Root Finding Methods

Reva Dhillon

September 6, 2022

1 Introduction

This report aims to give a brief introduction of the Bisection, Newton and Secant methods of finding roots and describes the general algorithms employed by them.
The following functions have been analysed in detail using these methods:

- $f_1(x) = x^3 - 3x^2 - x + 9$
- $f_2(x) = e^x f_1(x)$
- $f_3(x) = x^3 - 2x + 2$

In addition the variations in time taken and number of iterations for the methods with different starting points has been graphed and described.

2 Bisection Method (Interval Halving Method/Binary Search Method/Dichotomy Method)

In mathematics, the **bisection method** is a root-finding method that applies to any continuous function for which one knows two values with opposite signs.
The method consists of repeatedly bisecting the interval defined by these values and then selecting the subinterval in which the function changes sign, and therefore must contain a root.
It is a very simple and robust method, but it is also relatively slow. Because of this, it is often used to obtain a rough approximation to a solution which is then used as a starting point for more rapidly converging methods.

2.1 The algorithm:

The following are known:

- Function $f(x)$
- Interval endpoint values $x = a$, $x = b$ where $a < b$ and $f(a) \times f(b) < 0$.
- Tolerance ϵ

The method is as follows:

- Find c and compute $f(c)$.
$$c = \frac{a+b}{2} \tag{1}$$
- Pick a new subinterval with endpoints having opposite signs.If $f(a) \times f(c) < 0$, let $b = c$. Else let $a = c$. This gives the new subinterval $[a, c]$ or $[c, b]$.
- Repeat the steps listed above until you find an $x = c$ such that $|f(x)| < \epsilon$. If this condition is satisfied a root of $f(x)$ is c.

3 Newton Method

In numerical analysis, **Newton's method**, also known as the **Newton–Raphson method**, named after Isaac Newton and Joseph Raphson, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function.
The idea is to start with an initial guess, then to approximate the function by its tangent line, and finally to compute the x-intercept of this tangent line. This x-intercept will typically be a better approximation to the original function's root than the first guess, and the method can be iterated.
The most basic version starts with a single-variable function f defined for a real variable x, the function's derivative f', and an initial guess x_0 for a root of f.

3.1 The algorithm:

The following are known:

- Function $f(x)$
- Derivative of the function $f'(x)$.
- Initial guess x_0 for a root of f.
- Tolerance ϵ

The method is as follows:

- Start with initial guess x_0 and find the derivative $f'(x_0)$.
  Then the x-intercept of the tangent would be given by:

$$f'(x_0) = \frac{y - f(x_0)}{x - x_0} \tag{2}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{3}$$

- If the function satisfies sufficient assumptions and the initial guess is close, then x_1 is a better approximation of the root than x_0.
- Similarly the $(n+1)^{th}$ approximation of the root can be found by the equation:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4}$$

- If you find an $x = x_{n+1}$ such that $|f(x)| < \epsilon$, a root of $f(x)$ is x_{n+1}.

4 Secant Method

In numerical analysis, the **secant method** is a root-finding algorithm that uses a succession of roots of secant lines to better approximate a root of a function f.
The secant method can be thought of as a finite-difference approximation of Newton's method. However, the secant method predates Newton's method by over 3000 years.

4.1 The algorithm:

The secant method can be interpreted as a method in which the derivative is replaced by an approximation and is thus a quasi-Newton method. In this subsection, the recurrence formula of the secant method has been derived from the formula for Newton's method. The following are known:

- Function $f(x)$
- Two initial values x_0 and x_1.
- Tolerance ϵ

The method is as follows:

- According to Newton's method, the x-intercept of the tangent at $(x_1, f(x_1))$ would be given by:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \tag{5}$$

- We can approximate $f'(x_1)$ as follows:

$$f'(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{6}$$

- Therefore, by Secant method:

$$x_2 = x_1 - \frac{f(x_1)}{\frac{f(x_1)-f(x_0)}{x_1-x_0}} \tag{7}$$

- Therefore, the recurrence relation for Secant method is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(x_{n-1})}(x_n - x_{n-1}) \tag{8}$$

- If you find an $x = x_{n+1}$ such that $|f(x)| < \epsilon$, a root of $f(x)$ is x_{n+1}.

5 Comparative Study

In this section are given the plots of function values against number of iterations and wall-clock time for different starting sets and the inferences that can be made from them.
Values used:

- set1 $= [3.5, 4]$
- set2 $= [3.6, 5.1]$

5.1 Variations with initial guess:

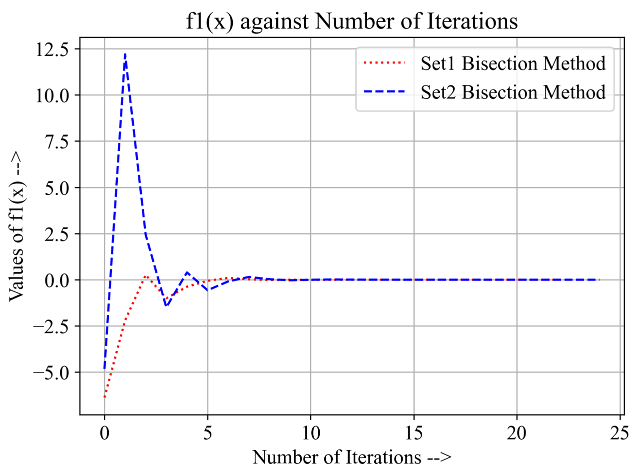

Figure 1: $f_1(x)$ against number of iterations for both starting sets(Bisection method)

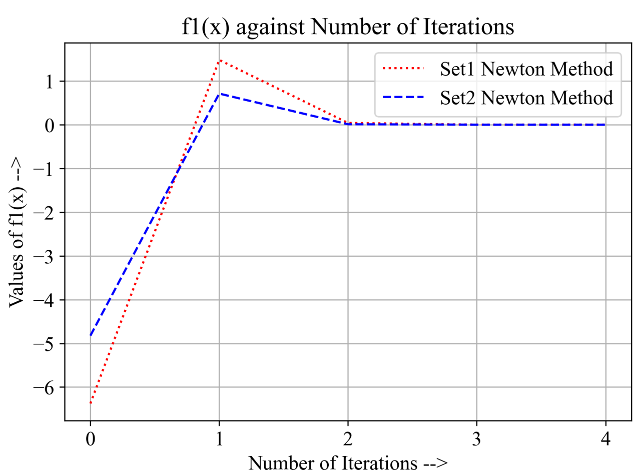

Figure 2: $f_1(x)$ against number of iterations for both starting sets(Newton Method)

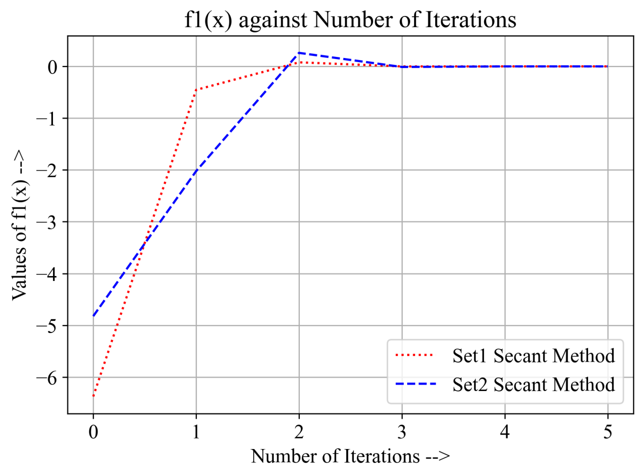

Figure 3: $f_1(x)$ against number of iterations for both starting sets(Secant Method)

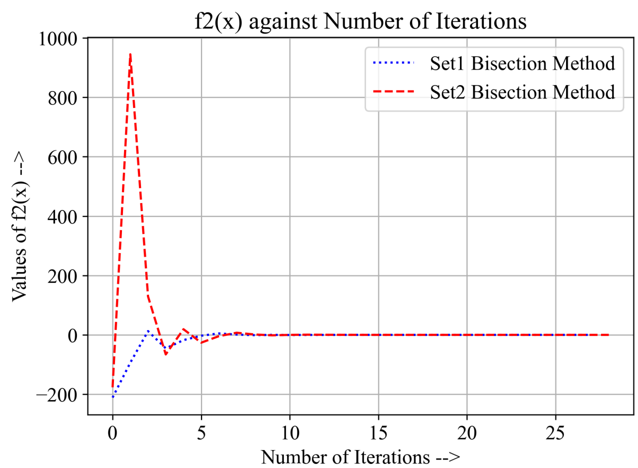

Figure 4: $f_2(x)$ against number of iterations for both starting sets(Bisection method)

From figures 1, 2, 3, 4, 5 and 6 we can clearly infer that for each method, every subsequent guess is a better approximation since, in general, the next guess is closer to the line $y = 0$. The number of iterations vary for each function as we change the initial guess. The number of iterations depend upon the nature of the function, the initial guess and the method chosen.

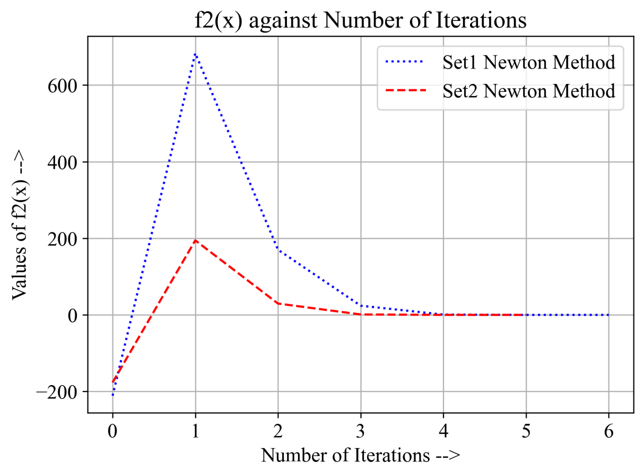

Figure 5: $f_2(x)$ against number of iterations for both starting sets(Newton Method)

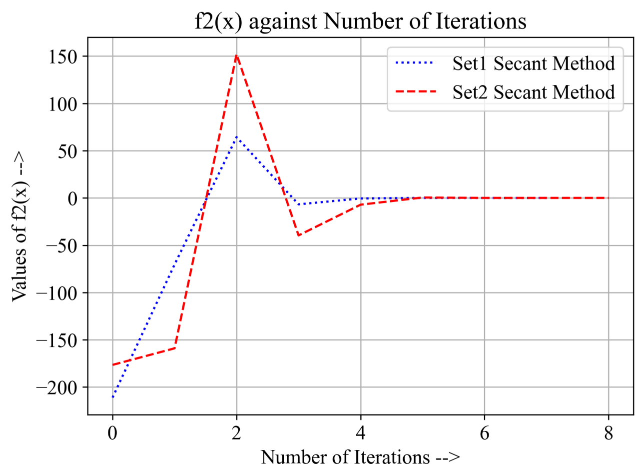

Figure 6: $f_2(x)$ against number of iterations for both starting sets(Secant Method)

5.2 Variations in number of iterations with method:

From figure 7 and figure 8 we can clearly infer that the number of iterations vary for each function as we change the method.
For both functions, **bisection method** has the slowest, **secant method** has a faster and **newton method** has the fastest convergence.
Therefore, bisection method requires the highest number of iterations while newton method requires very few iterations in general.

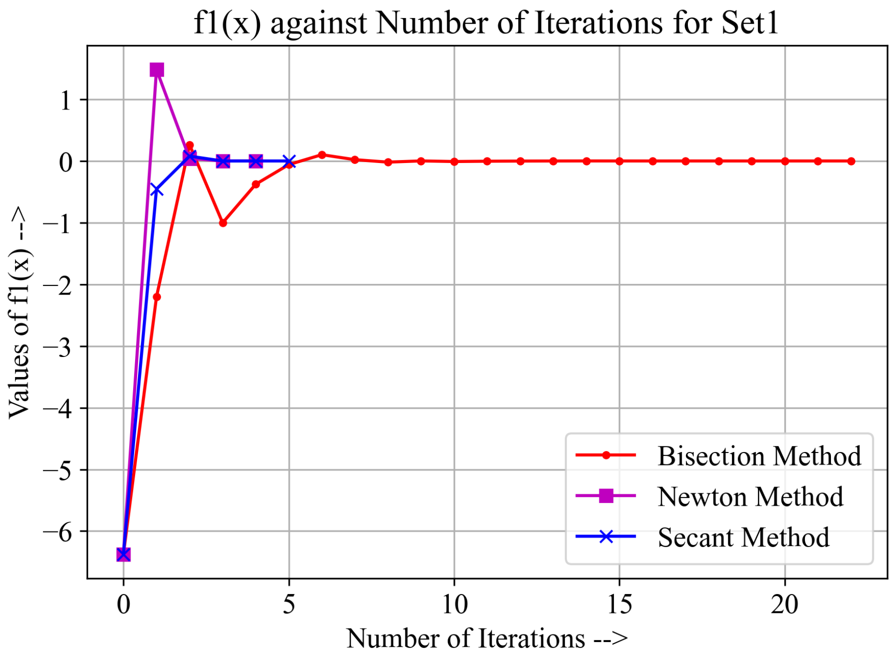

Figure 7: $f_1(x)$ against number of iterations for initial guess set1

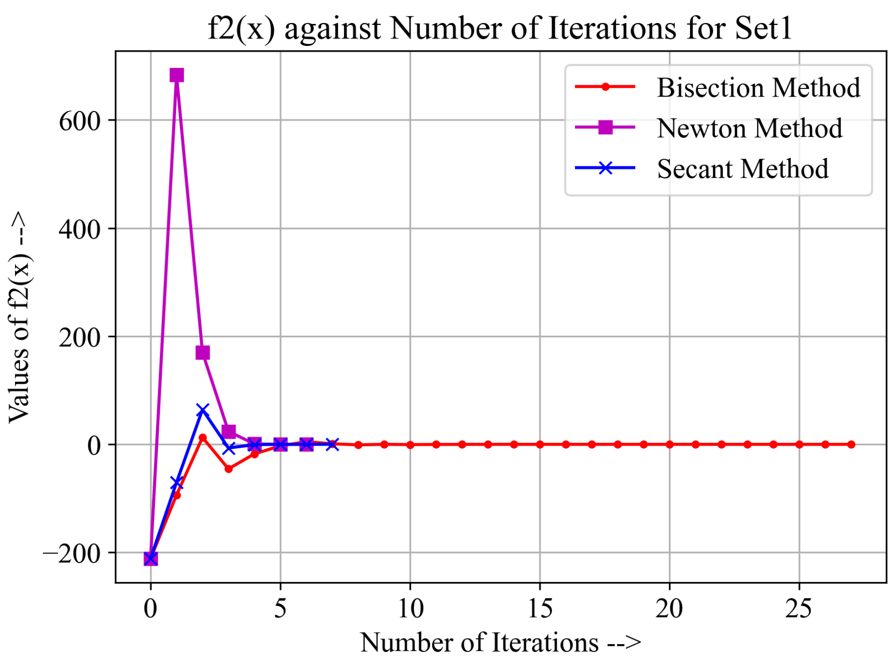

Figure 8: $f_2(x)$ against number of iterations for initial guess set1

5.3 Variations in time taken with method:

From figure 9 and figure 10 we can clearly infer that the time taken to obtain the root varies for each function as we change the method.

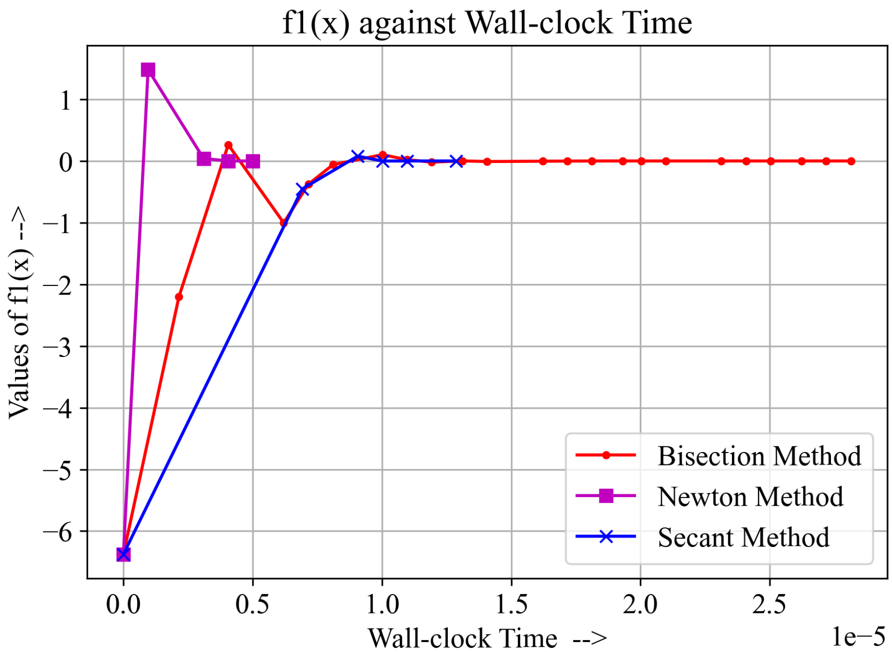

Figure 9: $f_1(x)$ against wall-clock time for initial guess set1

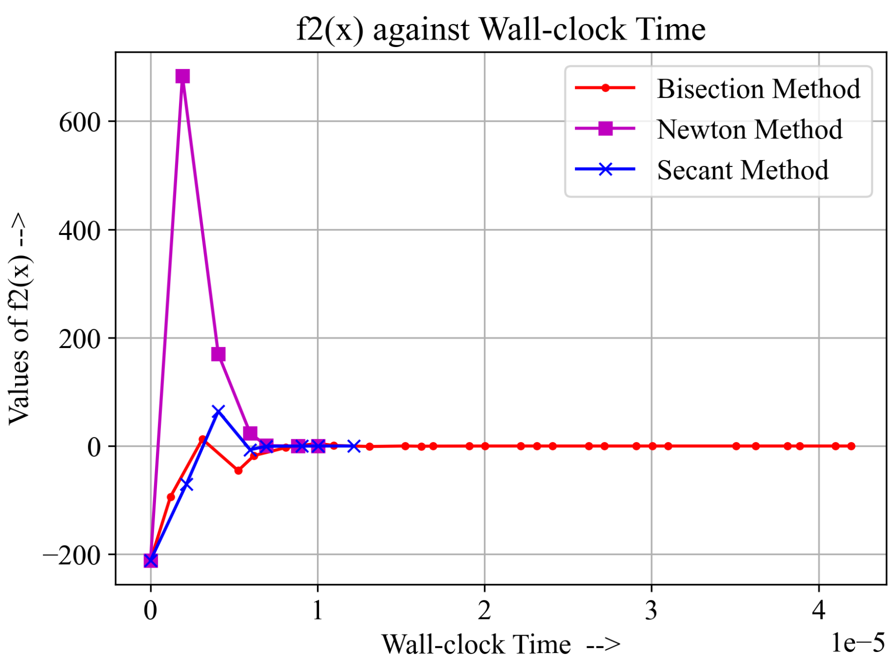

Figure 10: $f_2(x)$ against wall-clock time for initial guess set1

For both functions, **bisection method** has the slowest, **secant method** has a faster and **newton method** has the fastest convergence.

Therefore, bisection method requires the longest amount of time while newton method requires much lesser time in general.

6 Errors in result:

Analyse the function $f_3(x) = x^3 - 2x + 2$.
Upon applying Newton's method to this function, starting with initial guess $x = 0$, the result obtained, even after 10^7 iterations is 'None', that is no root found.
However, this function has a root at $x = -1.7692923542973595$.
Reason for error:

- Given: $x_0 = 0$ and $f'(x) = 3x^2 - 2$.
- Step 1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{9}$$

$$x_1 = 0 - \frac{2}{-2} \tag{10}$$

$$x_1 = 1 \tag{11}$$

- Step 2:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \tag{12}$$

$$x_2 = 1 - \frac{1}{1} \tag{13}$$

$$x_2 = 0 \tag{14}$$

- Since $x_2 = x_0$, for the initial guess $x = 0$, Newton's method runs into an infinite loop and when the maximum number of iterations is exceeded, the loop ends reporting that the function has no root.
- **CAUTION:** The initial guess for the function has to be picked with caution else the method can run infinitely without returning a root.

7 Conclusion

The 3 methods of finding roots for functions are extremely useful. They each have their own advantages and speeds with Newton's method having the fastest convergence.
However, they must be used with caution as they may return an erroneous result depending upon the properties of the function and the initial values used.
Overall, they are extremely helpful in the field of numerical analysis.