**Experiment No: 02**

**Name of the Experiment:** Study of Bisection Method to Obtain the Roots of a Nonlinear Equation.

**Objectives:**

The objective of this experiment is to apply bisection method to find out the very precise value of the root of an equation, using MATLAB.

**Theory:**

The bisection method, which is alternatively called binary chopping, interval halving, or Bolzano’s method, is one type of incremental search method in which the interval is always divided in half. If a function changes sign over an interval, the function value at the midpoint is evaluated. The location of the root is then determined as lying at the midpoint of the subinterval

within which the sign change occurs. The process is repeated to obtain refined estimates. [1]

If the function *f*(*x*) is continuous in [*a, b*] and *f*(*a*)*f*(*b*) *<* 0 (i.e. the function *f* has values with different signs at *a* and *b*), then a value *c ∈* (*a, b*) exists such that *f*(*c*) = 0.[2]

The bisection algorithm attempts to locate the value c where the plot of f  
crosses over zero, by checking whether it belongs to either of the two sub-intervals [a,c],[c,b],where c is the midpoint  
c=(a+b)/2 [3]

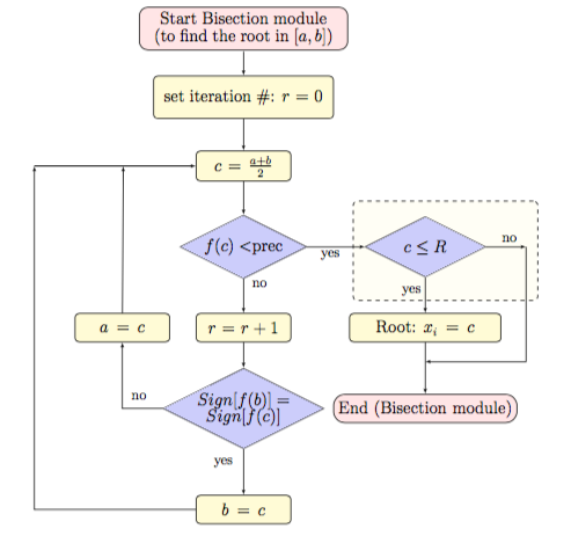
**Tool:** MATLAB

**Methodology:**

**(I) Algorithm:**

Step 1: Choose lower *a* and upper *b* guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that *f*(*a*)*f*(*b*) < 0.  
Step2 : An estimate of the root *c* is determined by *c=* (*a* + *b)/2*  
Step 3: Make the following evaluations to determine in which subinterval the root lies:  
(*a*) If *f*(*a*)*f*(*c*) < 0, the root lies in the lower subinterval. Therefore, set *b* = *c* and return to step 2.  
(*b*) If *f*(*a*)*f*(*c*) > 0, the root lies in the upper subinterval. Therefore, set *a* = *c* and return to step 2.  
(*c*) If *f*(*a*)*f*(*c*) = 0, the root equals *c*; terminate the computation.[4 chap]

**(II) Flowchart:**



**Figure 2.1 Flowchart of bisection method procedure** [4]

**(III) MATLAB Code:** The given function is f(x) =2x^2-15x+3

a=input('Enter the value of 1st assumption:'); //calling value from user

b=input('Enter the value of 2nd assumption:');

y= @(x) 2\*x^2-15\*x+3 ; //declaring function

if y(a)\*y(b)>0 //checking assumption

fprintf('WRONG!!');

return;

end

if y(a)==0

fprintf('Root')

return

elseif y(b)==0;

fprintf('Root')

return

end

display('No. a b c y')

display('------------------------------------------')

for i=1:1:100

c=(a+b)/2;

if y(a)\*y(c)>0 //checking signs

a=c;

else b=c;

end

if abs(y(a))<.001

break;

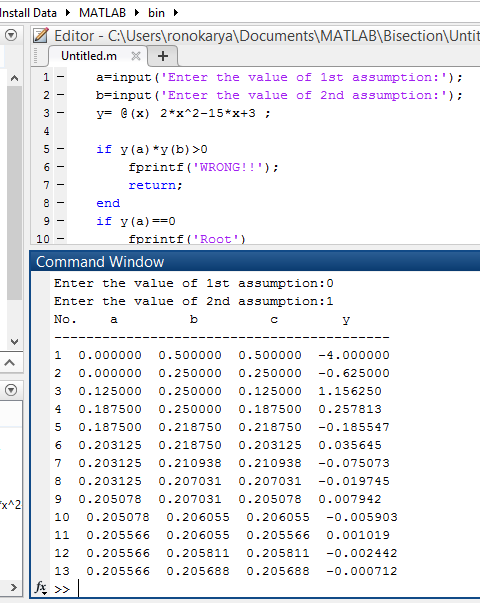
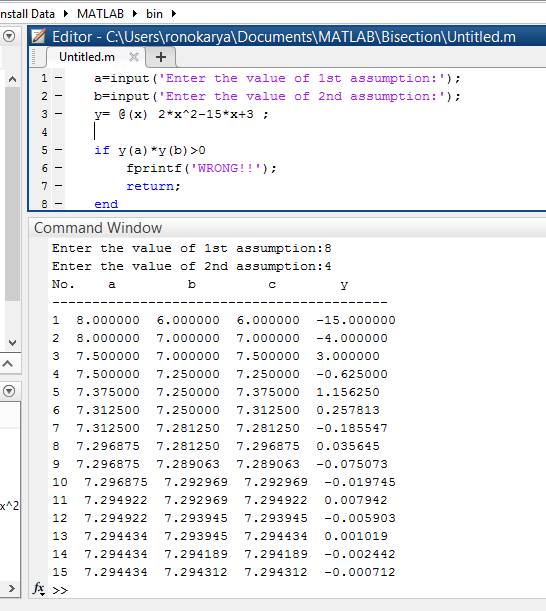
fprintf('%d',i);

end

fprintf('%d %f %f %f %f \n',i,a,b,c,y(c)); //printing iteration values

end

**Output:**

**Results & Discussion:**

The roots of the given function is 7.294312 and 0.205688. Which is nearly close to the original value (7.294361 & 0.205638) direct calculated by calculator.

**Precaution:**

1. Be careful with 2x because in MATLAB the syntax is 2\*x.

2. Variables are declared properly.

3. The accuracy was kept small to reduce iteration.

**Conclusion:**

So from the above test we saw that nearly 15th iteration we get the resultant value of two roots which is very close to the original roots.

**References:**

[1]C. Chapra and P. Canale Raymond , “*Numerical Methods for Engineers”,* 7th ed. McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121, 2015

[2]<http://pages.cs.wisc.edu/~sifakis/courses/cs412s13/lecture_notes/CS412_5_Feb_2013.pdf>

[3]<http://pages.cs.wisc.edu/~sifakis/courses/cs412s13/lecture_notes/CS412_5_Feb_2013.pdf>

[4] C. DENİZ Adnan Menderes University, “*A FAST BISECTION BASED ANALYZER DESIGN FOR THE DETERMINATION OF MODES IN CIRCULAR WAVEGUIDES*” ,page 108, Aytepe-09010, Aydin, TURKEY, Geliş/Received: 06.04.2017; Kabul/Accepted in Revised Form: 21.07.2017