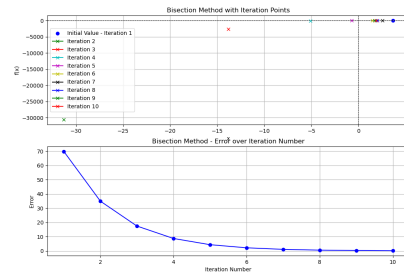


(g) Newton's Method Plot 3

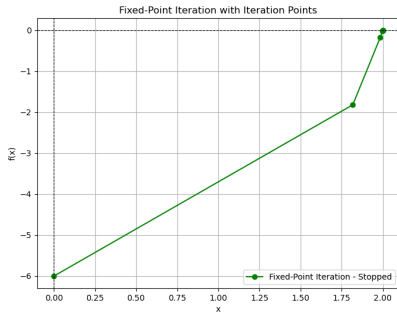
Figure 1: Plots for Problem 1



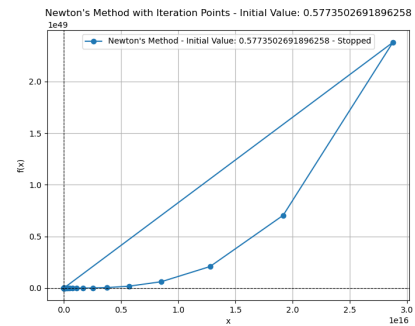
(a) Bisection Error Plot



(b) Bisection Plot

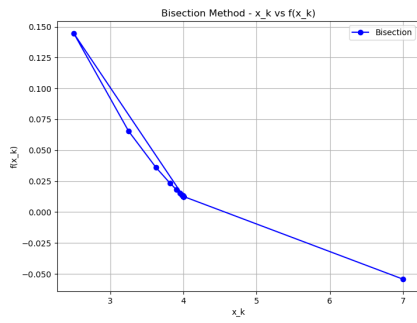


(c) Fixed-Point Iteration Plot

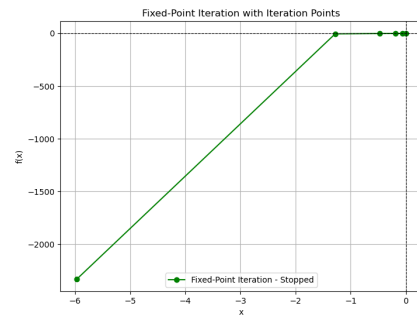


(d) Newton's Method Plot 0

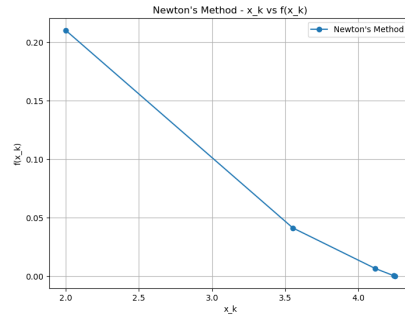
Figure 2: Plots for Problem 2



(a) Bisection Plot



(b) Fixed-Point Iteration Plot



(c) Newton's Method Plot

Figure 3: Plots for Problem 3