

# **Information Technology**

### **Numerical Methods**

Module 3: Iterative Methods to solve equation f(x) = 0:

### **Method of False Position**

### 3.1 Introduction

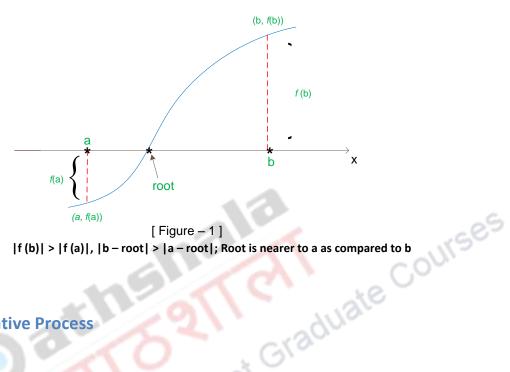
In our previous lecture (module), we learned about nature of iterative methods, classification of iterative methods (bracketing or open) to find roots of an equation f(x) = 0. We began with Bisection method to find roots of f(x) = 0. It was observed that through Bisection method is a very simple, steady and reliable method, it suffers from drawback of being slow. It would be advisable to seek its modification to make it faster. The regular falsi method or method of false position is an attempt in that direction. This also is closed method. As in, Bisection method, at the beginning of iterative process two end points a and b are found so that  $f(a) \cdot f(b) < 0$  to ensure that at least one root lies between a and b.

It was also pointed out in disadvantages of Bisection method that it does not take into account of magnitude of function values at end points to generate estimates for the next iteration. It just considers sign of function values. This makes it slow.

Thus, in method of false position, the function values at the end points a and b are taken into account to generate the estimates for next iteration.

Method of false position differs with Bisection method only in the methodology of finding c. In Bisection Method, interval was subdivided into equal parts, c was mid point given by (a+b)/2.

No account was taken of function values at a and b. As at the root, function value is zero, the point, where function value is smaller is expected to be nearer to the root. It can be visualized also from the graph below.



|f(b)| > |f(a)|, |b - root| > |a - root|; Root is nearer to a as compared to b

#### 3.2 **Iterative Process**

So, in regular falsi method, to take into consideration the function values at a and b, straight line is drawn, joining (a, f(a)) and (b, f(b)).

The point, where it cuts X axis is, the new estimate of the root.

Mathematically, estimate of formula for c can be derived as follows:

Equation of straight line joining (a, f(a)) and (b, f(b)) is given by :

$$y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a)$$
 ... (1)

Point of intersection of straight line with X - axis being on X-axis is (c, 0).

Thus (c, 0) satisfies equation (1). Substituting (c, 0) in place of (x, y) in equation (1) gives

$$0 - f(a) = \frac{f(b) - f(a)}{b - a}(c - a)$$

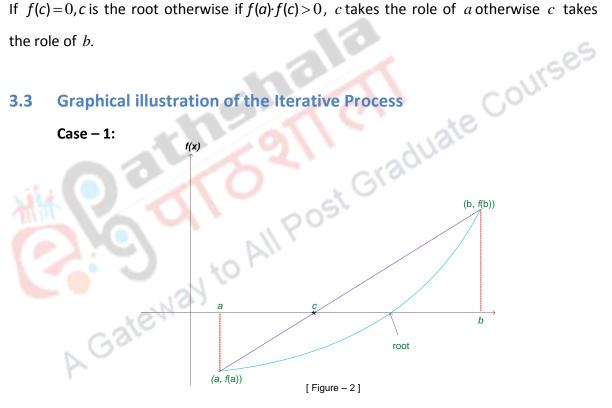
$$\therefore (c-a) = \frac{-(b-a)f(a)}{f(b) - f(a)}$$

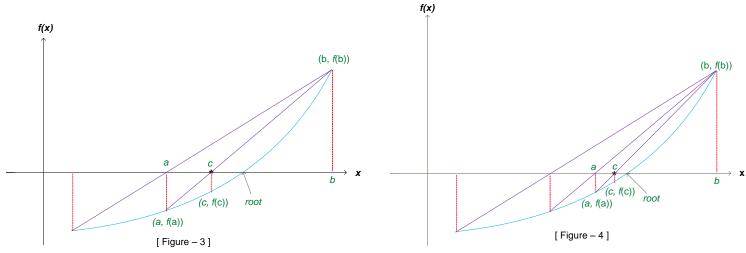
$$c = a - \frac{(b-a)f(a)}{f(b) - f(a)}$$

$$c = \frac{a f(b) - b f(a)}{f(b) - f(a)}$$
 ... (2)

If f(c) = 0, c is the root otherwise if  $f(a) \cdot f(c) > 0$ , c takes the role of a otherwise c takes the role of b.

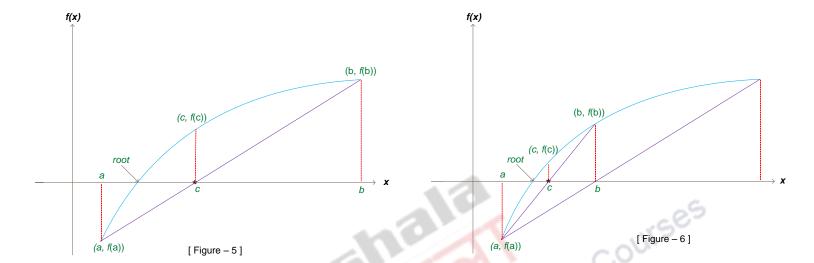
### 3.3





f(c) < 0;  $f(a) \cdot f(c) > 0$  ... c takes the role of a

### Case - 2:



f(c)>0;  $f(b)\circ f(c)>0$  . . c takes the role of b

# 3.4 Different Stopping Criterion

Now, once we have observed the iterative process, let us focus on different stopping criterions. The sequence of  $c_k$ 's approaches the root. That is why, in case of the Bisection section algorithm, iterative process was stopped, when

$$|c_{k} - c_{k-1}| < \epsilon$$
 ... (3)

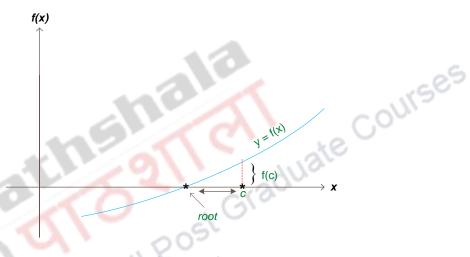
It is called x – tolerance criterion (**X-TOL**). The recent most  $c_k$  is the estimate of the root. There is another stopping criterion called function tolerance (**F-TOL**). At root, function value is zero, so if estimate is quite near the root then function value would be small. Hence, many times, one would like to stop when

$$|f(c_k)| < \delta ... (4)$$

where  $\delta$  is some small positive constant like  $\in$  .

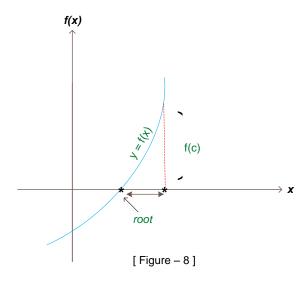
You would like to ask: which one is better or what basis decision should be made to choose the particular criterion? Well, it depends on the method, nature of function near the root, objective of finding the root etc. Many times, a combination of different criterions are also applied.

If slope of the curve is small near the root, curve is almost horizontal, and then function tolerance may not be appropriate stopping criterion, because curve is rising slowly, function values in neighborhood of the root are going to be small, so even if estimate c is not sufficiently near the root, one may stop. This is visible from the following graph:



[ Figure – 7 ]

On the other hand, if it is expected that slope is high near the root or sequence of approximations  $c_{\rm k}$ 's may converge to the root, like in case of almost vertical graph, then function tolerance ensures that estimate is good approximation to the root.



In simple words, for f(x), if |f'(x)| is high near the root, FTOL would be better to use as stopping criterion.

Many times instead of absolute error  $|c_k - c_{k-1}|$ , bound on Relative error  $\frac{|c_k - c_{k-1}|}{|c_k|}$  is

used .This needs to be applied when looking answer correct to certain number of significant digits. For example, the root could be like  $1.2749 \times 10^{-12}$ 

So here, if we go for  $|c_k - c_{k-1}| \le \frac{1}{2} \times 10^{-5}$ , we shall get answer zero, but if we go

for  $\frac{|c_k - c_{k-1}|}{|c_k|} \le \frac{1}{2} \times 10^{-5}$ , we would get root correct to 5 significant digits. It root is required

to be correct to N significant digits, then one should apply

$$\frac{\left|c_{k}-c_{k-1}\right|}{\left|c_{k}\right|} \le \epsilon$$
, where  $\epsilon = \frac{1}{2} \times 10^{-N}$ 

In sum, any or more than one of the following criterions can be selected to exit the iterative loop:

X-TOL: 
$$|c_k - c_{k-1}| \le c \dots (5)$$

F-TOL: 
$$|f(c_k)| \le \delta$$
 ... (6)

Relative Error: 
$$\frac{\left|c_{k}-c_{k-1}\right|}{\left|c_{k}\right|} \leq \in \qquad ... (7)$$

It may be noted that none of above inequalities can be used to interpret the actual error  $|c_{\mathbf{k}}$  -root.



### **Algorithm for Regular False Method** 3.5

Step 1 Choose a and b such that  $f(a) \cdot f(b) < 0$ 

Step 2 Input 
$$\in$$
,  $\delta$ ,  $a$ ,  $b$ 

Step 3 
$$c_0 = a, k = 1$$

Step 4 
$$c_k = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

Evaluate  $f(c_k)$ , If  $f(c_k)=0$ ;  $c_k$  is the desired root, exit Step 5

Step 6

Step 6 Otherwise, evaluate 
$$f(a) \cdot f(c_k)$$
. If  $f(a) \cdot f(c_k) > 0$ ,  $a = c_k$ ; else  $b = c_k$ 

Step 7 If  $|c_k - c_{k-1}| < \varepsilon$  or / and  $|f(c_k)| < \delta$  or / and  $\frac{|c_k - c_{k-1}|}{|c_k|} \le \varepsilon$ 

Step 8 Output: estimate of the desired root as value of  $c_k$ , Exit

ುot as value c Step 8

Step 9

#### 3.6 Illustrations

Let us solve examples to find root of an equation using method of false position. For comparison purpose, we shall find the roots of the same equations, used for illustrating Bisection Method.

# Example 1:

$$f(x) = 3x - \cos x - 1$$
Take  $a = 0, f(a) = -2.0, c_0 = a$ 

$$b = 1, f(b) = 1.459698$$

$$f(a).f(b) < 0$$

k	а	f(a)	b	f(b)	Ck	f(c <sub>k</sub> )	C <sub>k</sub> - C <sub>k-1</sub>
1	0.000000	-2.000000	1.000000	1.459698	0.578085	-0.103255	0.578085
2	0.578085	-0.103255	1.000000	1.459698	0.605959	-0.004081	0.027873
3	0.605959	-0.004081	1.000000	1.459698	0.607057	-0.000159	0.001099
4	0.607057	-0.000159	1.000000	1.459698	0.607100	-0.000006	0.000043
5	0.607100	-0.000006	1.000000	1.459698	0.607102	-0.000000	0.000002
6	0.607102	-0.000000	1.000000	1.459698	0.607102	-0.000000	0.000000
Example 2:							

$$f(x) = x^3 + x - 1$$
  
Take  $a = 0$ ,  $f(a) = -1$ ,  $c_0 = a = 0$   
 $b = 1$ ,  $f(b) = 1$ 

k	а	f(a)	b	f(b)	C <sub>k</sub>	f(c <sub>k</sub> )	C <sub>k</sub> - C <sub>k-1</sub>
1	0.000000	-1.000000	1.000000	1.000000	0.500000	-0.375000	0.500000
2	0.500000	-0.375000	1.000000	1.000000	0.636364	-0.105935	0.136364
3	0.636364	-0.105935	1.000000	1.000000	0.671196	-0.026428	0.034832

k	а	f(a)	b	f(b)	Ck	f(c <sub>k</sub> )	C <sub>k</sub> - C <sub>k-1</sub>
4	0.671196	-0.026428	1.000000	1.000000	0.679662	-0.006376	0.008466
5	0.679662	-0.006376	1.000000	1.000000	0.681691	-0.001525	0.002029
6	0.681691	-0.001525	1.000000	1.000000	0.682176	-0.000364	0.000485
7	0.682176	-0.000364	1.000000	1.000000	0.682292	-0.000087	0.000116
8	0.682292	-0.000087	1.000000	1.000000	0.682319	-0.000021	0.000028
9	0.682319	-0.000021	1.000000	1.000000	0.682326	-0.000005	0.000007
10	0.682326	-0.000005	1.000000	1.000000	0.682327	-0.000001	0.000002
11	0.682327	-0.000001	1.000000	1.000000	0.682328	-0.000000	0.000000

# **Programming Code in C:**

The above output was obtained by running the following C program.

```
Post Graduate Courses
//Regular falsi method
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define fnx(x) pow (X, 3)-X+1
void main()
       float a, b, c,fa, fb, fc, error,co;
       int i=0;
       int count=0;
       float err = pow(10.0,-6.0)/2;
              printf("\nEnter initial value of a = ");
              scanf("%f", &a);
              fa = fnx(a);
              printf("\n value of f(a) = %f", fa);
              printf("\nEnter initial value of b = ");
              scanf("%f", &b);
              fb = fnx(b);
              printf("\n value of f(b) = %f", fb);
              co=a;
       printf("\nNumber\t A\t F(A)\t B\t F(B)\t C\t F(C)\t Error");
```

```
{
              i++;
              c = (a*fb - b*fa) / (fb - fa);
              fc = fnx(c);
              if(fc==0)
                      printf("\ndesired root is %f", c);
                      //exit(0);
              error=fabs(co-c);
              co=c;
              printf("\n\n\%d\t \%f\t \%f\t \%f\t \%f\t \%f\t \%f', i, a, fa, b, fb, c, fc, error);
                              to All Post Graduate Courses
              if(fa*fc >0)
                      a = c;
                      fa= fc;
              }
              else
                      b = c;
                      fb = fc;
       }while(error >err);
       printf("\n\nThe root of the equation = %f", c);
       getch();
}
```

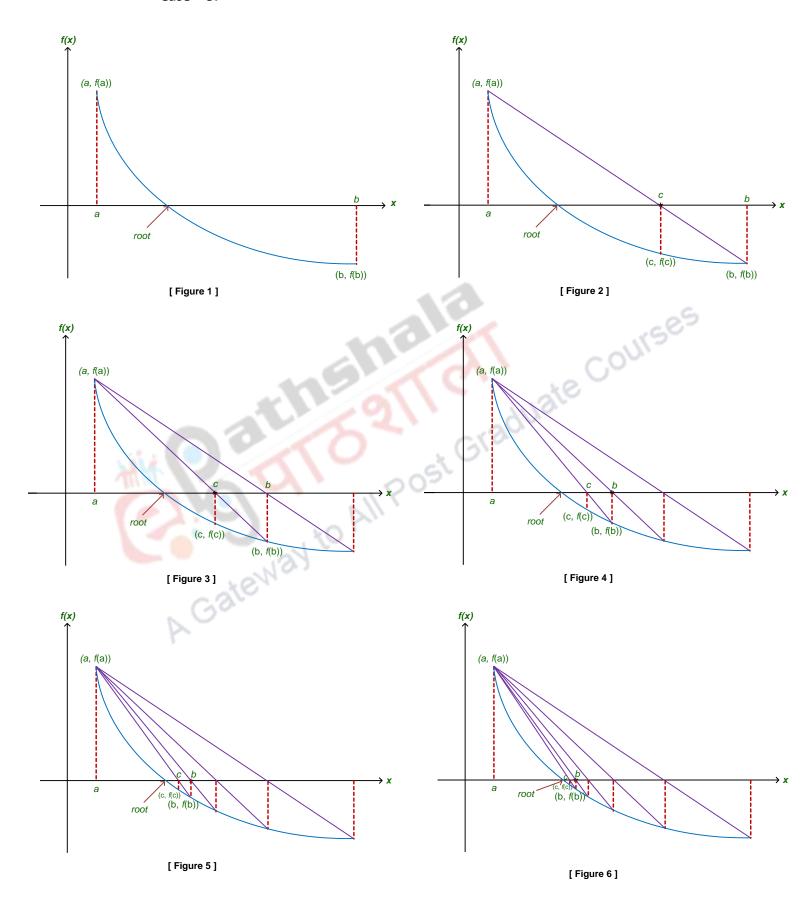
### 3.7 Observations

1) From the tables, it is clear that same desired accuracy is obtained in less number of iterations. Question is, is this the case for these two specific equations, or in general we can expect False Position Method to perform better as compared to Bisection Method. Well, in general method of false position is expected to converge faster to the root than the Bisection Method. That should be also, as to generate next approximation, it takes account of function values by the formula

$$c_k = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

and  $c_k$  is just not the mid point of the interval.  $c_k$  is the point, where straight line joining the two points (a, f(a)) and (b, f(b)) on the curve intersects with X axis. One very important measure of speed of convergence is "Order of Convergence". We shall discuss over "Order of Convergence" in one of next modules. Method of False Position is regarded as having superlinear convergence.

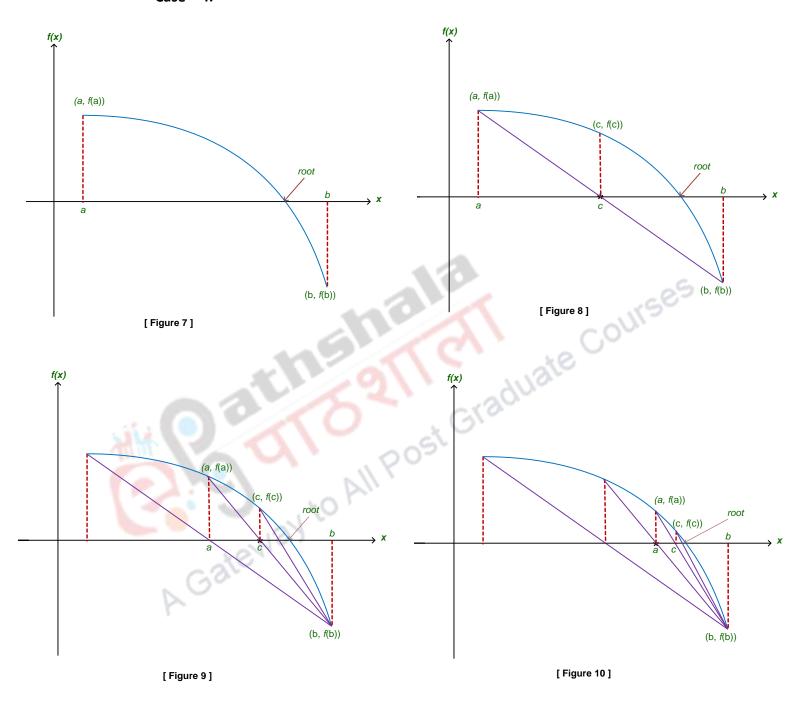
It can be seen, from the Figures [2,3,4,5,6], as well as computational tables above, one endpoint has become fixed. The moment curve does not change its shape from concave to convex or vice versa in the bracketing interval [a, b], always in this method,  $c_k$ 's will approach the root from one side. As a result  $c_k$ 's successively replace one of the two a's or b's, as the case may be, and the other remains fixed. For example, in example 1 as well as example 2, from the very beginning b is remaining constant. Other shape figures are also presented below for completeness.



[a remains fixed throughout the process. Only b is changing, sequence of c's that is b's approach the root]



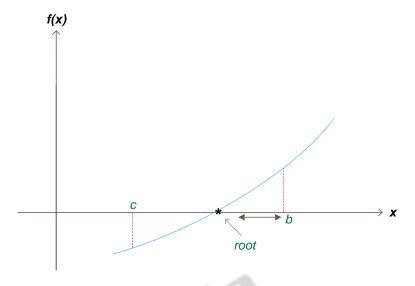




[ b remains fixed, a keeps on changing, sequence of  $c_{k}$  's (a's) approach the root ]

Thus, as  $c_k$ 's approach the root only from one side of the root, interval length no more approaches zero, it will always be greater than either |a - root| or |b - root| depending upon which of the endpoint is fixed.

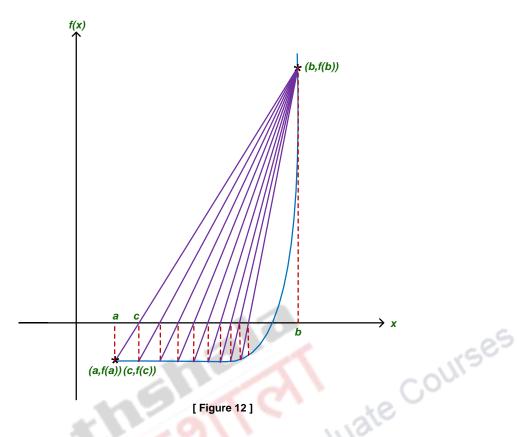




[ Figure – 11 ]

Thus, no more number of intervals can be predetermined to achieve desired accuracy.

2) There could be cases, though rare, where Method False Position may get slower as compared to Bisection Method. For example, look at the following graph of the equation  $f(x) = x^{10} - 1 = 0$ .



[ False position method is slower than Bisection method ]

In this example, sequence of approximations is converging slowly.  $c_k - c_{k-1}$  is becoming small, though not sufficiently near the root. The reason being, slope of the curve is large, at and in neighborhood of the root. That is why; many solvers apply F – TOL in conjunction with X–TOL stopping criterion to find roots of an equation using method of False Position Sequence Following solution tables give comparison of Bisection Method and method of False Position in this specific example, which confirms that for this example, Bisection Method is better.

### Table:

$$f(x) = x^{10} - 1$$

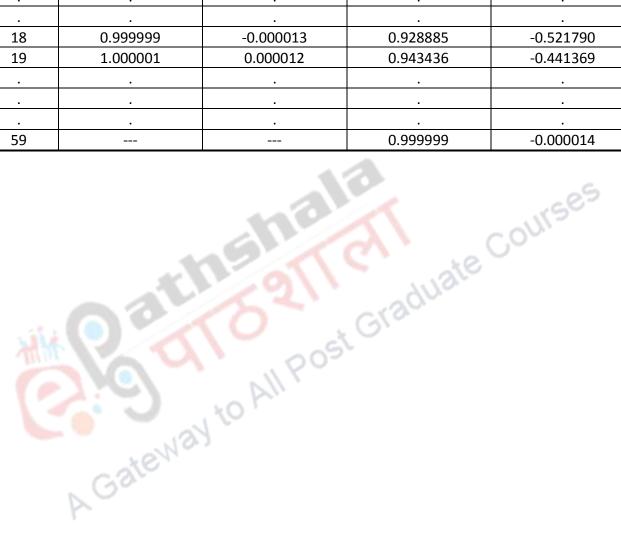
$$a = 0, f(a) = 1.00000$$

$$b = 1.3, f(b) = 12.785842$$

# **Comparison Table**

k	Bisection Method	False Position Method

	Ck	f(c <sub>k</sub> )	Ck	f(c <sub>k</sub> )
1	0.650000	-0.986537	0.094300	-1.000000
2	0.975000	-0.223671	0.181759	-1.000000
3	1.137500	2.626718	0.262874	-0.999998
	•	•	•	•
	•	•	•	•
	•	•	•	•
	•	•	•	•
18	0.999999	-0.000013	0.928885	-0.521790
19	1.000001	0.000012	0.943436	-0.441369
	•	•	•	•
	•	•	•	•
	•	•	•	•
59			0.99999	-0.000014



# **Advantages:**

- 1) It is faster than Bisection Method.
- 2) It is also simple.
- 3) It guarantees convergence.
- 4) Only one function evaluation per iteration is required.

# **Disadvantages:**

- 1) Not self starting. One needs two initial guesses a and b such that  $f(a) \cdot f(b) < 0$ .
- 2) Though, faster than Bisection Method, still regarded as slow.
- 3) In rare cases, it may become slower than Bisection Method.

