Problem No : 01

Problem Name : Implementation of Bisection Method for Finding Roots of Non-linear Equations.

Problem Definition :

The purpose of this lab is to implement Bisection method to find the root of a non-linear equation. The method requires initial guesses and a tolerable error. It iteratively narrows down the interval containing the root until the error becomes acceptable.

Algorithm :

1. Start
2. Define function f(x)
3. Choose initial guesses x0 and x1 such that f (x0) f(x1) < 0
4. Choose pre-specified tolerable error e.
5. Calculate new approximated root as x2 = (x0 + x1) / 2
6. Calculate f (x0) f(x2)
   a. If f (x0) f(x2) < 0 than x0 = x0 and x1 = x2
   b. If f (x0) f(x2) > 0 than x0 = x2 and x1 = x1
   c. If f (x0) f(x2) = 0 than goto (8)
   d.
7. If |f(x2)|> e then goto (5) otherwise goto (8)
8. Display x2 as root.
9. Stop.

Code :

Bisection_1028.m * ×

/MATLAB Drive/Bisection_1028.m

```
clc
syms x;
y=input('Enter your non-linear equation:');
a=input('Enter your first guess:');
b=input('Enter your second guess:');
e=input('Tolerable error:');
fa= eval(subs(y,x,a));
fb= eval(subs(y,x,b));
if fa*fb>0
    disp('Given initial values do not bracket the root');
else
    c=(a+b)/2;
    fc= eval(subs(y,x,c));
    fprintf('n\na\t\t\tb\t\t\tc\t\t\tf(c)\n');
    while abs(fc)>e
        fprintf('%f\t%f\t%f\t%f\n',a,b,c,fc);
        if fa*fc<0
            b=c;
        else
            a=c;
        end
        c=(a+b)/2;
        fc= eval(subs(y,x,c));
    end
    fprintf('\nRoot is: %f\n',c);
end
```

Input & Output:

Bisection_1028.m

Command Window

```
Enter your non-linear equation:
x^3+x+1
Enter your first guess:
-1
Enter your second guess:
1
Tolerable error:
0.001
n
a                       b                       c                       f(c)
-1.000000       1.000000        0.000000        1.000000
-1.000000       0.000000        -0.500000       0.375000
-1.000000       -0.500000       -0.750000       -0.171875
-0.750000       -0.500000       -0.625000       0.130859
-0.750000       -0.625000       -0.687500       -0.012451
-0.687500       -0.625000       -0.656250       0.061127
-0.687500       -0.656250       -0.671875       0.024830
-0.687500       -0.671875       -0.679688       0.006314
-0.687500       -0.679688       -0.683594       -0.003037
-0.683594       -0.679688       -0.681641       0.001646

Root is: -0.682617
>>
```

Discussion : The code begins by initializing symbolic variables and obtaining the non-linear equation, initial guesses, and tolerable error. In this lab, we can know the process how to find out root of any non-linear equations and determine the root in a short time by programming. Within the binary search loop, the midpoint of the interval is calculated, and the function value at that point is evaluated. The process continues until the absolute value of the function at the midpoint is less than the tolerable error. Throughout the iterations, the program prints the current interval endpoints, the midpoint, and the function value at the midpoint. Finally, it displays the root found. This implementation demonstrates the efficacy of the bisection method in finding roots of non-linear equations within a specified error margin.