Netrokona University

Department of Computer Science and Engineering

Laboratory Report - 02

False Position Method Implementation

Course: CSE-3212 (Numerical Methods Lab)

Submitted By:

Name: Eyasir Ahamed
Class Roll: 15
Exam Roll: 413
Reg. No: 202004017

Submitted To:

Dr. A F M Shahab Uddin
Assistant Professor
Dept. of CSE
Jashore University of Science & Technology

May 8, 2025

Contents

1 Introduction

The False Position (Regula Falsi) Method is an iterative root-finding algorithm that combines features of the Bisection Method and the Secant Method. This report presents the theoretical background, algorithm implementation, and results of applying the False Position Method to solve nonlinear equations, comparing its performance with the Bisection Method.

2 Theoretical Background

2.1 Mathematical Foundation

The False Position Method improves upon the Bisection Method by using linear interpolation between the interval endpoints rather than simple bisection. Given a continuous function $f(x)$ on $[a, b]$ where $f(a) \times f(b) < 0$, the next approximation is calculated as:

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)} \tag{1}$$

2.2 Convergence Analysis

While the Bisection Method guarantees linear convergence ($\alpha = 1$), the False Position Method typically achieves superlinear convergence ($\alpha \approx 1.618$) when the function is well-behaved.

3 Algorithm Design

Algorithm 1 False Position Method with Interval Validation

Require: Continuous function f, interval $[a, b]$, tolerance ϵ, max iterations N
Ensure: Approximate root c or error message

if $f(a) \times f(b) \geq 0$ **then**
    return "Error: No root in interval [a,b]"
end if
for $iter \leftarrow 1$ to N **do**
    Compute $c \leftarrow \frac{af(b)-bf(a)}{f(b)-f(a)}$
    Evaluate $f_c \leftarrow f(c)$
    if $|f_c| < \epsilon$ **then**
        return c
    end if
    if $f(a) \times f_c < 0$ **then**
        $b \leftarrow c$
    else
        $a \leftarrow c$
    end if
end for
return c (Best approximation)

4 Implementation

4.1 C++ Implementation

```cpp
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

const double epsilon = 1e-6;

double f(double x) {
    return x*x*x - 3*x + 1;
}

void false_position(double a, double b) {
    if (f(a) * f(b) >= 0) {
        cout << "Invalid interval\n";
        return;
    }

    cout << fixed << setprecision(6);
    cout << "Iter\ta\t\tb\t\tc\t\tf(c)\n";
    cout << "-------------------------------------------------\n";

    double c;
    for (int iter = 1; iter <= 50; ++iter) {
        c = (a*f(b) - b*f(a))/(f(b) - f(a));
        double fc = f(c);

        cout << setw(3) << iter << "\t"
             << setw(10) << a << "\t"
             << setw(10) << b << "\t"
             << setw(10) << c << "\t"
             << setw(12) << fc << endl;

        if (abs(fc) < epsilon) break;

        if (f(a)*fc < 0) b = c;
        else a = c;
    }

    cout << "\nApproximate root: " << c << endl;
    cout << "f(root) = " << f(c) << endl;
}

int main() {
    false_position(0.0, 1.0);
    return 0;
}
```

Listing 1: False Position Method Implementation

5 Results and Analysis

5.1 Execution Output

```
Iter    a           b           c           f(c)
-----------------------------------------------
   1   0.000000    1.000000    0.500000    -0.375000
   2   0.000000    0.500000    0.363636    -0.042825
   3   0.000000    0.363636    0.348703    -0.003709
   4   0.000000    0.348703    0.347414    -0.000312
   5   0.000000    0.347414    0.347306    -0.000026
   6   0.000000    0.347306    0.347297    -0.000002
   7   0.000000    0.347297    0.347296    -0.000000

Approximate root: 0.347296
f(root) = -0.000000
```

Figure: Program Output

5.2 Comparison with Bisection Method

Aspect	**False Position Method**	**Bisection Method**
Function	$f(x) = x^3 - 3x + 1$	$f(x) = x^3 - 3x + 1$
Initial Interval	[0, 1]	[-4, 4]
Root Found	≈ 0.347296	≈ -1.879385
Value of $f(\text{root})$	≈ 0	≈ 0
Number of Iterations	7	25
Convergence Speed	Faster for this interval	Slower but guaranteed convergence
Root Location	Between 0 and 1 (rightmost root)	Between -2 and -1 (leftmost root)
Dependence on Interval	Yes, finds root in the interval [0, 1]	Yes, finds root in the interval [-2, -1]
Method Type	Bracketing with interpolation	Bracketing with bisection (midpoint)
Remarks	May converge faster near linear roots	Always converges, slower near flat slopes

Table 1: Comparison between False Position and Bisection Method

6 Conclusion

The False Position Method demonstrated superior convergence compared to the Bisection Method for the function $f(x) = x^3 - 3x + 1$:

- Required only 7 iterations vs 25 for Bisection to reach similar accuracy

- Achieved faster convergence through linear interpolation
- Maintained the bracketing guarantee of the Bisection Method
- Showed the expected superlinear convergence behavior

This implementation confirms the theoretical advantages of the False Position Method while maintaining numerical stability. The method is particularly effective for well-behaved continuous functions where the root can be approximated through linear interpolation.