###### University of Dhaka

#### Department of Electrical and Electronic Engineering

EEE-3102: Numerical Technique Laboratory

Experiment-05: Determination of roots of a function by using bisection method.

|  |  |  |
| --- | --- | --- |
| Section | : | Group A |
| Registration No.. | : | Roll: JN-19, SH-11 |
| Name | : | Sriman Bidhan Baray, Sharar Muhtasim |
| Date | : | February 19,2020 |

**Roots of a function:**

Roots of a function f(x) relating variable x are the values of x for which f(x) =0. For example, the roots of the function is obtained by settingwhich gives

………………. (1)

Solution of Eq. (1) gives



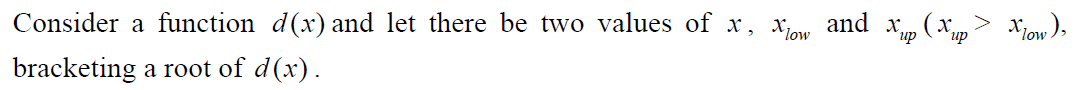
Thus

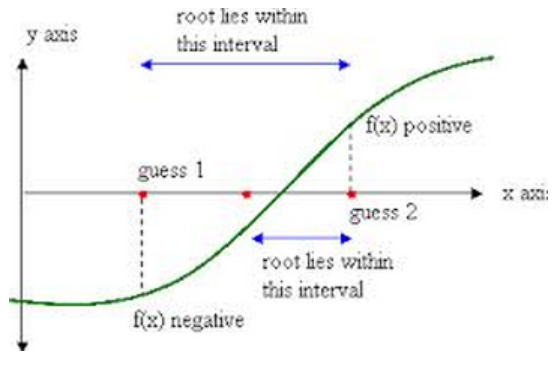


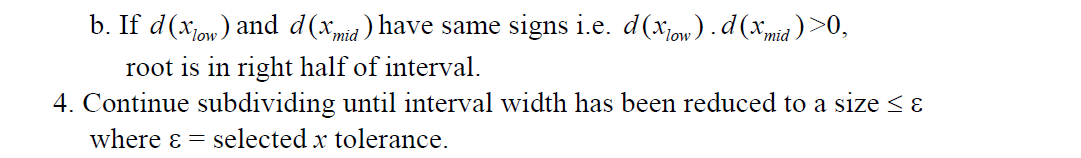
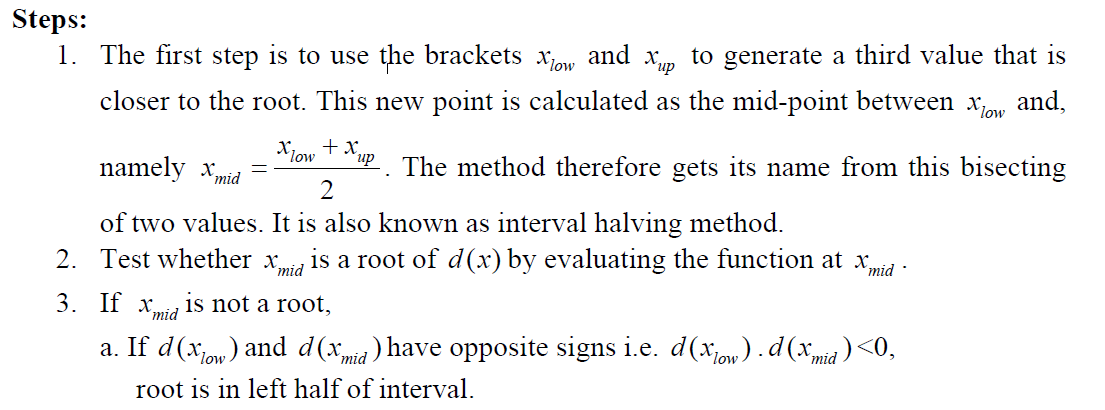
are two roots of f(x). Note that the order of Eq. (1) is two. Thus it has two roots. Similarly, a function of order n has n roots.

**Bisection method**

The Bisection method is one of the simplest procedures for finding root of a function in a given interval. The procedure is straightforward. The approximate location of the root is first determined by finding two values that bracket the root (a root is bracketed or enclosed if the function changes sign at the endpoints). Based on these a third value is calculated which is closer to the root than the original two value. A check is made to see if the new value is a root. Otherwise a new pair of bracket is generated from the three values, and the procedure is repeated.







**Example-1:** Suppose, we want to find the roots of the simple function f(x) = x-2. It is easy to find that f(x) has only one root which is 2. Now, we can write the following matlab code to find this root using bisection method.

clc; close all; clear all;

f=@(x)(x-2); % [0 3]

xl=input ('please input the value of Xl : ');

xu=input ('please input the value of Xu : ');

tol=input('Percentage of relative error tolerance : ');

rel\_error=10; % Error must be greater than tol

%%% To check if there is root in between xl and xu

a=subs(f,xl);

b=subs(f,xu);

if (a\*b>0)

disp ('wrong guess of xl and xu. Try other values')

end

xr=(xl+xu)/2;

if (a\*b<0 & subs(f,xr)==0);

fprintf('Root of given function is %f',xr)

else

i=1;

if (a\*b<0)

xr=(xl+xu)/2;

xri=(xl+xu)/2; %%% Initail value which will be upfated later

while(rel\_error>tol)

xr\_old=xr;

c=subs(f,xr);

d=subs(f,xl);

if(c\*d<0)

xu=xr;

else

xl=xr;

end

xr=(xl+xu)/2;

xrnew(i)=xr; %%%% xrupdatecd after ithiteratiom

rel\_error(i)=abs((xr-xr\_old)/xr\*100);

i=i+1;

end

end

end

fprintf('Root of given function is %f',xr)

n=1:1:i-1;

figure;

plot(n,rel\_error,'-bo')

grid on

xlabel('number of iteration n')

ylabel('percentage of relative error')

m=1:1:i;

xru=[xri xrnew];

f\_xru=subs(f,xru);

figure;

plot(m,xru,'-rs')

xlabel('number of iteration n')

ylabel('updated xr at each iteration')

grid on

figure;

plot(m,f\_xru,'-g\*')

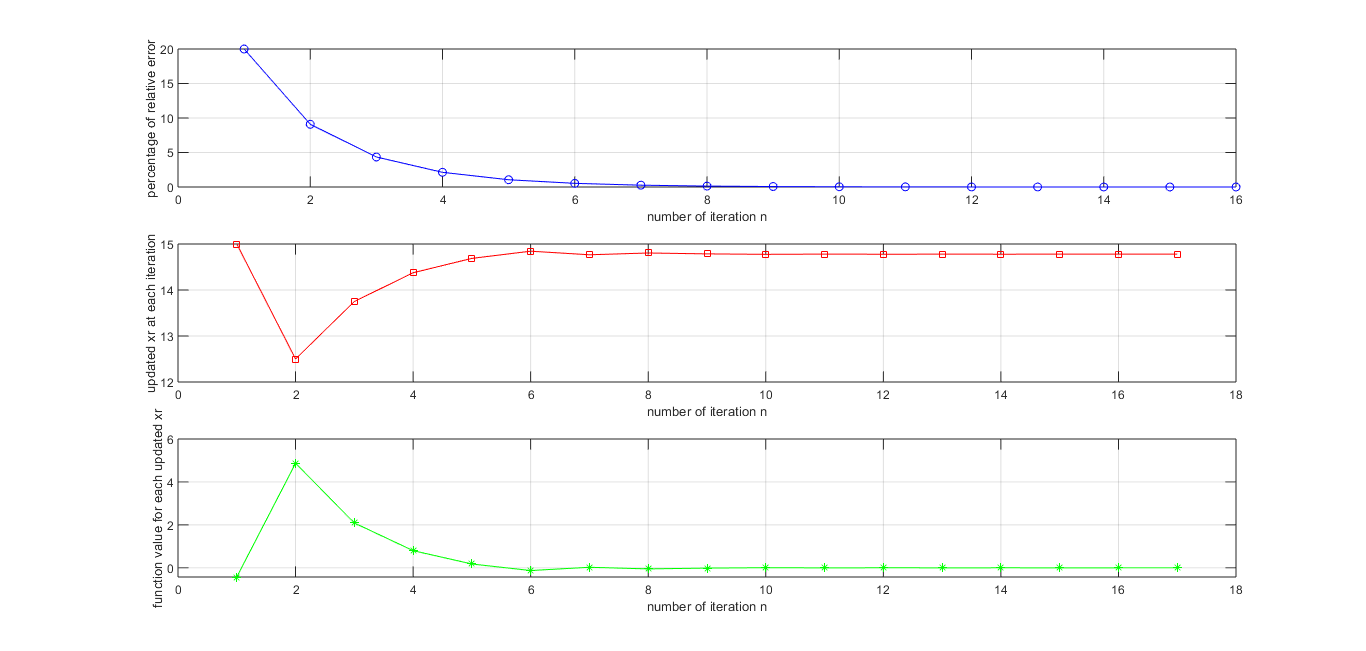
xlabel('number of iteration n')

ylabel('function value for each updated xr')

grid on

Exercise-1:Find the root of the function,. Guess [xl xu] = [10 20]. Also draw the plots to validate your results. Use tolerable relative percentage error of 0.001.

Ans: Root of given function is 14.778519



Exercise-2: Check if there is any root of the functionwithin the x interval of [-2 -1] ,

[-0.5 0.5] and [0 2]. If exists, find the roots. Use tolerable relative percentage error of 0.0005.

Ans: For interval [-2 -1] , root of given function is -1.447727 .

For interval [-0.5 0.5], there is no root in this interval for the given function.

For interval [0 2], , root of given function is 1.447727.

Exercise-3: Check if there is any root of the function*f*(*x*)=(*x*+5.2)(*x*+2.9)(x+1.1)(x-3.6)(x-4.9) within the x interval given in the following table. If exists, find the root.Use tolerable relative percentage error of 0.00002.

|  |  |  |
| --- | --- | --- |
| Interval of x | Root exists (Yes/Not) | Value of the root |
| [-7 -6] | No | - |
| [-6 -4] | Yes | -5.200000 |
| [-4 -2] | Yes | -2.900000 |
| [-2 -0] | Yes | -1.100000 |
| [0 1] | No |  |
| [1.5 4.5] | Yes | 3.600000 |
| [3 5] | No |  |
| [5 10] | No |  |

**The End**

**References**

[1] S.K. Mitra, Digital Signal Processing, 3rd Edition, McGraw-Hill Education (Asia), 2009.

[2] J.G. Proakis and D.G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th Edition, Pearson International Edition, 2007.

[3] McClellan, Schafer and Yoder, *DSP FIRST: A Multimedia Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1998 Prentice Hall.

[4] *Using Matlab*, The Math Works Inc.