Kathmandu University
Dhulikhel, Kavre

"False position Method"
[Code No.: MCSC 202]

Submitted by

Nikki Kayastha (22)

BSc. Computer Science

Submitted to

Khim Bahadur Khatri

Submission date: September 26, 2023

1. Introduction

1.1 Background

The regula Falsi method, also referred to as the false position method, is a numerical method for estimating the root of a nonlinear equation. It works by initially determining two points with different signs. The interval containing the root is then iteratively improved using linear interpolation, resulting in a more precise estimate with each iteration.

1.2 Objectives

The objectives of this project are:

- To implement the False Position Method algorithm using programming language.
- To comprehend the underlying theoretical ideas and mathematical foundations of the method.
- To be aware of the advantages and drawbacks of this numerical technique.

2. Theory

2.1 False Position Method

Consider an equation f(x) = 0 , which contains only one variable, i.e. x. To find the real root of the equation f(x)=0 we consider a sufficiently small interval(a,b) where a<b such that f(a) and f(b) will have opposite signs. If f(a).f(b) < 0 then this implies that root is present within the interval. Now, the equation of the chord joining the two points A[a,f(a)] and B[b,f(b)] is given by :

$$y - f(a) = \frac{f(b) - f(a)}{b-a}(x - a)$$

The method consists in replacing the part of the curve between the points [a.f(a)] and [b,f(b)] by means of the chord joining these points, and taking the point of intersection of the chord with the x axis as an approximation to the root. The point of intersection in the present case is obtained by putting y= 0 in the above equation . Thus, we obtain:

$$x1 = a - \frac{f(a)}{f(b)-f(a)}(b - a) = \frac{af(b) - bf(a)}{f(b) - f(a)} \text{——---------i)}$$

which is the first approximation to the root of f(x) = 0 . If now f(x1) and f(a) are of opposite signs, then the root lies between a and x1, and we replace b by x1 in equation 1, and obtain the next approximation. Otherwise, we replace a by x1 and generate the next approximation . The procedure is repeated till the root is obtained to the desired accuracy . The figure below gives the graphical representation of the method.

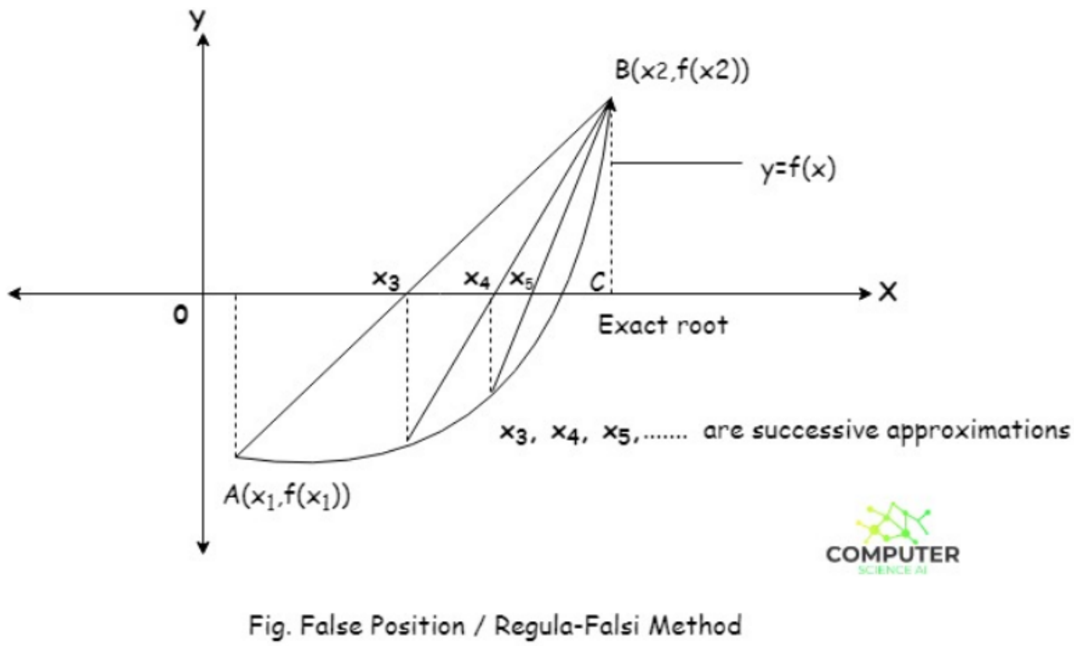

Fig: False Position Method Graph

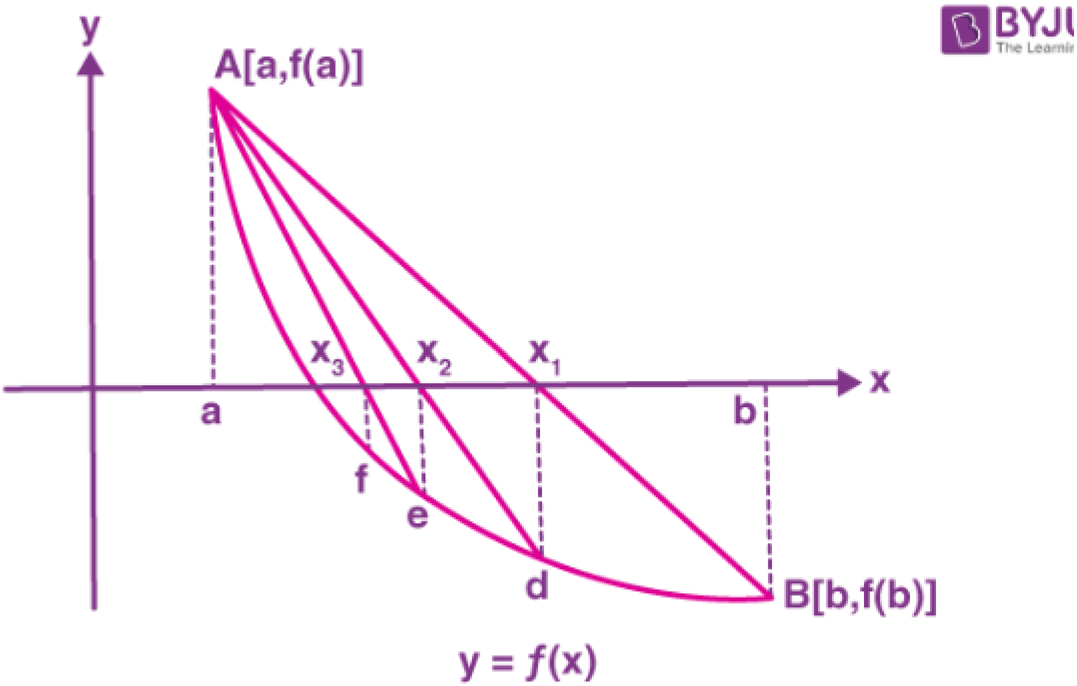

Fig: False Position Method Graph

2.2 Important Notes

Some important notes about the False Position Method are:

- To ensure the presence of a root within that interval [a,b], the method needs an initial interval where the function satisfies (f(a) * f(b) < 0).
- It usually converges slower but is a balance between speed and simplicity.
- The accuracy depends on the initial interval selection and the number of iterations; with enough iterations, it can produce approximations that are reasonably accurate.
- The root estimate is improved repeatedly until the desired level of accuracy is attained.

3. Implementation

```python
def f(x):
    return x**3-5*x-9

def falsePosition(x0,x1,e):
    step = 1
    print('\n\n*** FALSE POSITION METHOD IMPLEMENTATION ***')
    condition = True
    while condition:
        x2 = x0 - (x1-x0) * f(x0)/( f(x1) - f(x0) )
        print('Iteration-%d, x2 = %0.6f and f(x2) = %0.6f' % (step, x2, f(x2)))

        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2

        step = step + 1
        condition = abs(f(x2)) > e

    print('\nRequired root is: %0.8f' % x2)

x0 = input('First Guess: ')
x1 = input('Second Guess: ')
e = input('Tolerable Error: ')

x0 = float(x0)
x1 = float(x1)
e = float(e)

if f(x0) * f(x1) > 0.0:
    print('Given guess values do not bracket the root.')
    print('Try Again with different guess values.')
else:
    falsePosition(x0,x1,e)
```

```
First Guess: 1
Second Guess: 5
Tolerable Error: 0.001


*** FALSE POSITION METHOD IMPLEMENTATION ***
Iteration-1, x2 = 1.500000 and f(x2) = -13.125000
Iteration-2, x2 = 1.941176 and f(x2) = -11.391207
Iteration-3, x2 = 2.281476 and f(x2) = -8.531993
Iteration-4, x2 = 2.514511 and f(x2) = -5.673891
Iteration-5, x2 = 2.660387 and f(x2) = -3.472624
Iteration-6, x2 = 2.746386 and f(x2) = -2.016932
Iteration-7, x2 = 2.795253 and f(x2) = -1.135732
Iteration-8, x2 = 2.822430 and f(x2) = -0.628360
Iteration-9, x2 = 2.837363 and f(x2) = -0.344257
Iteration-10, x2 = 2.845514 and f(x2) = -0.187593
Iteration-11, x2 = 2.849946 and f(x2) = -0.101923
Iteration-12, x2 = 2.852351 and f(x2) = -0.055289
Iteration-13, x2 = 2.853655 and f(x2) = -0.029965
Iteration-14, x2 = 2.854362 and f(x2) = -0.016233
Iteration-15, x2 = 2.854745 and f(x2) = -0.008792
Iteration-16, x2 = 2.854952 and f(x2) = -0.004761
Iteration-17, x2 = 2.855064 and f(x2) = -0.002578
Iteration-18, x2 = 2.855125 and f(x2) = -0.001396
Iteration-19, x2 = 2.855158 and f(x2) = -0.000756

Required root is: 2.85515770
```

3.1 Overview of Code

The false position method for approximating the root of a given function is implemented in the provided code. The false position method, also referred to as regula Falsi, helps us to determine a real-valued function's approximative root. The procedure begins with two initial estimates that must bracket the root and iteratively improves the estimates until it finds a root that is within a given tolerance. The code is explained as follows:

The equation for which we are trying to find the root is represented by the mathematical function f(x) that is defined in the code. It is defined as x^3 - 5*x - 9 in this instance. Any other function for which you want to find the root can be used in place of this one.

The three parameters for the false Position function are:

x0: The initial guess.
x1: The second guess.
e: The acceptable mistake

It starts by asking the user for two initial guesses, x0 and x1, as well as a tolerance, e. By creating a new x2 through a linear calculation based on the function values at x0 and x1, the method incrementally improves its estimate. It keeps doing this until the function's absolute value at x2 is less than e. It prints the current estimate and the corresponding function value after each iteration.

The code first determines whether the initial hypotheses bracket the root (i.e., have opposite signs) before entering the loop. If they don't, it alerts the user to offer alternative hypotheses. Finally, the code invokes the false position method and prints the estimated root if the guesses do indeed bracket the root. In conclusion, the code uses the false position method to iteratively approximate the root of a function while supplying updates, making sure the guesses bracket the root for convergence.

4. Advantages and Disadvantages

4.1 Advantages of False Position Method

1. The interval containing the root is rapidly reduced by the False Position Method, which typically converges faster than the pure bisection method.
2. It ensures the existence of a root within the selected interval by requiring various signs of the function at the interval's endpoints.
3. This technique can be applied to a wide range of functions, including those with complex or discontinuous behavior, because it doesn't rely on understanding the function's derivative.
4. It is reliable and applicable to a wide range of continuous functions, demonstrating stability even for those that have irregular or discontinuous characteristics.

4.2 Disadvantages of False Position Method

1. If the initial interval choice is weak or the function exhibits complex behavior close to the root, the False Position Method may converge slowly or fail to converge.
2. When multiple roots are present in the same interval, it may run into problems, possibly causing slow convergence or convergence to the incorrect root.
3. Since it depends on the continuity of the function within the chosen interval, the method is not suitable for functions with large discontinuities or quickly varying slopes.

4. Conclusion

In conclusion, the False Position Method is a useful technique for locating equation roots. It's advantageous because it typically converges quickly, guarantees there is a root in the initial interval, and doesn't require knowledge of the derivative of the function.

This approach successfully balances simplicity and effectiveness. It continues to be a useful tool in a variety of fields because it offers a dependable way to approximate roots.