

The False Position Method

Theory

This is the oldest method for finding the real root of a non linear equation $f(x)=0$ and closely resembles the bisection method. In this method, also known as regula falsi or the method of chords. Here two point a and b are chosen in such a way that $f(a)$ and $f(b)$ are of opposite signs. Hence a root must lie in between these points. The equation of the chord joining the two points $[a,f(a)]$ and $[b,f(b)]$ is given by,

$$x = \frac{a*f(b) - b*f(a)}{f(b) - f(a)} \dots\dots\dots (1)$$

This method's procedure is same as the bisection method where the value of a and b changes with respect to the sign of the $f(a)$ and $f(x)$. The only difference between the false position and the bisection method is the equation for determining the root. (Equation 1).

Code

```
#include<bits/stdc++.h>
using namespace std;
double f(double x)
{
    return (x*x*x+x-1);
}

int main()
{
    double a,b,x1,x2;
    cout<<"Enter the value of a: ";
    cin>>a;
    cout<<"Enter the value of b: ";
    cin>>b;
    if(f(a)*f(b)>0)
        cout<<"Wrong Assumption"<<endl;
    else {
        cout<<"-----"<<endl;
        cout<<"  a      b      x1      f(x1) Error Rate \n";
        cout<<"-----"<<endl;
        x1=(a*f(b)-b*f(a))/(f(b)-f(a));
        do
        {
            printf(" %0.4lf    %0.4lf    %0.4lf    %0.4lf %0.6lf \n",a,b,x1,f(x1),fabs(x2-x1));
            x2=x1;
```

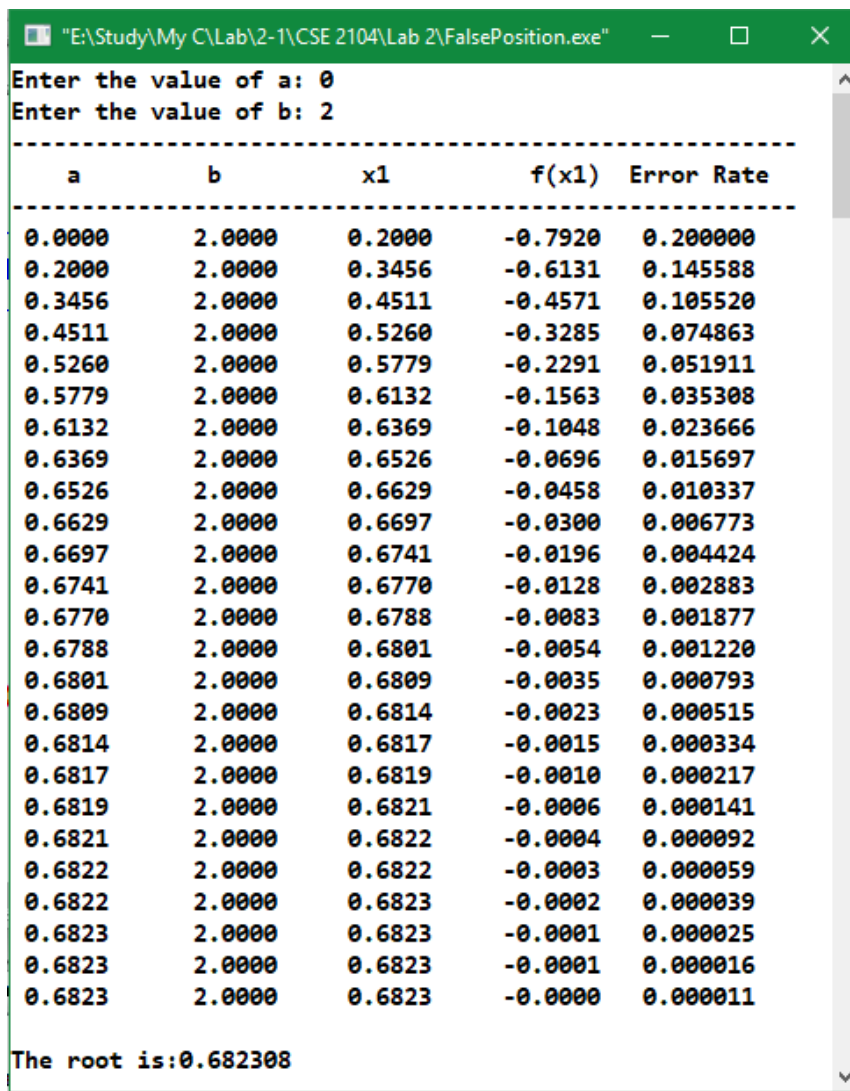
```

        if (f(a)*f(x1)<0)
            b=x1;
        if(f(a)*f(x1)>0)
            a=x1;
        x1=(a*f(b)-b*f(a))/(f(b)-f(a));
    }while(fabs(x2-x1)>=0.00001);
    cout<<"\nThe root is:"<<x2<<endl;
}

return 0;
}

```

Output



Enter the value of a: 0
Enter the value of b: 2

a	b	x1	f(x1)	Error Rate
0.0000	2.0000	0.2000	-0.7920	0.200000
0.2000	2.0000	0.3456	-0.6131	0.145588
0.3456	2.0000	0.4511	-0.4571	0.105520
0.4511	2.0000	0.5260	-0.3285	0.074863
0.5260	2.0000	0.5779	-0.2291	0.051911
0.5779	2.0000	0.6132	-0.1563	0.035308
0.6132	2.0000	0.6369	-0.1048	0.023666
0.6369	2.0000	0.6526	-0.0696	0.015697
0.6526	2.0000	0.6629	-0.0458	0.010337
0.6629	2.0000	0.6697	-0.0300	0.006773
0.6697	2.0000	0.6741	-0.0196	0.004424
0.6741	2.0000	0.6770	-0.0128	0.002883
0.6770	2.0000	0.6788	-0.0083	0.001877
0.6788	2.0000	0.6801	-0.0054	0.001220
0.6801	2.0000	0.6809	-0.0035	0.000793
0.6809	2.0000	0.6814	-0.0023	0.000515
0.6814	2.0000	0.6817	-0.0015	0.000334
0.6817	2.0000	0.6819	-0.0010	0.000217
0.6819	2.0000	0.6821	-0.0006	0.000141
0.6821	2.0000	0.6822	-0.0004	0.000092
0.6822	2.0000	0.6822	-0.0003	0.000059
0.6822	2.0000	0.6823	-0.0002	0.000039
0.6823	2.0000	0.6823	-0.0001	0.000025
0.6823	2.0000	0.6823	-0.0001	0.000016
0.6823	2.0000	0.6823	-0.0000	0.000011

The root is:0.682308

Discusssion

In the above code, firstly a function $f()$ was declared as it hold the main equation $f(x) = x^3 + x - 1 = 0$. Then in the main function a do while loop was executing until the difference of the two consecutive roots of the equation became less than 0.00001. Firstly the value of a and b were taken from the user. In the loop it was checked whether $f(a)$ and $f(x)$ are of opposite signs or not. If so then the value of x was assigned to b, otherwise the value of x is assigned to a. If $f(x)$ becomes 0 then the root $x=0$. And then this equation $x = (a*f(b)-b*f(a))/(f(b)-f(a))$ was used. Thus this is the process of False position method and the result and all the values were shown as a tabular form.