Islamic University of Technology (IUT)

Organization of Islamic Cooperation (OIC)
Department of Electrical and Electronic Engineering (EEE)

Course No.	Math 4522
Course Name	Numerical Methods Lab.
Experiment No.	03
Experiment Name	Introduction to Graphical method, Bisection method and False-Position method to find the roots of non-linear equation

Objectives

- To get familiarized with the bisection method and the false-position method to find out the root of the nonlinear equation.

Theory

A linear equation can be defined as an equation with maximum one degree which forms a straight line. A non-linear equation has the degree of two or more and it forms a curve.

The general representation of a linear equation is –

$$y = mx + c$$

The general representation of a non-linear equation is –

$$ax^2 + by^2 = c$$

We can easily find the roots of a linear equation. However, we need to use special techniques to find the roots of a non-linear equation. In this experiment, we will discuss three different methods for finding the roots of a non-linear equation.

Graphical Method

It is a simple method of finding the root of a non-linear equation from the graph. First, we plot the function f(x)=0 and see where it crosses the x-axis. This value of x for which f(x)=0 provides a rough approximation of the root.

Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT

Plot the following function and find its root using graphical approach.

$$f(c) = \frac{gm}{c}\left(1 - e^{-\frac{c}{m}*t}\right) - v$$

- mass, m = 68.1 kg,
- velocity after free falling, v= 40 m/s,
- time, t= 10s
- g=9.81m/s^2
- drag co-efficient, c =?

For a range of values of c, plot the function to observe the curve and find where it crosses the x-axis.

MATLAB Code

Function -

```
function out= drag(c)
x1= exp(-0.1468 .* c);
x2= 1-x1;
x3= 667.38./c;
out= x3.*x2 - 40;
end
```

Test -

```
c= 1:25

out= drag(c)

plot (c,out)

hold on

yline(0, '-g')

hold off
```

If you inspect the curve, it is quite clear that the curve crosses x-axis somewhere between 14 and 16. You can obtain a rough estimate from visual inspection. You can also take MATLAB's help to find the exact intersection point.

Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT

format long g

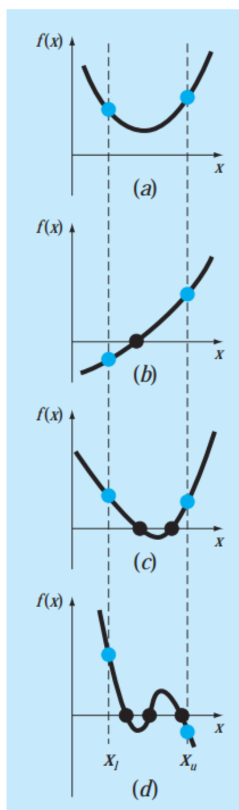
[x, y] = polyxpoly (1:25, drag(1:25), 1:25, zeros(1,25))

x= 14.7852505287737 and y=0.

So, this is your intersection point, and this value of x is root of the equation.

To find the root numerically, bracketing methods can be utilized. Before diving into that, first try to understand the limitations of the approach. Observe the following figure. It illustrates several ways a root may occur or may not occur in an interval $[x_l, x_u]$, the lower bound and the upper bound.

- if x_l and x_u are on the opposite sides of x-axis, odd number of roots exists in the interval.
- if x_l and x_u are on the same side of x-axis, either there is no root, or an even number of roots exist.



Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT

Bisection Method

If f(x) is real and continuous in the interval $[x_l, x_u]$ and f (x_l) and f (x_u) have opposite signs (different sides of x-axis), then there exists at least one root in the interval. The steps to find the root using bisection method are -

I. Choose $x\ell$ and xu as two guesses for the root such that $f(x_l).f(x_u) < 0$, or in other words, $f(x)$ changes sign between $x\ell$ and xu. You can obtain the guesses using graphical method.

II. Estimate the root (x_r) of the equation $f(x) = 0$ as the mid-point between $x\ell$ and xu.

$$x_r = \frac{x_l + x_u}{2}$$

III. Now check the following

a) If $f(x_l).f(x\mathrm{r}) < 0$, then the root lies between x_l and x_r; make - $x_l = x_l,\ x_u = x_r$

b) If $f(x_l).f(x\mathrm{r}) > 0$, then the root lies between x_r and x_u; make - $x_l = r,\ x_u = x_u$

c) If $f(x_l).f(x\mathrm{r}) = 0$; then the root is x_r. We have found the root. So, we can stop the iteration.

IV. Find the new estimate of the root –

$$x_r = \frac{x_l + x_u}{2}$$

V. Find the absolute relative approximate error

$$|\epsilon_a| = \left|\frac{x_r^{new} - x_r^{old}}{x_r^{new}}\right| * 100$$

VI. Compare this error with the error tolerance provided. If the obtained error value is higher than the tolerance, then repeat process from step – III. Otherwise, stop the iteration.

VII. Also check if the maximum number of iterations i_{max} is reached. If the no. of iterations > i_{max}, even when the desired tolerance is not achieved, stop the algorithm.

Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT

Sample MATLAB Program

```
% Bisection method

xl=12
xu=16

xr_prev=0
ea=100

while ea>1
    xr= (xl+xu)/2
    ea= abs((xr- xr_prev)/xr) *100
    if drag(xl)*drag(xr) <0
        xu=xr
    elseif drag(xl)*drag(xr) >0
        xl=xr
    else
        break;
    end
    xr_prev=xr
end
```

False – Position Method

A shortcoming of the bisection method is that, while dividing the interval from *xl* to *xu* into equal halves, no account is taken of the magnitudes of *f*(*xl*) and *f*(*xu*). This method attempts to overcome this limitation. The steps to find the root using the false-position method are -

I. Choose $x\ell$ and *xu* as two guesses for the root such that $f(x_l).f(x_u) < 0$, or in other words, *f* (*x*) changes sign between $x\ell$ and *xu.*

II. Estimate the root (x_r) of the equation *f* (*x*) = 0 as –

$$x_r = \frac{x_u\, f(x_l) - x_l\, f(x_u)}{f(x_l) - f(x_u)}$$

III. Now check the following

a) If $f(x_l).f(x\mathrm{r}) < 0$, then the root lies between x_l and x_r; make - $x_l = x_l,\ x_u = x_r$

b) If $f(x_l).f(x\mathrm{r}) > 0$, then the root lies between x_r and x_u; make - $x_l = r,\ x_u = x_u$

Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT

c) If $f(x_l).f(x\text{r}) = 0$; then the root is x_r. We have found the root. So, we can stop the iteration.

IV. Find the new estimate of the root –

$$x_r = \frac{x_u\, f(x_l) - x_l\, f(x_u)}{f(x_l) - f(x_u)}$$

V. Find the absolute relative approximate error

$$|\epsilon_a| = \left|\frac{x_r^{new} - x_r^{old}}{x_r^{new}}\right| * 100$$

VI. Compare this error with the error tolerance provided. If the obtained error value is higher than the tolerance, then repeat process from step – III. Otherwise, stop the iteration.

VII. Also check if the maximum number of iterations i_{max} is reached. If the no. of iterations > i_{max}, even when the desired tolerance is not achieved, stop the algorithm.

Lab Task

Write a program to find the root of a non-linear equation using False-position method.

Assignment

1. Explain the advantage of false-position method over bisection method.
2. Write the code to locate the root of $f(x) = x^{10} - 1$ using bisection and false position method. Show the approximation error and true error at each iteration using a table created using MATLAB program. Also provide illustration. Comment on which method is more appropriate for this example.

Prepared by

ASIF NEWAZ
Lecturer, Department of EEE, IUT