Journal of Innovative Applied Mathematics and Computational Sciences

J. Innov. Appl. Math. Comput. Sci. **5**(1) (2025), 169–195. DOI: 10.58205/jiamcs.v5i1.1921

http://jiamcs.centre-univ-mila.dz/

Introduction of modified root finding approaches and their comparative study with existing methods

Haputhanthrige Hashintha Dilan Kumara [1], Samoshi Shanika [2], Haputhanthrige Sachithra Gayashani [3], Withanage Ujitha Imeshan [1] and Sanjula Thilakarathne ✉[4]

[1]Department of Computer Systems Engineering, Faculty of Computing and Technology, University of Kelaniya, Kelaniya, 11600, Sri Lanka

[2]Department of Mathematical Sciences, Faculty of Applied Sciences, South Eastern University of Sri Lanka, Oluvil, 32360, Sri Lanka

[3]Department of Mathematics, Faculty of Applied Sciences, University of Sri Jayewardenepura, Nugegoda, 10250, Sri Lanka

[4]Department of Computer Systems Engineering, Faculty of Computing and Technology, University of Kelaniya, Kelaniya, 11600, Sri Lanka

Received February 24, 2025, Accepted July 16, 2025, Published July 25, 2025

Abstract. Root-finding in nonlinear equations is a fundamental problem in numerical analysis with applications in mathematics and engineering. Traditional methods like the Bisection and False Position methods have been widely used, but they often face challenges related to convergence speed, stability, and computational efficiency. This paper presents two novel numerical root-finding methods that combine the robustness of the Bisection method with the efficiency of the False Position method, improving both convergence rates and stability. Furthermore, we illustrate some numerical applications to discuss error analysis, convergence analysis, and comparisons with existing methods. These findings contribute to the advancement of numerical computation by providing more reliable and efficient root-finding techniques.

Keywords: Root-finding Techniques, Bisection method, False-Position Method, Modified Root-finding Methods, Numerical Algorithms.

2020 Mathematics Subject Classification: 65H04, 65H05, 65G50. MSC2020

1 Introduction

Numerical root-finding is a fundamental problem in computational and applied mathematics. It involves identifying the roots (or solutions) of equations of the form $f(x) = 0$, where $f(x)$ is a continuous function. Although analytical methods exist for solving such equations, they are typically limited to simple or specific forms. The significance of numerical root-finding lies in its ability to handle complex, nonlinear equations that arise across various domains.

✉Corresponding author. Email: sanjulaprsan@gmail.com

ISSN (electronic): 2773-4196

For example, in engineering, it is used to determine equilibrium points in structural analysis [18], [17]. In finance, it assists in calculating interest rates and investment returns [9], [2]. In machine learning, root-finding algorithms are used during the training phase of neural networks [19], [15].

Over the years, numerous numerical root-finding methods have been developed, each tailored to specific types of functions and computational requirements. Among the most widely used methods are the Bisection method [16], the Newton-Raphson method [14], the Secant method [5], the False Position method [13], Broyden's method [7], and the Fixed-point Iteration method [3]. Additionally, researchers in [1] introduced a new method for finding roots of nonlinear equations using decomposition techniques.

This study aims to explore whether combining the Bisection method with the False Position method can create a root-finding technique that is both faster and more reliable. Traditional methods often struggle with slow convergence or stability issues, especially when dealing with nonlinear equations. By integrating the strengths of both approaches, this research seeks to overcome these common problems and offer a better solution for finding the roots of equations.

The Bisection method is a simple and reliable technique that guarantees convergence to a root within a specified interval by repeatedly halving the interval. However, its convergence type is linear (order of convergence = 1), making it relatively slow for high-precision applications [16], and inefficient for functions with steep gradients or multiple roots. The False Position method improves upon the Bisection method by using linear interpolation to estimate the root, often resulting in faster convergence. Nonetheless, its convergence is also linear in most cases and is highly dependent on the nature of the function and the initial guesses [6]. These limitations make both methods less effective when used individually. This motivates the development of new numerical root-finding approaches that combine the strengths of both methods to achieve improved performance and reliability. Based on the strengths and weaknesses of the Bisection method and the False Position method, we hypothesize that their combination can lead to a more robust and efficient approach. The Bisection method is known for its stability, whereas the False Position method typically offers faster convergence. By merging these characteristics, we aim to enhance overall performance in terms of both speed and accuracy. This concept forms the foundation for the two novel root-finding approaches proposed in this study.

In this paper, we propose two novel numerical root-finding approaches that combine the robustness of the Bisection method with the efficiency of the False Position method. These approaches offer improved convergence rates and enhanced stability. They are particularly well-suited for solving a wide range of nonlinear equations, making them versatile tools for various applications. All calculations, simulations, and analyses in this study were performed using MATLAB and R. The use of MATLAB and R facilitates rapid prototyping and iterative refinement of the proposed approaches, allowing efficient exploration of parameter spaces and convergence behavior. Furthermore, MATLAB's built-in debugging and profiling tools facilitated the optimization of the algorithms' computational efficiency.

The remainder of this paper is organized as follows: In the next section, we present the methodology, detailing the algorithmic steps and theoretical foundations of the proposed approaches. Section 3 focuses on error analysis, convergence analysis, and comparisons with existing methods, supported by several numerical examples. Finally, we conclude the paper by addressing future research directions. Through this work, we aim to contribute to the ongoing advancement of numerical root-finding methods by providing practical and efficient solutions for solving complex equations.

2 Methodology

This section presents the numerical methods employed in this study, including the Bisection method, the False Position method, and two modified algorithms that combine these approaches in distinct ways. The notations used throughout the section are defined below.

$f(x)$ – The continuous function on $[a, b]$ for which we seek to find the root.

a – Lower bound of the interval where the root is sought.

b – Upper bound of the interval where the root is sought.

c – Estimated value calculated using the False Position method.

mid – Mid value of the interval $[a, b]$.

$iter$ – Next iteration value.

$f(iter)$ – Function value at the current iteration.

tol – Tolerance level for convergence.

i – Index.

root – Final approximated root.

2.1 Bisection method

The Bisection method is a bracketing technique for root finding in continuous functions, based on the Intermediate Value Theorem [3]. The method starts with an interval where the function changes sign, guaranteeing the existence of at least one root within that interval [4]. The interval is then iteratively bisected, and the midpoint is evaluated as a potential root. The subinterval containing the sign change is chosen for the next iteration. This process continues until the root is approximated within a specified tolerance.

2.1.1 Bisection method algorithm

The Bisection method is a fundamental iterative technique used to find the roots of a continuous function $f(x)$ over a given interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs, i.e., $f(a) \cdot f(b) < 0$. This condition, as ensured by the Intermediate Value Theorem, guarantees the existence of at least one root within the interval. The method operates by repeatedly halving the interval and selecting the subinterval that contains the root based on the sign of the function at the midpoint [4]. The iterative process of the Bisection method can be described as follows:

$$mid_n = \frac{a_n + b_n}{2}.$$

Here, a_n and b_n denote the endpoints of the interval at the n-th iteration, and mid_n is the midpoint. If $f(mid_n) = 0$, then mid_n is the exact root. Otherwise, the interval is updated as follows:

$$\begin{cases} [a_n, mid_n], & f(a_n) \cdot f(mid_n) < 0, \\ [mid_n, b_n], & \text{otherwise.} \end{cases}$$

The iterations continue until the interval length satisfies the stopping criterion:

$$|a_n - b_n| < \epsilon,$$

where ϵ is a predefined tolerance.

The Bisection method guarantees convergence as long as the function is continuous and the initial interval satisfies the sign-change condition. Its convergence is linear, and the error after n iterations can be expressed as:

$$|mid_n - r| \leq \frac{b-a}{2^n},$$

where r is the actual root and $[a, b]$ is the initial interval.

The algorithm can be summarized as follows:

Step 1: Initialize $f_{iter} = 1$, $iter = 1$, $i = 1$.

Step 2: Calculate $f(a)$ and $f(b)$.

Step 3: $mid = (a+b)/2$.

Step 4: Update $iter = mid$ and $f_{iter} = f(iter)$.

Step 5: If $f(iter) = 0$ or $|f_{iter}| < tol$, then go to Step 8.

Step 6: If $f(a) \cdot f(iter) \leq 0$, update $b = iter$; otherwise, $a = iter$.

Step 7: $i = i + 1$, and go to Step 3.

Step 8: Stop iteration and output, root $= iter$.

2.2 False Position method

The False Position method is another bracketing technique for root finding, similar to the Bisection method, but it employs linear interpolation to estimate the root. Although typically faster than the Bisection method, the False Position method may fail to converge if the function values at two consecutive approximations are identical [12], [5]. Furthermore, if the secant line becomes tangent to the given nonlinear function at any step, the method may stagnate and fail to converge to the root [12].

2.2.1 False Position method algorithm

Consider a continuous function $f(x)$ defined on an interval $[a, b]$ where $f(a) \cdot f(b) < 0$, ensuring that at least one root exists in the interval for the equation $f(x) = 0$. Unlike the Bisection method, which uses the midpoint, the False Position method estimates the root by computing the intersection point of the secant line connecting the points $(a, f(a))$ and $(b, f(b))$. Once the first approximation c is obtained, the method proceeds by updating the interval based on the sign of $f(c)$. Specifically, either $(a, f(a))$ or $(b, f(b))$ is replaced with $(c, f(c))$, depending on where the sign change occurs. The iterations continue until $f(c)$ is within a predefined tolerance.

The algorithm can be summarized as follows:

Step 1: Initialize $f_{iter} = 1$, $iter = 1$, $i = 1$.

Step 2: Calculate $f(a)$ and $f(b)$.

Step 3: $c = \frac{f(a) \cdot b - f(b) \cdot a}{f(a) - f(b)}$.

Step 4: Update $iter = c$ and $f_{iter} = f(iter)$.

Step 5: If $f(iter) = 0$ or $|f_{iter}| < tol$, then go to Step 8.

Step 6: If $f(a) \cdot f(iter) \leq 0$, update $b = iter$; otherwise, $a = iter$.

Step 7: $i = i + 1$, and go to Step 3.

Step 8: Stop iteration and output, root $= iter$.

2.3 Newton's method

The Newton-Raphson method is one of the most powerful numerical methods for solving a root-finding problem of the form $f(x) = 0$ [4]. It is based on the concept of successive linearization, in which a complex nonlinear problem is transformed into a sequence of simpler linear problems. The solutions of these linear problems iteratively converge toward the root of the original nonlinear equation [8]. Unlike the Secant method, which uses two initial points to construct a secant line, Newton's method requires only a single initial guess and uses the tangent line at that point to estimate the root. This method relies on the derivative of the function.

2.3.1 Newton's method algorithm

Let $x_0 \in [a, b]$ be a suitable initial approximation to the root such that $f'(x_0) \neq 0$. Here, $f'(x_0)$ denotes the first derivative of the function $f(x)$ at x_0, and c' represents the estimated root obtained using Newton's method.

The algorithm can be summarized as follows:

Step 1: Initialize $f_{iter} = 1$, $iter = 1$, $i = 1$.

Step 2: Calculate $f(x_0)$ and $f'(x_0)$.

Step 3: $c' = x_0 - \frac{f(x_0)}{f'(x_0)}$.

Step 4: Update $iter = c'$ and $f_{iter} = f(iter)$.

Step 5: If $f(iter) = 0$ or $|f_{iter}| < tol$, then go to Step 8.

Step 6: Update $x_0 = iter$.

Step 7: $i = i + 1$, and go to Step 2.

Step 8: Stop iteration and output, root $= iter$.

2.4 Modified methods

To leverage the strengths of both the Bisection method and the False Position method, two novel algorithms were developed for computing the roots of nonlinear equations. These algorithms combine the reliability of the Bisection method with the faster convergence of the False Position method in two distinct ways.

2.4.1 Modified algorithm 1 (BF approach)

The procedure begins by selecting an initial interval $[a, b]$ such that $f(a).f(b) < 0$, ensuring the existence of at least one root within the interval. The midpoint of this interval, $mid = (a + b)/2$, is then computed. Based on the sign of the function at the midpoint, the interval is updated to either $[a, mid]$ or $[mid, b]$. Subsequently, the False Position method is applied to the updated interval to obtain the first approximation (c). If c satisfies the convergence criterion, either $f(c) = 0$ or $|f(c)| < \epsilon$, then c is accepted as the root. Otherwise, the process is repeated by updating the interval and reapplying the same steps until the desired level of accuracy is achieved. Figure 2.1 illustrates the first iteration of the BF approach.

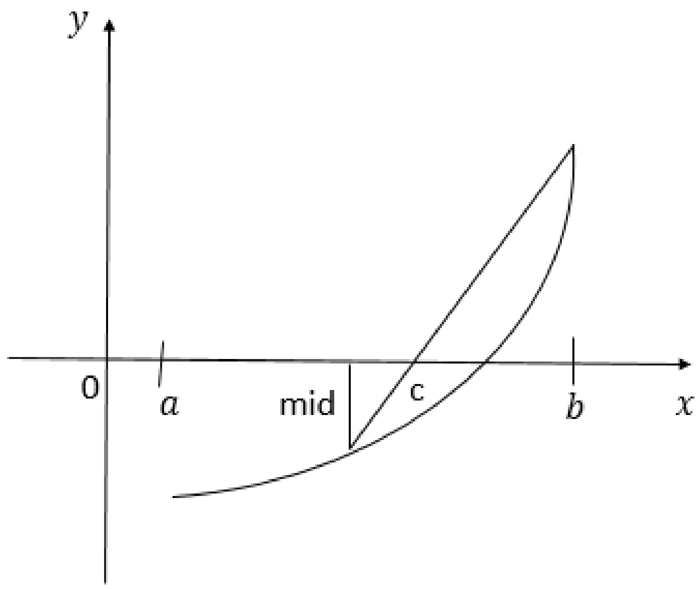

Figure 2.1: First Iteration of Modified Algorithm 1 (BF Approach)

Key features:

- **Interval refinement:** The Bisection method is used to refine the interval $[a, b]$ and ensure the root is bracketed.
- **Root estimation:** The False Position method is used to estimate the root within the refined interval.
- **Tolerance check:** The algorithm terminates when the function value at the estimated root is within the specified tolerance tol.

The algorithm can be summarized as follows:

Step 1: Initialize $f_{iter} = 1$, $iter = 1$, $i = 1$.

Step 2: Calculate $f(a)$ and $f(b)$.

Step 3: $mid = (a + b)/2$.

Step 4: If $f(b).f(mid) < 0$, set $a = mid$; otherwise, $b = mid$.

Step 5: $c = \frac{f(a)\cdot b - f(b)\cdot a}{f(a)-f(b)}$.

Step 6: Update $iter = c$ and $f_{iter} = f(iter)$.

Step 7: If $f(iter) = 0$ or $|f_{\text{iter}}| < tol$, then go to Step 10.

Step 8: If $f(a) \cdot f(iter) < 0$, update $b = iter$; otherwise, $a = iter$.

Step 9: $i = i + 1$, and go to Step 2.

Step 10: Stop iteration and output, root $= iter$.

Figure 2.2 presents a detailed flowchart illustrating the complete implementation of the algorithm.

2.4.2 Modified algorithm 2 (FB approach)

The procedure begins by selecting an initial interval $[a, b]$ such that $f(a).f(b) < 0$, ensuring the existence of a root within the interval. Next, the intersection point c of the secant line connecting the points $(a, f(a))$ and $(b, f(b))$ with the x-axis is computed, as in the False Position method. Based on the sign of the function at c, the interval is updated to either $[a, c]$ or $[c, b]$. The first approximation is then obtained by calculating the midpoint (*mid*) of the updated interval, similar to the Bisection method. If *mid* satisfies the convergence criterion, i.e., $f(mid) = 0$ or $|f(mid)| < \epsilon$, it is accepted as the root. Otherwise, the process is repeated by updating the interval and reapplying the same steps until convergence is achieved. Figure 2.3 illustrates the first iteration of the FB approach.

Key features:

- **Interval refinement:** The False Position method is used to refine the interval $[a, b]$ and ensure the root is bracketed.
- **Root estimation:** The Bisection method is used to estimate the root within the refined interval.
- **Tolerance check:** The algorithm terminates when the function value at the estimated root is within the specified tolerance tol.

The algorithm can be summarized as follows:

Step 1: Initialize $f_{iter} = 1$, $iter = 1$, $i = 1$.

Step 2: Calculate $f(a)$ and $f(b)$.

Step 3: $c = \frac{f(a)\cdot b - f(b)\cdot a}{f(a)-f(b)}$.

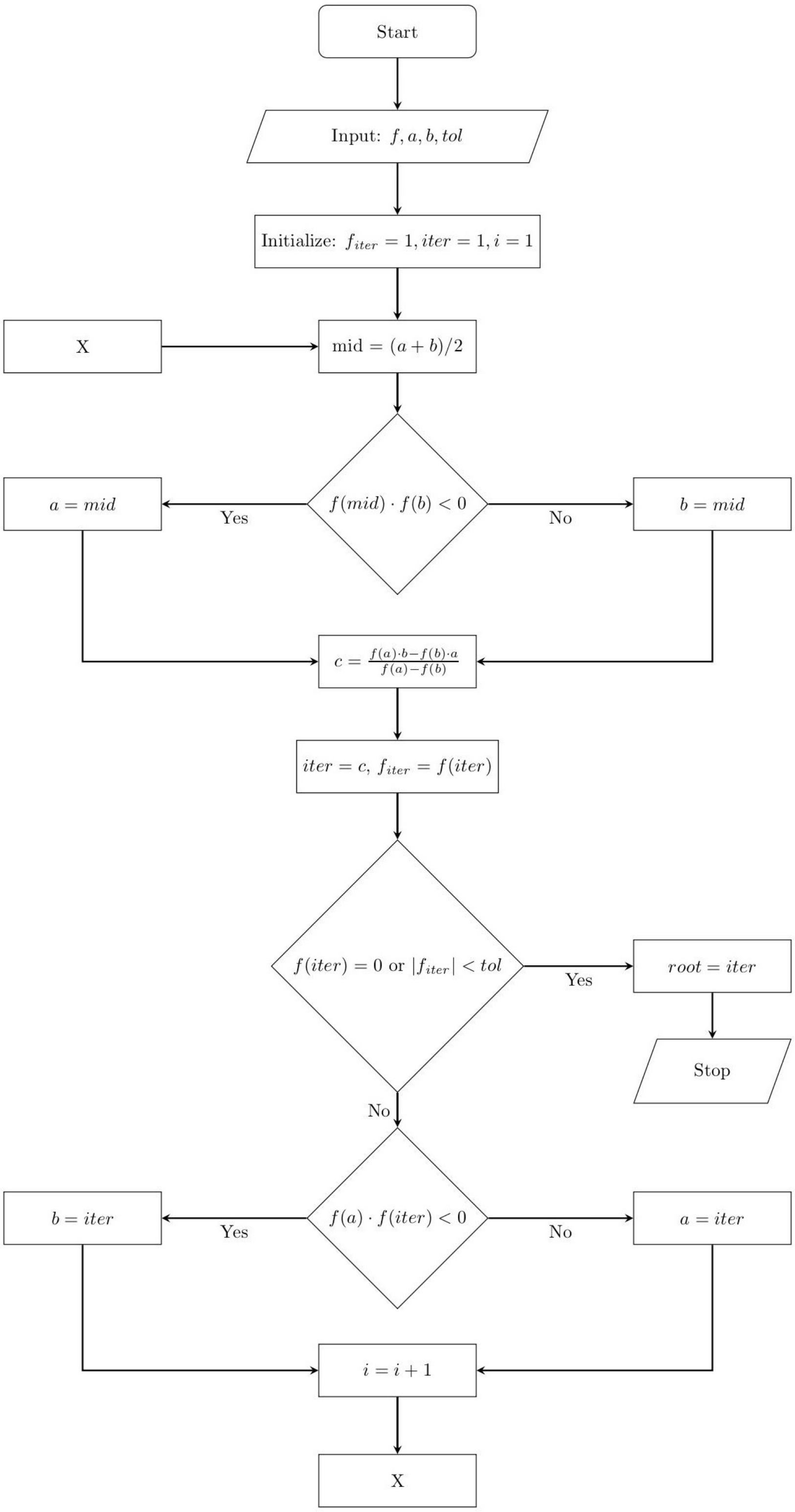

Figure 2.2: Steps of Modified Algorithm 1 (BF Approach)

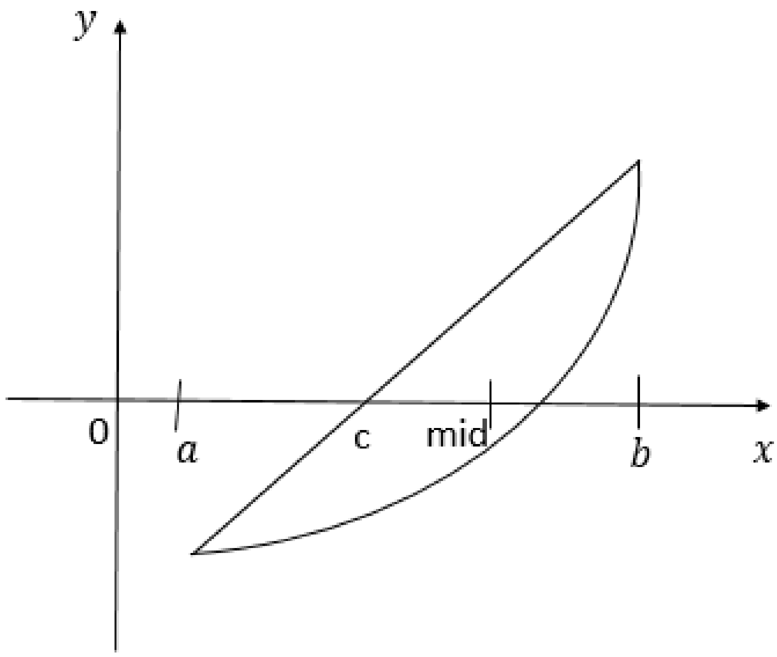

Figure 2.3: First Iteration of Modified Algorithm 2 (FB Approach)

Step 4: If $f(c) \cdot f(b) < 0$, set $a = c$; otherwise, $b = c$.

Step 5: $mid = (a + b)/2$.

Step 6: Update $iter = mid$ and $f_{iter} = f(iter)$.

Step 7: If $f(iter) = 0$ or $|f_{\text{iter}}| < tol$, then go to Step 10.

Step 8: If $f(a) \cdot f(iter) < 0$, update $b = iter$; otherwise, $a = iter$.

Step 9: $i = i + 1$, and go to Step 2.

Step 10: Stop iteration and output, root $= iter$.

Figure 2.4 presents a detailed flowchart illustrating the complete implementation of the algorithm.

2.5 Convergence rate estimation

Assume that r is the exact root and x_n is the nth approximation of the equation $f(x) = 0$. The absolute error at the nth iteration is given by:

$$e_n = |x_n - r|.$$

The error at the $(n + 1)$th step should satisfy the following condition:

$$e_{n+1} = Ce_n^p,$$

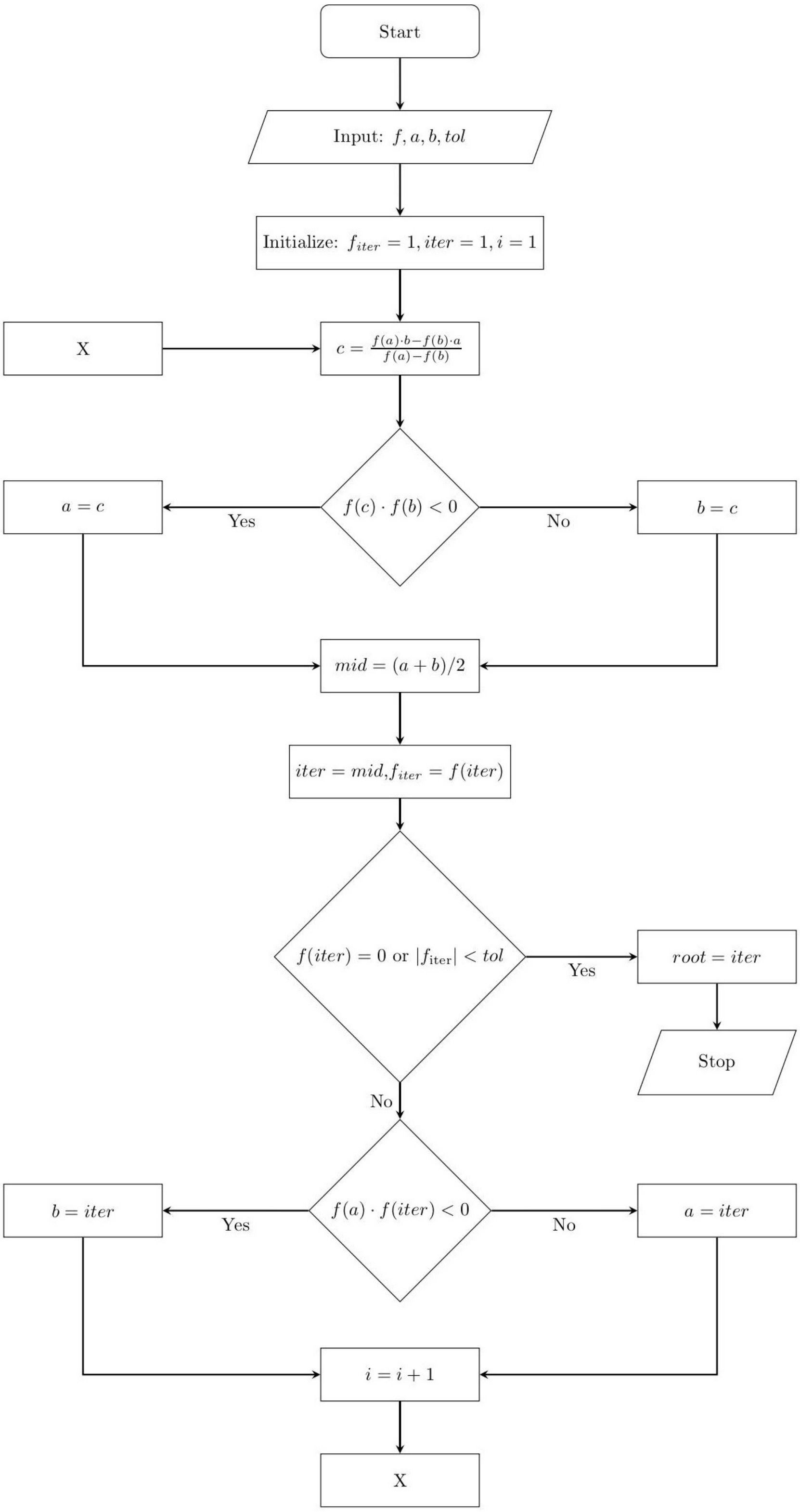

Figure 2.4: Steps of Modified Algorithm 2 (FB Approach)

where p is the order of convergence and C is a constant that depends on the method and the function.

The error at the nth step is given by:

$$e_n = Ce_{n-1}^p.$$

From equations (1) and (2),

$$\frac{e_{n+1}}{e_n} = \frac{e_n^p}{e_{n-1}^p}.$$

Taking ln of both sides,

$$p \approx \frac{\ln\left(\frac{e_{n+1}}{e_n}\right)}{\ln\left(\frac{e_n}{e_{n-1}}\right)}.$$

This relation, denoted as equation (3), can be used to estimate the stepwise convergence rate of an iterative method [10].

3 Results and discussion

This section describes the traditional root-finding methods-Bisection, False Position, and Newton-Raphson as well as the newly proposed FB and BF approaches. These methods are analyzed and compared using two arbitrary equations and two engineering applications. The accuracy and efficiency of each method are evaluated using established root-finding criteria. All tables and graphs were generated using MATLAB.

3.1 Arbitrary equations and engineering applications

Example 1. *Arbitrary problems*

Two arbitrary test problems were considered to analyze the numerical behavior of the proposed algorithms, and the results are presented in Tables 3.1-3.4.

$$f_1(u) = u^2 - 5u + 6,$$

$$f_2(u) = e^u - 2u - 5.$$

Example 2. *Solving for volume in the Van der Waals equation*

The Van der Waals equation is a mathematical formula used to describe the behavior of real gases. From this equation, we derive the following:

$$\left(P + \frac{K_1 n^2}{V^2}\right)(V - nK_2) = nRT.$$

By assuming appropriate values for the given parameters, we formulate the following nonlinear problem:

$$f_3(u) = 0.986u^3 - 5.181u^2 + 9.067u - 5.289,$$

where u represents the volume, and the root of this equation can be found by solving $f_3 = 0$. As it is a cubic equation, it possesses a richer mathematical structure, typically having three roots. However, only one of these is a feasible positive real root "1.9298462428" since the volume of a gas must be positive [10]. To initiate the iteration process, we set $u_0 = 1.00$, and the results are recorded in Tables 3.5 and 3.6.

Example 3. *Beam Designing model*

Beam design is a fundamental aspect of structural engineering. In civil engineering, bricks or stones often serve as structural horizontal elements to span gaps and support loads in the upper portions of walls.

$$f_4(u) = u^4 + 4u^3 - 24u^2 + 16u + 16,$$

The governing equation in this case is a fourth-order polynomial with roots at 2 (with multiplicity 2) and $-4 \pm 2\sqrt{3}$ [11]. The initial guess for the numerical method is set as $u_0 = -0.8$, and all computed numerical results are presented in Tables 3.7 and 3.8.

3.2 Comparison of iteration data and error analysis

Tables 3.1-3.8 present a detailed comparison of absolute errors to assess the repeatability and performance of various root-finding methods. The two proposed arbitrary equations, $f_1 = 0$ and $f_2 = 0$, along with two engineering applications, $f_3 = 0$ and $f_4 = 0$, were analyzed using well-known root-finding methods: Bisection, False Position, Newton-Raphson, as well as the newly proposed FB and BF approaches. The tables report the number of iterations required, the final approximate root obtained, and the corresponding absolute error for each method.

Table 3.1: Approximate Roots of $f_1 = 0$ with Respect to Each Method

Iteration	**Root**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	1.750000	2.333333	1.666667	2.166667	1.666667
2	2.125000	2.200000	1.933333	2.007937	1.888889
3	1.937500	2.111111	1.996078	2.000133	1.950617
4	2.031250	2.058824	1.999985		1.975603
5	1.984375	2.030303			1.987808
6	2.007812	2.015385			1.993904
7	1.996094	2.007752			1.996952
8	2.001953	2.003891			1.998476
9	1.999023	2.001949			1.999238
10		2.000976			

Table 3.2 and Table 3.2 illustrate the approximate roots and their corresponding absolute errors of $f_1 = 0$ at each iteration for the methods under consideration.

The BF approach converges in just 3 iterations with a finite error and a computation time of 0.00094 seconds, indicating high efficiency. The FB approach, while requiring slightly more iterations (9 iterations), ensures stable and accurate convergence within 0.0038 seconds. This method effectively balances the robustness of the Bisection method with the fast convergence of the False Position method.

The Bisection method requires 9 iterations and a longer computation time of 0.0023 seconds to converge. Although the False Position method is reliable, it takes 10 iterations and

Table 3.2: Absolute Errors of the Roots of $f_1 = 0$ with Respect to Each Method

Iteration	**Absolute Error**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	0.2500	0.3333	0.3333	0.1667	0.3333
2	0.1250	0.2000	0.0667	0.0079	0.1111
3	0.0625	0.1111	0.0039	0.0001	0.0494
4	0.0312	0.0588	0.0000		0.0244
5	0.0156	0.0303			0.0122
6	0.0078	0.0154			0.0061
7	0.0039	0.0078			0.0030
8	0.0020	0.0039			0.0015
9	0.0010	0.0019			0.0008
10		0.0010			

results in a larger final error compared to the newly proposed approaches. When comparing the results of $f_1 = 0$, the Newton-Raphson method is the fastest, converging in 0.00065 seconds. However, it requires derivative evaluations, which can reduce its practicality in certain applications.

Overall, the newly proposed BF and FB approaches provide improved accuracy and efficiency by strategically integrating the strengths of the False Position and Bisection methods.

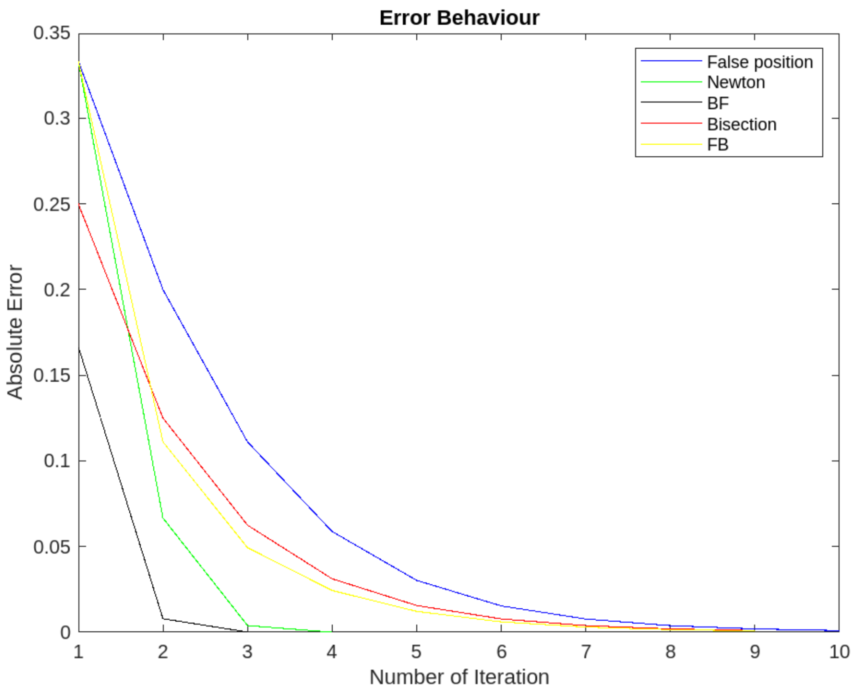

Figure 3.1: Error Behavior of the Equation $f_1 = 0$

Figure 3.1 illustrates the error behavior of the root-finding methods related to the equation $f_1 = 0$. The number of iterations is shown on the x-axis, and the absolute error is plotted on the y-axis. As the number of iterations increases, the error generally decreases across all methods.

The Newton-Raphson method demonstrates the fastest convergence, achieving the lowest error within the first three iterations. The BF approach also shows rapid convergence, reaching near-zero error by the second iteration. In contrast, the False Position and FB methods exhibit moderate convergence rates.

Overall, the Newton-Raphson method and the BF approach can be considered among the most efficient techniques for solving the example equation $f_1 = 0$.

Table 3.3: Approximate Roots of $f_2 = 0$ with Respect to Each Method.

Iteration	**Root**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	2.500000	2.150605	2.298929	2.212333	2.575303
2	2.250000	2.212264	2.253014	2.249086	2.403814
3	2.375000	2.236479	2.251637	2.251555	2.326831
4	2.312500	2.245829			2.289192
5	2.281250	2.249416			2.270413
6	2.265625	2.250788			2.261025
7	2.257812	2.251312			2.256330
8	2.253906	2.251512			2.253983
9	2.251953				2.252810
10	2.250977				2.251636
11	2.251465				
12	2.251709				

Table 3.4: Absolute Errors of the Roots of $f_2 = 0$ with Respect to Each Method

Iteration	**Absolute Error**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	0.2484	0.1010	0.0473	0.0393	0.3237
2	0.0016	0.0394	0.0014	0.0026	0.1522
3	0.1234	0.0152	0.0000	0.0001	0.0752
4	0.0609	0.0058			0.0376
5	0.0296	0.0022			0.0188
6	0.0140	0.0008			0.0094
7	0.0062	0.0003			0.0047
8	0.0023	0.0001			0.0023
9	0.0003				0.0012
10	0.0007				0.0000
11	0.0002				
12	0.0001				

Table 3.3 and Table 3.4 present the approximate roots and their corresponding absolute errors of $f_2 = 0$ at each iteration for the methods under consideration.

According to the results from the False Position method, it takes 8 iterations to reach the root, yielding an absolute error of 0.0001 and a computation time of 0.0044 seconds. The

Newton-Raphson method shows the fastest convergence, reaching the root in just 3 iterations with an absolute error of 0.0000 in 0.00076 seconds. While the Newton-Raphson method is known for its rapid convergence near the root under favorable conditions, it can sometimes diverge. This typically occurs when the derivative $f'(x)$ is very small or zero, for instance, in the function $f(x) = x^{1/3}$, leading to large or undefined steps. Divergence may also occur if the initial guess is far from the actual root, causing the tangent line to point away from the root instead of toward it. The Bisection method requires 12 iterations to converge, with a computation time of 0.0027 seconds and an absolute error of 0.0001.

In contrast, the BF approach achieves an accurate root approximation in just 3 iterations, with a computation time of 0.0012 seconds and an absolute error of 0.0001. Similarly, the FB approach finds the root in 10 iterations, requiring 0.0045 seconds to reach an absolute error of 0.0000.

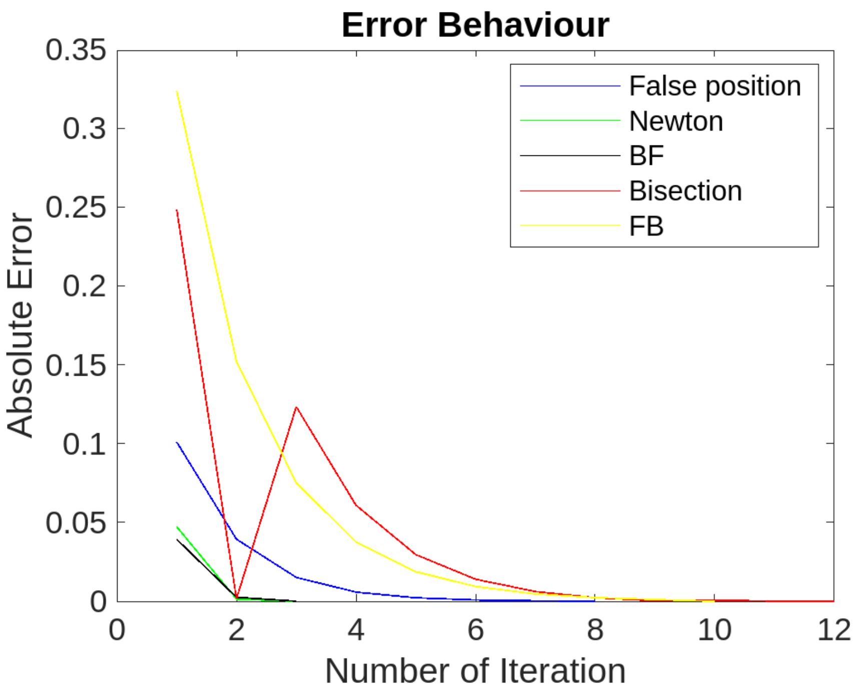

Figure 3.2: Error Behavior of the Equation $f_2 = 0$

Figure 3.2 illustrates the error behavior of the methods applied to find the root of the equation $f_2 = 0$. The graph shows that the Newton-Raphson method achieves the fastest convergence. The False Position method also demonstrates stable convergence at a slightly slower rate. The BF approach follows a pattern similar to that of the Newton-Raphson method, effectively minimizing error with high efficiency. In contrast, the Bisection and FB methods exhibit noticeable fluctuations before eventually converging and stabilizing.

Table 3.5 and Table 3.6 illustrate the approximate roots and their corresponding absolute errors of $f_3 = 0$ at each iteration for the methods under consideration.

The Bisection method required 3 iterations to find the root, yielding an absolute error of 0.0077. While this method guarantees convergence, its rate of convergence is relatively slow. The False Position method required 31 iterations to meet the error tolerance of 0.001, demonstrating a steady decrease in error over time, despite the higher number of iterations. The newly introduced BF and FB approaches demonstrated superior performance. The BF approach converged in just 4 iterations with an absolute error of 0.0008. The FB approach

Table 3.5: Approximate Roots of $f_3 = 0$ with Respect to Each Method

Iteration	**Root**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	1.750000	1.762340	1.600279	1.757760	2.131170
2	2.125000	1.770139	1.708344	1.789416	1.962304
3	1.937500	1.777952	-0.171433	1.890805	1.925894
4		1.785765	0.469044	1.929060	
5		1.793560	0.896064		
6		1.801314	1.181160		
7		1.809000	1.372672		
8		1.816591	1.504478		
9		1.824055	1.604024		
10		1.831359	1.714541		
11		1.838471	0.599854		
12		1.845359	0.983327		
13		1.851994	1.239577		
14		1.858351	1.412343		
15		1.864405	1.532931		
16		1.870139	1.629116		
17		1.875541	1.768249		
18		1.880600	1.128434		
19		1.885315	1.337059		
20		1.889686	1.479452		
21		1.893719	1.583610		
22		1.897424	1.684139		
23		1.900812	2.377007		
24		1.903898	2.174960		
25		1.906699	2.046028		
26		1.909232	1.970873		
27		1.911516	1.937449		
28		1.913570			
29		1.915412			
30		1.917060			
31		1.918531			

showed improved efficiency with 3 iterations and a final absolute error of 0.0040, indicating faster convergence than conventional methods.

A sharp decrease in absolute error was observed during the initial iterations, suggesting that the error behavior of the proposed approaches is more stable. These results indicate that the BF and FB approaches can achieve accurate solutions with minimal computational effort, making them particularly suitable for solving nonlinear equations in contexts where computational resources are limited and rapid convergence is essential.

Figure 3.3 illustrates the absolute error behavior of various numerical methods used to find the root of the Van der Waals equation, $f_3 = 0$.

The error behavior of the BF and FB approaches demonstrates high convergence rates,

Table 3.6: Absolute Errors of the Roots of $f_3 = 0$ with Respect to Each Method

Iteration	**Absolute Error**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	0.1798	0.1675	0.3296	0.1721	0.2013
2	0.1952	0.1597	0.2215	0.1404	0.0325
3	0.0077	0.1519	2.1013	0.0390	0.0040
4		0.1441	1.4608	0.0008	
5		0.1363	1.0338		
6		0.1285	0.7487		
7		0.1208	0.5572		
8		0.1133	0.4254		
9		0.1058	0.3258		
10		0.0985	0.2153		
11		0.0914	1.3300		
12		0.0845	0.9465		
13		0.0779	0.6903		
14		0.0715	0.5175		
15		0.0654	0.3969		
16		0.0597	0.3007		
17		0.0543	0.1616		
18		0.0492	0.8014		
19		0.0445	0.5928		
20		0.0402	0.4504		
21		0.0361	0.3462		
22		0.0324	0.2457		
23		0.0290	0.4472		
24		0.0259	0.2451		
25		0.0231	0.1162		
26		0.0206	0.0410		
27		0.0183	0.0076		
28		0.0163			
29		0.0144			
30		0.0128			
31		0.0113			

making them suitable for fast approximation. While the Newton-Raphson method converges more quickly, it exhibits less stability due to fluctuations in the error. Although the Bisection method is slower, it remains a reliable approach, providing a consistent and systematic reduction in error. In contrast, the False Position method shows lower reliability, with significant error fluctuations compared to the other methods.

Table 3.7 and Table 3.8 present the approximate roots and their corresponding absolute errors of $f_4 = 0$ at each iteration for the methods under consideration.

Using the Bisection method, the root is reached in 13 iterations with an absolute error of approximately 0.0000. However, the method is relatively slow due to its linear convergence rate. The False Position method converges in 6 iterations, achieving an absolute error of

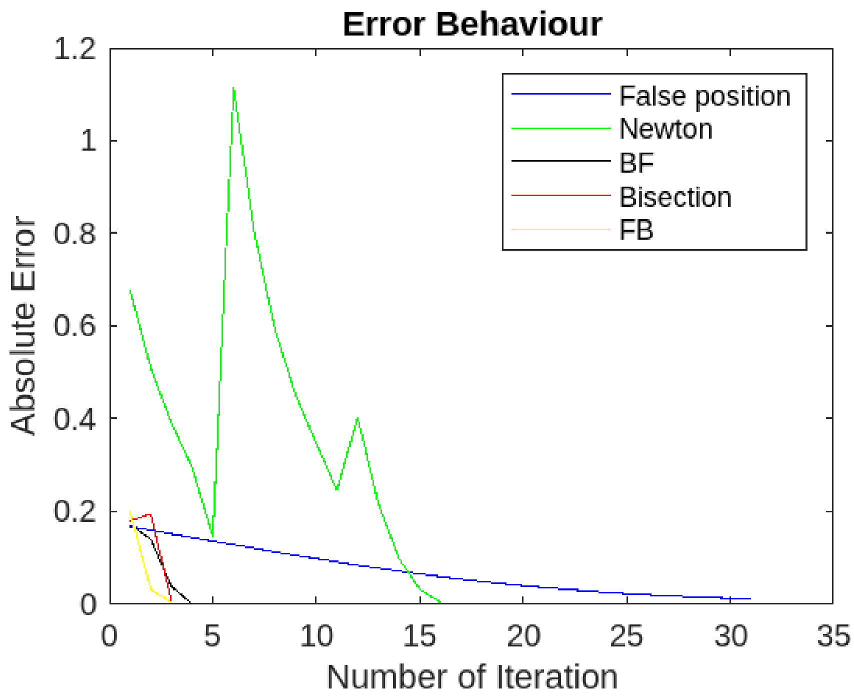

Figure 3.3: Error Behavior of the Van der Waals Equation

Table 3.7: Approximate Roots of $f_4 = 0$ with Respect to Each Method

Iteration	**Root**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	-0.450000	-0.454479	-0.625000	-0.522684	-0.627239
2	-0.625000	-0.523402	-0.540532	-0.534896	-0.579188
3	-0.537500	-0.534047	-0.535912	-0.535860	-0.557478
4	-0.493750	-0.535625		-0.535898	-0.546687
5	-0.515625	-0.535858			-0.541293
6	-0.526563	-0.535892			-0.538596
7	-0.532031				-0.537247
8	-0.534766				-0.536573
9	-0.536133				-0.536236
10	-0.535449				-0.536067
11	-0.535791				-0.535983
12	-0.535962				-0.535941
13	-0.535876				-0.535919

0.0000. Compared to the Bisection method, it shows greater efficiency by reducing the number of iterations, although it requires a slightly higher computation time of 0.0178 seconds.

The Newton-Raphson method exhibits the highest efficiency, reaching the root in just 3 iterations with an absolute error of 0.0000 and a computation time of 0.0020 seconds. This method demonstrates quadratic convergence but requires the computation of derivatives, which can limit its applicability in certain contexts.

The newly introduced BF approach also performs competitively, reaching the root in 4

Table 3.8: Absolute Errors of the Roots of $f_4 = 0$ with Respect to Each Method

Iteration	**Absolute Error**				
	Bisection	False Position	Newton-Raphson	BF Approach	FB Approach
1	0.0859	0.0814	0.0891	0.0132	0.0913
2	0.0891	0.0125	0.0046	0.0010	0.0433
3	0.0016	0.0019	0.0000	0.0000	0.0216
4	0.0421	0.0003		0.0000	0.0108
5	0.0203	0.0000			0.0054
6	0.0093	0.0000			0.0027
7	0.0039				0.0013
8	0.0011				0.0007
9	0.0002				0.0003
10	0.0004				0.0002
11	0.0001				0.0001
12	0.0001				0.0000
13	0.0000				0.0000

iterations with an absolute error of 0.0000 and a computation time of 0.0034 seconds. This method offers a strong balance between computational efficiency and rapid convergence.

The FB approach achieved convergence in 13 iterations. However, it shows improved efficiency by achieving a slightly faster computation time of 0.0047 seconds. While not as fast as the BF or Newton-Raphson methods, the FB approach maintains stable convergence with a consistent stepwise reduction in the absolute error.

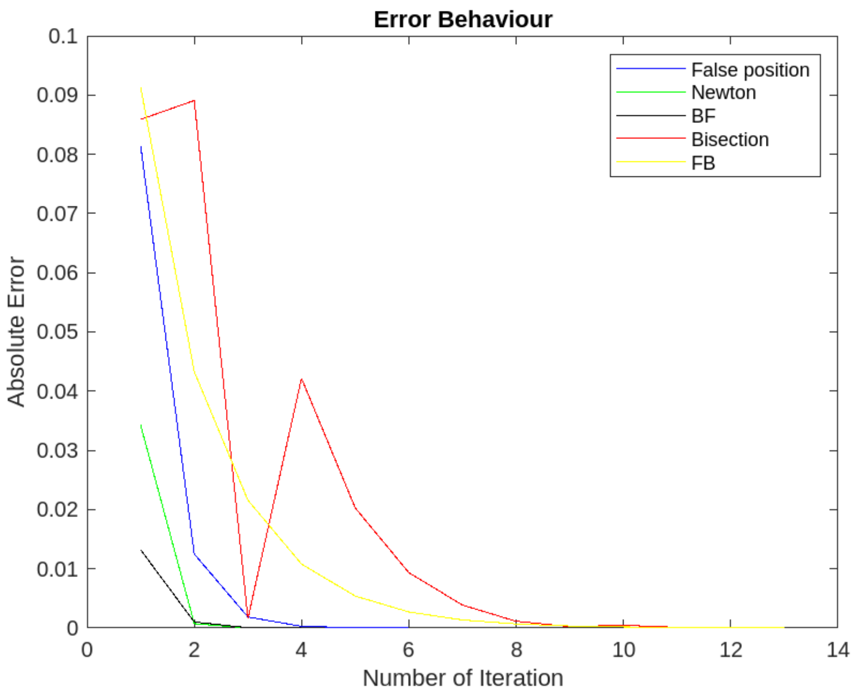

Figure 3.4: Error Behavior of the Beam Designing Model

Figure 3.4 illustrates the absolute error behavior of various numerical methods applied to solve the Beam Design Model equation, $f_4 = 0$.

The Newton-Raphson method demonstrates the fastest convergence due to its quadratic nature, with significantly improved efficiency when an appropriate initial guess is provided. The False Position and FB methods exhibit moderate convergence, effectively balancing speed and reliability. The Bisection method ensures stable and consistent convergence by progressively reducing the absolute error, though at a slower rate. The BF approach shows a rapid initial error reduction, followed by a steady convergence phase, indicating both stability and efficiency.

3.3 Convergence analysis

Tables 3.9–3.12 illustrate the convergence behavior of the example equations $f_1 = 0$, $f_2 = 0$, $f_3 = 0$, and $f_4 = 0$. Each function exhibits distinct convergence patterns, with varying levels of stability and efficiency. The equations $f_1 = 0$ and $f_2 = 0$ demonstrate linear convergence, gradually stabilizing toward the root. The equation $f_3 = 0$ displays quadratic convergence, characterized by rapid error reduction, whereas $f_4 = 0$ follows an exponential convergence trend.

These results highlight the importance of selecting an appropriate root-finding method to optimize computational efficiency and ensure reliable performance across different types of equations.

Table 3.9: Convergence Behavior of Equation $f_1 = 0$

Steps	**Bisection**	**False Position**	**Newton**	**BF**	**FB**
1.000000	1.000000	1.150660	1.913814	1.343856	0.738140
2.000000	1.000000	1.082006		0.869524	
3.000000	1.000000	1.042934		0.983830	
4.000000	1.000000	1.021989		0.999192	
5.000000	1.000000	0.011131		0.999980	
6.000000	1.000000	1.005600		1.000000	
7.000000	1.000000	1.002809		1.000000	
8.000000		1.001407			

Table 3.9 shows the convergence behavior of approximate roots of $f_1 = 0$ at each iteration for the methods under consideration. The results clearly reflect significant differences in the convergence rates among the root-finding methods. The Newton-Raphson method demonstrates the fastest convergence during initial iterations and achieves a higher overall convergence rate compared to other methods.

The newly introduced BF and FB approaches also exhibit strong performance. The BF approach begins with the Bisection method and transitions to the False Position method, resulting in stable and reliable convergence. In contrast, the FB approach starts with the False Position method before switching to the Bisection method. This configuration enables faster initial convergence than traditional methods, such as the standard False Position technique.

While the classic Bisection method maintains a constant convergence rate of 1.000000, the BF and FB approaches outperform it in terms of efficiency. The BF approach shows consistent performance with a steadily increasing convergence rate approaching 1.000000. Similarly, the FB approach achieves rapid convergence because False Position begins at an early stage. However, the standard False Position method shows low efficiency, especially in the later iterations, due to fluctuating convergence rates. This variability is effectively mitigated by the hybrid nature of the BF and FB approaches.

Table 3.10: Convergence Behavior of Equation $f_2 = 0$

Steps	Bisection	False Position	Newton	BF	FB
1.000000	-0.860676	1.012985	1.994147	1.260410	0.934148
2.000000	-0.163437	1.005039			0.984806
3.000000	1.019673	1.001936			0.998479
4.000000	1.041068	1.000738			0.999927
5.000000	1.090075	1.000274			0.999999
6.000000	1.224310	1.000084			1.000001
7.000000	1.967111				1.000002
8.000000	-0.372324				1.000000
9.000000	-1.838690				
10.000000	0.635400				

Table 3.10 illustrates the convergence behavior of the approximate roots of $f_2 = 0$ at each iteration for the methods under consideration. The results reveal distinct convergence patterns, highlighting differences in efficiency and stability among the methods.

The Bisection method consistently achieves convergence; however, the approximate roots exhibit oscillations between positive and negative values in successive iterations, indicating fluctuations in the convergence rate. The False Position method demonstrates a steady convergence trend, gradually approaching a rate of 1, although its overall progress toward the root is comparatively slow. In contrast, the Newton-Raphson method, known for its quadratic convergence, outperforms both the Bisection and False Position methods in terms of speed and accuracy.

The newly introduced BF and FB approaches exhibit high convergence tendencies. The BF approach, in particular, achieves rapid convergence, reaching a rate close to 1 as early as the first iteration and outperforming other methods in later iterations. By the 8th iteration, it successfully reached the root with a notably high convergence rate. The FB approach also demonstrates a robust and stable convergence pattern with minimal fluctuation, offering a reliable alternative to classical techniques.

Table 3.11 presents the convergence behavior of various root-finding methods applied to the Van der Waals equation represented by $f_3 = 0$.

The Newton-Raphson method demonstrates rapid convergence in the initial iterations; however, it exhibits an oscillatory pattern, indicating instability. While the Bisection method maintains stable convergence, it fails to demonstrate sufficient progress toward the root within the recorded iterations, highlighting its slower performance.

The newly proposed BF approach achieves faster convergence in the early stages but displays slight fluctuations before stabilizing. In contrast, the FB approach demonstrates a more accurate and stable convergence trend, outperforming both the Newton-Raphson and False Position methods in terms of consistency and precision.

These results suggest that the BF and FB approaches offer effective alternatives for solving nonlinear equations, especially in cases where traditional methods struggle to maintain stable

Table 3.11: Convergence Behavior of Equation $f_3 = 0$

Steps	Bisection	False Position	Newton	BF	FB
1.000000	-39.647090	1.051992	0.923174	6.297153	1.153829
2.000000		1.052824	1.062005	3.050221	
3.000000		1.053220	2.503878		
4.000000		1.053198	-2.888425		
5.000000		1.052778	-0.161569		
6.000000		1.051985	0.919108		
7.000000		1.050848	0.910983		
8.000000		1.049401	0.956485		
9.000000		1.047681	1.297493		
10.000000		1.045727	-1.440426		
11.000000		1.043581	-1.264434		
12.000000		1.041287	1.264679		
13.000000		1.038887	1.430498		
14.000000		1.036422	1.666126		
15.000000		1.033932			
16.000000		1.031452			
17.000000		1.029014			
18.000000		1.026644			
19.000000		1.024365			
20.000000		1.022192			
21.000000		1.020140			
22.000000		1.018217			
23.000000		1.016426			
24.000000		1.014770			
25.000000		1.013247			
26.000000		1.011853			
27.000000		1.010584			
28.000000		1.009433			
29.000000		1.008392			

convergence or efficient performance.

Table 3.12 presents the convergence behavior of different root-finding methods applied to the Beam Designing model equation represented by $f_4 = 0$.

The Bisection method demonstrates a slow and oscillatory convergence pattern, with some intermediate steps yielding negative values before approaching the root. The False Position method, although requiring more iterations, exhibits stable and consistent convergence. Notably, the Newton-Raphson method achieves a high convergence rate of 1.990116 in the first iteration, indicating its rapid progression toward the root under favorable conditions.

The newly introduced BF and FB approaches demonstrate exceptionally strong convergence characteristics. The FB approach quickly approximates the correct root, reaching a convergence value close to 1.000000 within three iterations. Similarly, the BF approach achieves rapid steady-state convergence and high accuracy in the final steps. Overall, the BF and FB methods prove to be efficient and robust alternatives, outperforming traditional root-finding

Table 3.12: Convergence Behavior of Equation $f_4 = 0$

Steps	Bisection	False Position	Newton	BF	FB
1.000000	-109.765028	1.018806	1.990116	1.260791	0.932321
2.000000	-0.813729	1.002811		1.208076	0.995801
3.000000	-0.223806	1.000414			0.999882
4.000000	1.059512	1.000059			0.999998
5.000000	1.136578				1.000000
6.000000	1.393152				1.000000
7.000000	1.282939				1.000000
8.000000	-0.412785				1.000000
9.000000	-2.200888				1.000000
10.000000	0.366689				1.000000
11.000000	2.027720				1.000001

techniques in both speed and stability.

3.4 Validation of method accuracy using statistical tests

In this section, two statistical tests are applied to evaluate the variance in performance among the numerical methods under consideration.

Table 3.13: One-way ANOVA Summary for log(RMSE) Across Root-Finding Methods

Source of Variation	Df	Sum of Squares (SS)	Mean Square (MS)	F-value	p-value
Method	4	4.972	1.2429	11.19	0.000208
Residuals	15	1.666	0.1111		

Note: The p-value indicates that differences in log(RMSE) among the methods are statistically significant at the 0.05 level.

Table 3.13 presents the results of the one-way ANOVA conducted to evaluate the variance in performance among the numerical methods under consideration.

To assess variations in performance among the five methods, a one-way analysis of variance (ANOVA) was conducted on the log-transformed root mean square error (RMSE) values derived from the error data. The log transformation was applied to stabilize the variance and improve the normality of the data distribution. The ANOVA results, $F(4,15)=11.19$, $p=0.0002$, indicate a statistically significant difference among the methods, suggesting that at least one method differs significantly in terms of accuracy.

Table 3.14 presents the results of pairwise comparisons using Tukey's Honest Significant Difference (HSD) test. This post-hoc analysis identified specific pairs of methods with statistically significant differences. The proposed BF method performed significantly better than the Bisection ($p=0.0063$), False Position ($p=0.011$), and Newton-Raphson ($p=0.0013$) methods. Likewise, the FB method outperformed the Bisection ($p=0.0086$) and False Position ($p=0.0154$) methods. However, the difference between BF and FB methods was not statistically significant ($p=0.9998$), suggesting comparable performance between these two hybrid approaches.

Table 3.14: Tukey HSD Pairwise Comparisons of log(RMSE) Between Methods

Comparison	Mean Difference	95% Confidence Interval	Adjusted p-value
Bisection – BF	0.98228102	[0.2545496, 1.7100125]	0.0062820
False Position – BF	0.91204081	[0.1843094, 1.6397723]	0.0112030
FB – BF	0.03845372	[-0.6892777, 0.7661852]	0.9998208
Newton-Raphson – BF	1.17116641	[0.4434350, 1.8988979]	0.0013483
False Position – Bisection	-0.07024021	[-0.7979717, 0.6574912]	0.9980768
FB – Bisection	-0.94382730	[-1.6715588, -0.2160959]	0.0086225
Newton-Raphson – Bisection	0.18888539	[-0.5388461, 0.9166168]	0.9261313
FB – False Position	-0.87358709	[-1.6013185, -0.1458556]	0.0153690
Newton-Raphson – False Position	0.25912560	[-0.4686059, 0.9868570]	0.8041633
Newton-Raphson – FB	1.13271269	[0.404981, 1.8604441]	0.0018378

4 Conclusion

In this study, we introduced two novel approaches, namely the BF and FB methods, for solving nonlinear equations. These techniques were developed by modifying two classical root-finding methods: the Bisection and the False Position. To evaluate the performance of the proposed approaches, we applied them to four distinct nonlinear equations, including two arbitrary problems and two practical applications. Additionally, we compared the modified approaches with three existing methods (Bisection, False Position, and Newton-Raphson) to analyze their convergence behavior.

For the equation $f_1(u) = 0$, the BF method demonstrated rapid convergence to the root with minimal error and reduced computation time compared to the Bisection and False Position methods. In contrast, the FB and Bisection methods exhibited similar convergence behavior, while the False Position method required more iterations and resulted in higher errors at each step. Compared to the Newton-Raphson method, the BF approach proved to be more efficient, whereas the FB method showed slightly inferior performance.

For the equation $f_2(u) = 0$, both the BF and Newton-Raphson methods converged to the root with the same number of iterations and nearly identical error values at each step. The FB approach required significantly more iterations than the False Position, Newton-Raphson, and BF methods, although it outperformed the Bisection method in this instance.

In the Van der Waals equation $f_3(u) = 0$, the BF, FB, and Bisection methods achieved faster convergence to the root than both the Newton-Raphson and False Position methods. Although the False Position method required more iterations than Newton-Raphson, the Newton method exhibited higher errors at each step in this case.

For the Beam Design Model equation $f_4(u) = 0$, the BF and Newton-Raphson methods required nearly the same number of iterations to converge; however, the BF approach achieved higher accuracy. The FB approach, while converging with better accuracy than the Bisection method, demonstrated lower precision compared to the False Position method.

A numerical convergence analysis was conducted to examine the convergence behavior of each method. The estimated average orders of convergence for the existing methods were consistent with their theoretically established values. In contrast, the proposed hybrid methods exhibited average orders of convergence generally greater than or equal to one, indicating super-linear convergence. Furthermore, statistical analysis revealed that the proposed BF method achieved significantly better performance than all other methods, except for the FB approach. Additionally, the FB method demonstrated improved efficiency compared to the traditional Bisection and False Position methods.

The BF method demonstrates significant advantages in scenarios involving functions with flat or shallow roots (i.e., when the derivative of the function near the root is very small or close to zero), where conventional methods often exhibit slow initial convergence. For example, the Van der Waals equation $f_3 = 0$ benefits from the BF method's combined strategy of interval reduction through Bisection, followed by accelerated refinement via the False Position technique. Furthermore, the method is well-suited for highly nonlinear or oscillatory functions, as observed in $f_2 = 0$ and the Beam Design model $f_4 = 0$, where abrupt slope variations and multiple inflection points can hinder convergence if not effectively stabilized in the initial stages.

Additionally, in cases characterized by poorly bracketed roots or wide initial intervals, such as $f_1 = 0$ and again $f_4 = 0$, the initial Bisection step ensures that the root remains confined within a progressively narrowing interval, thereby enhancing robustness and reliability. Conversely, the FB method is particularly effective for functions that are near-linear, monotonic, or exhibit gentle curvature, where the False Position method enables rapid initial progress before transitioning to Bisection for convergence stabilization.

For instance, both $f_1 = 0$ and $f_2 = 0$, which are monotonic within the tested intervals, benefit from the FB strategy: the initial False Position step efficiently reduces the root-containing interval, while the subsequent Bisection step ensures convergence without the risk of overshooting or divergence. This dual-phase approach highlights the importance of method selection based on the underlying characteristics of the equationfavoring BF for highly irregular or higher-order polynomials, and FB for well-behaved transcendental or nearly linear functions.

Based on the overall results from the selected examples, we can conclude that the proposed BF approach provides accurate approximation with high efficiency and requires fewer iterations compared to the Bisection, False Position, and occasionally the Newton-Raphson methods. However, the FB approach converges more slowly compared to the Newton-Raphson method.

5 Future work

Future research can focus on providing rigorous theoretical foundations for the proposed BF and FB methods, including formal proofs of their convergence order and conditions under which they exhibit super-linear behavior. Additionally, deriving precise error bounds would help quantify the accuracy at each iteration. A comprehensive stability analysis, considering variations in initial intervals and function perturbations, would further enhance the reliability of these techniques.

Beyond theoretical improvements, the methods can be extended to address more complex problem classes, such as systems of nonlinear equations, where multidimensional root-finding poses additional challenges. Furthermore, exploring their integration into numerical schemes for ordinary and partial differential equations, as well as their application in optimization frameworks, would significantly expand the practical relevance of these hybrid approaches.

Declarations

Availability of data and materials

Data sharing is not applicable to this article.

Funding

Not applicable. No funds, grants, or other support was received.

Authors' contributions

All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflicts of interest.

Appendix

The MATLAB codes used to support the findings of this study are available at the following link:

https://drive.google.com/drive/folders/1VT99mzh51YjtBfR90cFWOfnYWkWpfpIg?usp=sharing

Researchers are welcome to access the repository for implementation details and to facilitate further analysis or replication of the results presented in this paper.

References

[1] S. Abbasbandy, *Improving Newton-Raphson method for nonlinear equations by modied Adomian decomposition method*, Appl. Math. Comput., **145** (2003), 887–893. DOI

[2] M.N.I. Alonso, *Numerical Methods in Quantitative Finance*, SSRN, 2025.

[3] A.H. Altaee, H.K. Hoomod, and K.L. Hussein, *A New Approach to Find Roots of Nonlinear Equations by Hybrid Algorithm to Bisection and Newton-Raphson Algorithms*, Iraqi J. Inf. Technol., **7**(1) (2015), 75–82. DOI

[4] R. Chaudhary, S. Rawat, P. Chauhan, and A.S. Malik, *Unraveling the Roots: A Comprehensive Review of Numerical Methods for Root Finding*, J. Emerg. Technol. Innov. Res., **11**(1) (2024).

[5] J.C. Ehiwario and S.O. Aghamie, *Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root- Finding Problems*, IOSR J. Eng., **4**(4) (2014), 01–07. DOI

[6] GeeksforGeeks, *Regula Falsi Method*, Online (2023). Available at: https://www.geeksforgeeks.org/regula-falsi-method/.

[7] E. Jarlebring, *Broyden's method for nonlinear eigenproblems*, J. Differ. Equ. Appl., (2018). Preprint at http://arxiv.org/abs/1802.07322.

[8] K.A. Kopecky, *Root-finding methods*, Economics, **613**(614) (2007).

[9] N. Na, *Mathematical Economics*, Springer, New York, 2016.

[10] A. Naseem, M.A. Rehman, and T. Abdeljawad, *Real-World Applications of a Newly Designed Root-Finding Algorithm and Its Polynomiography*, IEEE Access, **9** (2021). DOI

[11] A. Naseem, M.A. Rehman, S. Qureshi, and N.A.D. Ide, *Graphical and Numerical Study of a Newly Developed Root-Finding Algorithm and Its Engineering Applications*, IEEE Access, **11** (2023), 2375–2383. DOI

[12] A.W. Nwry, H.M. Kareem, R.B. Ibrahim, and M. Mohammed, *Comparison Between Bisection, Newton and Secant Methods for determining the root of the Non-Linear equation using MATLAB*, Turk. J. Comput. Math. Educ., **12**(14) (2021), 01–07.

[13] P.K. Parida and D.K. Gupta, *An improved regula-falsi method for enclosing simple zeros of nonlinear equations*, Appl. Math. Comput., **177**(2) (2006), 769–776. DOI

[14] K. Pho, *Improvements of the Newton-Raphson method*, J. Comput. Appl. Math., **408** (2022). DOI

[15] A. Pozdnyakov, *Predicting root numbers with neural networks*, Int. J. Data Sci. Math. Sci., **2**(1) (2024), 1–30. DOI

[16] P. Thapliyal, K. Tomar, and C. Solanki, *Role of Bisection Method*, Int. J. Comput. Appl. Technol. Res., **3**(8) (2014), 533–535.

[17] R.R. Vasiljevi, *Numerical methods and their application in dynamics of structures*, Vojnoteh. Glas./Mil. Tech. Courier, **71**(2) (2023), 452–472.

[18] M.A. Wadee, *Nonlinear Mathematics in Structural Engineering*, Math. Today-Bull. Inst. Math. Appl., **43**(3) (2007), 104–108.

[19] C. Wang, K.L. Tan, and C. Lin, *Newton Methods for Convolutional Neural Networks*, ACM Trans. Intell. Syst. Technol., **11**(2) (2020), 1–30. DOI