Numerical Methods: Comparative Analysis of Different Methods for Non-Linear Equations

Carreon, R.M.M.[1], Culaste, G.C.B.[1*], Iradiel, C.A.D.[1], Mozar, R. M.[1], Roy, F.A. Jr[1], Harith, Z.[2**]

[1]College of Engineering, Pamantasan ng Lungsod ng Maynila, Manila, Philippines
[2]FEQS, INTI International University, Persiaran Perdana BBN, Putra Nilai, Malaysia

Email: cgerbhert@gmail.com[*], zuraidah.harith@newinti.edu.my[**]

Abstract

Solving nonlinear equations analytically becomes increasingly complex as functions grow in difficulty or when multiple nonlinear components are involved. This study aims to address that challenge by applying and comparing two well-established numerical methods—the Bisection Method and the False Position Method—in approximating the real roots of nonlinear equations. These iterative techniques are evaluated based on their accuracy, convergence rate, and computational efficiency. Specifically, the study investigates the number of iterations required, the magnitude of relative errors, and the number of significant digits in the final approximations. The results show that while both methods are capable of reaching the desired tolerance, the False Position Method converges faster and yields a higher accuracy score. The findings contribute to the practical selection of numerical methods by providing a comparative analysis that guides users in choosing the most appropriate technique based on the nature of the nonlinear function.

Keywords

Iteration, Bisection, False Position, Newton-Raphson, Secant

Introduction

A system of nonlinear equations consists of two or more equations, with at least one being nonlinear, that are solved simultaneously. Nonlinear equations do not produce a straight line on a graph; instead, they appear as curves with slopes that vary at different points.
The general form of a nonlinear equation is,

$$ax^2 + by^2 = c \tag{1}$$

where a, b, c are constants and a0 and x and y are variables. For example:

$$2x^2 + 3y^2 = 7 \tag{2}$$

$$a^2 + 2ab + b^2 = 0 \tag{3}$$

Submission: 27 January 2025; **Acceptance:** 5 May 2025; **Available online:** August 2025

Analytically solving such systems becomes increasingly difficult as the equations grow in complexity or when multiple nonlinear equations are coupled together. As a result, numerical methods have become essential tools for approximating the solutions of nonlinear systems. These methods aim to find roots of equations through iterative procedures, gradually converging to an acceptable solution within a specified tolerance.

Two widely used numerical techniques for solving single-variable nonlinear equations are the Bisection Method and the False Position Method (Regula Falsi). Both are categorized as bracketing methods, which require that the initial interval contains at least one root (i.e., the function changes sign within the interval). These methods are valued for their reliability and simplicity, particularly when an approximate location of the root is already known.

Objectives of the Study:

To perform a comparative analysis of numerical methods—specifically the Bisection Method and the False Position Method—in solving nonlinear equations, with emphasis on their accuracy, convergence rate, and computational efficiency.

1. To implement the Bisection and False Position methods for solving nonlinear equations.
2. To evaluate the performance of each method in terms of: Number of iterations required to converge, accuracy of the approximated root, and its efficiency.
3. To provide recommendations on the suitable use cases for each method based on the nature of the problem and required precision.

A Bisection Method for Systems of Nonlinear Equations

Eiger, Sikorski, and Stenger (1984) introduced a significant extension of the classical bisection method to systems of nonlinear equations, presenting a multidimensional bracketing technique that guarantees convergence under the assumption of function continuity. Unlike traditional root-finding methods that require derivatives, their approach relies solely on function evaluations, making it robust and suitable for complex or poorly behaved systems. By dividing an initial hyper-rectangle into smaller subregions while preserving the root-bracketing condition, the method iteratively narrows the solution space. This work has influenced numerous subsequent studies in numerical analysis, particularly in interval methods and derivative-free solvers, and remains foundational in the development of reliable and rigorously convergent algorithms (Eiger, Sikorski, & Stenger, 1984).

Nonlinear Equation by Using False Position Method

In their 2016 study, Tasleem and Joythi applied the False Position Method (Regula Falsi) to solve a nonlinear equation modeling the vertical deflection of a wooden bookshelf, using Young's modulus as a key parameter. Their work highlights the method's faster convergence when compared to the classical bisection method, particularly in structural analysis problems where computational efficiency is desirable. The paper emphasizes that while the bisection method is reliable, it often requires more iterations due to its uniform interval halving. In contrast, the false position method's use of linear interpolation leads to quicker root estimation when the function's behavior is favorable. The study demonstrates the method's effectiveness in practical engineering scenarios, supporting its continued use in solving continuous, real-valued nonlinear problems (Tasleem & Joythi, 2016).

Methodology

This section outlines the numerical methods used to solve nonlinear equations: the Bisection Method and the False Position Method. These techniques are iterative and fall under the category of *bracketing methods,* which require the solution to lie within a specified interval.

Numerical Methods

- Bisection Method

In the Bisection Method, we begin with an interval that contains a solution. This interval is then divided in half, and one of the resulting halves will still contain the solution, while the other will not. We continue selecting the half that contains the solution and repeat the process until the interval becomes sufficiently small. However, if no solution is bracketed within the initial interval, the Bisection Method fails to locate one. It is classified as a bracketing method because it requires the solution to be enclosed within the initial interval. The method works by repeatedly solving for the midpoint of the selected interval.

$$N_{n+1} = \frac{N_0 + N_1}{2} \quad (4)$$

- False Position Method

The False Position Method, also known as the Regula Falsi Method, is an improvement over the Bisection Method. Like Bisection, it starts with an interval that contains a root, but instead of simply halving the interval, it estimates the root using linear interpolation.

This approach provides an approximate (or 'false') solution at each step. For the method to work correctly, it is essential that the function's curve remains continuous within the initial interval. The iterative formula is given as follows:

$$N = N_0 - f(N_0) \times \frac{N_1 + N_0}{f(N_1) - f(N_0)} \quad (5)$$

- Algorithmic Approach for Numerical Methods

Algorithms of Bisection Method:

i. Begin the algorithm.
ii. Read the interval endpoints a and b, the error threshold e and the maximum number of iterations N.
iii. Calculate $fa = f(a)$ and $fb = f(b)$.
iv. If $f(a) \cdot f(b) > 0$, it means that a and b do not bracket a root. In this case, output an error message, as this method cannot find the root, hence, print the error and go directly to the end.
v. Set up variables to store the midpoint x and its function value $f(x)$.
vi. Find the midpoint $x = (a + b)/2$ and calculate $f(x)$.

vii. If $fa \cdot fx < 0$, then assign $b \leftarrow x$, $fb \leftarrow fx$, else assign $a \leftarrow x$, $fa \leftarrow fx$.
viii. If $|(a-b)/a|$ is greater than or equal to the error tolerance e, repeat the iteration.
ix. Stop the algorithm.

Algorithms of False Position Method:

i. Begin the algorithm
ii. Read the initial value $x0$, $x1$, error threshold e and set number of iterations.
iii. Compute $fx0 = f(x0)$ and $fx1 = f(x1)$.
iv. If $fx0 \cdot fx1 > 0$, then the interval $[x0, x1]$ does not bracket the root. Print an error message and terminate the process.
v. Declare $x, fx, xprev$ and set the iteration counter $c = 0$.
vi. Calculate the new iteration of x using $x = x0 - f(x0)(x1 - x0)/(f(x1) - f(x0))$
vii. and evaluate $fx = f(x)$.
viii. If $fx0 \cdot fx < 0$ then assign $x1 \leftarrow x$, $fx1 \leftarrow fx$, else assign $x0 \leftarrow x$, $fx0 \leftarrow fx$.
ix. If $c > 0$ and $|xprev - x| < e|$, then assign values.
x. Assign $xprev \leftarrow x$, $c \leftarrow c + 1$ and repeat the iteration.
xi. Output the solution x as the root.
xii. Stop the algorithm.

Application for Numerical Methods

$$0 = \left[(1.7 \times 10^{-19})(73.81971171)\left(\frac{1}{2}\left(N + \sqrt{N^2 + 4(6.21 \times 10^9)^2}\right)\right)(6.5 \times 10^6)\right] - 1$$

$$\rho = \left[(1.7 \times 10^{-19})(73.81971171)\left(\frac{1}{2}\left(N + \sqrt{N^2 + 4(6.21 \times 10^9)^2}\right)\right)(6.5 \times 10^6)\right] - 1$$

$$f(N) = \left[(1.7 \times 10^{-19})(73.81971171)\left(\frac{1}{2}\left(N + \sqrt{N^2 + 4(6.21 \times 10^9)^2}\right)\right)(6.5 \times 10^6)\right] - 1$$

Assume that $N = 0$ and 2.5×10^{10}
Note: The approximate root should satisfy $\in_a \% \leq \in_s \%$; $\in_s \% = 0.5\%$ before making a conclusion. The formula for the relative percentage error is:

$$\in_a \% = \left|\frac{M_{new} - M_{old}}{M_{new}}\right| \times 100\% \quad (6)$$

Bisection Method:

Using equation 4, roots between can be calculated as

$N_0 = 0; f(N_0) = -0.49344544726039 < 0$

$N_1 = 2.5x10^{10}; f(N_1) = 1.15816565251362 > 0$

1st iteration:

$N_2 = \frac{0 + 2.5x10^{10}}{2} = 1.25x10^{10}$

$f(N_2) = 0.22850457598474 > 0$

Roots between

$N_0 = 0\ ; f(N_0) = -0.49344544726039 < 0$

$N_2 = 1.25x10^{10}; f(N_2) = 0.22850457598474 > 0$

$N_2 \rightarrow N_1$

2nd iteration:

$$N_2 = \frac{0 + 1.25x10^{10}}{2} = 6.25x10^9$$

$f(N_2) = -0.17801463227082 < 0$

Using equation 6, relative percent error is:

$$\in_a \% = \left|\frac{6.25x10^9 - 1.25x10^{10}}{6.25x10^9}\right| \times 100\% = 100\%$$

Number of significant digits at least correct to 0.

$|\in_a \%| \leq 0.5 \text{ x } 10^{\,2-\mathrm{m}} \rightarrow 100 \leq 0.5 \text{ x } 10^{\,2-\mathrm{m}} \rightarrow = -0.3010$

Roots between

$N_2 = 6.25x10^9; f(N_2) = -0.17801463227082 < 0$

$N_1 = 1.25x10^{10}; f(N_1) = 0.22850457598474 \quad > 0$

$N_2 \rightarrow N_0$

Note: As we keep repeating the same process of iteration, we will come up with 10th iteration.

10th iteration:

$$N_2 = \frac{0.9082031250x10^{10} + 9130859375}{2} = 9106445313$$

$f(N_2) = -0.00046360875066 < 0$

Using equation 6, relative percent error is:

$$\in_a \% = \left|\frac{9106445313 - 9130859375}{9106445313}\right| = 0.2681\%$$

Number of significant digits at least correct

$$|\in_a \%| \leq 0.5 \text{ x } 10^{\,2-\mathrm{m}} \rightarrow 0.2681 \leq 0.5 \text{ x } 10^{\,2-\mathrm{m}} \rightarrow = 2.2707$$

Therefore, the number of significant digits is at least correct to 2.

The approximate root satisfies: $0.2681\% \leq 0.5\%$. Therefore, the approximate root of the function using the Bisection Method is 9106445313, obtained at the 10th iteration, and is accurate to at least two significant digits: ≈ 9106445313 (10th iteration).

False Position Method:

Using equation 5, $N = N_0 - f(N_0) \times \frac{N_1 - N_0}{f(N_1) - f(N_0)}$, limiting it to 6 decimal places, root between is:

$N_0 = 0; f(N_0) = -0.493445 < 0$

$N_1 = 2.5x10^{10}; f(N_1) = 1.158165 \quad > 0$

1st iteration:

$$N_2 = 0 - (-0.493445) \times \frac{2.5\,x10^{10} - 0}{1.158165 + 0.493445} = 7469146791$$

$N_2 = 7469146791 \;\; ; f(N_2) = -0.104269 < 0$

$N_1 = 2.5x10^{10} \; ; f(N_1) = 1.158165 \;\; > 0$

$N_2 \rightarrow N_0$

2nd iteration:

$$N_2 = 7469146791 - (-0.104269) \times \frac{2.5\,x10^{10} - 7469146791}{1.158165 - (-0.104269)} = 8917082355$$

$f(N_2) = -0.012722 < 0$

Using equation 6, relative percent error is:

$$\in_a \% = \left|\frac{8917082355 - 7469146791}{8917082355}\right| \times 100\% = 16.2378\%$$

Number of significant digits at least correct:

$| \in_a \%| \leq 0.5 \text{ x } 10^{\,2-\text{m}} \rightarrow 16.2378\ 0.5\text{x } 10^{\,2-\text{m}} \rightarrow = 0.4884$

Therefore, the number of significant digits is at least correct to 0.

Roots between

$N_2 = 8917082355; f(N_2) = -0.012722 < 0$

$N_1 = 2.5x10^{10}; f(N_1) = 1.158165 \; > 0$

$N_2 \rightarrow N_0$

3rd iteration:

$$N_2 = 8917082355 - (-0.012722) \times \frac{2.5\,x10^{10} - 8917082355}{1.158165 - (-0.012722)} = 9091827402$$

$f(N_2) = -0.001412 < 0$

Using equation 6, relative percent error is:

$$\in_a \% = \left|\frac{9091827402 - 8917082355}{9091827402}\right| \times 100\% = 1.9220\%$$

Number of significant digits at least correct:

$|\in_a \%| \leq 0.5 \text{ x } 10^{2-\mathrm{m}} \rightarrow 1.9220 \leq 0.5 \text{ x } 10^{2-\mathrm{m}} \rightarrow = 1.4152$

Therefore, the number of significant digits is at least correct to 1.

Roots between

$N_2 = 9091827402; f(N_2) = -0.001412 < 0$

$N_1 = 2.5x10^{10}; f(N_1) = 1.158165 > 0$

$N_2 \rightarrow N_0$

4th iteration:

$$N_2 = 9091827402 - (-0.001412) \times \frac{2.5\,x10^{10} - 9091827402}{1.158165 - (-0.001412)} = 9111198535$$

$f(N_2) = -0.000155 < 0$

Using equation 6, relative percent error is:

$$\in_a \% = \left|\frac{8917082355 - 7469146791}{8917082355}\right| \times 100\% = 0.2126\%$$

Number of significant digits at least correct:

$|\in_a \%| \leq 0.5 \text{ x } 10^{2-\mathrm{m}} \rightarrow 0.2126 \leq 0.5 \text{ x } 10^{2-\mathrm{m}} \rightarrow = 2.3714$

Therefore, the number of significant digits is at least correct to 2.

The approximate relative error is 0.2126%, which is less than the tolerance of 0.5%. Therefore, the approximate root of the function using the False Position Method (FPM) is 9111198535, obtained at the 4th iteration, and accurate to at least two significant digits: ≈9111198535 (4th iteration).

Results and Discussion

This section discusses the outcomes obtained from applying the Bisection and False Position Methods to solve a given nonlinear equation. The performance of each method was evaluated in terms of the number of iterations required for convergence, relative percent error, and the number of significant digits in the approximated root.

Application of the Bisection Method

Table 1.1: Summary table of Bisection Method.

Iteration #	N_0	$f(N_0)$	N_1	$f(N_1)$
1	0	-0.49344544726039	$2.5x10^{10}$	1.15816565251362
2	0	-0.49344544726039	$1.25x10^{10}$	0.22850457598474
3	$6.25x10^9$	-0.17801463227082	$1.25x10^{10}$	0.22850457598474

4	$6.25x10^9$	-0.17801463227082	$0.9375x10^{10}$	0.01702752237385
5	$0.78125x10^{10}$	-0.08292756412361	$0.9375x10^{10}$	0.01702752237385
6	$0.859375x10^{10}$	-0.03350754496888	$0.9375x10^{10}$	0.01702752237385
7	$0.8984375x10^{10}$	-0.0083732881949	$0.9375x10^{10}$	0.01702752237385
8	$0.8984375x10^{10}$	-0.0083732881949	$0.91796875x10^{10}$	0.00429454124806
9	$0.9082031250x10^{10}$	-0.00220476094393	$0.91796875x10^{10}$	0.00429454124806
10	$0.9082031250x10^{10}$	-0.00220476094393	9130859375	0.0011214184718

Table 1.2: Summary table of Bisection Method.

Iteration #	N_2	$f(N_2)$	Update	$\in_a$ %	Correct Sig. Digits
1	$1.25x10^{10}$	0.22850457598474	$N_2 \rightarrow N_1$	N/A	N/A
2	$6.25x10^9$	-0.17801463227082	$N_2 \rightarrow N_0$	100	0
3	$0.9375x10^{10}$	0.01702752237385	$N_2 \rightarrow N_1$	33.3333	0
4	$0.78125x10^{10}$	-0.08292756412361	$N_2 \rightarrow N_0$	20	0
5	$0.859375x10^{10}$	-0.03350754496888	$N_2 \rightarrow N_0$	9.0909	0
6	$0.8984375x10^{10}$	-0.0083732881949	$N_2 \rightarrow N_0$	4.3478	1
7	$0.91796875x10^{10}$	0.00429454124806	$N_2 \rightarrow N_1$	2.1277	1
8	$0.9082031250x10^{10}$	-0.00220476094393	$N_2 \rightarrow N_0$	1.0753	1
9	9130859375	0.0011214184718	$N_2 \rightarrow N_1$	0.5348	1
10	9106445313	-0.00046360875066	$N_2 \rightarrow N_0$	0.2681	2

The Bisection Method began with an interval known to contain the root. Each iteration halved the interval, narrowing down the root's location. After 10 iterations, the method yielded an approximate root of 9106445313, with a relative percent error below 0.5%, and accuracy correct to at least two significant digits. While this method is slow due to uniform interval halving, it is highly reliable and guarantees convergence if the root is bracketed.

Application of the False Position Method

Table 2.1: Summary table of False Position Method.

Iteration #	N_0	$f(N_0)$	N_1	$f(N_1)$
1	0	-0.493445	$2.5x10^{10}$	1.158165
2	7469146791	-0.104269	$2.5x10^{10}$	1.158165
3	8917082355	-0.012722	$2.5x10^{10}$	1.158165
4	9091827402	-0.001412	$2.5x10^{10}$	1.158165

Table 2.2: Summary table of False Position Method.

Iteration#	N_2	$f(N_2)$	Update	$\in_a$ %	Correct Sig. Digits
1	7469146791	-0.104269	$N_2 \rightarrow N_0$	N/A	N/A
2	8917082355	-0.012722	$N_2 \rightarrow N_0$	16.2378	0
3	9091827402	-0.001412	$N_2 \rightarrow N_0$	1.9220	1
4	9111198535	-0.000155	$N_2 \rightarrow N_0$	0.2126	2

The False Position Method also started with the same bracketing interval but used linear interpolation to estimate the root. The method converged in just 4 iterations, yielding an approximate root of 9111198535 with a relative error of 0.2126% and two correct significant digits. The use of slope-based interpolation allowed for faster convergence, especially when the function behaved smoothly within the interval.

Accuracy Computation

Accuracy Criteria/Weight:

f(approx. root): 30% (closer to zero is better)

Relative absolute percent error ($\in_a$%): 30% (smaller is better)

Number of iterations: 20% (smaller is better)

Number of at least Correct Significant digits: 20% (higher is better)

Total weight: 100%

Formula:

$$S = \left[0.30\left(1 - \frac{|f(approx.root)|}{0.000705}\right) + 0.30\left(1 - \frac{\in_a \%}{0.3211}\right) + 0.20\left(1 - \frac{\text{No. of Iterations}}{11}\right) + 0.20\left(\frac{\text{Correct digits}}{4}\right)\right]10$$

where:

$S = Accoracy\ level$ (10 as the highest)

$|f(approx.root)|$ = The absolute value of the $\boldsymbol{f(approx.root)}$ in the table

$\in_a$ % = **Relative absolute percent error**

No. of Iterations = Number of iterations

Correct digits = Number of at least correct significant digits

Accuracy Computation:

Bisection Method:

$$S = \left[0.30\left(1 - \frac{0.00046360875066}{0.000705}\right) + 0.30\left(1 - \frac{0.2681}{0.3211}\right) + 0.20\left(1 - \frac{10}{11}\right) + 0.20\left(\frac{2}{4}\right)\right]10 = 2.7042$$

False Position Method (FPM):

$$S = \left[0.30\left(1 - \frac{0.000155}{0.000705}\right) + 0.30\left(1 - \frac{0.2126}{0.3211}\right) + 0.20\left(1 - \frac{4}{11}\right) + 0.20\left(\frac{2}{4}\right)\right]10 = 5.6269$$

Comparative Analysis

This section presents a side-by-side comparison of the Bisection Method and the False Position Method based on their computed results, including number of iterations, relative percent error, significant digits, and final accuracy scores.

Table 3: Comparison table of the two methods.

Method	Approximate Roots	Iterations	Relative Error (%)	Significant Errors	Accuracy Score
Bisection Method	9106445313	10	0.2681	2	2.7042
False Position Method	9111198535	4	0.2126	2	5.6269

The results of the comparative analysis demonstrate that both the Bisection Method and the False Position Method (FPM) were able to approximate the root of the nonlinear equation to at least two correct significant digits. However, the methods differed significantly in terms of their efficiency and overall performance. The False Position Method achieved convergence in just four iterations with a relative error of 0.2126%, while the Bisection Method required ten iterations to meet the same tolerance threshold. These numerical results are reflected in the computed accuracy scores: 5.6269 for the False Position Method and 2.7042 for the Bisection Method, indicating that FPM outperformed Bisection in terms of computational efficiency and accuracy under the given conditions. Despite this, the Bisection Method maintains an advantage in terms of simplicity and reliability, particularly when applied to functions with irregular or unpredictable behavior. Therefore, the findings support the notion that while the False Position Method is preferable for rapid convergence with smooth and continuous functions, the Bisection Method remains more robust and dependable when dealing with functions that may not conform to ideal conditions. Method selection, thus, should be informed by the specific characteristics of the function being solved.

Conclusion

This study evaluated and compared the performance of the Bisection Method and the False Position Method (FPM) in solving nonlinear equations, with specific emphasis on convergence speed and accuracy. The analysis demonstrated that both methods are reliable root-finding techniques based on bracketing approaches, each with distinct advantages depending on the nature of the function involved.

In terms of convergence speed, the False Position Method significantly outperformed the Bisection Method. FPM achieved the desired tolerance level in only four iterations, whereas the Bisection Method required ten iterations to reach a comparable level of precision. This difference in iteration count reflects the more dynamic nature of the False Position Method, which utilizes linear interpolation to accelerate convergence, in contrast to the uniform interval halving of the Bisection approach.

About accuracy, both methods were able to produce approximations with at least two correct significant digits, satisfying the required error tolerance. However, the computed accuracy score for the False Position Method was 5.6269, more than double that of the Bisection Method's 2.7042, indicating superior performance in terms of overall precision and computational efficiency.

Ultimately, the choice between methods should be based on the characteristics of the function to be solved. The False Position Method is more suitable for continuous and functions where rapid convergence is desirable. In contrast, the Bisection Method remains advantageous for cases involving irregular function behavior, offering greater stability even when fewer assumptions can be made about the function's structure. The findings affirm that while both methods are effective, their efficiency and accuracy vary depending on the problem context.

Acknowledgements

There are no grant or funding bodies to be acknowledged for preparing this paper.

References

Eiger, S., Sikorski, K., & Stenger, F. (1984). A bisection method for systems of nonlinear equations. *ACM Transactions on Mathematical Software (TOMS)*, *10*(4), 367–377. https://doi.org/10.1145/2701.2705

Ersöz, A., & Kurban, M. (2013). Bisection method and algorithm for solving the electrical circuits. In *2nd International Eurasian Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia-Herzegovina*. https://www.researchgate.net/publication/259849595_Bisection_Method_and_Algorithm_for_Solving_The_Electrical_Circuits

Roots of Non-Linear Equation | GRAPHOE. (n.d.). Retrieved April 14, 2025, from https://graphoe.com/resources/numerical-methods/non-linear/

Srivastava, R. B., & Dixit, N. D. (2012). Numerical accuracy of method of false position. International Multidisciplinary Research Journal, 2(1). https://updatepublishing.com/journal/index.php/imrj/article/view/1537

Tasleem, S., & Joythi, S. (2016). *Non-Linear Equation by Using False Position Method*. *International Journal of Scientific Engineering and Technology*, 5(5), 256–258. https://www.researchgate.net/publication/361523850_Non-Linear_Equation_by_using_False_Position_Method