Jahangirnagar University
Department of Computer Science and Engineering

LAB REPORT
ON
CSE-206 (NUMERICAL METHODS LAB)

EXPERIMENT No.: 04

EXPERIMENT NAME: Determining The Root of a Non-Linear Equation Using Secant Method.

SUBMITTED BY			SUBMITTED TO
Sumaita Binte Shorif			**Dr. Md. Golam Moazzam**
Class Roll: 357			**Professor**
Exam Roll: 191338			**Dept. of Computer Science & Engineering**
Submission Date:			**Jahangirnagar University**
Sept 04, 2021			

Experiment No: 04

Name of the Experiment: Determining the root of a non-linear equation using secant method.

Objectives:

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

Theory:

Secant method uses two initial estimates but does not require that they must bracket the root. The basic concept is illustrated in the following figure:

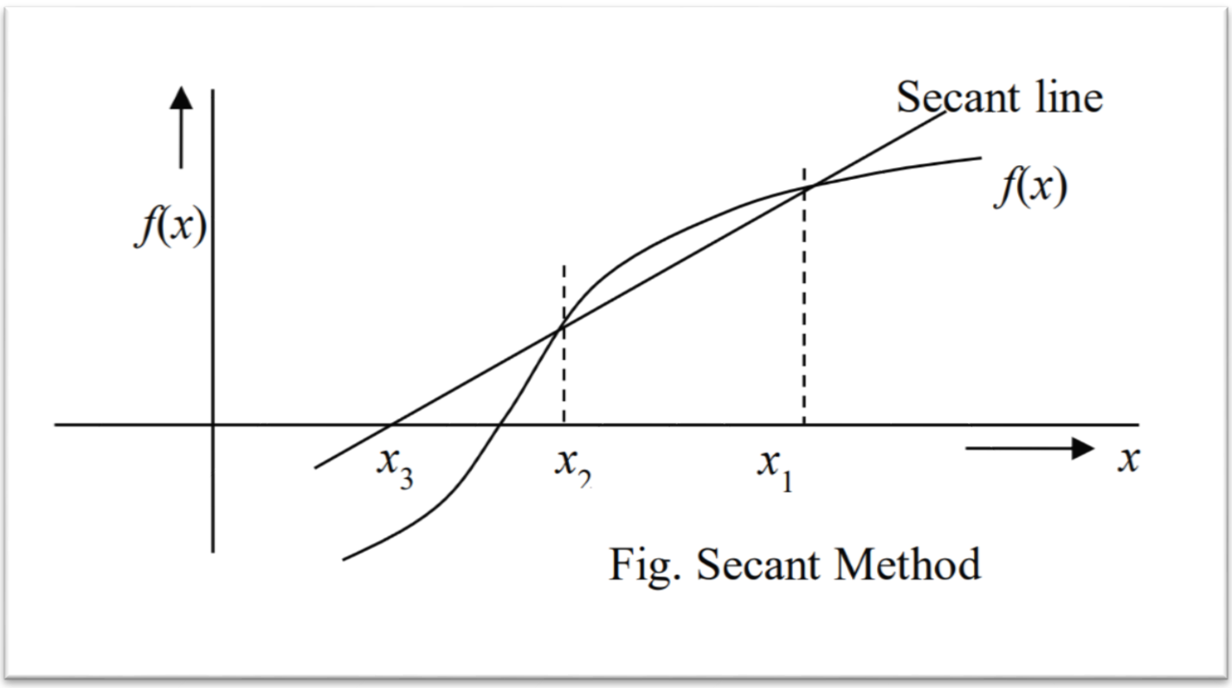


Fig 01: Graphical Depiction of Secant Method.

Consider the points x1 and x2 as starting values. They don't bracket the root. Slope of the secant line passing through x1 and x2 is given by

$$\frac{f(x_1)}{x_1 - x_3} = \frac{f(x_2)}{x_2 - x_3}$$

or,

$$f(x_1)(x_2 - x_3) = f(x_2)(x_1 - x_3)$$

or,

$$x_3[f(x_2) - f(x_1)] = f(x_2)x_1 - f(x_1)x_2$$

Therefore,

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$$

In general form,

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Algorithm:

Secant Method:

1. Choose two initial points x_1 and x_2, accuracy level required, E.
2. Compute f1=f(x_1) and f2=f(x_2).
3. Compute

$$x_3 = x_2 - \frac{f_2 x_1 - f_1 x_2}{f_2 - f_1}$$

4. Test for accuracy of x_3.

   If $| \frac{x_3 - x_2}{x_3} | > E$, then Set x_1=x_2 and f_1=f_2 Set x_2=x_3 and f_2=f_3 Go to step 3

   Otherwise, Set root = x_3 Print results.
5. Stop.

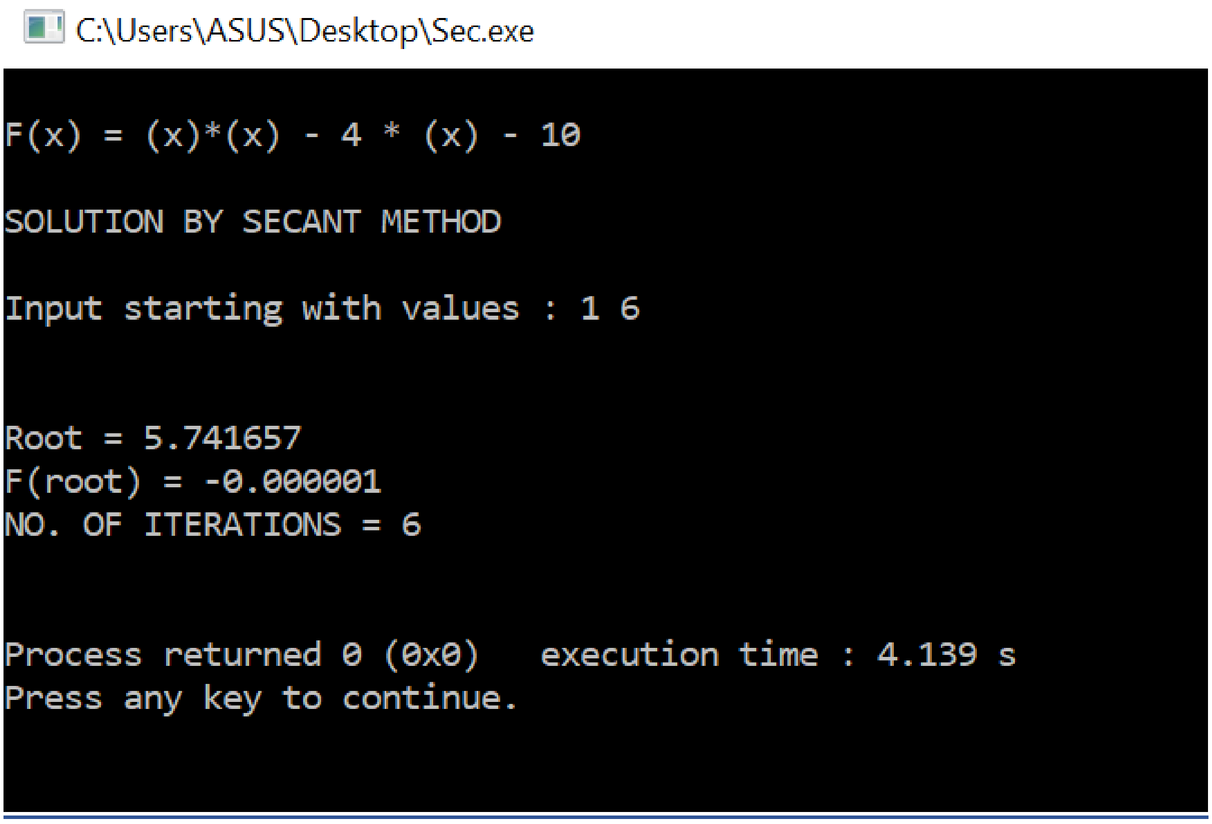
Coding in C:

```
#include<stdio.h>
#include<math.h>
#define EPS 0.000001
#define MAXIT 50
#define F(x)  (x)*(x) - 4 * (x) - 10
int cnt;
float root, x1, x2;
int sec(float a, float b)
{
    float x3, f1, f2, err;
    x1 = a;
    x2 = b;
    cnt = 1;

    int c;
    do
    {
        f1 = F(x1);
        f2 = F(x2);
        if(fabs(f1 - f2) <=EPS) return 1;
        x3 = x2 - (f2 * (x2 - x1) / (f2 - f1));
        x1 = x2;
        f1 = f2;
        x2 = x3;
        f2 = F(x3);
        cnt++;
        if(cnt > MAXIT)
        {
            return 2;
        }
    }
    while(fabs(f1 - f2) >= EPS);
    root = x3;
    return 3;
}
int main()
{
    int s = 0;
    float a, b, x1, x2;
    printf("\n");
    printf("F(x) = (x)*(x) - 4 * (x) - 10\n\n");
    printf("SOLUTION BY SECANT METHOD\n\n");
    printf("Input starting with values : ");
    scanf("%f%f",&a,&b);
    printf("\n");
    s = sec(a, b);
    if(s == 1)
    {
        printf("\nDIVISION BY ZERO\n");
        printf("Last x1 = %.6f\n",x1);
        printf("Last x2 = %.6f\n",x2);
        printf("NO. OF ITERATIONS = %d\n\n",cnt - 1);
    }
    else if(s == 2)
    {

        printf("\nNO COVERGENCE IN %d ITERATIONS\n\n",MAXIT);
    }
    else
    {
        printf("\nRoot = %.6f\n",root);
        printf("F(root) = %.6f\n",F(root));
        printf("NO. OF ITERATIONS = %d\n\n",cnt - 1);
    }
    return 0;
}
```

Output:


C:\Users\ASUS\Desktop\Sec.exe

```
F(x) = (x)*(x) - 4 * (x) - 10

SOLUTION BY SECANT METHOD

Input starting with values : 1 6


Root = 5.741657
F(root) = -0.000001
NO. OF ITERATIONS = 6


Process returned 0 (0x0)   execution time : 4.139 s
Press any key to continue.
```

Fig 02: Output Obtained.

Discussion:

From the above experiment, we can observe that, Secant method doesn't need to bracket the root. Its convergence is slower than Newton-Raphson method. It doesn't need derivative to find the root of a nonlinear equation which makes it more convenient than Newton-Raphson method.

As the initial guesses don't bracket the root, it is uncertain whether the method will converge or not.

This method is easier to implement. If time is not the issue, then we can use this method to easily determine a root of a non-linear equation.
