MA 322 Lab Assignment

Name: Anubhav Bajaj

Roll no. : 190123007

Lab no. : 01

Q1

Using Bisection Method we get approximate answer in 15 iteration and using Newton method we get using 4 iteration.

Bisection Method

n	x(n)	f(x(n))
1	1.000000	-0.473592
2	0.500000	0.127105
3	0.750000	-0.209272
4	0.625000	-0.049836
5	0.562500	0.036480
6	0.593750	-0.007221
7	0.578125	0.014495
8	0.585938	0.003603
9	0.589844	-0.001817
10	0.587891	0.000891
11	0.588867	-0.000464
12	0.588379	0.000213
13	0.588623	-0.000125
14	0.588501	0.000044
15	0.588562	-0.000041

Newton Method

n	x(n)	f(x(n))
1	0.478528	0.159222
2	0.584157	0.006079
3	0.588525	0.000011
4	0.588533	0.000000

```
ans =

    0.5885
```

Q2

```
           x(n)          f(x(n))           Error
                    -1.414214e+00
    1.414214e+00    -4.335455e-01    2.346331e-01
    1.847759e+00    -1.138115e-01    5.802060e-02
    1.961571e+00    -2.879889e-02    1.446912e-02
    1.990369e+00    -7.221459e-03    3.615084e-03
    1.997591e+00    -1.806725e-03    9.036346e-04

ans =

    1.999397637392408
```

Complete Code

Q1

Code is divided into function each being in separate file named as function name.

```
clear all;
bisection = @biSection;
newtonmethod = @newtonMethod;
a = 0;
b = 2;
x = 1;
epsilon = 1e-5;
max_iter = 1000;

bisection(@f, a, b, epsilon, max_iter)
newtonmethod(@f,@df,x,epsilon,max_iter)

function [y] = f(x)
      y = exp(-x) - sin(x);
end

function [y] = df(x)
      y = -exp(-x) - cos(x);
end

function [root, val, status, data] = biSection(f, a, b, epsilon, max_iter)
      data = [];
      fa = f(a);
      fb = f(b);
    fprintf('%2s %15s %15s \n','n','x(n)','f(x(n))');
      for i = 1:max_iter
            c = (a + b) / 2;
            fc = f(c);
            err = abs(c-a);

            if(abs(fc) < epsilon)
                  break;
            elseif (fa*fc < 0)
                  b = c;
```

```
                fb = fc;
            else
                a = c;
                fa = fc;
        end
        fprintf('%2d %15f %15f \n',i,c,fc);

    end
end

function [fixed, status, iterations, data] = newtonMethod(f, df, x, epsilon,
max_iter)
      data = [];
    fprintf('%2s %15s %15s \n','n','x(n)','f(x(n))');
      for i = 1:max_iter
            y = x - f(x)/df(x);
            err = abs(x-y);
            temp = [i, x, y];
            data = [data; temp];
            if(abs(x - y) < epsilon)
                break;
            end
            x = y;
        fprintf('%2d %15f %15f \n',i,x,f(x));
      end

      fixed = y;
      iterations = i;
    fprintf('%2d %15f %15f \n',iterations,fixed,f(fixed));
end
```

Q2

```
clear all;
format long;
fixed_point = @fixedPoint;

x = 0;
epsilon = 1e-3;
max_iter = 1000;

fixed_point(@g, x, epsilon, max_iter)


function [y] = f(x)
      y = x^2 - x - 2;
end


function [y] = g(x)
      y = (x+2)^(1/2);
end


function [fixed, status, data] = fixedPoint(g, x, epsilon, max_iter)
      data = [];
    fprintf('%15s %15s %15s \n','x(n)','f(x(n))','Error');
      for i = 1:max_iter
```

```
            y = g(x);
          err=abs(x - y)/y;
          fprintf('%15s %15s %15s \n',x,f(y),err);
            if(err < epsilon | abs(f(y)) < epsilon)
                  break;
            end
            x = y;
      end

      fixed = y;

      %%data

      if(abs(x - y) < epsilon)
            status = 0;
      else
            status = 1;
end
```