

Application of Bisection Method

Prakhar Gupta, Prof. Vishal V. Mehtre

Department of Electrical Engineering, Bharati Vidyapeeth (Deemed to be University) college of Engineering, Pune

Abstract : The method in which this project is based is called as bisection method, which states “The bisection method is an approximation method to find the roots of the given equation by repeatedly dividing the interval.” This method will divide the interval until the resulting interval is found, which is extremely small.

INTRODUCTION

Bisection method is a popular root finding method of mathematics and numerical methods. This method is applicable to find the root of any polynomial equation $f(x) = 0$, provided that the roots lie within the interval $[a, b]$ and $f(x)$ is continuous in the interval.

This method is closed bracket type, requiring two initial guesses. The convergence is linear and it gives good accuracy overall. Compared to other rooting finding methods, bisection method is considered to be relatively slow because of its slow and steady rate of convergence.

Bisection Method

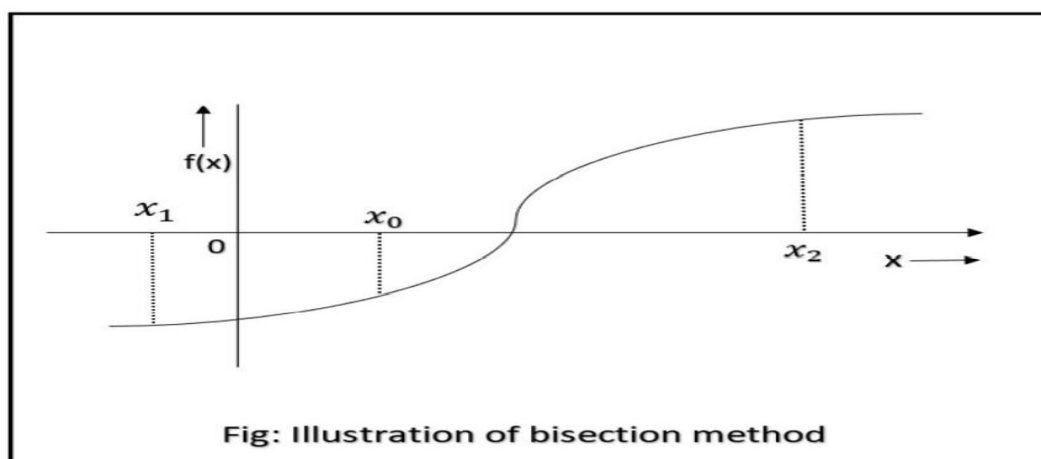
The Bisection method is one of the simplest and most reliable of iterative methods for the solutions of nonlinear equations. This method, also known as binary chopping or half interval method, relies on the fact that if $f(x)$ is real and continuous in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs, that is, $f(a)f(b) < 0$. Then there is at least one real root between a and b . There may be more than one root in the interval. Let, $x_1 = a$ and $x_2 = b$. Let us also define another point x_0 to be the middle point between a and b , that is, $x_0 = (x_1 + x_2) / 2$. Now, there exists the following three conditions:

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

It follows that by testing the sign of the function at midpoint, we can deduce which part of the interval contains the root. This is illustrated in Fig given below. It shows that, since $f(x_0)$ and $f(x_2)$ are of opposite sign, a root lies between x_0 and x_2 . We can further divide this subinterval into two halves to locate a new subinterval containing the root. This process can be repeated until the interval containing the root is as small as we desire.

Bisection method is based on Intermediate Value Theorem. According to the theorem: “If there exists a continuous function $f(x)$ in the interval $[a, b]$ and c is any number between $f(a)$ and $f(b)$, then there exists at least one number x in that interval such that $f(x) = c$.”

The intermediate value theorem can be presented graphically as follows:



Bisection Method Algorithm

Follow the below procedure to get the solution for the continuous function:

For any continuous function $f(x)$,

Find two points, say a and b such that $a < b$ and $f(a) \cdot f(b) < 0$

Find the midpoint of a and b , say “ t ”

t is the root of the given function if $f(t) = 0$; else follow the next step

Divide the interval $[a, b]$ – If $f(t) \cdot f(a) < 0$, there exist a root between t and a

– else if $f(t) \cdot f(b) < 0$, there exist a root between t and b Repeat above three steps until $f(t) = 0$.

The bisection method is an approximation method to find the roots of the given equation by repeatedly dividing the interval. This method will divide the interval until the resulting interval is found, which is extremely small.

Bisection Method Example:

Now, lets analyze the above program of bisection method in Matlab mathematically. We have to find the root of $x^2 - 3 = 0$, starting with the interval $[1, 2]$ and tolerable error 0.01.

Given that,

$$f(x) = x^2 - 3 \text{ and } a = 1 \text{ \& } b = 2$$

$$\text{Mid-value of the interval, } c = (a+b)/2 = (1+2)/2 = 1.5$$

$$f(c) = 1.5^2 - 3 = -0.75$$

Check for the following cases:

1. $f(c) \neq 0$: c is not the root of given equation.
2. $f(c) \cdot f(a) = -0.75 \cdot -2 = 1.5 > 0$: root doesn't lie in $[1, 1.5]$
3. $f(c) \cdot f(b) = -0.75 \cdot 1 = -0.75 < 0$: root lies in $[1.5, 2]$

The process is then repeated for the new interval $[1.5, 2]$. The table shows the entire iteration procedure of bisection method and its MATLAB program:

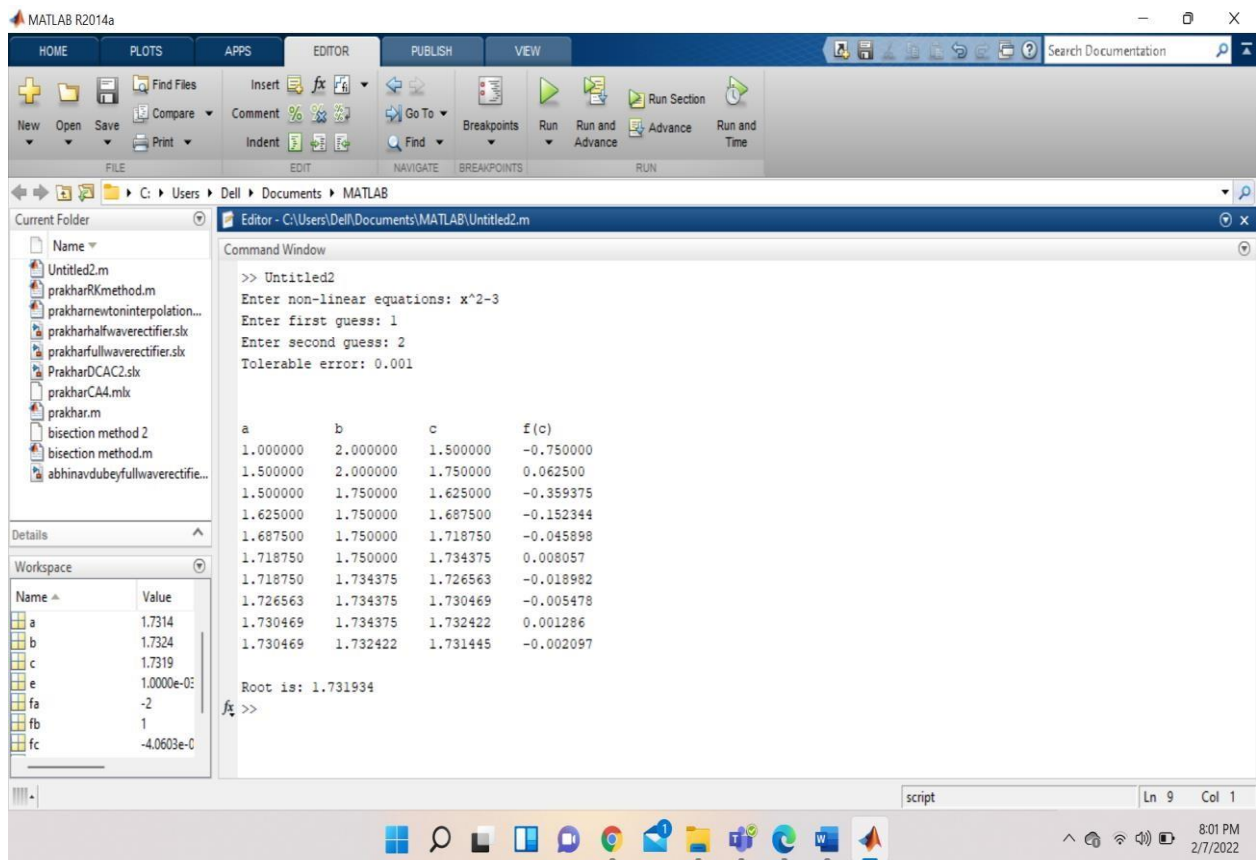
| a | b | $c = (a + b)/2$ | $f(a)$ | $f(b)$ | $f(c)$ | Swap | new $b - a$ |
|---------|--------|-----------------|---------|--------|---------|---------|-------------|
| 1.0 | 2.0 | 1.5 | -2.0 | 1.0 | -0.75 | $a = c$ | 0.5 |
| 1.5 | 2.0 | 1.75 | -0.75 | 1.0 | 0.062 | $b = c$ | 0.25 |
| 1.5 | 1.75 | 1.625 | -0.75 | 0.0625 | -0.359 | $a = c$ | 0.125 |
| 1.625 | 1.75 | 1.6875 | -0.3594 | 0.0625 | -0.1523 | $a = c$ | 0.0625 |
| 1.6875 | 1.75 | 1.7188 | -0.1523 | 0.0625 | -0.0457 | $a = c$ | 0.0313 |
| 1.7188 | 1.75 | 1.7344 | -0.0457 | 0.0625 | 0.0081 | $b = c$ | 0.0156 |
| 1.71988 | 1.7321 | 1.7266 | -0.0457 | 0.0081 | -0.0189 | $a = c$ | 0.0078 |

Thus, the root of $x^2 - 3 = 0$ is 1.7321. It is slightly different from the one obtained using MATLAB program. But, this root can be further refined by changing the tolerable error and hence the number of iteration.

```
1 % Setting x as symbolic variable
2 syms x;
3
4 % Input Section
5 y = input('Enter non-linear equations: ');
6 a = input('Enter first guess: ');
7 b = input('Enter second guess: ');
8 e = input('Tolerable error: ');
9
10 % Finding Functional Value
11 fa = eval(subs(y,x,a));
12 fb = eval(subs(y,x,b));
13
14 % Implementing Bisection Method
15 if fa*fb > 0
16     disp('Given initial values do not bracket the root.');
```

```
17 else
18     c = (a+b)/2;
19     fc = eval(subs(y,x,c));
20     fprintf('\n\na\t\tb\t\tc\t\tf(c)\n');
21     while abs(fc)>e
22         fprintf('%f\t%f\t%f\t%f\n',a,b,c,fc);
23         if fa*fc < 0
24             b = c;
25         else
26             a = c;
27         end
28         c = (a+b)/2;
29         fc = eval(subs(y,x,c));
30     end
31     fprintf('\nRoot is: %f\n', c);
32 end
```

algorithm



```

>> Untitled2
Enter non-linear equations: x^2-3
Enter first guess: 1
Enter second guess: 2
Tolerable error: 0.001

a         b         c         f(c)
1.000000  2.000000  1.500000  -0.750000
1.500000  2.000000  1.750000  0.062500
1.500000  1.750000  1.625000  -0.359375
1.625000  1.750000  1.687500  -0.152344
1.687500  1.750000  1.718750  -0.045898
1.718750  1.750000  1.734375  0.008057
1.718750  1.734375  1.726563  -0.018982
1.726563  1.734375  1.730469  -0.005478
1.730469  1.734375  1.732422  0.001286
1.730469  1.732422  1.731445  -0.002097

Root is: 1.731934
fx >>
    
```

In this code for bisection method in Matlab, first the equation to be solved is defined, and it is then assigned with a variable **f** using **inline()** command. The program then asks for the values of guess intervals and allowable error. The iteration process is similar to that described in the theory above. Or, you can go through this algorithm to see how the iteration is done in bisection method. Iteration continues till the desired root is allocated within the allowable error.

CONCLUSION

Bisection method is the safest and it always converges. The bisection method is the simplest of all other methods and is guaranteed to converge for a continuous function. It is always possible to find the number of steps required for a given accuracy. and the new methods can also be developed from bisection method and bisection method plays a very crucial role in computer science research.

REFERENCES

- [1] Chapter 03.03 Bisection Method of Solving a Nonlinear Equation
- [2] <http://www.cadence.com>
- [3] <http://www.mathcad.com>
- [4] <https://in.mathworks.com/>