

Jahangirnagar University
Department of Computer Science and Engineering

LAB REPORT
ON
CSE-206 (NUMERICAL METHODS LAB)

EXPERIMENT No.: 01

EXPERIMENT NAME: Determining the Root of a Non-Linear Equation Using Bisection Method.



SUBMITTED BY Sumaita Binte Shorif Class Roll: 357 Exam Roll: 191338 Submission Date: Sept 07, 2021			SUBMITTED TO Dr. Md. Golam Moazzam Professor Dept. of Computer Science & Engineering Jahangirnagar University
---	--	--	--

Experiment No: 01

Name of the Experiment: Determining the Root of a Non-Linear Equation Using Bisection Method.

Objectives:

- ✚ Understanding the process of finding the root of non-linear equation using Bisection Method.
- ✚ Executing the implementation of the algorithm of Bisection Method.
- ✚ To achieve the accurate root and expected output of a given nonlinear function.
- ✚ To be able to interpret the advantages and disadvantages of the Bisection method.

The bisection method is used to find the roots of a polynomial equation. It separates the interval and subdivides the interval in which the root of the equation lies.

Theory:

Bisection method being one of the most comprehensible and definitive iterative methods for the solution of nonlinear equations is also known as **binary chopping** or **half-interval method**, 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 .

Let $x_1 = a$ and $x_2 = b$ Let us also define another point x_0 to be the midpoint between a and b . That is, there now exists the following three conditions:

- ✚ If $f(x_0) = 0$, we have a root at x_0 .
- ✚ If $f(x_0).f(x_1) < 0$, there is a root between x_0 and x_1 .
- ✚ If $f(x_0).f(x_2) < 0$, there is a root between x_0 and x_2 .

Testing the sign of the function at midpoint, we can deduce which part of the interval contains the root. This is illustrated in the following figure:

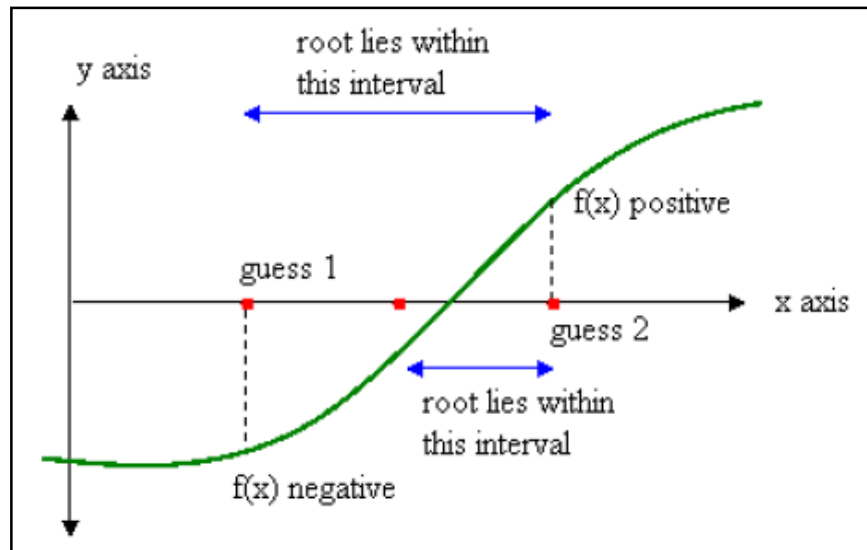


Fig 01: Graphical Depiction of Bisection Method.

Algorithm:

Bisection Method:

1. Deciding initial values for x_1 and x_2 and stopping criterion, E .
2. Computing $f_1 = f(x_1)$ and $f_2 = f(x_2)$.
3. If $f_1 \times f_2 > 0$, then x_1 and x_2 do not bracket any root and go to step 7; otherwise continue.
4. Compute $x_0 = (x_1 + x_2)/2$ and compute $f_0 = f(x_0)$.
5. If $f_1 \times f_0 < 0$ 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 = $(x_2 - x_1)/2$
 Write the value of root
 Go to step 7

else

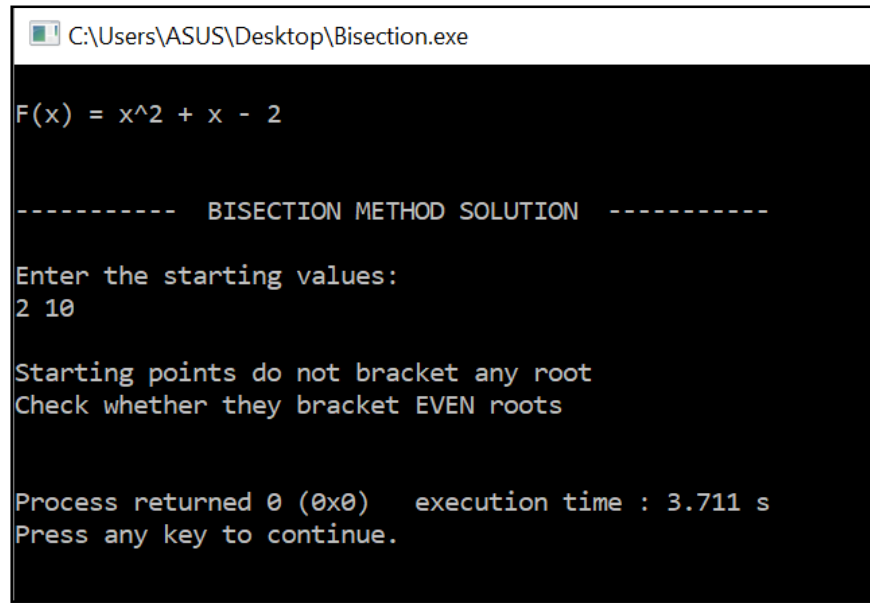
Go to step 4
7. Stop.

Coding in C:

```
#include<stdio.h>
#include<math.h>
#define EPS 0.000001
#define F(x) (x)*(x)+(x)-2
void Bisect(float *a,float *b,float *root,int *s,int *count)
{
    float x1,x2,x0,f0,f1,f2;
    x1=*a;
    x2=*b;
    f1=F(x1);
    f2=F(x2);
    if(f1*f2>0)
    {
        *s=0;
        return ;
    }
    else
    {
        *count = 0;
        while(1)
        {
            *count=*count+1;
            x0=(x1+x2)/2.0;
            f0=F(x0);
            if(f0==0)
            {
                *s=1;
                *root=x0;
                return ;
            }
            if(f1*f0<0)
            {
                x2=x0;
            }
            else
            {
                x1=x0;
                f1=f0;
            }
            if(fabs((x2-x1)/x2)<EPS)
            {
                *s=1;
                *root=(x1+x2)/2.0;
                return;
            }
            else
            {
                continue;
            }
        }
    }
}
```

```
int main()
{
    int s,count;
    float a,b,root;
    printf("\n");
    printf("F(x) = x^2 + x - 2\n\n");
    printf("\n");
    printf("-----  BISECTION METHOD SOLUTION  ----- \n");
    printf("\n");
    printf("Enter the starting values:\n");
    scanf("%f %f",&a,&b);
    Bisect(&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\n");
        printf("\n");
    }
    else
    {
        printf("\nRoot = %f \n",root);
        printf("F(root) = %f\n",F(root));
        printf("\n");
        printf("Iterations = %d\n",count);
        printf("\n");
    }
    return 0;
}
```

Output:



```
C:\Users\ASUS\Desktop\Bisection.exe

F(x) = x^2 + x - 2

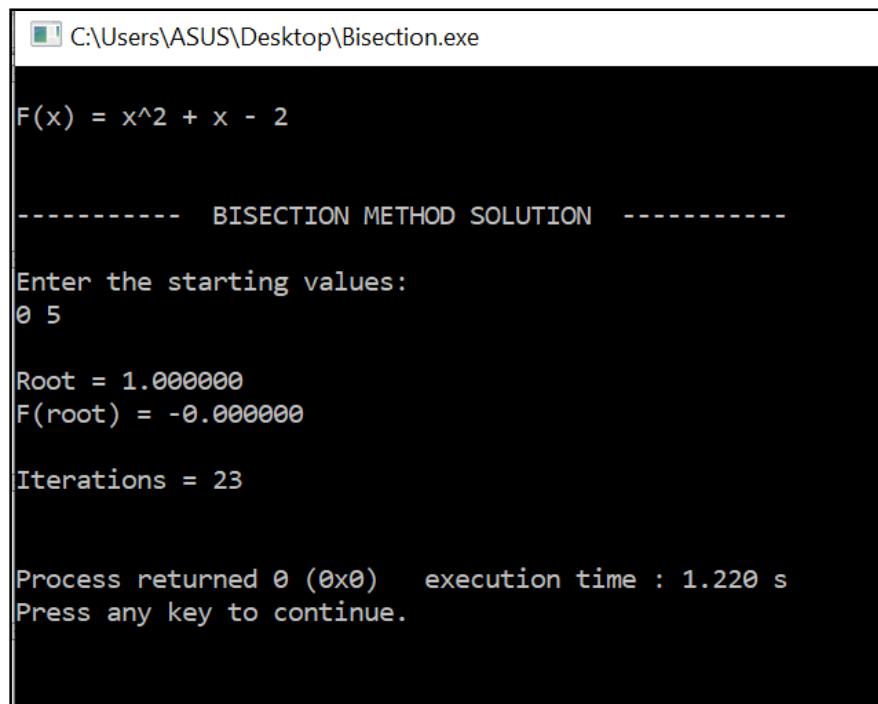
-----  BISECTION METHOD SOLUTION  -----

Enter the starting values:
2 10

Starting points do not bracket any root
Check whether they bracket EVEN roots

Process returned 0 (0x0)   execution time : 3.711 s
Press any key to continue.
```

Fig 02: Output obtained when the starting points do not bracket any root.



```
C:\Users\ASUS\Desktop\Bisection.exe

F(x) = x^2 + x - 2

-----  BISECTION METHOD SOLUTION  -----

Enter the starting values:
0 5

Root = 1.000000
F(root) = -0.000000

Iterations = 23

Process returned 0 (0x0)   execution time : 1.220 s
Press any key to continue.
```

Fig 03: Output Obtained.

Discussion:

From the above Bisection method, we can say that bisection method is linearly convergent. Since the convergence is slow to achieve a high degree of accuracy, a large number of iterations may be needed. However, the bisection algorithm is guaranteed to converge. In bisection method, the interval between x_1 and x_2 is divided into two equal halves, irrespective of location of the root. It may be possible that the root is closer to one end than the other.

After applying this bracketing iterative method, we finally get the expected result which is the root of the polynomial equation. Thus using bisection method, we can find roots of nonlinear equation as it is difficult to solve normally we use numerical mathematics.
