University of Asia Pacific

Department of Computer Science and Engineering

Course Code : CSE 314

Course Title : Numerical Methods Lab

Final Assignment

Submitted by

Uniza Hossain Arpa

Registration No.: 19201061

Group: A2, Department: CSE

Submitted to

Nadeem Ahmed

Assistant Professor

Department of CSE, UAP

Question 1:

Please take a 4th degree equation by your own. Solve the equation with both Bisection Method and Newton-Raphson Method. Then provide a comparison (analysis) between the two methods.

Solution:

Let's take a 2nd-degree equation for both Bisection and Newton Rapson Method.

Equation = $25x^4 + 5x^3 - 10x^2 - 148x - 150 = 0$

Bisection Method:

For solving with the Bisection Method, I have to find a root. So, that's why for solving that I have taken two-point which are the lower point and upper point.

Here, Lower Mid = 0

Upper Mid = 5.5

Code:

```
clc
clear

i = 10;
bisection = @(x) 25*x.^4 + 5*x.^3 - 10*x.^2 - 148*x - 150
lower = 0;
upper = 5.5;
array = [];

if bisection(lower)*bisection(upper) < 0
    for i = 1:i
    midpoint = (lower+upper)/2;
    array(i) = double(midpoint);
    if bisection(lower)*bisection(midpoint) < 0
        upper = midpoint;
    else
        lower = midpoint;
    endif
    endfor
else
    disp("No Root Found.");
end
disp("Value of Mid in every iteration are : ");
disp(array);
```

Output:

```
bisection =

@(x) 25 * x .^ 4 + 5 * x .^ 3 - 10 * x .^ 2 - 148 * x - 150

Value of Mid in every iteration are :
 Columns 1 through 8:

   2.7500    1.3750    2.0625    2.4062    2.2344    2.1484    2.1055    2.0840

 Columns 9 and 10:

   2.0732    2.0679
```

Newton-Raphson Method:

For Solving with Newton-Raphson Method I need to find an estimated root value which is (0 + 5.4)/2 or 2.7. And find the second derivation of the chosen equation.

Code:

```
clc
clear

i = 10;
func = @(x)x.^4 + 6*x.^3 - 9*x.^2 - 162*x-243
send_d = @(x) 4*x.^3 + 18*x.^2 - 18*x - 162;
firstroot(1) = 2.7

for i=1: i-1
    firstroot(i+1) = firstroot(i) - (func(firstroot(i))/send_d(firstroot(i)));
end
disp("Value of x_m in every iteration are: ");
disp(firstroot);
```

Output:

```
func =

@(x) x .^ 4 + 6 * x .^ 3 - 9 * x .^ 2 - 162 * x - 243

firstroot = 2.7000
Value of x_m in every iteration are:
 Columns 1 through 7:

     2.7000  -884.2875  -663.5938  -498.0745  -373.9365  -280.8348  -211.0109

 Columns 8 through 10:

  -158.6461  -119.3767   -89.9299
```

In the Bisection Method, the rate of convergence is linear thus it is slow. In the Newton Raphson method, the rate of convergence is second-order or quadratic. The Newton and Secant are speedy to converge with very small error part and requiring a few steps of iterations while the bisection method is converged with taking too much computing of iterations.

Question 2:

Please take a Linear equation by your own. Write the code (you may show the plot) and then Find an estimated value of 'y 'for a 'x' for Linear Interpolation.

Solution:

My linear equation: y => f(x) = 3.79x + 26

Analysis: Creating a vector from 1 to 126 with 10 differences between each value, i.e., step size 10.

Generating a second vector from the first one as y = f(x) where x is the first vector.

Then interpolating f(x) = y where x = 26 (x = 26 does not exist and has to be interpolated.)

The interpolated estimated value is f(26) = 124.54

Code:

```
clc
clear
fn_y = @(x) 3.79*x + 26;
x = [1:10:126];
y = fn_y(x);
ans = interp1(x, y, 26);
plot(x, y, ':.o', 26, ans, 'x');
title("Linear interpolation for '3.79*x + 26'");
xlabel("Values of x");
ylabel("Values of y = f(x)");
legend("Values of 'y = f(x)'", "Interpolation for point X = 26");
disp(ans);
```

Output: