CSE330(Numerical Methods)

LECTURE 7 – ROOTS OF EQUATIONS

[BRACKETING METHOD: FALSE POSITION METHOD]

INSTRUCTOR: NAHIAN IBN HASAN

LECTURER, DEPARTMENT OF EEE,

BRAC UNIVERSITY, DHAKA, BANGLADESH

Overview

- Although bisection is a perfectly valid technique for determining roots, its "brute-force" approach is relatively inefficient. False position is an alternative based on a graphical insight.

- A shortcoming of the bisection method is that, in dividing the interval from x_l to x_u into equal halves, no account is taken of the magnitudes of $f(x_l)$ and $f(x_u)$.

- False Position method exploits the graphical insight to join $f(x_l)$ and $f(x_u)$ by a straight line. The intersection of this line with the x axis represents an improved estimate of the root(x_r)

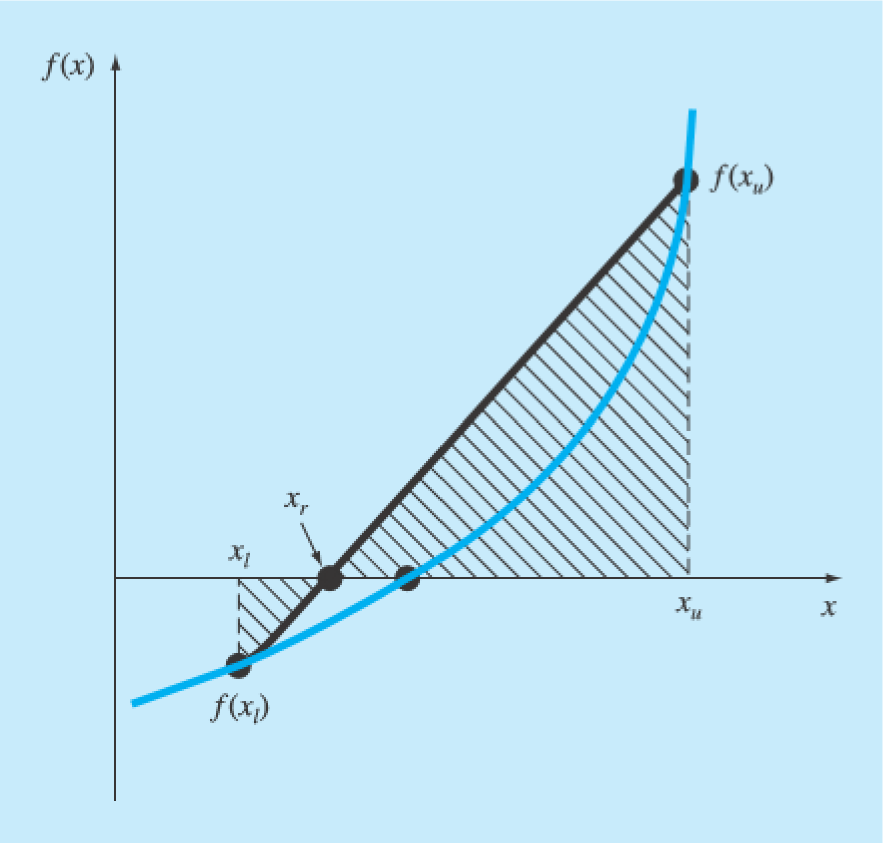
- The fact that the replacement of the curve by a straight line gives a "false position" of the root is the origin of the name, **method of false position**, or in Latin, **regula falsi**. It is also called the **linear interpolation method**.

False Position Method: Estimating Root

- Since, 'False Position Method' is a **Bracketing method** of estimating roots, it will require two initial boundary points (x_l, x_u). We then find the function value at these two points (i.e; $f(x_l)$ and $f(x_u)$).

- We then join the two points [x_l, $f(x_l)$] and [x_u, $f(x_u)$] with a straight line. The joining line intersects the x-axis at a point. The intersection point is referred to as our new root estimation(x_r).

- The rest of the process is similar to Bisection method.



False Position Method: Estimating Root

❑ How do we find out the x_r, intersection point?

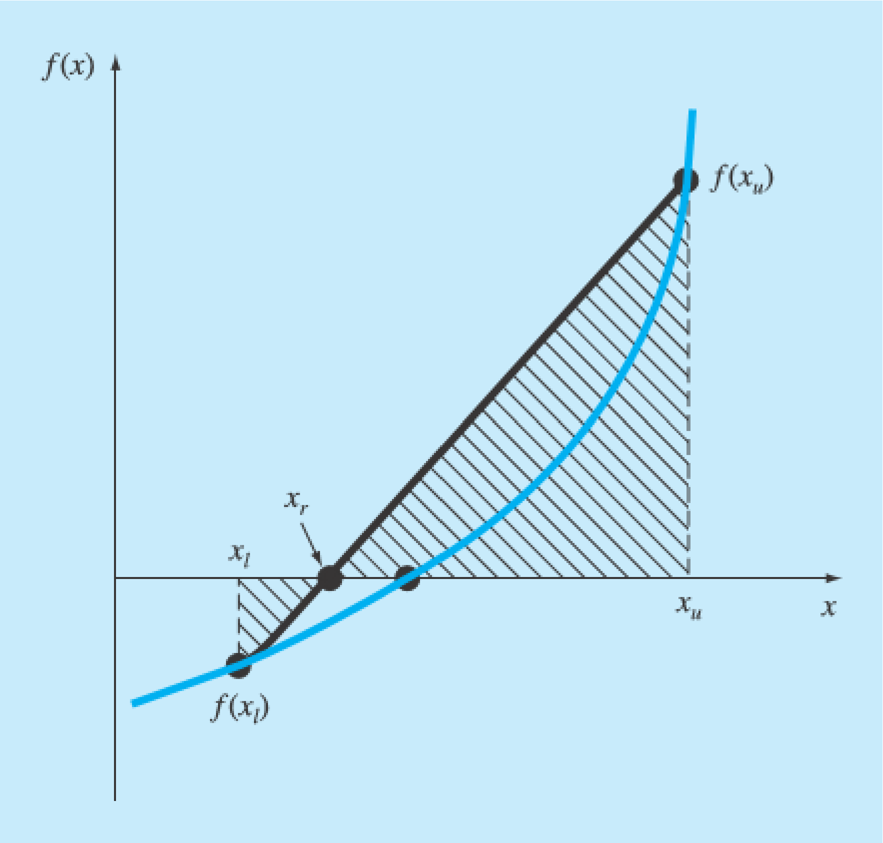
- ❑ The value of the slope of the straight line determined from both of the upper triangle and the lower triangle should be same.

Slope in upper triangle = $\frac{height}{base} = \frac{f(x_u)-0}{x_u - xr} = \frac{f(x_u)}{x_u - xr}$

Slope in lower triangle = $\frac{height}{base} = \frac{0 - f(x_l)}{x_r - xl} = \frac{-f(x_l)}{x_r - xl}$

The slopes should be similar. Hence,

$$\frac{f(x_u)}{x_u - xr} = \frac{-f(x_l)}{x_r - xl}$$



False Position Method: Estimating Root

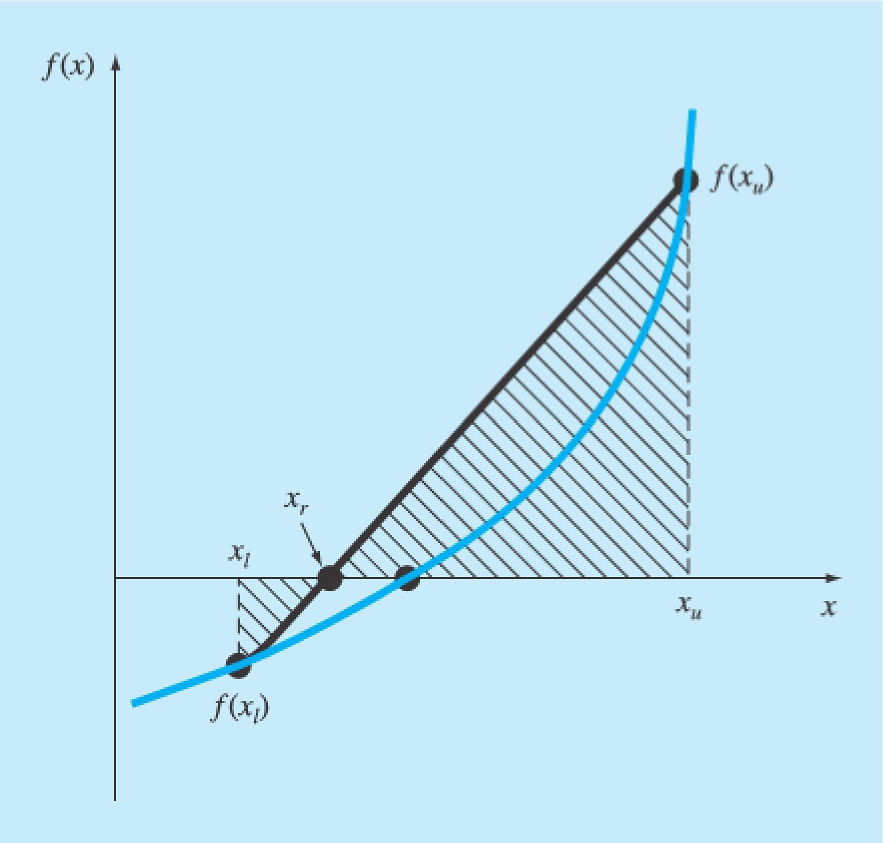
❑ How do we find out the x_r, intersection point?

- ❑ Cross multiplying both sides of the equation and solving for x_r, we would get the following equation –

$$x_r = \frac{x_u f(x_l) - xlf(xu)}{f(x_l) - f(xu)} \text{......................(1)}$$

Equation (1) can be modified further into the following form [You derive at home how the following equation can be deduced from equation 1]-

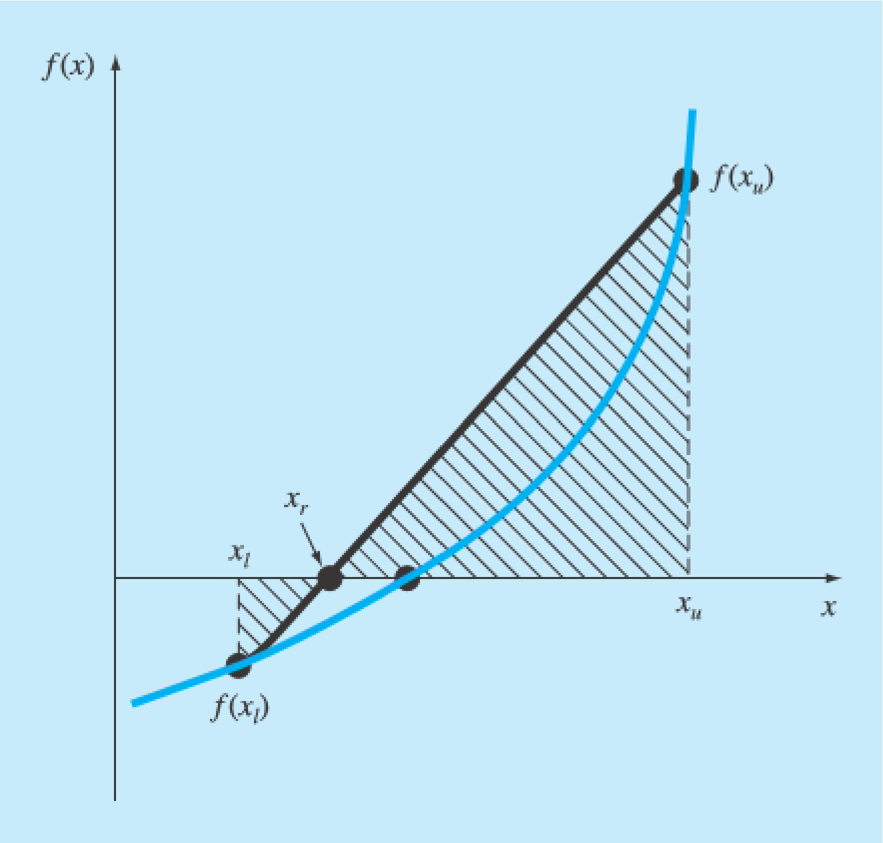
$$x_r = x_u - \frac{f(x_u)[xl - x_u]}{f(x_l) - f(xu)} \text{........................(2)}$$



False Position Method: Estimating Root

- Any one of equation 1 and equation 2 can be used as the '**False Position Formula**' for determining the root.

- Equation 2 is better from the perspective of computational resource allocation. Because it involves one less function evaluation and one less multiplication.



False Position Method: Algorithm

Step 1 :

Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)\ .\ f(x_u) < 0$. Also, you can take help using graphical approach from lesson 5.

Step 2:

An estimate of the root x_r is determined by $x_r = x_u - \dfrac{f(x_u)[xl - xu]}{f(x_l) - f(xu)}$

Step 3:

Make the following evaluations to determine in which subinterval the root lies:

a) If $f(x_l).f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

b) If $f(x_l).f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

c) If $f(x_l).f(x_r) = 0$, the root equals x_r; terminate the computation.

False Position Method: Termination Criteria

- The step 3 of false-position method mentions the termination criteria. But, the code only terminates when the x_r exactly overlaps with the true value. Otherwise, the code will continue until the computer memory is fully consumed.

- Hence, we always calculate the approximate error percentage in every iteration using the **percentage approximate relative error** from lesson 2.

$$\epsilon_a = \left| \frac{x_r^{new} - xrold}{x_r^{new}} \right| \times 100\%$$

Here, x_r^{new} is the root for the present iteration, and x_r^{old} is the root for the previous iteration. When ϵ_a is lower than a pre-specified tolerance limit of ϵ_s , then the iteration stops and x_r^{new} is considered as the root.

- In the exam, you will not have access to any computer, hence, you will have to try for various values of x_l and x_u initially, so that $f(x_l) \, . \, f(x_u) < 0$.

False Position Method: Example

Example 1:

Find out the root of the equation f(x) = x^2 - $8x$ + 12 using False Position method.

Solution:

Iteration 1:

- Initially we need to take two approximate values x_l and x_u. Say, x_l = 1 and x_u = 4.
- Hence, $f(x_l)$ = 1 – 8 + 12 = 5 and $f(x_u)$ = 16 – 8*4 + 12 = -4. Since, $f(x_l)$. $f(x_u)$ < 0, our approximation is correct. So, we have a root between these two points [1,4].
- Now, new approximated root, $x_r = 4 - \frac{-4 \times (1-4)}{5-(-4)} = 2.666667$
- $f(x_r)$ = f(2.5) = -2.22222
- $f(x_l)$. $f(x_r)$ = 5 * - 2.22222 = -11.111111 < 0. Hence, the root is in the lower half. So, we set $x_u = x_r$ = 2.666667

False Position Method: Example

Iteration 2:

- Now, after first iteration, we have $x_l = 1$ and $x_u = 2.666667$
- Hence, $f(x_l) = f(1) = 5$ and $f(x_u) = f(2.666667) = -2.22222$

- New approximated root, $x_r = 2.666667 - \frac{-2.22222 \times (1-2.666667)}{5-(-2.22222)} = 2.15384674$
- $f(x_r) = f(2.15384674) = -0.591718139$

- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(2.15384674 - 2.666667)/\ 2.15384674| \times 100\% = 23.81\%$
- $f(x_l)\ .\ f(x_r) = 5 * -0.591718139 < 0$. Hence, the root is in the lower half. So, we set $x_u = x_r = 2.15384674$

False Position Method: Example

Iteration 3:

- Now, after first iteration, we have $x_l = 1$ and $x_u = 2.15384674$
- Hence, $f(x_l) = f(1) = 5$ and $f(x_u) = f(2.15384674) = -0.591718139$
- New approximated root, $x_r = 2.15384674 - \frac{-0.591718139 \times (1-2.15384674)}{5-(-0.591718139)} = 2.031746157$
- $f(x_r) = f(2.031746157) = -0.125976808$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(2.031746157 - 2.15384674)/ 2.031746157| \times 100\% = 6.01\%$
- $f(x_l) \,.\, f(x_r) = 5 * -0.125976808 < 0$. Hence, the root is in the lower half. So, we set $x_u = x_r = 2.031746157$

False Position Method: Example

Iteration 4:

- Now, after first iteration, we have $x_l = 1$ and $x_u = 2.031746157$
- Hence, $f(x_l) = f(1) = 5$ and $f(x_u) = f(f(2.031746157) = - 0.125976808$
- New approximated root, $x_r = 2.031746157 - \frac{-0.125976808 \times (1-2.031746157)}{5-(-0.125976808)} = 2.006389802$
- $f(x_r) = f(2.006389802) = -0.025518378$
- $\epsilon_a = |(x_r^{new} - x_r^{old})/x_r^{new}| \times 100\% = |(2.006389802 - 2.031746157)/ 2.006389802| \times 100\% = 1.264\%$
- $f(x_l) . f(x_r) = 5 * -0.025518378 < 0$. Hence, the root is in the lower half. So, we set $x_u = x_r = 2.006389802$

False Position Method: Example

Result:

- Observe the value of approximate error (ϵ_a). The error decreases with increasing iterations. If we continue with more iterations, we will see that the error decreases further. In our code, we will set a pre-specified value ϵ_s . When the error ϵ_a gets lower than ϵ_s , we would consider the last iteration's x_r as our result. For example, if ϵ_s = 2%; then we see in the 4th iteration our error (ϵ_a) is lower (1.264%). Hence, we consider x_r in the 4th iteration (2.006389802) as our final root.

- Obviously, if we want more precision, we would set the ϵ_s at lower value. Then more iterations will be needed. But we will have better approximation to the real root.

Comparison Between Bisection and False Position Method

- We have solved the same problem using bisection method in lesson 6. we see that for this problem the false position method converges faster than the bisection method. In the 4^{th} iteration the false position method has an error rate of 1.264%; whereas even in the 5^{th} iteration of bisection method the error was still greater than 4%.

- **For most of the cases we prefer False Position Method.**

- You might want to draw the error propagation throughout iterations using MATLAB and compare the result between bisection and false position method.

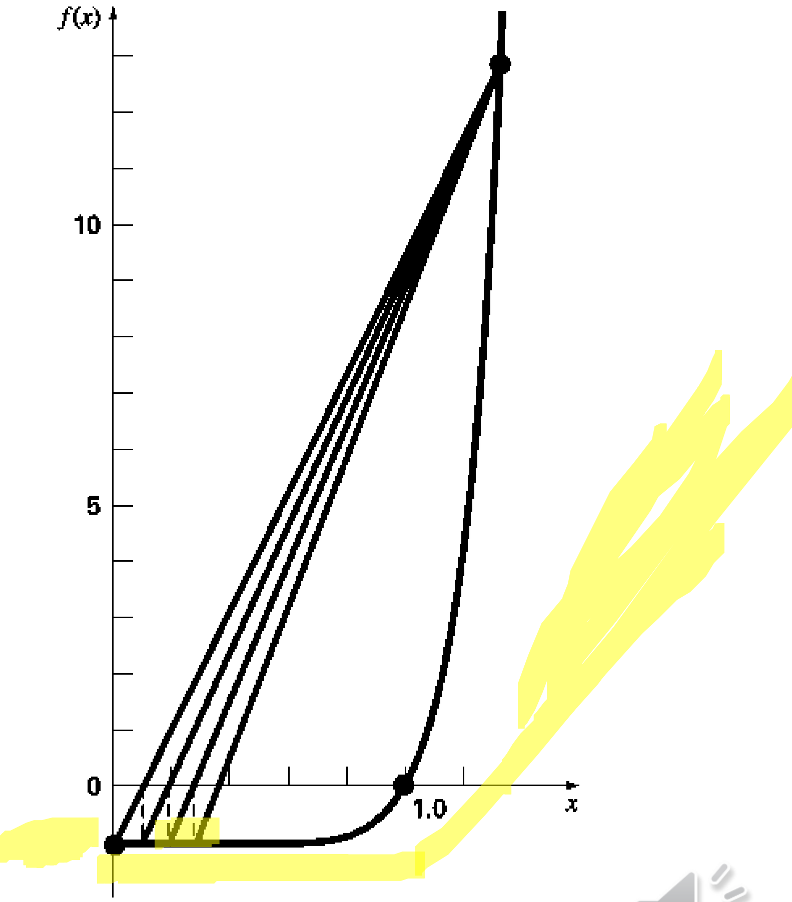
False Position Method: Limitations

- Although the false-position method would seem to always be the bracketing method of preference, there are cases where it performs poorly.
- Also, the method can find additional roots using different initial guesses of x_l and x_u like the bisection method.

- Sometimes, bisection method performs better than the False Position method. For example, in the figure, bisection method would perform better than false position method.

- **Modified False Position Method**: One way to mitigate the "one-sided" nature of false position (like in the figure) is to have the algorithm detect when one of the bounds is stuck. If this occurs, the function value at the stagnant bound can be divided in half. This is called the modified false-position method.

- **Remember:** In using **modified False Position method**, we halve the function value at the stagnant bound, not the stagnant bound itself.



Conclusion

In this lesson we have

- learnt the algorithm of false position method for finding roots of a function
- implemented the false position algorithm for solving a function practically
- known about the limitations of false-position algorithm
- learned about a remedy of one-sided function (**modified false position method**)
- There is no better method. Bisection method is better in solving some problems than false position method (vice versa)

- Please try out the MATLAB code given with this lesson implementing the false position and modified false position method.

In the next lesson we will learn about incremental search. Incremental search is important for determining the initial guesses of x_l and x_u in both bisection and false position method.

Thank You