



Application of Search Algorithms to Root-Finding Problems in Algebraic and Transcendental Contexts

Bayram Köse¹, Bahar Demirtürk^{2,*}, Şükran Konca²

¹ *Electric-Electronic Engineering Department, Engineering and Architecture Faculty, Izmir Bakırçay University, Izmir, Türkiye*

² *Fundamental Sciences Department, Engineering and Architecture Faculty, Izmir Bakırçay University, Izmir, Türkiye*

Abstract. In this study, the bisection algorithm, golden ratio and Fibonacci search algorithms are used to find the roots of logarithmic, trigonometric, exponential, and polynomial functions. The results obtained from these methods are compared, and the fundamental principles, advantages, and disadvantages of each algorithm are discussed in terms of numerical stability. The approximation to the root with minimal error and the fewest steps is analyzed. The study concludes with results and recommendations based on comparative analysis.

2020 Mathematics Subject Classifications: 65H04, 65HXX, 65K10, 68W25

Key Words and Phrases: Root-finding algorithms, Optimization, Binary search algorithm, Golden ratio search algorithm, Fibonacci search algorithm

1. Introduction

An algorithm is a specific step-by-step procedure for solving a problem or completing a task. Focuses on the process and logic to arrive at a solution. Root-finding algorithms are essential tools in mathematics and computational sciences for locating the zeros of continuous functions within a given interval. The goal is to identify the value x such that $f(x) = 0$, or to approximate this root when an exact solution is not possible. These algorithms are particularly useful for solving equations of the form $g(x)=h(x)$, which can be transformed into finding the root of $f(x)=g(x)-h(x)$. In many mathematical problems that cannot be solved analytically to find an exact solution, the only way to predict the solution of this problem is to obtain values that can represent the solution of the problem by approaching the problem with numerical methods and computational science using algorithms. The root finding algorithms aim to find the minimum value of the absolute value of a function. The root search method calculates the values of the functions at various

*Corresponding author.

DOI: <https://doi.org/10.29020/nybg.ejpam.v18i3.6351>

Email addresses: bayram.kose@bakircay.edu.tr (B. Köse),
bahar.demirturk@bakircay.edu.tr (B. Demirtürk), sukran.konca@bakircay.edu.tr (Ş. Konca)

points within an interval to locate the point closest to zero, indicating the minimum of the function. Most algorithms do not guarantee finding all possible roots, and the absence of a root in the results does not necessarily imply that no roots exist and potentially may require many calculations. Therefore, more efficient and faster optimization algorithms have been developed.

Root finding plays a critical role in various engineering applications. In electrical circuits, it helps to analyze transfer functions and characteristic equations, while in control systems, it is crucial for stability analysis. Polynomial roots are also used in vibration analysis, chemical reaction modeling, and mechanical system dynamics, providing valuable information on system behavior. Furthermore, polynomial roots are essential in fields such as regression analysis, image filtering, signal correction, and data compression [1–6]. Some methods, like Newton-Raphson, use derivatives, while others work in small intervals to converge to a root. Algorithms are classified as bounded (e.g., bisection, Regula Falsi) or unbounded (e.g., fixed-point iteration). The choice of algorithm depends on the characteristics of the problem [7]. In [8], a method is proposed to compute all the roots of systems of nonlinear algebraic equations in a multidimensional interval. Moreover, a graphical interface program that includes Newton-based methods is developed and a performance analysis of the methods is also performed comparatively in [7]. In addition to these, new iteration algorithms are developed to find the roots of a given nonlinear equation using exponential series in [9]. In addition, a new algorithm that is a dynamic mixture of bisection and Regula Falsi algorithms is designed and implemented in [10]. A derivative-free root finding algorithm is proposed using the forward difference method in [11].

In this study, three derivative-free root-finding and optimization algorithms; Bisection, Golden Section and Fibonacci Search, are evaluated in terms of both convergence behavior and error analysis. These optimization techniques are used to find the roots of different type functions such as logarithmic, trigonometric, exponential, and polynomial functions. The results are analyzed to determine the most efficient approach, minimizing the steps with the least error. The basic principles, advantages, and disadvantages of each algorithm are discussed in terms of numerical stability, results, and suggestions are presented.

2. Materials and Methods

Bisection, Golden Ratio, and Fibonacci Search Algorithms are used to find the roots of exponential, logarithmic, trigonometric, and polynomial functions. MATLAB is used for the performance analysis of the algorithms in the study.

2.1. Existence of Roots

Determining the existence and number of roots is crucial, as roots form the set of solutions and validate the model. Roots indicate equilibrium points in systems and are essential for the accuracy of the model in representing real-world phenomena. Solving a system of equations reveals the meaning, validity, stability points, and set of solution points

of the model. Finding roots is also important in optimization problems and engineering applications. Let us give two important theorems in the following on the existence and finding the roots. These theorems are essential for root-approximation methods that rely on analyzing extrema and boundedness, especially in simulation-based approaches to minimization or optimization problems.

Theorem 1. (*Bolzano Theorem*) *If the function f is continuous on the interval $[a, b]$ and $f(a) \cdot f(b) < 0$, then there exists a $c \in (a, b)$ such that $f(c) = 0$ [12],[13].*

The following theorem is used in the context of analysis and should not be confused with the extreme value theorem in statistics. This theorem was first proved partially by Bolzano. Before giving Theorem 2, the following nuance should be noted. Bolzano proved that a continuous function is bounded on a closed interval, but he did not guarantee that the function can reach extreme values. This gap was filled by Weierstrass, who showed that a continuous function defined on a closed and bounded interval reaches both its minimum and maximum value. Therefore, the following theorem is called a theorem proved by Bolzano-Weierstrass.

Theorem 2. (*Extreme Value Theorem*) *Let $f : I \rightarrow \mathbb{R}$ be a continuous function on a closed and bounded interval $I = [a, b] \subset \mathbb{R}$. In the interval $[a, b]$ there exist such real numbers c and d such that $f(c) \leq f(x) \leq f(d)$, $\forall x \in [a, b]$. That is, there exist numbers $c, d \in I$ such that $f(c) = \inf f(I) = \min f(I)$ and $f(d) = \sup f(I) = \max f(I)$. That is, a continuous function takes its maximum and minimum values in a closed interval. In other words, the image of a closed and bounded interval under a continuous function is also a closed and bounded interval. Since boundedness and continuity are equivalent in the space of real numbers, it is possible to express this as follows: A continuous function f defined on the interval $[a, b]$ is bounded on this interval. That is, there exist such real numbers m and M such that $m \leq f(x) \leq M$ for $\forall x \in [a, b]$ in [12],[13].*

Theorem 3. *If the function f is continuous, the function $|f|$ is also continuous [12].*

We define and prove the following theorem because of the basic theorems above. This theorem is used in MATLAB calculations throughout the paper as the root search method in the optimization algorithm.

Theorem 4. (*Root Search in Optimization Algorithm*) *For $I = [a, b] \subset \mathbb{R}$, if the function $f : I \rightarrow \mathbb{R}$ is continuous, it has at least one minimum in this interval, and if $|f(x_i)| = 0$, there is at least one $x_i \in I$, ($i \in \mathbb{N}$) satisfying this equality.*

Proof. From the Extreme Value Theorem, if the function is continuous, it has at least one minimum. If $|f(x_i)| = 0$, then $f(x_i) = 0$. Since this continuous function f has at least one minimum in the given closed-limited interval, and since $|f(x_i)| = 0$, this minimum takes its zero value at the roots $x_i \in I$, $i \in \mathbb{N}$, in the given interval. Thus, there is at least one $x_i \in I$ satisfying the equality $f(x_i) = 0$.

2.2. Optimization Approaches For Finding the Roots

Optimization aims to find the best value of an objective function based on certain criteria, and optimization methods use numerical techniques to find the roots of equations. Root-finding typically involves an iterative process in which candidate root values are updated until the objective function reaches its minimum or maximum or a set tolerance is met. Root-finding algorithms, such as bisection, golden ratio, and Fibonacci search, are common optimization techniques that iteratively identify and refine candidate roots, quickly and accurately locating the solution of the equation. In this section, the mathematical foundations, accuracy levels, and error bounds of these algorithms are presented.

Consider a function $f(x)$ in one variable. Suppose that we want to solve the following optimization model

$$\text{Max} f(x)$$

$$a \leq x \leq b.$$

For some x , $f'(x)$ may not exist or it may be very difficult to solve the equation $f'(x) = 0$. How can the optimal value of the function $f(x)$ be found if the given interval contains only one vertex? For the answer to this question, the following optimization techniques can be analyzed. Root-finding methods applied to both a function and its first derivative help to identify critical points. For multimodal functions, these points may represent local maxima or minima, while for unimodal functions the critical point indicates a potential absolute maximum or minimum. The methods considered in this study are constructed as derivative-free methods that do not require knowledge of the derivative of the function. The extreme value theorem and Bolzano's theorem, which support the theoretical background of the paper, are used to support the process of reaching the minimum value with simulations, making the mathematical analysis more concrete. The mean absolute error (MAE) and the mean absolute approximation error (MAAE) calculated in the study are used to analyze the precision and error of the methods, revealing which method is more advantageous under which conditions. In this way, it has been prepared as a source of guidance for researchers on method selection for both theoretical and applied research.

2.2.1. Bisection Algorithm (BM)

The bisection method is a simple and efficient numerical method based on the existence of a root for a continuous function $f(x)$ in the interval $[a, b]$ satisfying the condition $f(a) \cdot f(b) < 0$ given in the Bolzano Theorem, by Theorem 1. The method iteratively approaches the root by dividing the interval in half at each step. This algorithm shows linear convergence and the error is halved with each iteration.

This algorithm follows the following steps: There exist points a and b such that $a < b$ and $f(a) \cdot f(b) < 0$. Then calculate the midpoint of points a and b and call this point t . Let $t = (a + b)/2$. If $f(t) = 0$, t is the root of the given function. If not, proceed to the next step. Then divide the interval $[a, b]$ by two. If $f(t) \cdot f(a) < 0$ then there is a root between a and t , otherwise if $f(t) \cdot f(b) < 0$ then there is a root between t and b . Continue the above steps until $f(t) = 0$.

The error bound after the n -th iteration is given by

$$x_n = \frac{a+b}{2}, \quad |x_n - x_r| \leq \frac{b-a}{2^n}, \quad (1)$$

where x_r denotes the true root and n is the number of iterations. The minimum number of iterations required to achieve a tolerance ε satisfies

$$\frac{\log(b-a) - \log \varepsilon}{\log 2} \leq n. \quad (2)$$

This method is highly stable, but exhibits a slow convergence rate, although it guarantees convergence [14]. However, it cannot be applied to discontinuous functions, and it fails to start if

$$f(a)f(b) \geq 0.$$

Let us give the flow diagram for the bisection algorithm in Figure 1.

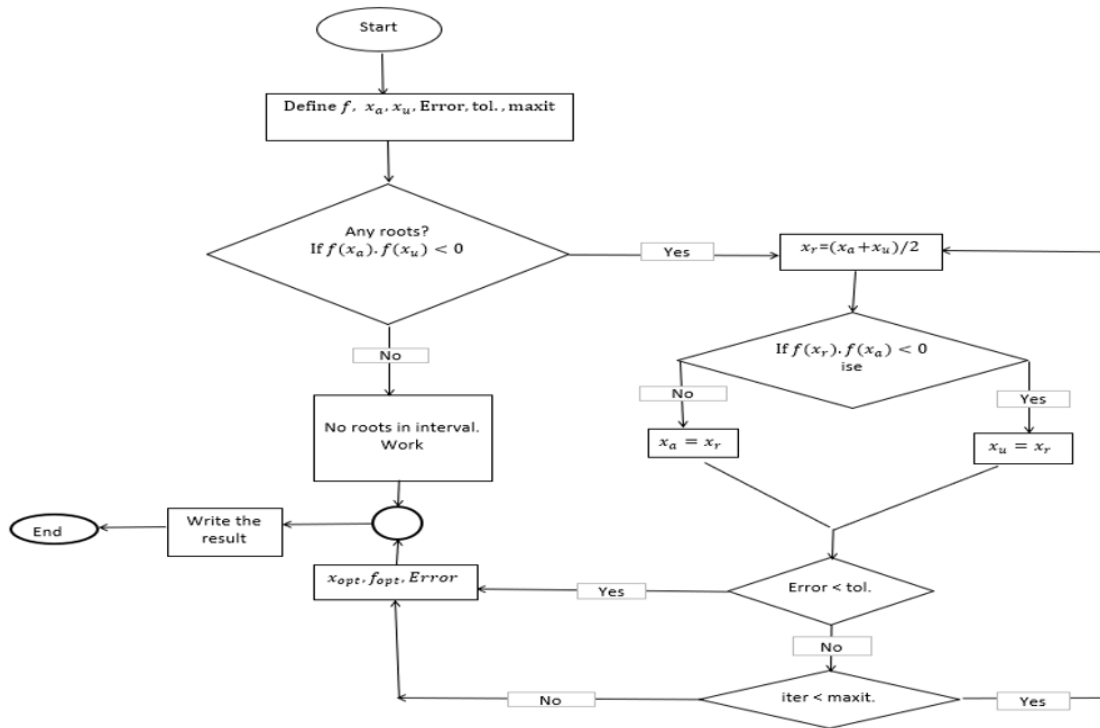


Figure 1: The flowchart of the BM algorithm [15]

We suggest consulting [14–17] for more details on this algorithm.

2.2.2. Golden Ratio Search Algorithm (GR)

The golden ratio search method is a reduction method used to find the minimum value, especially for unimodal functions. The golden ratio is a special ratio found in nature,

art, and architecture, often considered aesthetically harmonious. It appears in the human body, plants, and shells. This ratio, discovered by ancient Egyptians and Greeks, divides a line so that the ratio of the smaller part to the larger part is the same as the ratio of the larger part to the whole line. According to Plato, this ratio is the key to cosmic physics.

Let the length of the interval $[x_a, x_u]$ be l_0 . Divide this interval into l_1 and l_2 subintervals such that

$$l_0 = l_1 + l_2 \quad \text{and} \quad \frac{l_1}{l_0} = \frac{l_2}{l_1}.$$

From

$$\frac{l_1}{l_0} = \frac{l_2}{l_1} \implies \frac{l_1}{l_1 + l_2} = \frac{l_2}{l_1}.$$

Taking $\frac{l_2}{l_1} = R$ in the above equation, we have $1 + R = \frac{1}{R}$. Thus, we get the characteristic equation

$$R^2 + R - 1 = 0.$$

The positive root of this equation is the golden ratio $R = \frac{\sqrt{5}-1}{2} = 0.61803\dots$

The golden ratio is also related to Fibonacci numbers, an important mathematical series:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

The Fibonacci series given by the recurrence relation

$$F_{n+1} = F_n + F_{n-1}, \quad \forall n \geq 1$$

with the initial conditions $F_0 = 0$ and $F_1 = 1$. F_n is called the n -th Fibonacci number. The ratio of the consecutive Fibonacci numbers are:

$$0/1 = 0, \quad 1/1 = 1, \quad 1/2 = 0.5, \quad 2/3 \approx 0.667, \quad 3/5 = 0.6, \quad 5/8 = 0.625, \quad 8/13 \approx 0.615, \dots$$

continued in this way, eventually the ratio of consecutive Fibonacci numbers reaches the golden ratio:

$$\lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_n} \approx 0.61803\dots$$

Consider an interval $[x_a, x_u]$, where x_a is the lower bound and x_u is the upper bound, and the function is unimodal and has only one maximum. In the bisection algorithm, the interval is divided in half, while the golden ratio search divides it according to the golden ratio. This method requires three function values, so a third point must be selected. After selecting a fourth point, a test determines if the maximum lies between the first three or the last three points. The advantage of the golden ratio search is that, since the original points x_1 and x_2 are selected using the golden ratio, function values don't need to be recalculated in each iteration, and a wider range is explored compared to bisection. Besides the advantages, the failure of this method is that the function $f(x)$ must be unimodal.

Let the function $y = f(x)$ be continuous in the interval $[x_a, x_u]$ while x_a is the lower bound, x_u is the upper bound, and the points x_1 and x_2 are chosen according to the golden ratio rule as follows: Let

$$x_1 = x_a + \frac{\sqrt{5}-1}{2} (x_u - x_a) \quad \text{and} \quad x_2 = x_u - \frac{\sqrt{5}-1}{2} (x_u - x_a).$$

Firstly, calculate $f(x_1)$ and $f(x_2)$. If $f(x_1) < f(x_2)$, then the search for roots in the interval $[x_a, x_1]$ continues by setting $x_u = x_1$. If $f(x_2) < f(x_1)$, then the search in $[x_2, x_u]$ continues with setting $x_a = x_2$.

This method recalculates only a fraction of new search points by reducing the interval at a constant rate $R = \frac{\sqrt{5}-1}{2} \approx 0.61803$, thus reducing the computational cost. The convergence behavior of this method is expressed as follows: if

$$|I_n| = R |I_{n-1}|,$$

then

$$|I_n| = R^n |I_0|,$$

where I_0 is the initial interval and I_n is the interval obtained at the n -th iteration.

The minimum number of iterations required to reach a tolerance ε is

$$\frac{\log(\varepsilon/(b-a))}{\log R} \leq n.$$

For some functions, the golden ratio method can achieve similar accuracy levels with fewer iterations compared to the bisection method. This method is often preferred in the optimization process both to reduce the number of operations and to provide high accuracy [18].

Now let us give the flow chart for the golden ratio Algorithm in Figure 2.

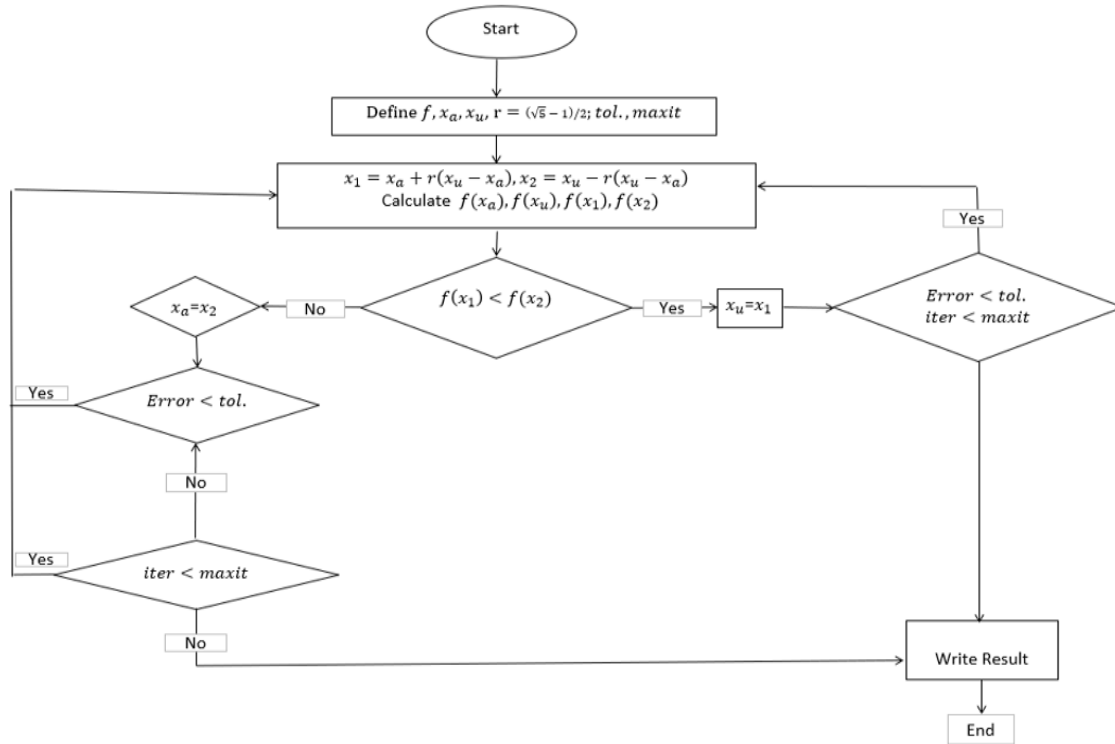


Figure 2: The flowchart of GR algorithm

For more details on this algorithm, we may recommend referring to [19] and [20].

2.2.3. Fibonacci Search Algorithm (FIB)

Although the Fibonacci search method is structurally similar to the golden section, it determines the number of iterations from the beginning and ensures that the algorithm is completed in a certain number of steps. At each iteration, the range is narrowed by Fibonacci numbers. This approach offers high precision in finite iterations. The interval minimization rate is

$$|I_n| = \frac{F_{n-k}}{F_n} |I_0|,$$

where I_0 is the initial interval, I_n is the interval obtained at the n -th iteration and F_n is the n -th Fibonacci number. Moreover, the minimum Fibonacci term F_n required for the desired margin of error ε is found from the equality

$$\frac{b-a}{\varepsilon} \leq F_n.$$

Since Fibonacci numbers need to be calculated in each iteration, this method requires more preparation in terms of applicability. Besides the advantages, the failure of this method is that the function $f(x)$ must be unimodal and continuous.

Let us give the flowchart of the Fibonacci search Algorithm below.

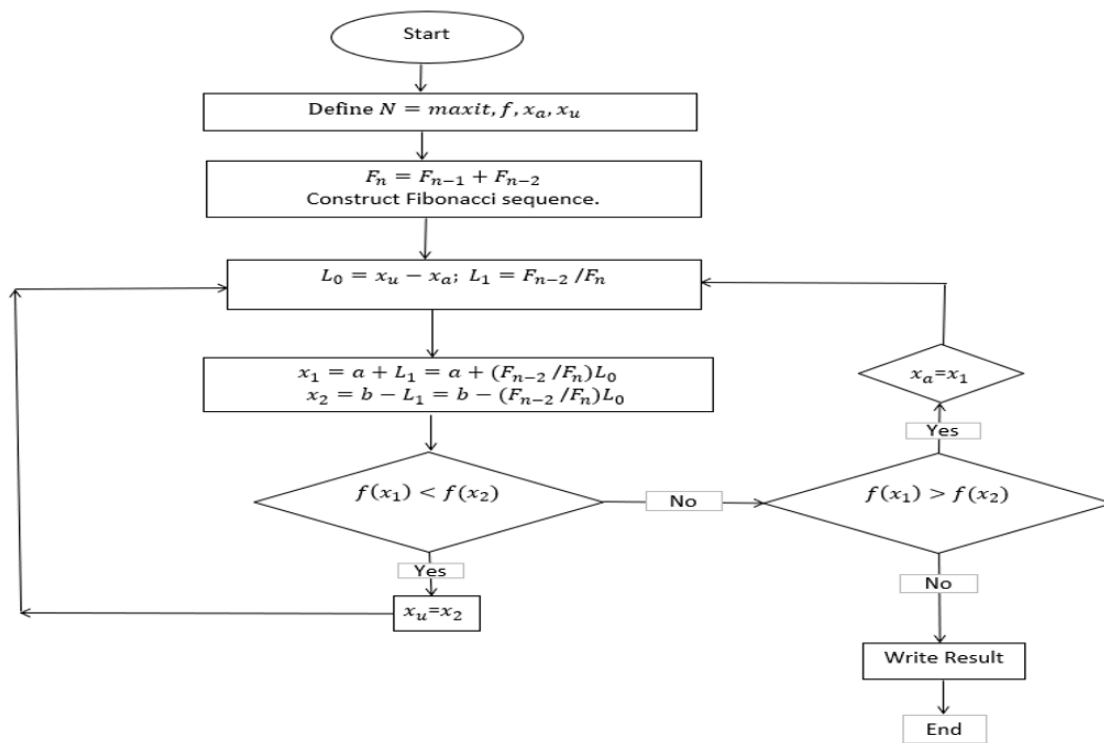


Figure 3: FIB Algorithm Flow Chart

For further information on this algorithm, we recommend referring to [18, 21].

3. Performance Criteria and Results

3.1. Test Functions and Related Graphics

In the comparative analysis to evaluate the performance of the algorithms used to find the roots of the four different types of functions considered in this study, we took into account both the achievement of the result and the behavior of the algorithms and their responses to the conditions. Therefore, firstly, the number of roots and the sensitivity to the root location are evaluated by analyzing which root different algorithms converge to on the same function. In addition, the number of iterations required for the algorithms to reach a certain error tolerance and their convergence speed are analyzed comparatively. The distances of the obtained roots from the true root, i.e. the approximation errors, are important because they reveal the accuracy of the methods. Furthermore, the sensitivity to initial conditions was measured by testing how small changes to the initial interval affect the algorithm's result. In addition, in future studies, different types of functions can be used to analyze the response of algorithms to the functional structure. Such analyzes

will reveal both the success rates of the algorithms and the types of problem for which they are more suitable.

In this study four univariate and nonlinear functions are tested using the BM, GR and FIB search algorithms to find their roots. The test functions were evaluated over the specified intervals as follows:

$$\begin{aligned} f_1 &= \log(x^2 + 1) + \sin(x - 1), [-2.5, 2.5] \\ f_2 &= \sin(x) - x^2/8, [-4.5, 4.5] \\ f_3 &= e^{-x} - x, [-2.5, 2.5] \\ f_4 &= x^3 - 13x + 12, [-4.5, 4.5] \end{aligned}$$

Here we can explain what we considered when choosing these four functions.

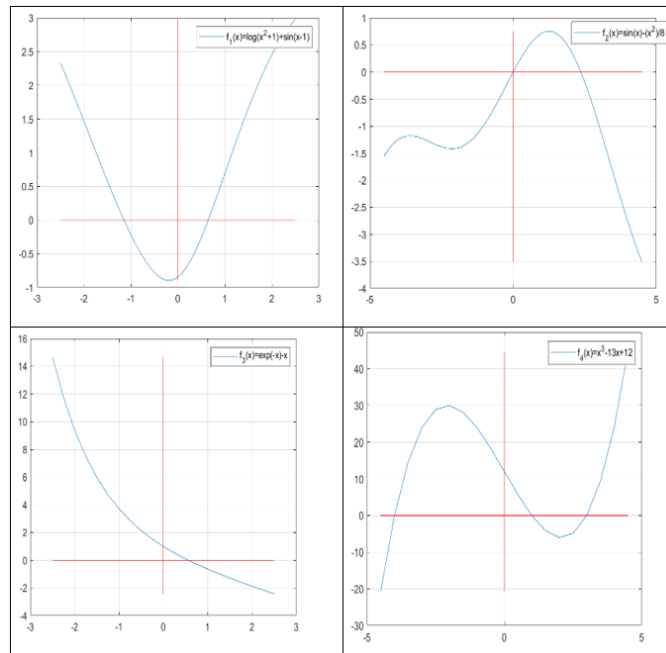
The function f_1 consists of both a logarithmic and a periodic function and contains two roots. This function is considered to compare the convergence speed of the algorithms in particular GR and FIB algorithms.

The function f_2 contains both periodic and linear components and has more than one root. The aim is to analyze which root the algorithms tend towards in the selection of this function.

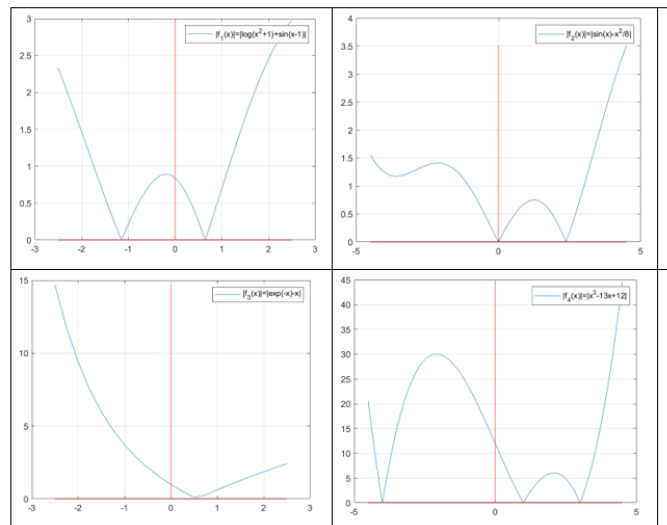
The reason for choosing the function f_3 to contain exponential and polynomial expressions is the idea that this function contains both fast-changing and constant-behaving terms. Thus, it is aimed to compare the ability of the algorithms to automatically change strategies against the selected function structure.

Although the function f_4 appears to be a simple polynomial function, it is actually a function with three roots and opposite slope. Such functions reveal the algorithms' sensitivity, stability, and sensitivity to the initial interval.

The graphics of these functions are shown in Figure 4.

Figure 4: The graphics of the functions f_1, f_2, f_3 and f_4 .

The graphics of the absolute values of these functions are shown in Figure 5.

Figure 5: The graphics of the functions $|f_1|, |f_2|, |f_3|$ and $|f_4|$.

3.1.1. Graphics of Approaching Minimum Value with BM Algorithm

The graphics of $|f_1|, |f_2|, |f_3|$ and $|f_4|$ approaching the minimum value with the bisection algorithm are shown in Figure 6.

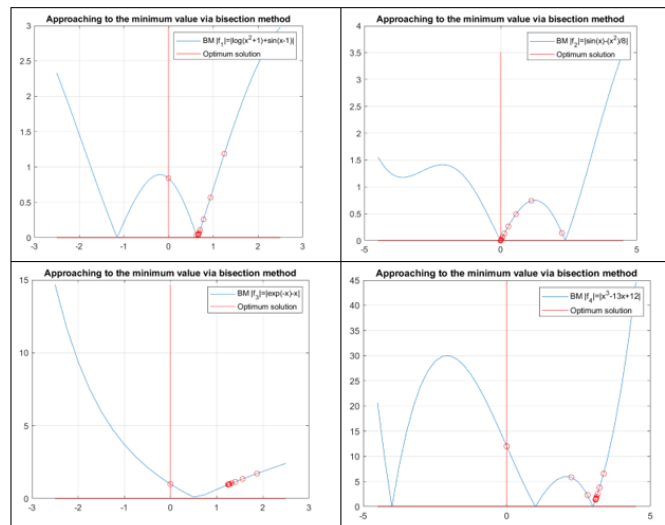


Figure 6: The graphics of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ approaching the minimum value with the BM algorithm

Moreover, Figure 7 shows the rate of convergence of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ to the optimum value using the BM Algorithm with the number of iterations.

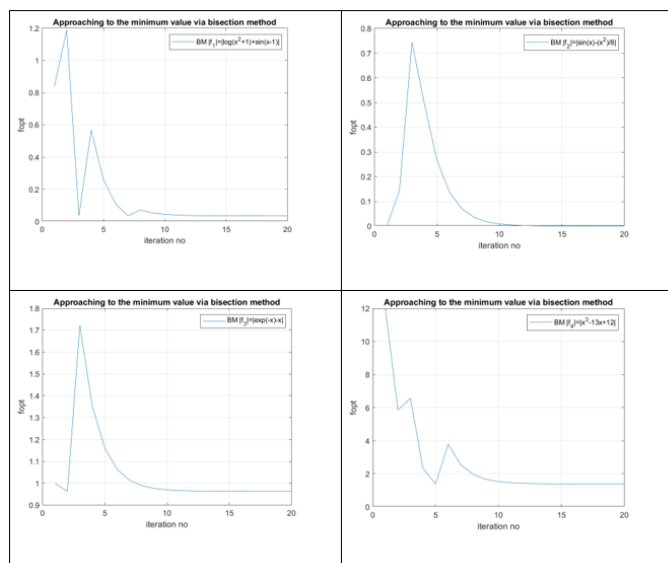
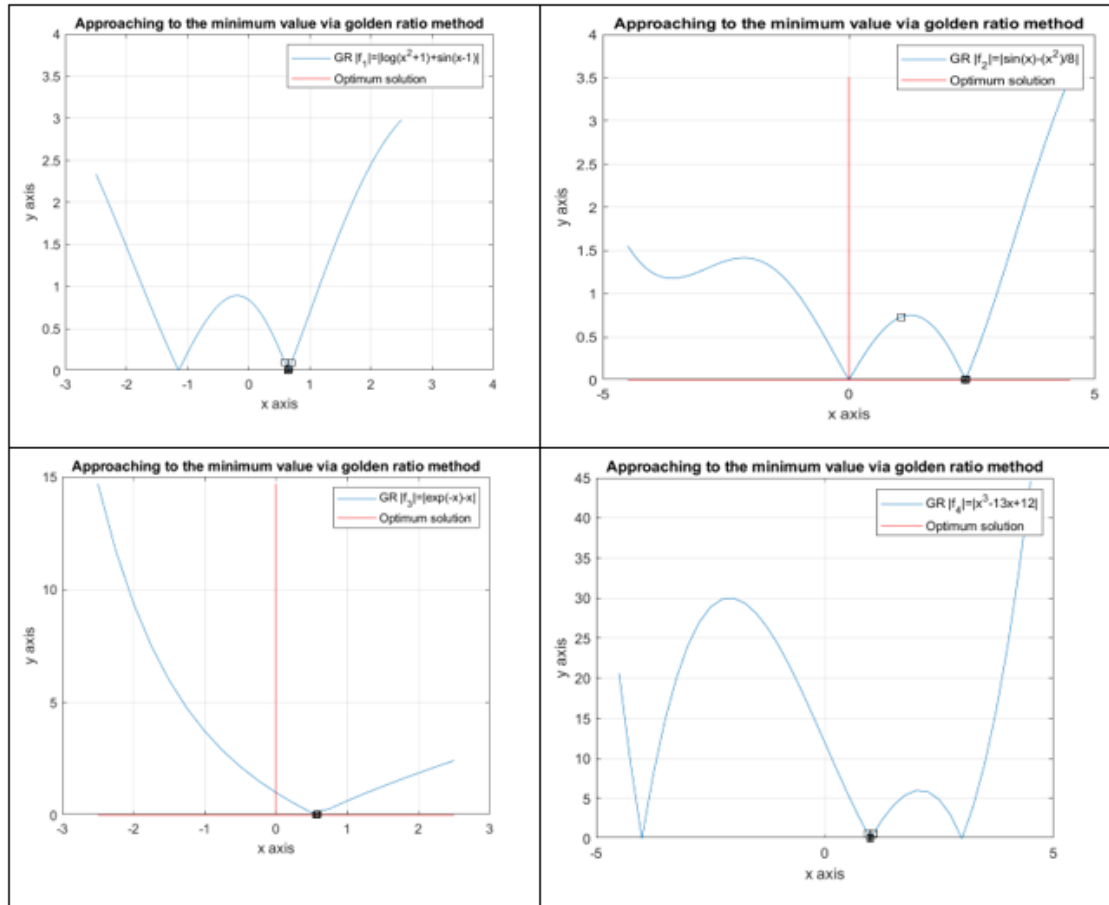


Figure 7: The rate of convergence of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ to the optimum value using the BM Algorithm (with the number of iterations).

3.1.2. Graphics of Approaching the Minimum Value with the GR Algorithm

The graphics of $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ approaching the minimum value with the bisection algorithm are shown in Figure 8.

Figure 8: Graphics of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ approaching the minimum value with the GR algorithm

In Figure 9, the speed of approaching the minimum value of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ using the GR Algorithm is given by the number of iterations.

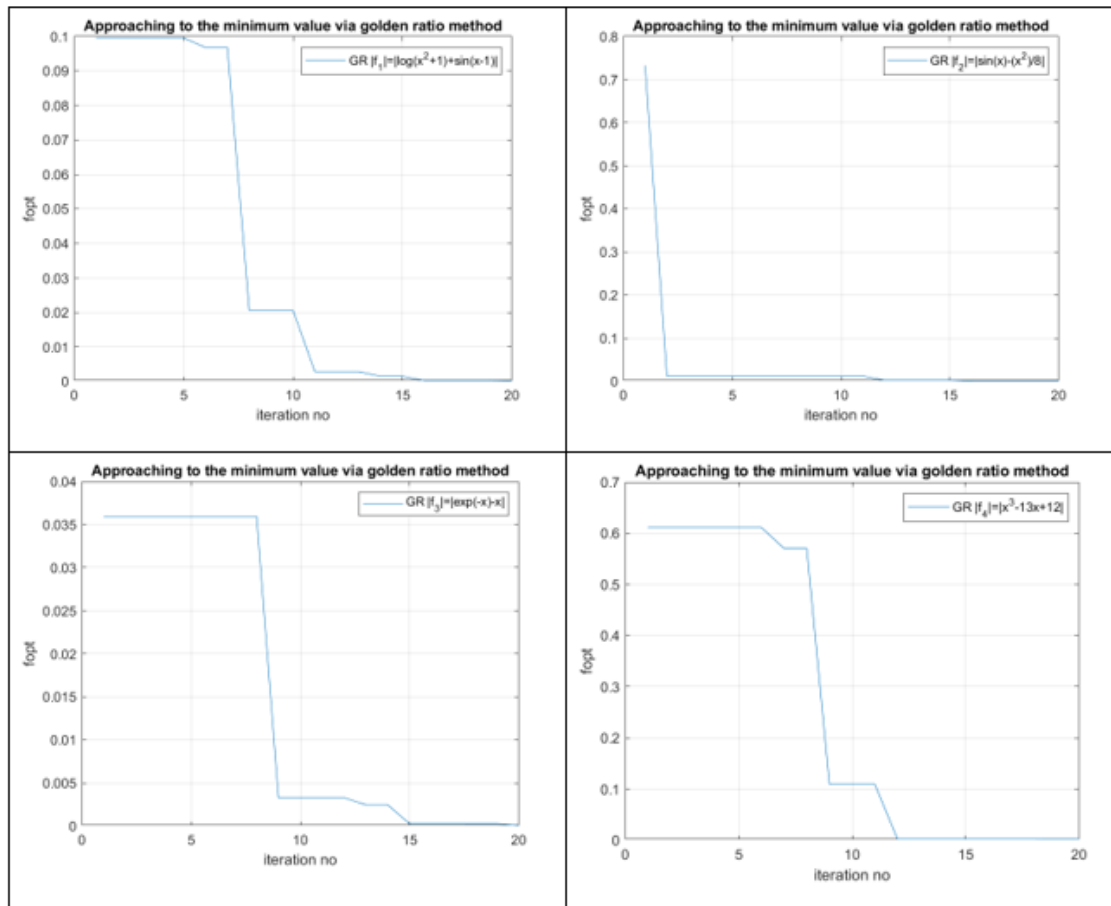


Figure 9: The speed of approaching the minimum value of functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ using the GR algorithm with the number of iterations

3.1.3. Graphics of Approaching the Minimum Value with FIB Algorithm

The graphics of $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ approaching the minimum value with the bisection algorithm are shown in Figure 10.

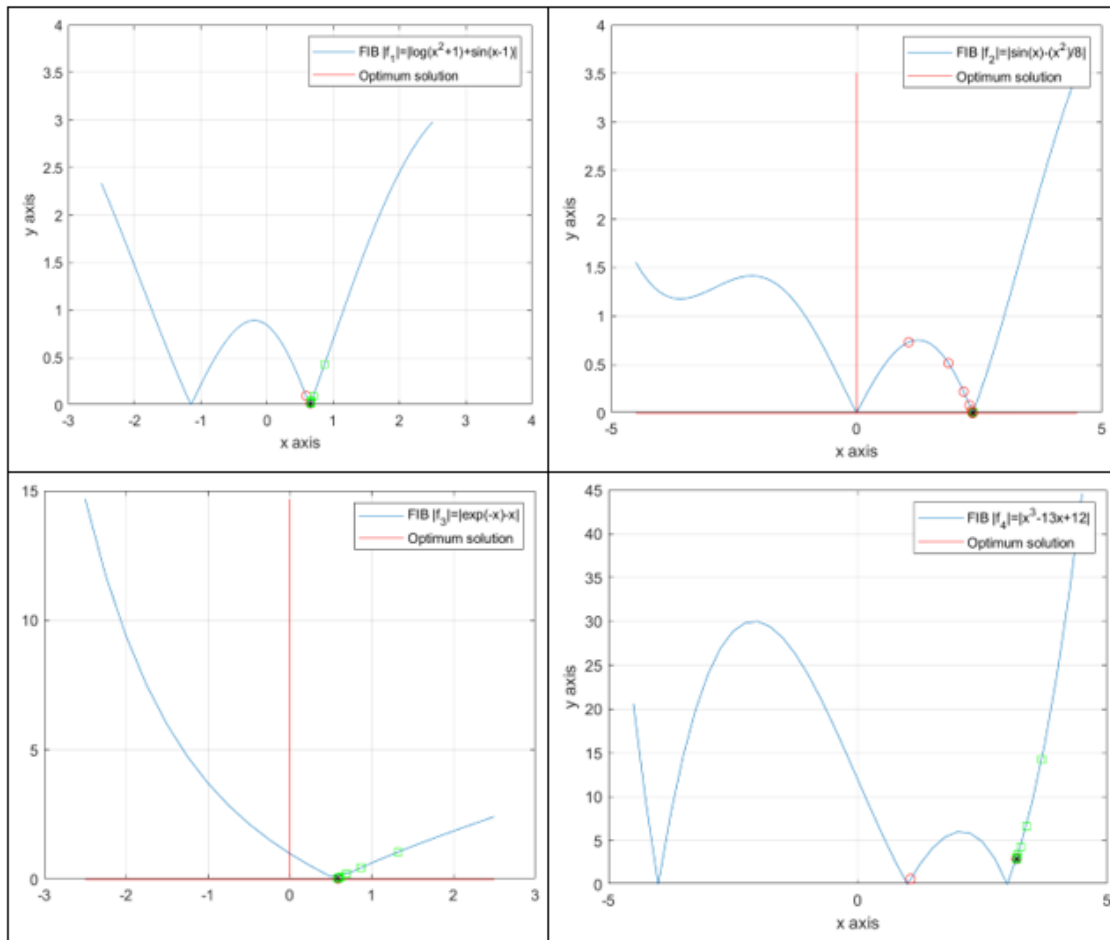


Figure 10: Graphics of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ approaching the minimum value with the FIB algorithm

Figure 11 shows the rate of convergence of functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ to the optimal value using the FIB Algorithm with the number of iterations.

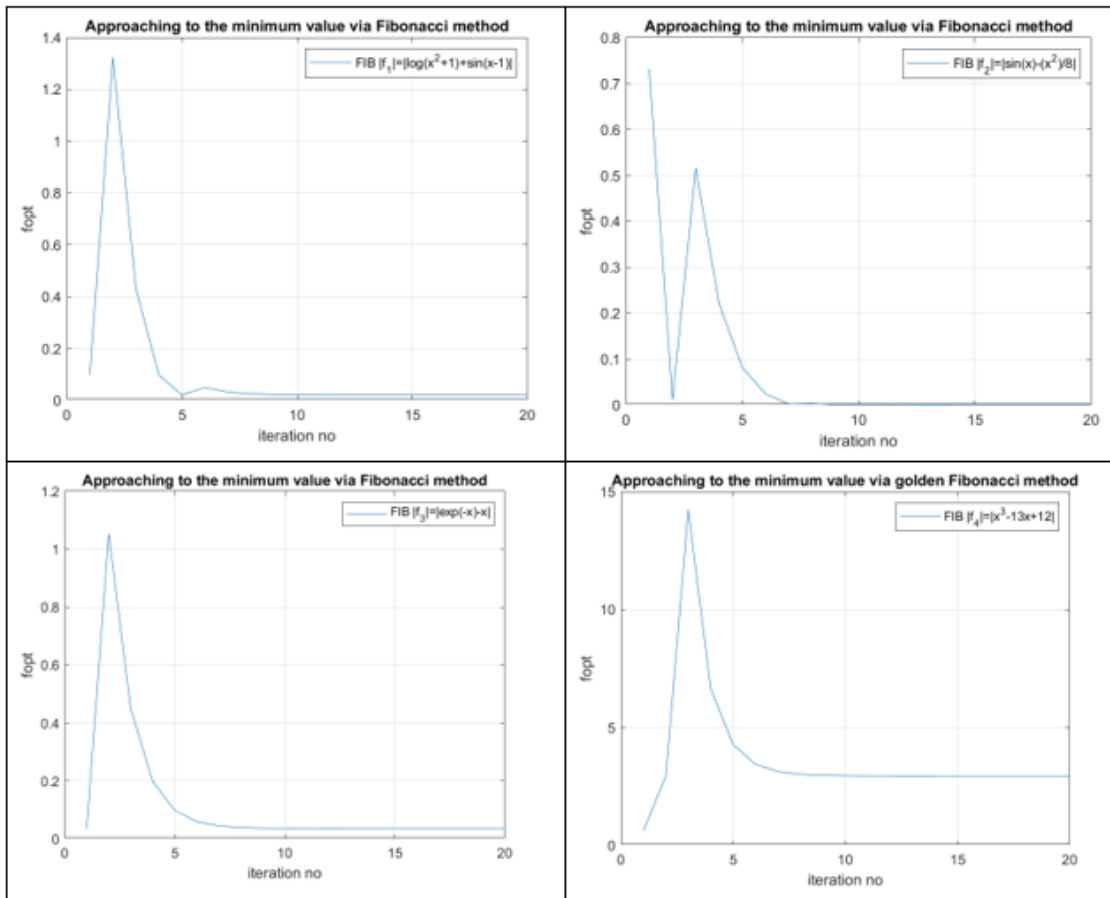


Figure 11: The speed of approaching the minimum value of the functions $|f_1|$, $|f_2|$, $|f_3|$ and $|f_4|$ using the FIB algorithm with the number of iterations

3.2. Error Calculations in Performance Evaluation

The problem of finding the root of algebraic equations is crucial in numerical analysis and computer science. The approximate values obtained by numerical and optimization methods and the difference between these values and the actual values, i.e. the error values, are of critical importance. In this section, the equations representing the errors are discussed and finally the computational results of these errors are presented for the performance evaluation of the algorithms.

3.2.1. Mean absolute error (MAE)

The mean absolute error (MAE) is frequently used in regression and time series problems because of its easy interpretation. MAE is a measure of the difference between two continuous variables, i.e. the sum of the absolute values of the differences between the actual and predicted values, and measures the magnitude of deviations between predictions and

actual values. The formula of mean absolute error is given by

$$MAE = \frac{1}{n} \sum_{i=0}^n |x_g - x_i|. \quad (1)$$

In Equation (1), x_g, x_i and n is defined as;

x_g : real root,

x_i : approximate root obtained in step i ,

n : number of the total iteration [22].

3.2.2. Mean absolute approximation error (MAAE)

The mean absolute approximation error (MAAE) measures the accuracy of approximations when the true value is unknown but can be estimated. It is calculated as the average of the absolute values of the differences between the values obtained at each step of an iteration and the values from the previous step. This error measure is widely used in numerical analysis, optimization, and engineering to assess the accuracy and stability of estimates. MAAE is particularly useful in computational and modeling processes, where it helps evaluate the performance of iterative algorithms. In addition, it is used in scientific research to determine the reliability of the data by measuring the insignificance of the differences between the measurements. This improves the accuracy and reliability of the results. The mean absolute error of the approximation is given in Equation (2)

$$MAAE = \frac{1}{k} \sum_{i=0}^k \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| \quad (2)$$

with x_i : approximate root obtained in step i , and k : number of iterations with minimum error.

Throughout this study, while calculating the mean absolute approximate error, since the root x_{i+1} in the denominator of Equation (2) is zero in some cases, this term will be ignored in order not to be undefined, and the mean absolute approximate error will be calculated with the formula given by Equation (3), given in [22].

$$MAAE = \frac{1}{k} \sum_{i=0}^k |x_{i+1} - x_i|. \quad (3)$$

3.3. Analysis and Comparison

Since it is necessary to consider both error magnitudes and error reduction rates when evaluating the effectiveness of the algorithm, especially for complex problems and large data sets, simulations are performed for the test functions selected here. An efficient algorithm is compared with the test set with the values produced by acceptable and evaluated with the performance metrics Mean Absolute Error (MAE) and Mean Absolute Approximation Error (MAAE). Since it is expected to reach a level of precision using a

minimum of iterations and computational resources, the following evaluations are made with the performance metrics of the algorithms' estimates. The rate of convergence to the minimum value, optimum points and global minimum values of the nonlinear test functions used in the test phase are given in the table below, and absolute errors are calculated.

| Test Functions | Iterations, roots, values, error calculations | BM | GR | FIB |
|--|--|----------|----------|----------|
| f_1 Real Roots $x_1 = -1.15$ $x_2 = 0.6449$ | i | 7 | 20 | 13 |
| | x_{f_1} | 0.6449 | 0.6449 | 0.6449 |
| | $optx_{f_1}$ | 0.6641 | 0.6448 | 0.6559 |
| | $opty_{f_1}$ | 0.0357 | 0.0001 | 0.0206 |
| | Absolute error | 0.0008 | 0.0001 | 0.0110 |
| | Mean absolute error | 0.10325 | 0.0208 | 0.0604 |
| | Mean absolute error depending on the number of iterations | 0.2538 | 0.0208 | 0.0869 |
| | Mean absolute approximation error | 0.1250 | 0.0080 | 0.1578 |
| | Average absolute approximation error depending on the number of iterations | 0.7087 | 0.6545 | 0.2377 |
| | | | | |
| f_2 Real Roots $x_1 = 0$ $x_2 = 0.2375$ | i | 1 | 17 | 15 |
| | x_{f_2} | 0 | 0 | 0 |
| | $optx_{f_2}$ | 0 | 2.3663 | 2.3663 |
| | $opty_{f_2}$ | 0 | 0 | 0 |
| | Absolute error | 0 | 2.3663 | 2.3663 |
| | Mean absolute error | 0.2249 | 2.3053 | 2.2636 |
| | Mean absolute error depending on the number of iterations | 0 | 2.2946 | 2.2294 |
| | Mean absolute approximation error | 0.2250 | 0.0663 | 0.3433 |
| | Average absolute approximation error depending on the number of iterations | 4.5 | 1.3856 | 0.3708 |
| | | | | |
| f_3 Real Root $x_1=0.5693$ | i | 2 | 20 | 15 |
| | x_{f_3} | 0.5693 | 0.5693 | 0.5693 |
| | $optx_{f_3}$ | 1.2500 | 0.5671 | 0.5902 |
| | $opty_{f_3}$ | 0.9635 | 0.0001 | 0.0359 |
| | Absolute error | 0.6807 | 0.0022 | 0.0209 |
| | Mean absolute error | 0.7376 | 0.0099 | 0.0799 |
| | Mean absolute error depending on the number of iterations | 0.6250 | 0.0099 | 0.0996 |
| | Mean absolute approximation error | 0.1250 | 0.0015 | 0.1545 |
| | Average absolute approximation error depending on the number of iterations | 1.8750 | 0.6545 | 0.2060 |
| | | | | |
| f_4 Real Roots $x_1 = -4$ $x_2 = 1$ $x_3 = 3$ | i | 17 | 19 | 17 |
| | x_{f_4} | 3 | 1 | 3 |
| | $optx_{f_4}$ | 3.0938 | 1.0002 | 3.1869 |
| | $opty_{f_4}$ | 1.3926 | 0.0017 | 2.9378 |
| | Absolute error | 0.0938 | 0.0002 | 0.1869 |
| | Mean absolute error | 0.304687 | 0.026017 | 0.315033 |
| | Mean absolute error depending on the number of iterations | 0.3419 | 0.0274 | 0.3376 |
| | Mean absolute approximation error | 0.2250 | 0.0087 | 0.3843 |
| | Average absolute approximation error depending on the number of iterations | 0.5294 | 1.240 | 0.3272 |
| | | | | |

Table 1: Statistics of the test functions

In Table 1, i is the number of iterations, f_i are the test functions for $1 \leq i \leq n$, x_{f_i}

are the true roots of the f_i functions, $optx_{f_i}$ are the optimal points of the f_i functions and $opty_{f_i}$ are the optimal values of the f_i functions. In Table 1, various statistics such as best solution, optimum solution, absolute error, average absolute error, average absolute error, average absolute approximation error, average absolute error depending on the number of iterations and average absolute approximation error depending on the number of iterations are presented for the BM, GR and FIB algorithms. For the function f_1 , the BM achieves the lowest absolute error value (0.00008) with the least number of iterations (7 iterations), while the GR method has the lowest average absolute error (0.0208), while the FIB method has an intermediate performance (0.0604). In this case, considering the number of iterations and the average absolute error, it is shown that the BM can produce faster and more accurate results. Again, when the average absolute errors depending on the number of iterations of the function f_1 are compared, it is seen that the GR method produces more precise results. When the average absolute approximation errors are considered, it is seen that the GR method gives the most successful result. It is also seen that the FIB method gives more precise results in the relative average absolute approximation error according to the number of iterations.

For the function f_2 , the GR method has the highest average absolute error (2.3053), while the BM performed better with the fewest iterations and the lowest average absolute error (0.2249). When the average absolute approximation errors were considered, the GR method achieved the most accurate result. While the average absolute error depending on the number of iterations reached the most successful result with BM, the average absolute approximation error depending on the number of iterations reached the most precise value with the FIB method. This shows that the performance of the algorithms may vary according to different types of functions.

In the tests performed on the function f_3 , the GR method stood out with the lowest value of the mean absolute error (0.0099) and the mean absolute approximation error value (0.0015) with 20 iterations, while the FIB method showed average performance in the calculations of the mean absolute error and the mean absolute approximation error with 15 iterations. In the calculation of the average absolute error according to the number of iterations, GR again stood out as a successful method, while the FIB method again showed average performance. In the calculation of the average absolute approximation error based on the number of iterations, the FIB method gave the most successful result.

For the function f_4 , although the GR method was slower than the other methods in terms of the number of iterations, it was the most successful method in the calculations of the average absolute error and the average absolute approximation error. In these two error calculations, GR showed average performance. Although GR achieved the best result in the calculation of the average absolute error depending on the number of iterations, it was noteworthy that this method achieved the worst result in the calculation of the average absolute approximation error depending on the number of iterations.

4. Conclusion and Evaluation

The results given in Table 1 show that the performance of different optimization algorithms on various test functions can vary greatly, and that the choice of the optimal root-finding method should be carefully evaluated depending on the problem. According to these results, all three methods have advantages and disadvantages in certain situations. Below are some observations based on the test functions and the results obtained:

Bisection Search Method (BM): In general, BM gave good results with very low absolute errors for the functions f_1 and f_4 . Especially for the function f_1 , it provided a very accurate approximation using the minimum number of iterations. This shows that BM can provide fast and efficient results in some cases.

Golden Ratio Search Method (GR): The GR method has attracted attention with low absolute errors, especially for functions f_3 and f_4 . By providing the lowest absolute error for the function f_3 , it has shown that the GR method can be effective for this type of functions.

Fibonacci Search Method (FIB): The FIB method shows relatively high average absolute approximation errors for all four functions compared to the other methods. The FIB method gives the best result for all four functions in the absolute approximation error calculation depending on the number of iterations.

In general, the performance of each method varies depending on the properties of the function under test. This shows that when choosing an optimal root-finding method, the properties of the function should be carefully analyzed. In addition, other factors of each method, such as the number of iterations and computational cost, should be considered. The differences between the methods are due to the approximation strategies and mathematical structures of the algorithms, so it is important to choose the method that best suits the nature of a particular problem.

In this study, the performance analyses of Bisection, Fibonacci and Golden Ratio search methods are comprehensively evaluated on different functions and the strengths and weaknesses of these methods are systematically revealed. Identification of the differences between classical and modern optimization techniques provides direction for theoretical and applied research on algorithm selection processes.

The study highlights the mathematical foundations of the minimum value approach and adds a new dimension to the analytical analysis processes by using the mathematical foundations of the Extreme Value Theorem and the Intermediate Value Theorem and simulations to approximate the true root with minimum error through the optimization approach. Accuracy analyzes of the methods using metrics such as mean absolute error and mean absolute approximation error are performed, thus providing a comparative evaluation of the optimization techniques in terms of their error tolerance.

Especially in engineering design processes, the Fibonacci and Golden Ratio search methods used to determine the optimum parameters offer significant advantages in terms of both processing efficiency and calculation speed. These methods can be effectively applied in engineering fields such as aerodynamic modeling, structural analysis, and mechanical system optimization that require high precision. In addition, in decision processes

involving uncertainty such as financial analysis and risk management, these optimization approaches can be used in applications such as portfolio optimization, risk minimization and development of investment strategies. Especially Fibonacci and Golden Ratio based algorithms facilitate more accurate predictions in areas with unpredictable fluctuations such as stock price prediction and macroeconomic modeling.

References

- [1] H. Bhasin and S. Bhatia. Use of genetic algorithms for finding roots of algebraic equations. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2(4):1693–1696, 2011.
- [2] P. Erdogmuş. Finding nonlinear equation roots with particle swarm optimization and its comparison with genetic algorithm. *Journal of Advanced Technology Sciences*, 5(1):14, 2016.
- [3] K. Gdawiec. Fractal patterns from the dynamics of combined polynomial root methods. *Nonlinear Dynamics*, 90(4):2457–2479, 2017.
- [4] T. Gemechu and S. Thota. On new root finding algorithm for solving nonlinear transcendental equations. *International Journal of Chemistry, Mathematics and Physics (IJCMP)*, 4(2):18–24, 2020.
- [5] T. Nayak and T. Dash. Solution to quadratic equation using genetic algorithm. In *Proceedings of National Conference on AIRES-2012*. Andhra University, 2012.
- [6] V. Y. Pan. Root-finding and root-refining for a polynomial equation. Computer science technical reports, City University of New York (CUNY), 2012.
- [7] F. Vatansever and M. Hatun. Simulator design for newton-based root finding methods. *SAÜ Journal of Science and Technology*, 19(3):327–337, 2015.
- [8] V. Yu. Semenov. A method to find all the roots of the system of nonlinear algebraic equations based on the krawczyk operator. *Cybernetic and Systems Analysis*, 51(5):819–825, 2015.
- [9] S. Thota. A new root finding algorithm using exponential series. *Ural Mathematical Journal*, 5(1):83–90, 2019.
- [10] C. L. Sabharwa. Blended root finding algorithm outperforms bisection and regula falsi algorithms. *Mathematics*, 7:1118, 2019.
- [11] S. Thota, M. M. Awad, S. Palanisamy, and L. Rathour. A derivative-free root-finding algorithm using exponential method and its implementation. *BMC Research Notes*, 16(1):276, 2023.
- [12] R. G. Bartle and D. R. Sherbert. *Introduction to Real Analysis*. Wiley, 4 edition, 2011.
- [13] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3 edition, 1976.
- [14] R. L. Burden, J. D. Faires, and A. M. Burden. *Numerical Analysis*. Cengage Learning, 10 edition, 2016.
- [15] J. C. Ehiwario. *Lecture Notes on MTH213: Introduction to Numerical Analysis*. College of Education Agbor, Affiliate of Delta State University, Abraka, 2013.
- [16] J. C. Ehiwario and S. O. Aghamie. Comparative study of bisection, newton-raphson

- and secant methods of root-finding problems. *IOSR Journal of Engineering*, 4(4):1–7, 2014.
- [17] A. Hasan. Numerical study of some iterative methods for solving nonlinear equations. *International Journal of Engineering Science Invention*, 5(2):1–10, 2016.
- [18] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.
- [19] T. Koshy. *Fibonacci and Lucas Numbers with Applications*. Wiley, New York, 2001.
- [20] G. Markowsky. Misconceptions about the golden ratio. *The College Mathematics Journal*, 23(1):2–19, 1992.
- [21] L. R. Foulds. *Optimization Techniques: An Introduction*. Springer-Verlag, 1981.
- [22] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? - arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7:1247–1250, 2014.