LAB REPORT 2 OF DSL

Praveen Kumar Yadav

EE22M308

MATLAB R2014a

**Q2:** Consider the function f(x)=x4 -x-10. Find out the roots of the function f(x)=0 using the methods mentioned below:

a) Plot the function and see where the value is becoming zero.

Code:

clear all;

x=[-4 : 4];

y=[x.^4-x-10];

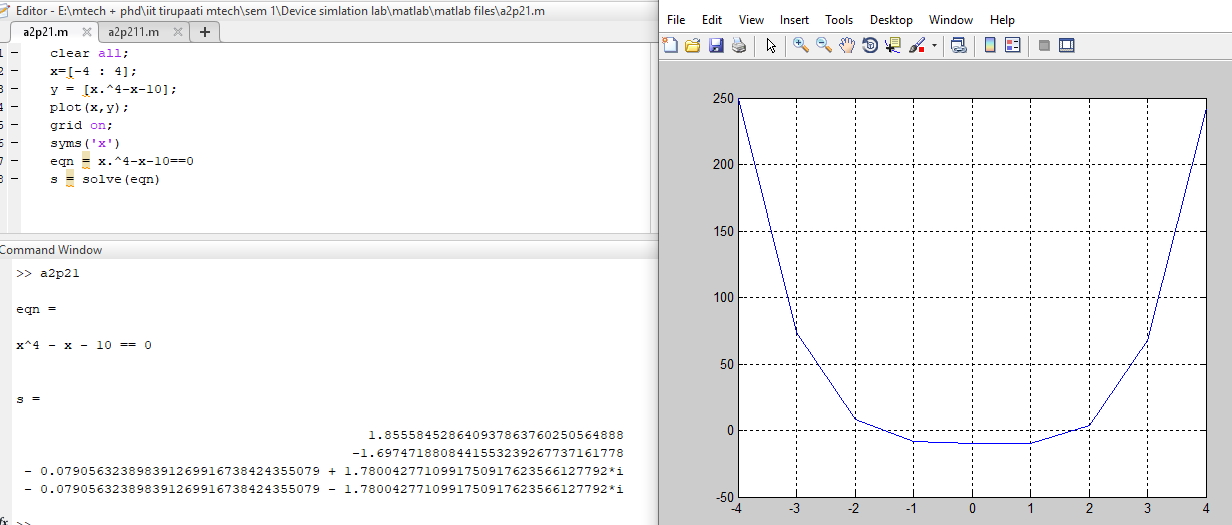
plot(x,y);

grid on;

syms('x')

eqn = x.^4-x-10==0

s=solve(eqn)



In this plot we got 2 real points and 2 complex roots shown above.

b) Bisection method (You can take a=1 and b=3. Can you obtain some other solution by choosing different values of a & b?). Plot the rate of convergence (that is value of c=0.5\*(a+b) vs the iteration no.)

code:

%ingrdients

f = @(x) x^4-x-10;

a=1;b=4;

n=4;

e=0.0001;

itrt=[];value\_c=[];

if f(a)\*f(b)<0

for i=1:n

c=(a+b)/2;

fprintf('p%d=%.4f\n',i,c)

if abs(c-b)<e || abs(c-a)<e

break

end

if f(a)\*f(b)<0

b=c;

elseif f(b)\*f(c)<0

a=c;

end

itrt=[itrt,i];

value\_c=[value\_c,c];

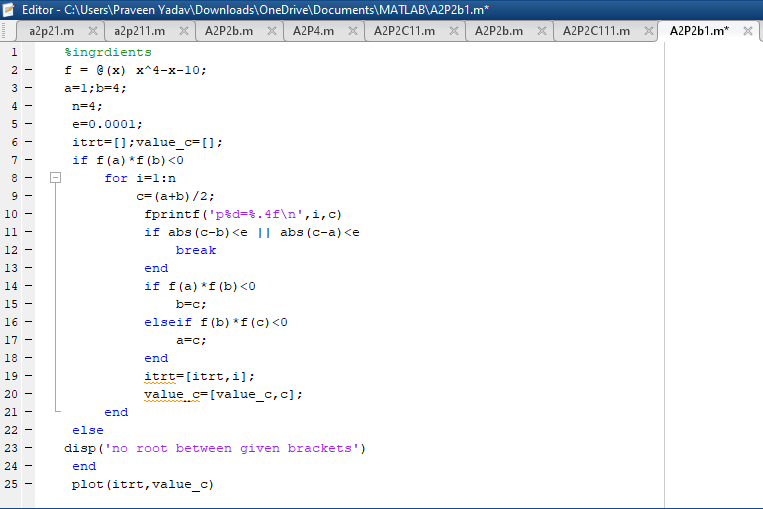
end

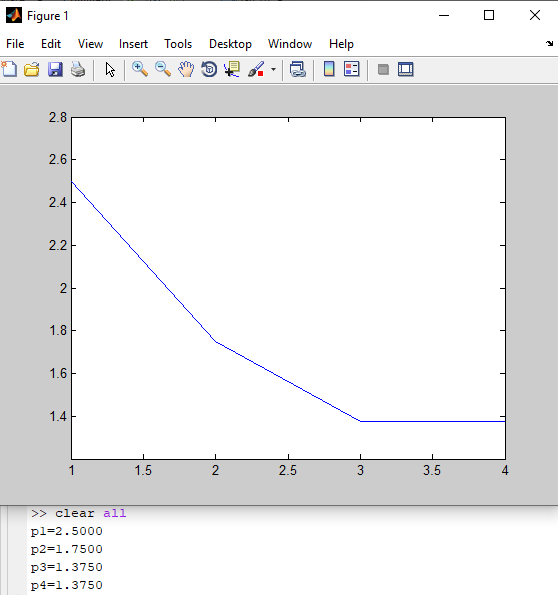
else

disp('no root between given brackets')

end

plot(itrt,value\_c)





c) Fixed point iteration method (Take g(x) as (x+10)0.25 and initial guess of 2. Do you see convergence to one solution? Take g(x) as 10/(x3 -1) and initial guess of 2. Do you see convergence? Take g(x) as (x+10)0.5 /x and initial guess of 2. Do you see convergence? Plot the rate of convergence for the cases where the solution is converging. Comment on the nature of convergence by finding out g’(x) at the initial guess value of x=2)

code: for g(x) as (x+10)0.25

clear all;

%ingredients

f = @(x) (x+10)^(1/4);

x0=2;

n=4;

iteration=[];

output=[];

e=0.0001;

for i=1:1:n

x1 = f(x0);

fprintf('x%d = %.4f\n',i,x1)

if abs(x1-x0)<e

break

end

x0=x1;

iteration=[iteration,i];

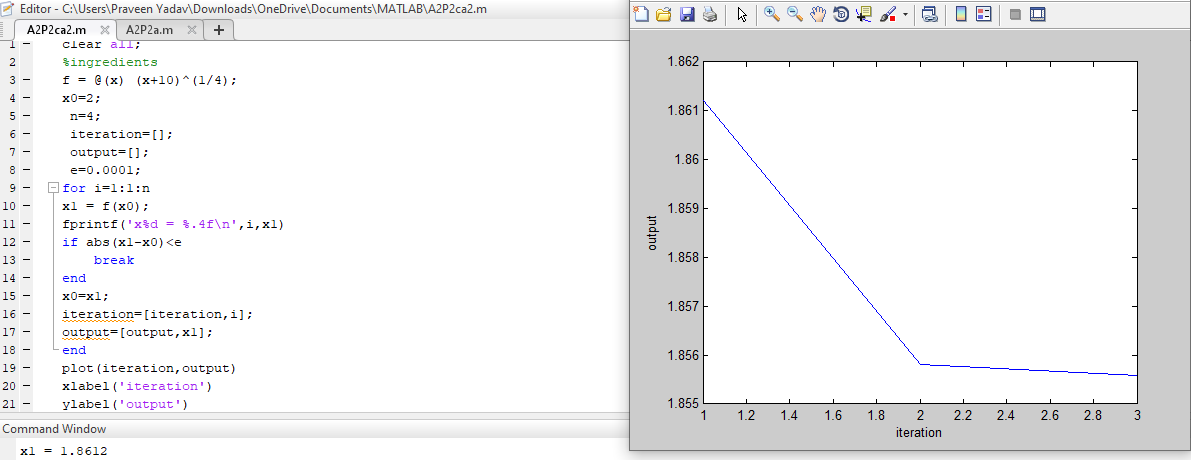
output=[output,x1];

end

plot(iteration,output)

xlabel('iteration')

ylabel('output')



Code for g(x) as 10/(x3 -1) :

clear all;

%ingredients

f = @(x) 10/(x.^3-1);

x0=2;

n=4;

iteration=[];

output=[];

e=0.0001;

for i=1:1:n

x1 = f(x0);

fprintf('x%d = %.4f\n',i,x1)

if abs(x1-x0)<e

break

end

x0=x1;

iteration=[iteration,i];

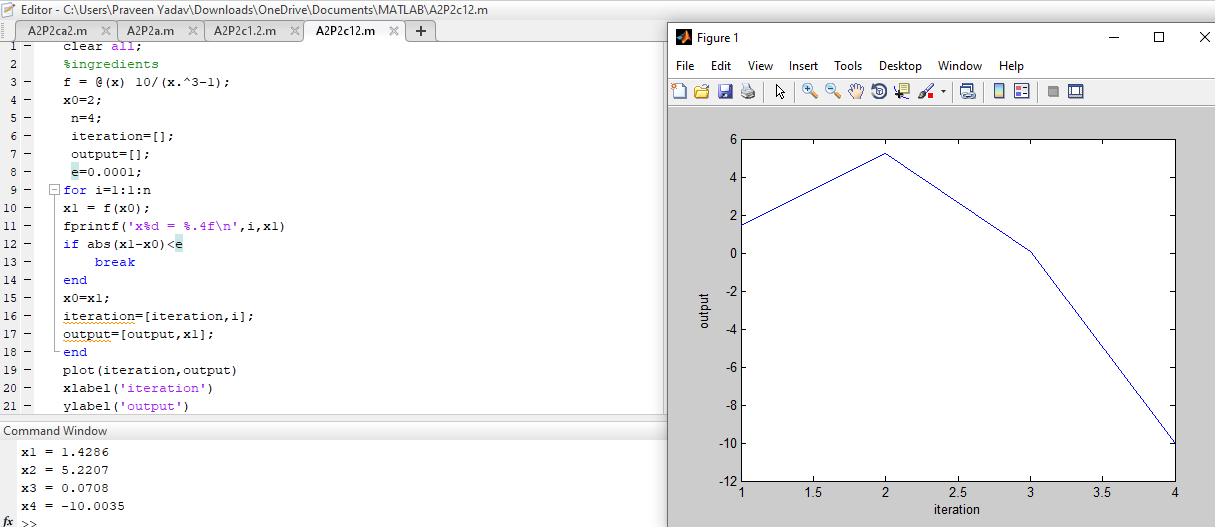
output=[output,x1];

end

plot(iteration,output)

xlabel('iteration')

ylabel('output')



Code for g(x) as (x+10)0.5 /x

clear all;

%ingredients

f = @(x) (x+10).^0.5/x;

x0=2;

n=4;

iteration=[];

output=[];

e=0.0001;

for i=1:1:n

x1 = f(x0);

fprintf('x%d = %.4f\n',i,x1)

if abs(x1-x0)<e

break

end

x0=x1;

iteration=[iteration,i];

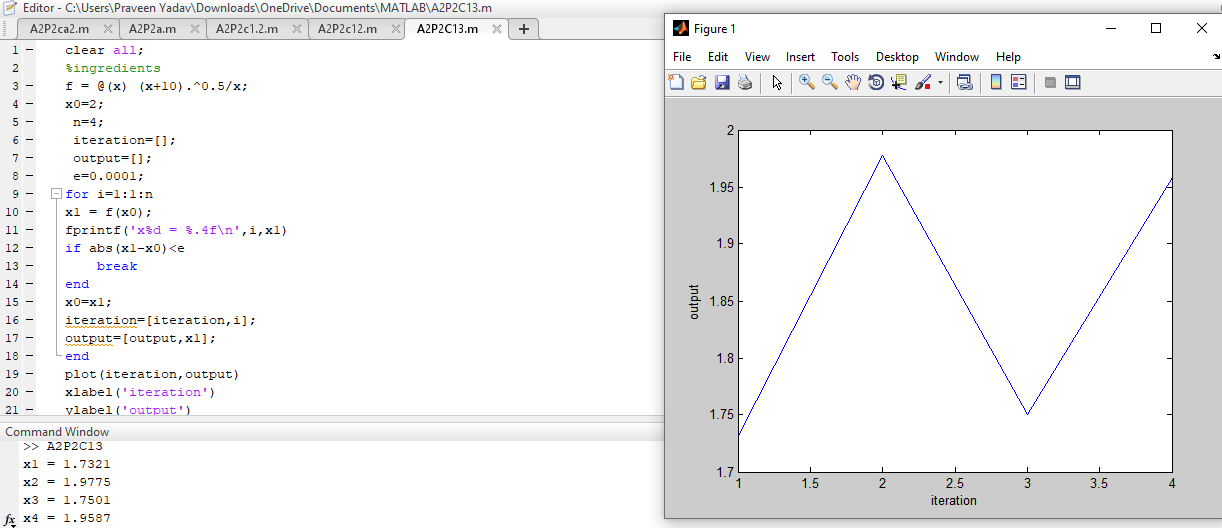
output=[output,x1];

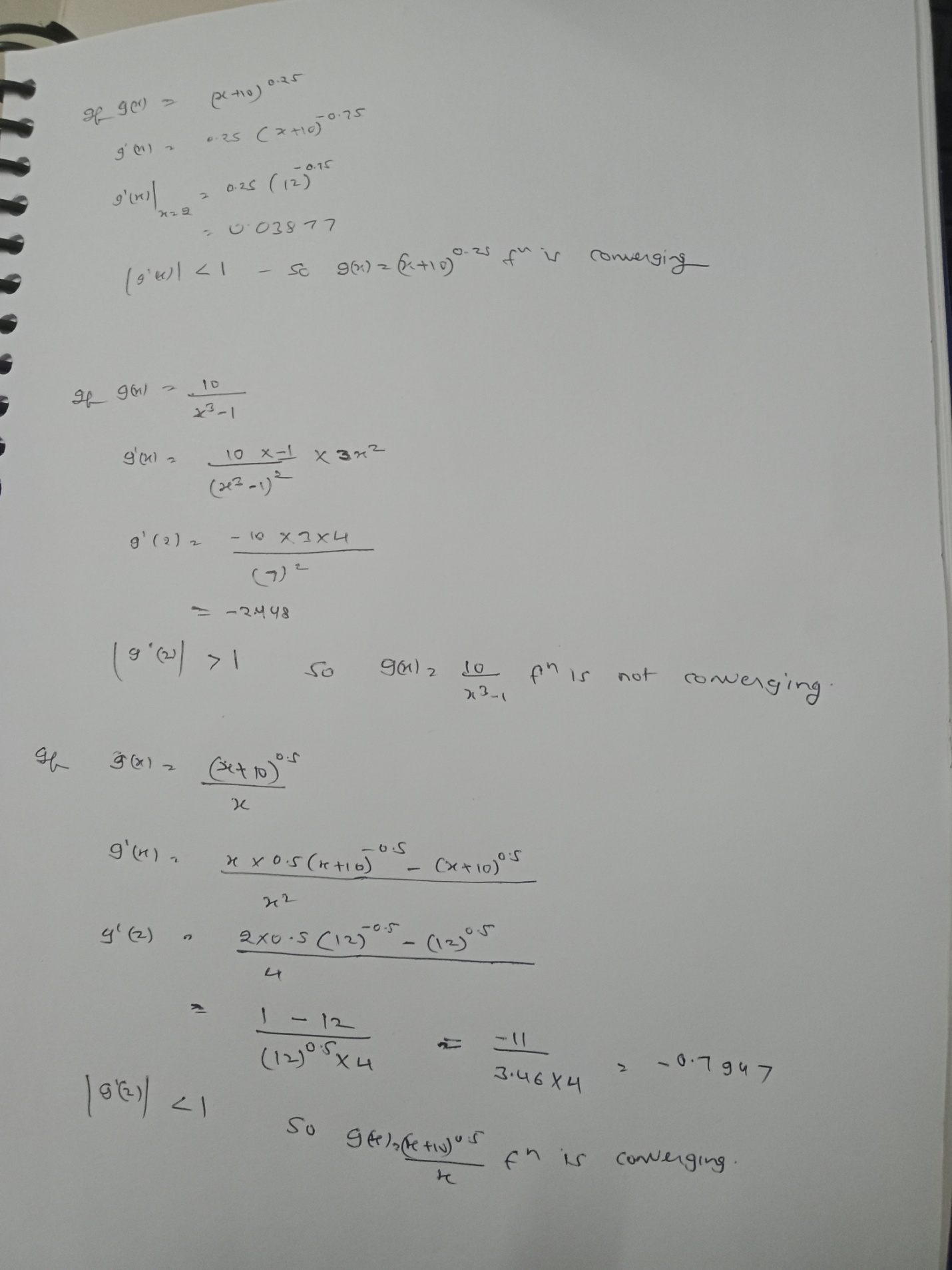
end

plot(iteration,output)

xlabel('iteration')

ylabel('output')





d) Newton’s method (Take initial guess of 2 and see what solution you get. Can you obtain some other solution by choosing a different initial guess? Also plot the rate of convergence.)

code:

f=@(x) x.^4-x-10;

fdiff = @(x) 4\*x.^3-1;

iterations =[];

output=[];

x0 = 2;

maxIterations = 10;

e = 0.0001;

for i=1:maxIterations

x1 = x0 - (f(x0)/fdiff(x0))

if abs(x1-x0)<e

break;

end

x0 = x1;

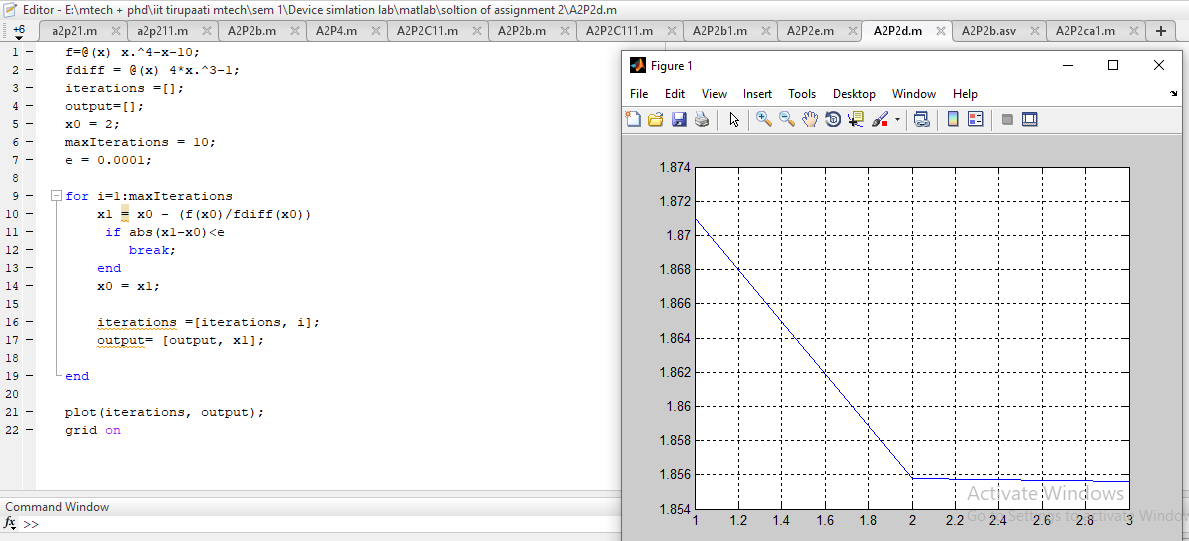
iterations =[iterations, i];

output= [output, x1];

end

plot(iterations, output);

grid on





(e) Secant method

Code:

%ingredients

f = @(x) x^4-x-10;

a=1;

b=4;

n=4;

e=0.0001;

for i=1:n

x2=(x0\*f(x1)-x1f(x0))/(f(x1)-f(x0));

fprintf('x%d = %.50f\n',i,x2)

if abs(x2-x1)<e

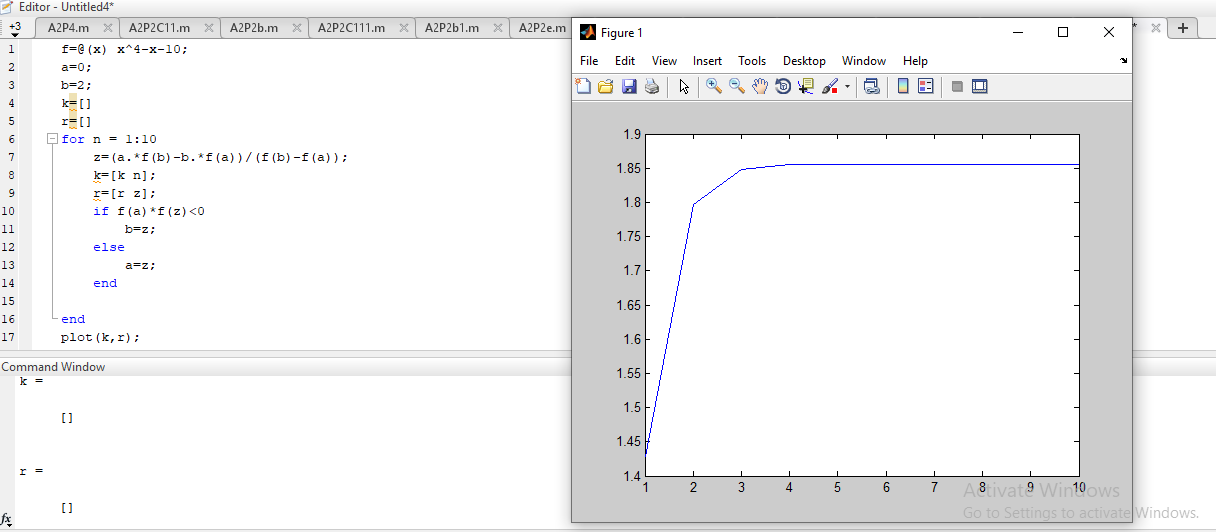
break

end

x0 = x1;

x1 = x2;

end



**Q4:** Use the Newton’s method to solve a system of non-linear equations given below

x1 + 2x2 = 2

x12 + 4x22 = 4

code:

clear all; clc;

syms x1 x2;

f1(x1,x2)=x1+2\*x2-2;

f2(x1,x2)=x1^2+4\*x2^2-4;

x=[10;-10];

e=0.0001;

n=10;

f1x1(x1,x2)=diff(f1,x1);

f1x2(x1,x2)=diff(f1,x2);

f2x1(x1,x2)=diff(f2,x1);

f2x2(x1,x2)=diff(f2,x2);

f11=matlabFunction(f1);

f22=matlabFunction(f2);

f1x11=matlabFunction(f1x1);

f1x22=matlabFunction(f1x2);

f2x11=matlabFunction(f2x1);

f2x22=matlabFunction(f2x2);

for i=1:n;

f=[f11(x(1),x(2));f22(x(1),x(2))];

j=[f1x11(x(1),x(2)),f1x22(x(1),x(2));...

f2x11(x(1),x(2)),f2x22(x(1),x(2))];

y=-j\f;

if norm(y)<e

fprintf('solution found in %d iterations\n',i)

break

end

x=x+y

end

