Experiment No.: 02

Name of the Experiment:

Determining the root of a non-linear equation using False Position Method.

Objectives:

- Getting introduced with False Position Method.
- Determining the roots of non-linear equations in C.
- Determining the roots of non-linear equations in Microsoft Excel.
- Making comparison of experimental results in C and in Microsoft Excel.

Theory:

The False Position method is one of the simplest and most reliable of iterative methods for the solution of nonlinear equations. This method is also known as binary chopping or half interval method. It relies on the fact that if $f(x)$ is real and continuous in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs, that is,

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

Then there is at least one real root in the interval between a and b. That is,

$$x_0=(x_1+x_2)/2$$

Now there exist following three conditions:

1. If $f(x_0) = 0$, we have a root at x_0.
2. If $f(x_0)\, f(x_1) < 0$, there is a root between x_0 and x_1
3. If $f(x_0)\, f(x_2) < 0$, there is a root between x_0 and x_2

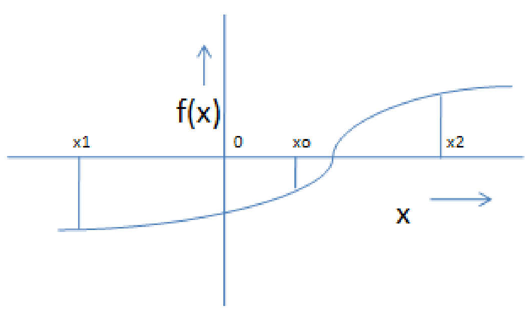


Figure: Illustration of False Position Method

Algorithm for False Position Method:

1. Decide initial values for x_1 and x_2 and stopping criterion, E.
2. Computing f1=f(x1) and f2=f(x2)
3. If $f_1 * f_2 > 0$, x_1 and x_2 do not bracket any root and go to step 7.

4. Compute $x_0=(x_1+x_2)/2$ and compute $f_0=f(x_0)$
5. If $f_1*f_0 <0$ then

   set $x_2=x_0$

   else

   set $x_1=x_0$

   set $f_1=f_0$
6. If absolute value of $(x_2 – x_1)/x_2$ is less than error E, then

   root= $(x_1 +x_2)/2$

   write the value of root,

   go to step 7

   else

   go to step 4
7. Stop.

C code of False Position Method:

```
/* Write a C program to find out a real root of the following non-linear equation using False Position method:
        x2 - 4x - 10 = 0
    Done by: Meraj al Maksud, Class Roll: 320
    Date:
*/
/*

#include<stdio.h>
#include<math.h>
#define ERROR 0.000001
double F(double x)
{
        double y;
        y=(x)*(x)-4*(x)-10;
        return(y);
}

main()
{
        int s, count;
        double a, b, root;
        printf("\n");
```

```
	printf("SOLUTION BY FALSE POSITION METHOD \n");
	printf("\n");
	printf("Input starting values: ");
	scanf("%lf%lf",&a,&b);

	/*calling the subroutine bim() */
	bim(&a, &b, &root, &s, &count);

	if(s == 0)
	{
		printf("\n");
		printf("Starting points do not bracket any root \n");
		printf("Check whether they bracket EVEN roots");
		printf("\n");
	}
	else
	{
		printf("\nRoot = %lf \n", root);
		printf("F(root)  = %lf\n", F(root));
		printf("\n");
		printf("Iterations = %d\n", count);
		printf("\n");
	}
}

/*End of main program */

/* ------------------------------------------------------------------------------------------------------- */
/* Defining the subroutine bim() */

bim(double  *a, double *b, double * root, int *s, int *count)
{
	double x1, x2, x0, f0, f1, f2;
	x1 = *a;
	x2 = *b;
	f1 = F(x1);
	f2 = F(x2);

/*Test if initial values bracket a SINGLE root */
	if(f1 * f2 > 0)
```

```
{
        *s = 0;
        return ;            /*Program terminated*/
}
else
{
        *count = 0;

        begin:
        x0 = (x1 + x2)/2.0;
        f0 = F(x0);
        if(f0 == 0)
        {
                *s = 1;
                *root = x0;
                return ;
        }
        if(f1 * f0 < 0)
        {
                x2 = x0;
                f2 = f0;
        }
        else
        {
                x1 = x0;
                f1 = f0;
        }

/*Testfor accuracy  and repeat the process,if necessary */

        if(fabs(x2 - x1) < ERROR)
        {
                *s = 1;
                *root = (x1 + x2) / 2.0;
                return ;            /*Iteration ends */
        }
        else
        {
                *count = *count + 1;
                goto begin;
```

```
                }
        }
}
```

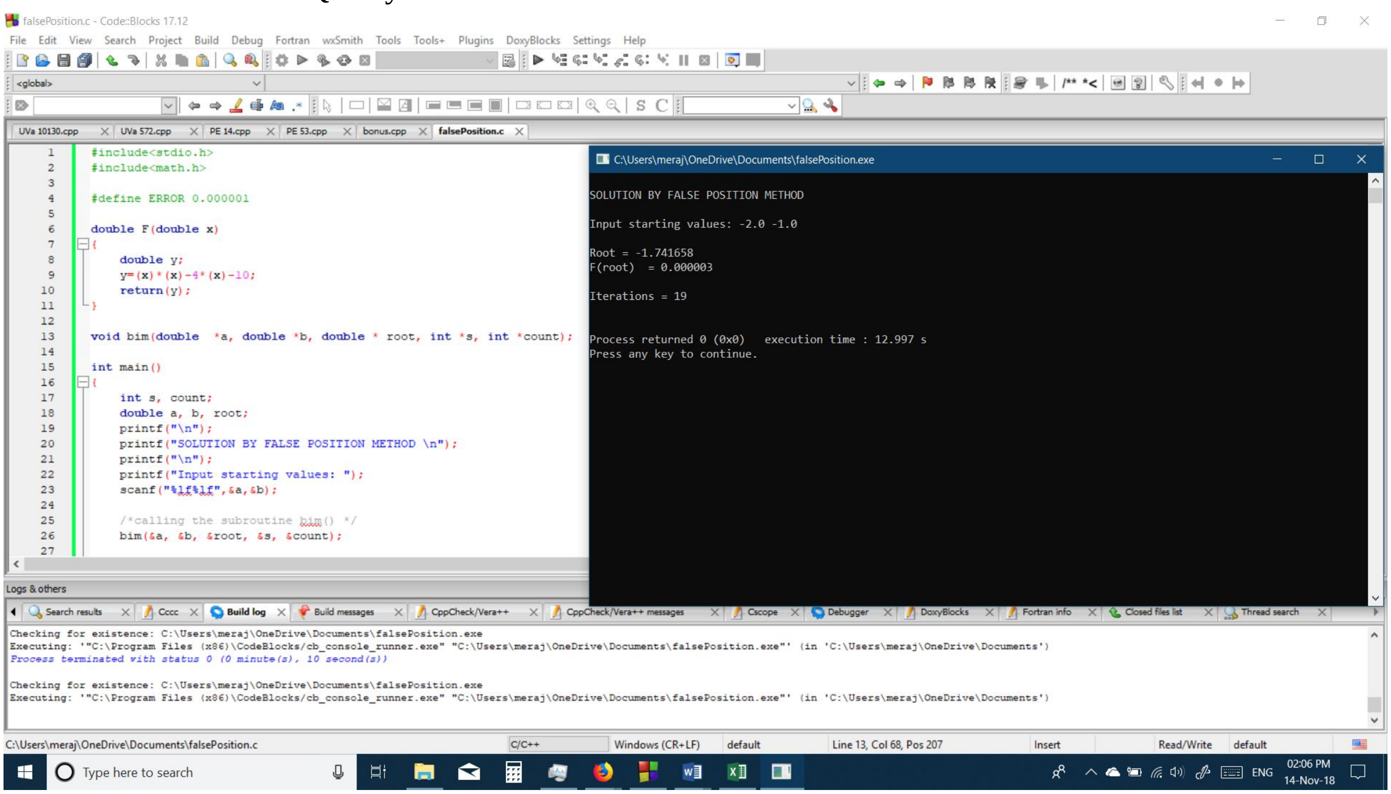
/*End of subroutine bim ()*/

Output:

SOLUTION BY FALSE POSITION METHOD
Input starting values: -2.0 -1.0
Root = -1.741658
F(root) = 0.000003

Iterations = 19

Press Enter to return to Quincy...



False Position Method in Microsoft Excel:

Experiment Name: Find the root of the following equation using False Position Method:

$$f(x) = x^2 - 4x - 10$$

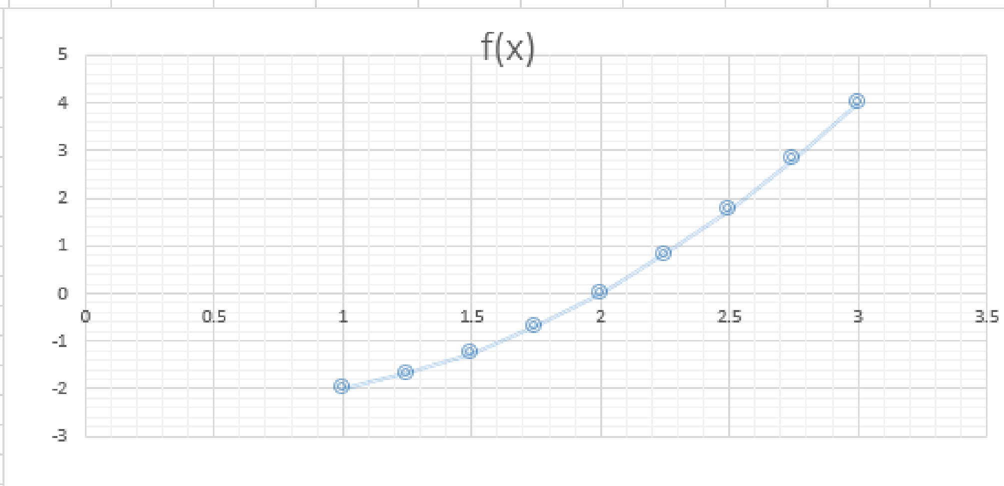
$$Therefore, range\,of\;\; X = \sqrt{\left(\frac{a_{n-1}}{a_n}\right)^2 - 2\left(\frac{a_{n-2}}{a_n}\right)}$$

= 6

$$f(x) = x^2 - x - 2$$

x	f(x)
1	-2
1.25	-1.6875
1.5	-1.25
1.75	-0.6875
2	0
2.25	0.8125
2.5	1.75
2.75	2.8125
3	4

x1	x2	f(x1)	f(x2)	x0	f(x0)	f(x1)f(x0)	f(x2)f(x0)
1	3	-2	4	1.66667	-0.8889	1.77778	-3.5556
1.66667	3	-0.8889	4	1.90909	-0.2645	0.23508	-1.0579
1	3	-2	4	1.66667	-0.8889	1.77778	-3.5556
1.66667	3	-0.8889	4	1.90909	-0.2645	0.23508	-1.0579
1	3	-2	4	1.66667	-0.8889	1.77778	-3.5556
1.66667	3	-0.8889	4	1.90909	-0.2645	0.23508	-1.0579
1	3	-2	4	1.66667	-0.8889	1.77778	-3.5556
1.66667	3	-0.8889	4	1.90909	-0.2645	0.23508	-1.0579
1	3	-2	4	1.66667	-0.8889	1.77778	-3.5556



Result:

After 1st iteration the root is -1.5
After 2nd iteration the root is -1.75
After 3rd iteration the root is -1.625
After 5th iteration the root is -1.71875

After 10^{th} iteration the root is -1.74121
After 15^{th} iteration the root is -1.74167
Approximately the root is -1.74166

Discussion:

The root is not totally accurate. The root has been taken when the interval between x1 and x2 is equal to 1.91×10^{-06}. After 20^{th} iteration the difference is 1.91^{-06}. This is the error of this calculation. The amount of error is too little that it can be avoided. So, -1.74166 can be considered as the root of the equation $x^2 - 4x - 10 = 0$.