# Solving polynomials

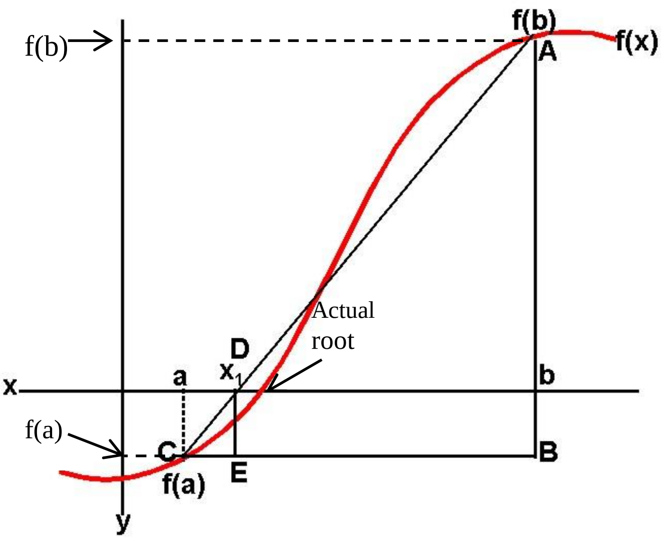
Part-2

## Regula–Falsi Method

It is a numerical method for estimating the roots of a polynomial **f(x)**. It is similar to Binary bisection method. Let one of the roots of the polynomial lie in the interval [a, b]. Here, the values of a and b are such that $a < b$ and only one root lies in the range.

The root is that value of *x* for which $f(x) = 0$. For a continuous polynomial, it means that the curve of f(*x*) vs *x* crosses x-axis at this value of *x*. This also implies that the sign of the functions on the two sides of *x* are different, on one side it is +ve and on the other side it is negative. So, if the root lies in the range [a, b], then, the product of f(a)and f(b) is less than zero.

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




In the figure, the red curve is the plot of the function f(x) for different values of x. The root is where the curve is crossing x-axis. a and b are lower and higher values respectively, of the range. Draw a line joining points (a,f(a)) to (b, f(b)). This line AC is known as **interpolation line.** It crosses the x-axis at $x_1$. Here, the value of y is zero. Draw the lines joining various points as shown in the figure. Observe the triangles DEC and ABC which are similar triangles. Therefore,

$$EC / BC = DE / AB$$

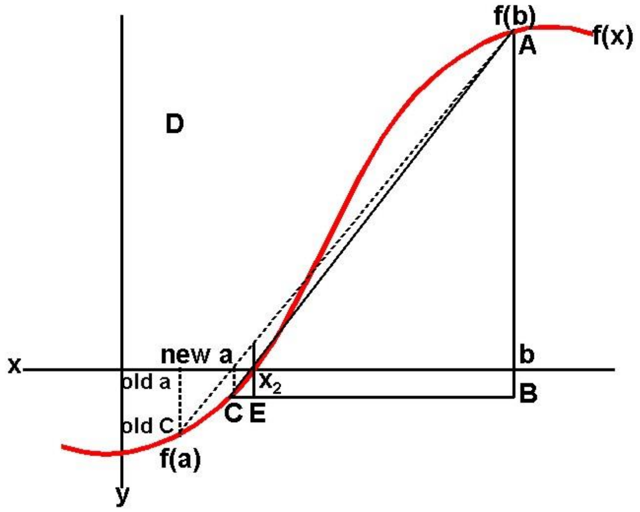
$$(x_1 - a) / (b - a) = [f(x_1) - f(a)] / [f(b) - f(a)]$$

$$x_1 - a = (b - a)[0 - f(a)] / [f(b) - f(a)]$$

$$x_1 = a + [b - a][-f(a)] / [f(b) - f(a)]$$

$$x_1 = a - [\,b - a\,]\, f(a) / [\, f(b) - f(a)\,]$$

Graphically, if the root is in **[ a, $x_i$ ]**, then the next interpolation line is drawn between **(a, f(a) )** and **( $x_i$, f($x_i$) )**; otherwise, if the root is in **[ $x_i$, b ]**, then the next interpolation line is drawn between **( $x_i$, f($x_i$) )** and **(b, f(b))**.




To find the value of root, the value of $x_1$ is taken which is slightly greater than a but it is less than b. If f($x_1$) = 0, $x_1$ is the root. If $x_1$ is not the root, then we check whether, root will lie between a and $x_1$ or between $x_1$ and b. If, f(a)×f($x_1$) < 0 , then root lies between a and $x_1$, else root lies between $x_1$ and b.

**Algorithm for the Regula–Falsi Method**: Given a continuous function **f(x)**

1. Find points **a** and **b** such that **a** < **b** and **f(a) * f(b)** < **0**.
2. Take the interval **[a, b]** and determine the next value of $x_1$.
3. If **f($x_1$) = 0** then $x_1$ is an exact root, else if **f($x_1$) * f(b) < 0** then let **a** = $x_1$, else if **f(a) * f($x_1$) < 0** then let **b** = $x_1$.
4. Repeat steps 2 & 3 until **f($x_i$) = 0** or **|f($x_i$)| ≤ degree of accuracy** (DOA).

EXAMPLE: Consider **f(x) = $x^3$ + 3x – 5**, where **[ a = 1, b = 2 ]** and **DOA = 0.001**.

| i | A | $x_1$ | b | f(a) | f($x_1$) | f(b) |
|---|---|---|---|---|---|---|
| 1 | 1 | 1.1 | 2 | – 1 | – 0.369 | 9 |
| 2 | 1.1 | 1.1354466858 | 2 | – 0.369 | – 0.1297975921309 | 9 |
| 3 | 1.1354466858 | 1.1477379702 | 2 | – 0.1297975921309 | – 0.04486805098 | 9 |
| 4 | 1.1477379702 | 1.1519657086 | 2 | – 0.044868050981328 | – 0.015415586390 | 9 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **5** | **1.1519657086** | **1.1534157744** | **2** | **– 0.0154155863909** | **– 0.005285298529** | **9** |
| **6** | **1.1534157744** | **1.1539126438** | **2** | **– 0.0052852985292** | **– 0.001810778834** | **9** |
| **7** | **1.1539126438** | **1.1540828403** | **2** | **– 0.0018107788348** | **– 0.000620231485** | **9** |

In $7^{th}$ step the value of $|f(x_1) \leq$ **.001** so 1.15408 is the root.

**Problem:** Write a program in BASIC to find the root of the given polynomial in the range[1,2] using Regula Falsi method. $f(x) = x^3 + 3x - 5$. Take the degree of accuracy (DOA) as 0.001

## Solve the same equation using binary bisection method.

We are going to use the logic explained above to make the program using Regula Falsi method.

$$x_1 = a - ( b - a ) f(a) / [ f(b) - f(a) ]$$

Program:

```
10 CLS
20 REM ***REGULA FALSI METHOD***
30 INPUT "Enter lower value of the range"; A
40 INPUT "Enter Higher value of the range"; B
50 DEF FNA (X) = X ^ 3 + 3 * X – 5          BASIC expression of function on RHS. For solving a
                                            different polynomial equation edit the RHS of this statement
60 FA = FNA(A): FB = FNA(B)                 Calculation of f(a) and f(b)
70 IF FA = 0 THEN PRINT "ROOT="; A
80 IF FB = 0 THEN PRINT "ROOT="; B
90 IF FA * FB > 0 THEN 100 ELSE 130         This means that the signs of f(a) and f(b) are same
100 PRINT "NO ROOT IN THE RANGE."
110 PRINT "CHOOSE DIFFERENT RANGE"
120 GOTO 30
130 X1 = A - (B - A) * FA / (FB - FA)       Calculating x1 using the formula
140 Y1 = FNA(X1)                            Calculation of f(x1)
150 IF Y1 = 0 THEN 160
155 IF ABS(Y1) <= .001 THEN 160 ELSE 170    Checking whether |f(xi)| ≤ DOA
160 PRINT "ROOT ="; X1: END
170 IF Y1 * FA > 0 THEN 180 ELSE 200
180 A = X1: FA = Y1
190 GOTO 130
200 B = X1: FB = Y1
220 GOTO 130
230 PRINT "ROOT="; X3
240 END
```

Program for solving the equation using Binary bisection method. In this program I have used a simpler algorithm. This method is used when the range [a, b] is small. You can solve the equation given in previous part of polynomial solving using this method. Just replace the equation given in this program with the equation you want to solve.

```
10 CLS
20 REM ***Binary bisection METHOD***
30 INPUT "lower value of the range"; X1
40 INPUT "Higher value of the range"; X2
50 DEF FNA (X) = X ^ 3 + 3 * X - 5
60 Y1 = FNA(X1): Y2 = FNA(X2)
70 IF Y1 = 0 THEN PRINT "ROOT="; X1
80 IF Y2 = 0 THEN PRINT "ROOT="; X2
90 IF SGN(Y1) = SGN(Y2) THEN 100 ELSE 130
100 PRINT "NO ROOT IN THE RANGE."
110 PRINT "CHOOSE DIFFERENT RANGE"
120 GOTO 30
130 X3 = (X1 + X2) / 2
140 Y3 = FNA(X3)
150 IF Y3 = 0 THEN PRINT "ROOT="; X3
160 IF ABS((X1 - X2) / X3) < .001 THEN 230
170 IF SGN(Y3) = SGN(Y1) THEN 180 ELSE 200
180 X1 = X3: Y1 = Y3
190 GOTO 130
200 X2 = X3: Y2 = Y3
220 GOTO 130
230 PRINT "ROOT="; X3
240 END
```

**SECANT method**

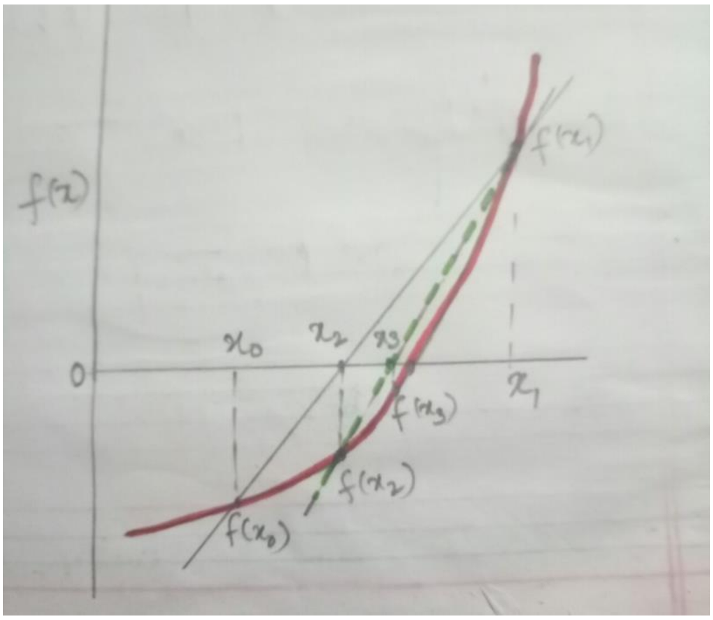
It is another method of solving the polynomial f(x). It also uses a recursive formula like Newton-Raphson's method. Newton Raphson's method is based on the fact that the function approximates as a tangent drawn on the curve near the root. In secant method instead of tangent, a secant line is drawn. Let the range in which the root lies is $x_0$, $x_1$. These must lie near to the root. Plot the graph of the function f(x) vs x. Draw a line between the points ($x_0$, $f(x_0)$) and ($x_1$, $f(x_1)$). The line crosses the curve at two points, therefore, it is known as secant. Since, it is a straight line, it can be written in the form of slope and the intercept as

$$y = \frac{\left(f(x_1) - f(x_0)\right)}{(x_1 - x_0)}(x - x_1) + f(x_1)$$

It intercepts x-axis at $x_2$ and therefore, the value of y at $x_2$ will be zero and can be calculated by the formula

$$\boldsymbol{x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1 - f(x_0)}}$$

This is the recursive formula that is used to calculate the root. In the second step $x_0$ is replaced by $x_1$ and $x_1$ by $x_2$ to calculate $x_3$. This process is repeated until $x_n$ is approximately equal to $x_{n-1}$. You can see from the figure given below. Red curve is the function plotted. The line drawn in the first step crosses x-axis at $x_2$. The value of function at $x_2$ is $f(x_2)$. Then for the second step, the line is drawn in green. It is crossing x-axis at $x_3$ and the value of the function is $f(x_3)$. As you can see, $x_3$ is closer to the root. So, the line drawn in the third step will almost cross the x-axis at the point where the curve crosses the x-axis. So it will be the root of the polynomial.



**Problem:** Write a program in BASIC to find the root of the given polynomial in the range[1,2] using Secant method. $f(x) = x^4 - x - 10.$

Program:

```
10 CLS
20 REM secant method
30 INPUT "X0="; X0
40 INPUT "X1="; X1
50 DEF FNA (X) = X ^ 4 - X - 10
60 Y0 = FNA(X0): Y1 = FNA(X1)
70 FOR I = 1 TO 50
80 X2 = X1 - (Y1 * (X1 - X0)) / (Y1 - Y0)
90 IF ABS((X2 - X1) / X1) < .00001 THEN 140
100 Y2 = FNA(X2)
110 X0 = X1: X1 = X2: Y0 = Y1: Y1 = Y2
120 NEXT I
130 PRINT "VALUES DO NOT CONVERGE. TAKE ANOTHER INITIAL VALUES": END
140 PRINT X2
150 END
```