NAME: MANI GARG ROLL NO: 102003470 BATCH: 2COE19

EXPERIMENT-2

Q1

2. The equation f(x) = $x^3 + 4x^2 - 10 = 0$ has a unique root in [1,2]. There are many ways to change the equation to the fixed-point form $x = g(x)$ using simple algebraic manipulation. Let g_1, g_2, g_3, g_4 and g_5 are iteration functions obtained by the given function, then check which of the following iteration functions will converge to the fixed point? (Tolerance $\epsilon = 10^{-3}$)

(a) $g_1(x) = x - x^3 - 4x^2 + 10$

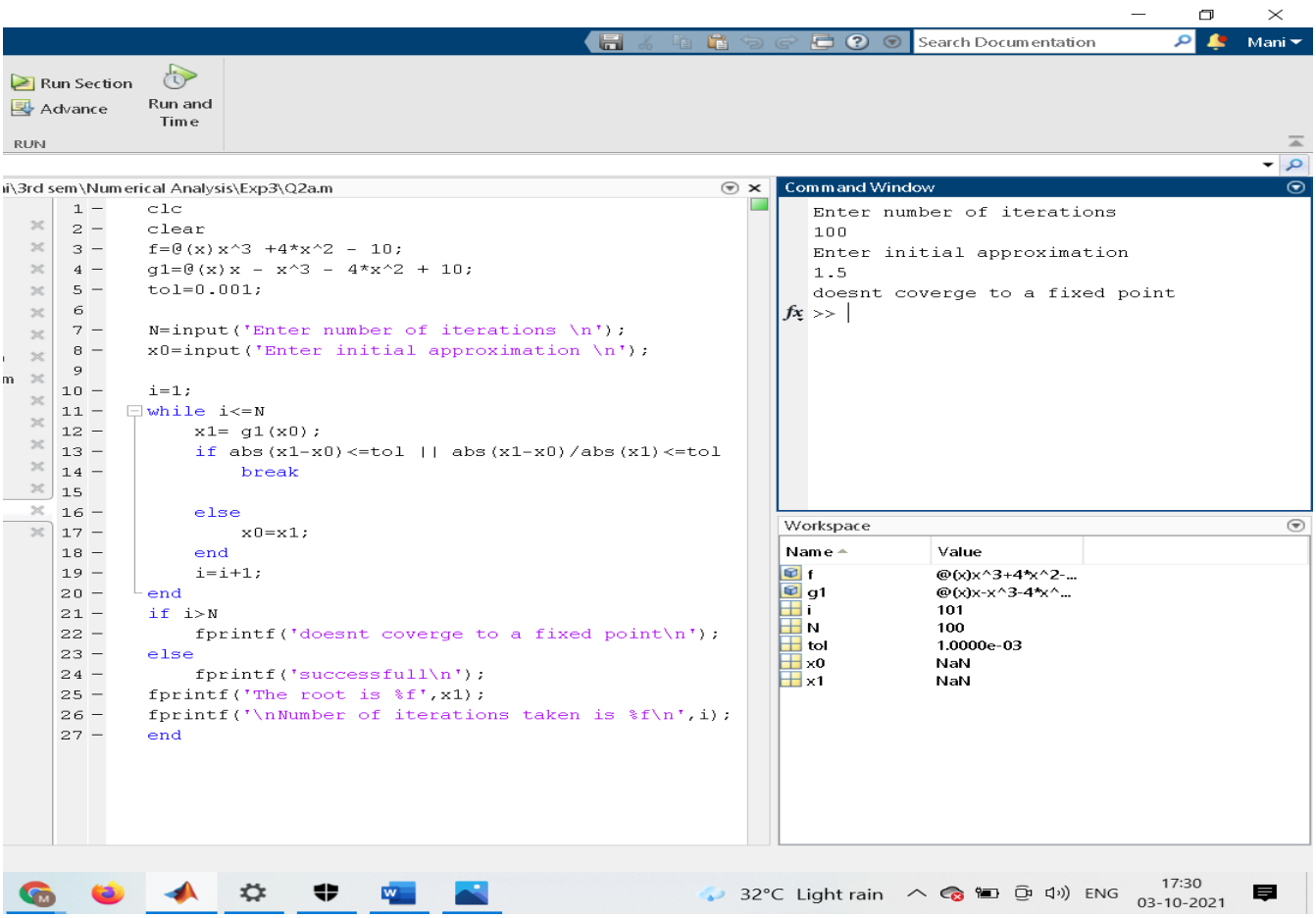
(b) $g_2(x) = \sqrt{\frac{10}{x} - 4x}$

(c) $g_3(x) = 0.5\sqrt{10 - x^3}$

(d) $g_4(x) = \sqrt{\frac{10}{4+x}}$

(e) $g_5(x) = x - \frac{x^3+4x^2-10}{3x^2+8x}$

Ans1(a)

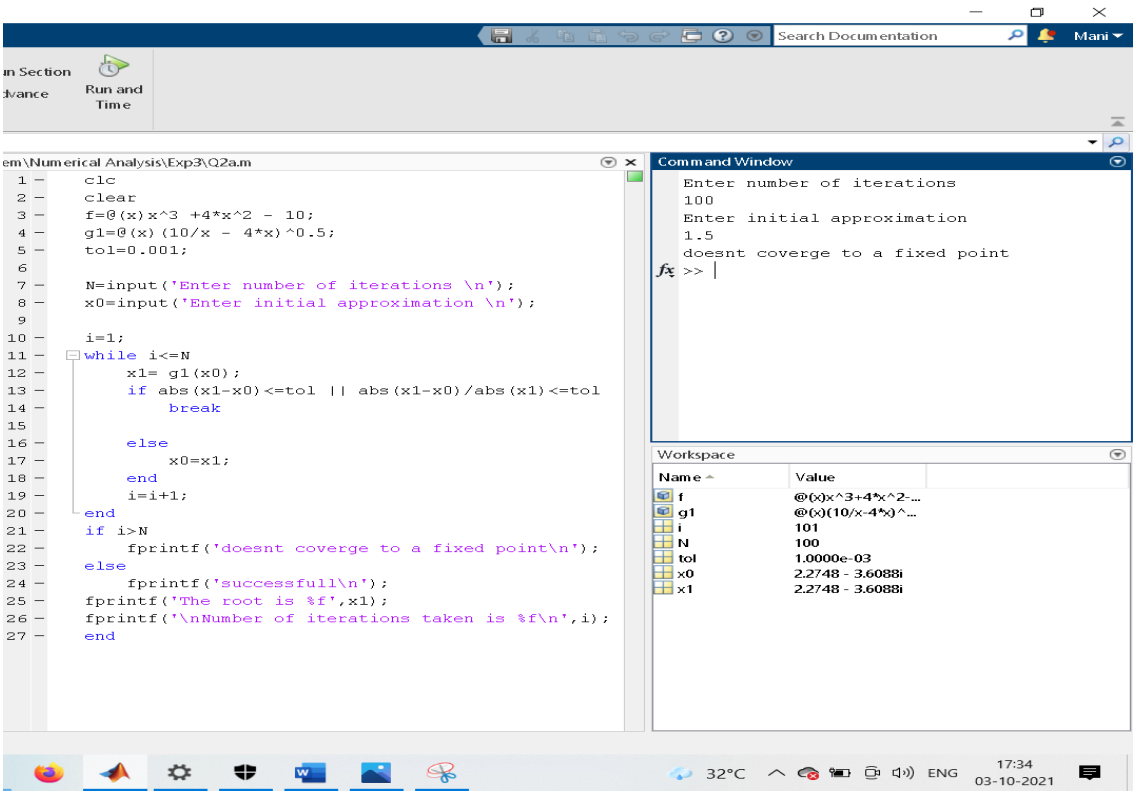


CODE:

```
clc
clear
f=@(x)x^3 +4*x^2 - 10;
g1=@(x)x - x^3 - 4*x^2 + 10;
tol=0.001;

N=input('Enter number of iterations \n');
x0=input('Enter initial approximation \n');

i=1;
while i<=N
    x1= g1(x0);
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
```

```
        break

    else
        x0=x1;
    end
    i=i+1;
end
if i>N
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The root is %f',x1);
fprintf('\nNumber of iterations taken is
%f\n',i);
end
```

Ans1(b)



CODE:

```
clc

clear

f=@(x)x^3 +4*x^2 - 10;

g1=@(x)(10/x - 4*x)^0.5;

tol=0.001;

N=input('Enter number of iterations \n');
```

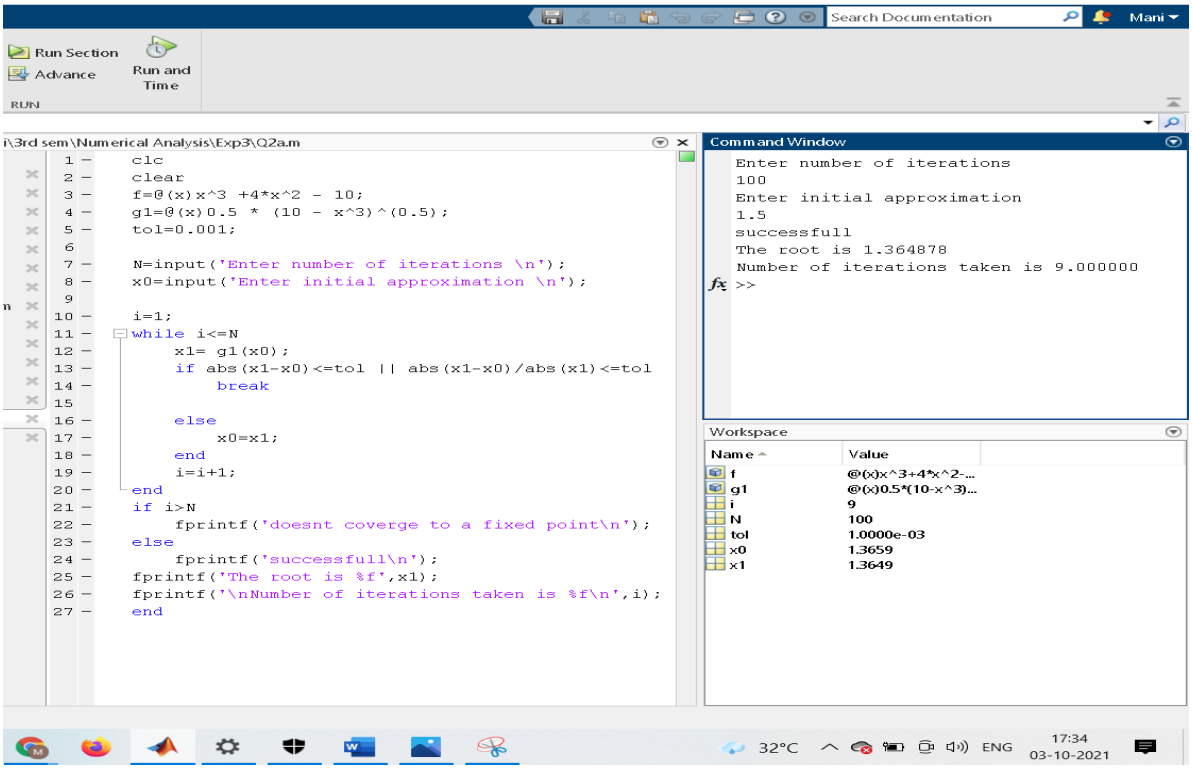
```
x0=input('Enter initial approximation \n');

i=1;
while i<=N
   x1= g1(x0);
   if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
      break

   else
      x0=x1;
   end
   i=i+1;
end
if i>N
   fprintf('doesnt coverge to a fixed point\n');
else
   fprintf('successfull\n');
```

fprintf('The root is %f',x1);

fprintf('\nNumber of iterations taken is %f\n',i);

end

Ans1(c)



CODE:

clc

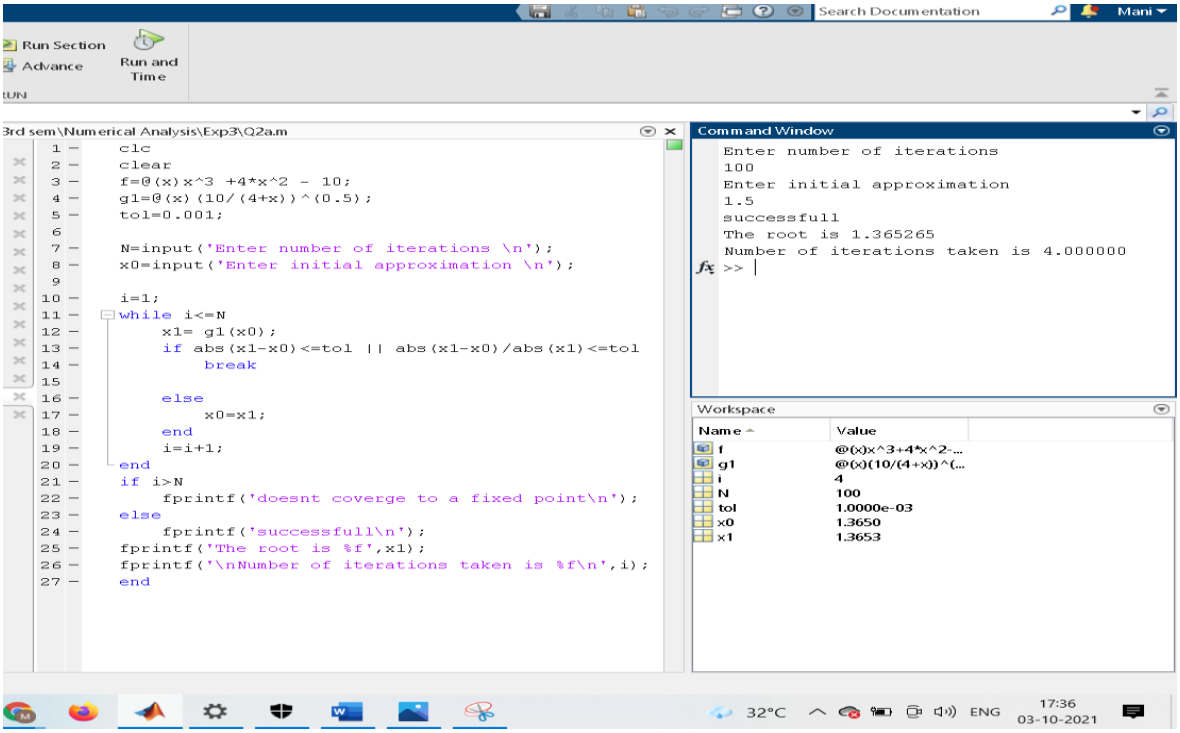
clear

f=@(x)x^3 +4*x^2 - 10;

```
g1=@(x)0.5 * (10 - x^3)^(0.5);
tol=0.001;

N=input('Enter number of iterations \n');
x0=input('Enter initial approximation \n');

i=1;
while i<=N
   x1= g1(x0);
   if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
      break

   else
      x0=x1;
   end
   i=i+1;
end
```

```
if i>N
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The root is %f',x1);
fprintf('\nNumber of iterations taken is %f\n',i);
end
```

Ans1(d)

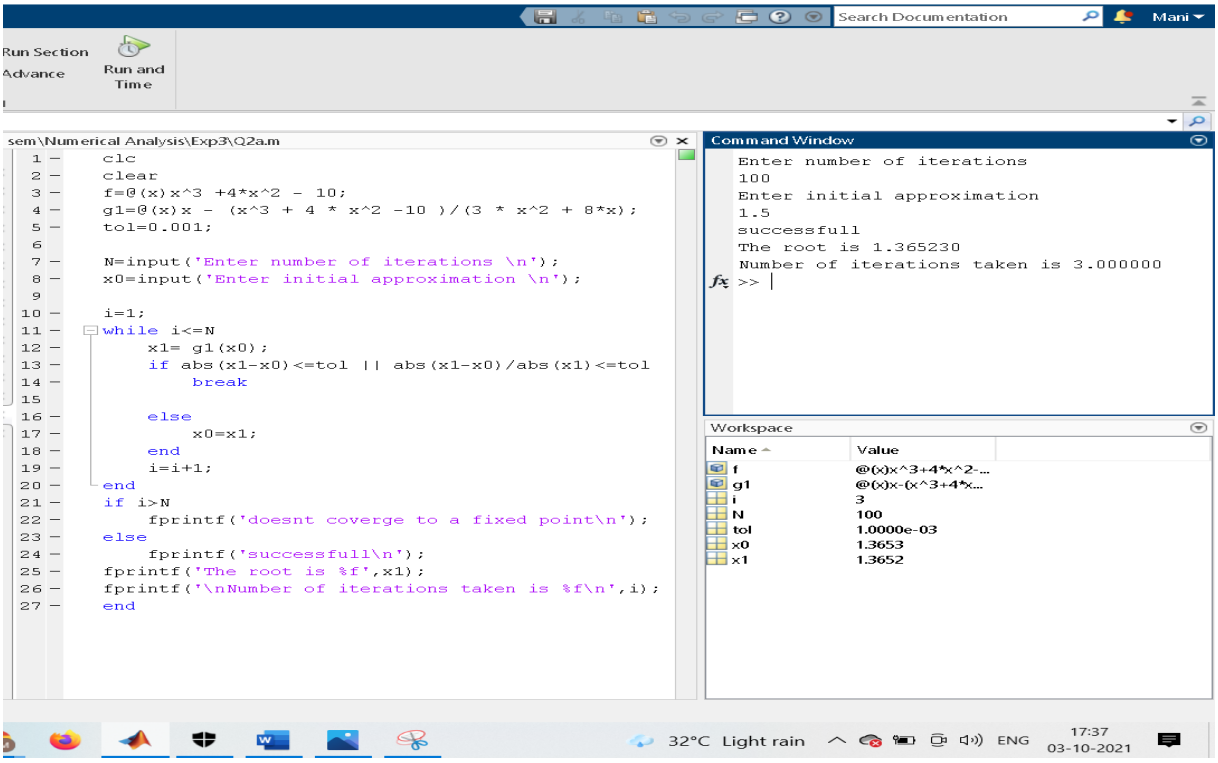


CODE:

```
clc
clear
f=@(x)x^3 +4*x^2 - 10;
g1=@(x)(10/(4+x))^(0.5);
tol=0.001;

N=input('Enter number of iterations \n');
x0=input('Enter initial approximation \n');

i=1;
while i<=N
   x1= g1(x0);
   if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
      break

   else
```

```
        x0=x1;
    end
    i=i+1;
end
if i>N
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The root is %f',x1);
fprintf('\nNumber of iterations taken is
%f\n',i);
end
```

Ans1(e)



CODE:

```
clc
clear
f=@(x)x^3 +4*x^2 - 10;
g1=@(x)x - (x^3 + 4 * x^2 -10 )/(3 * x^2 + 8*x);
tol=0.001;

N=input('Enter number of iterations \n');
x0=input('Enter initial approximation \n');
```

```
i=1;
while i<=N
    x1= g1(x0);
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        break

    else
        x0=x1;
    end
    i=i+1;
end
if i>N
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The root is %f',x1);
```

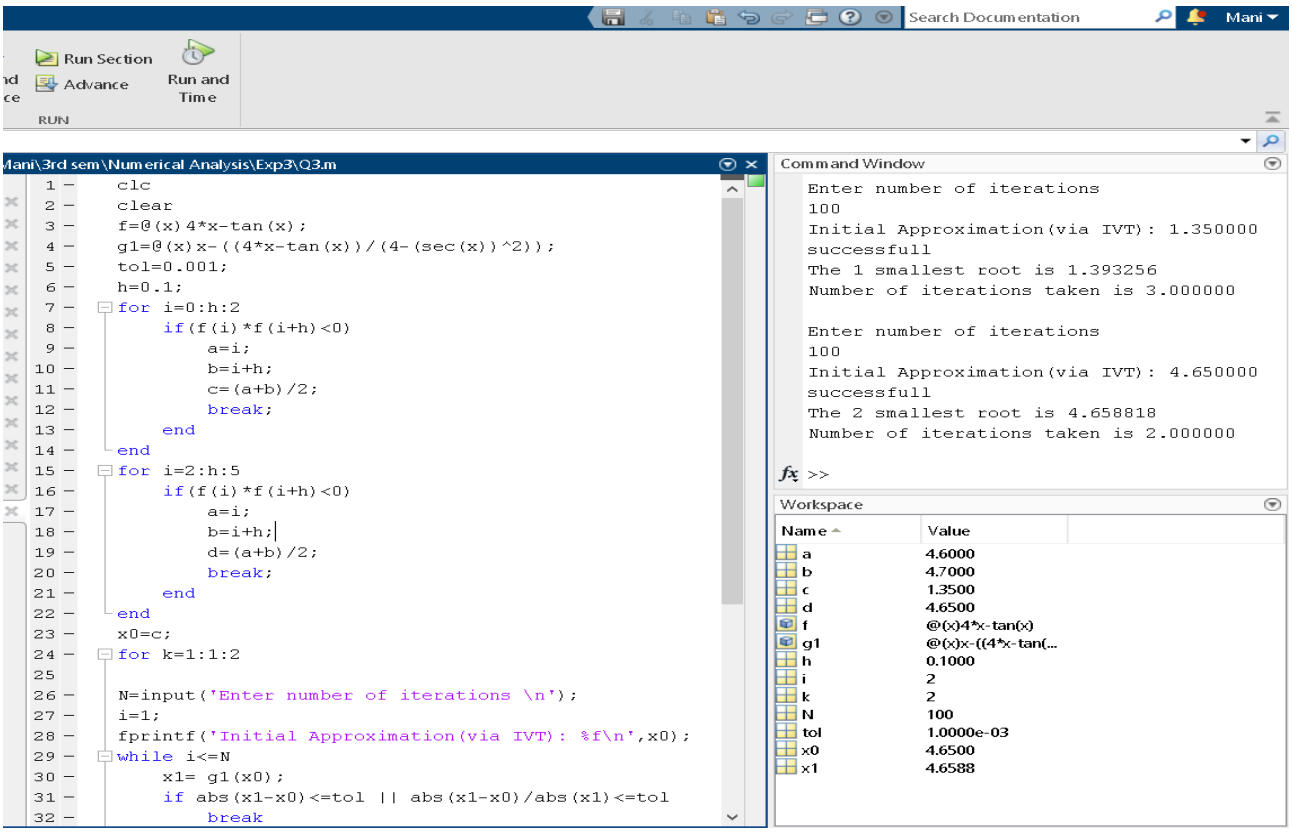
fprintf('\nNumber of iterations taken is %f\n',i);

end

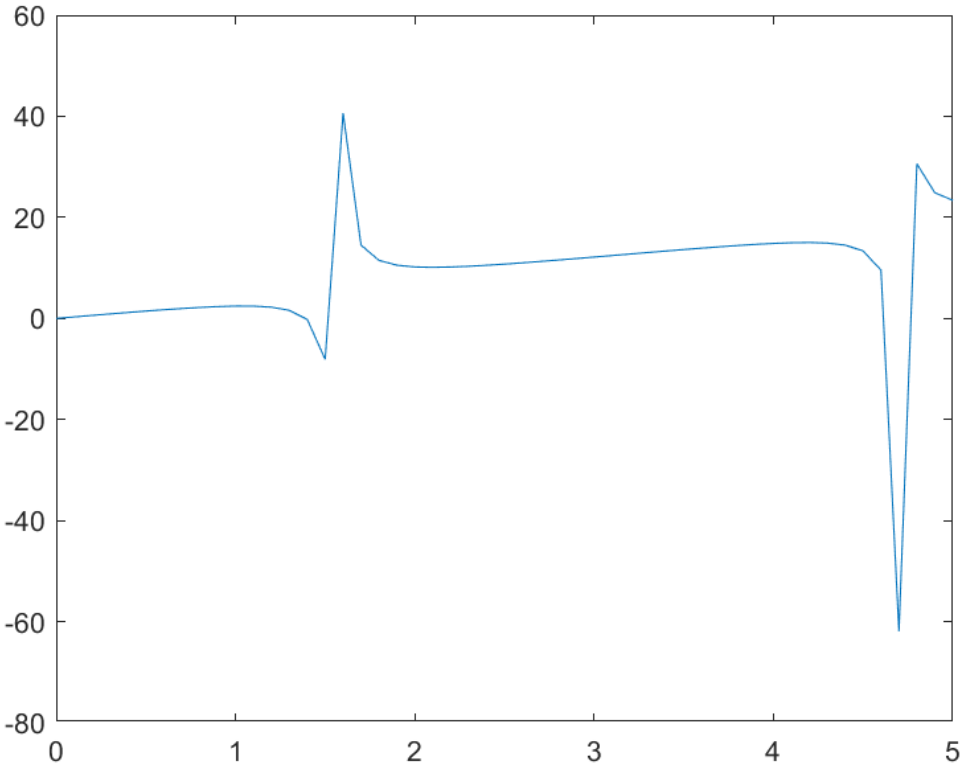
Q2

3. Find the smallest and second smallest positive roots of the equation tan(x) = 4x, with an accuracy of 10^{-3} using fixed-point iterations.

Ans2



GRAPH:



CODE:

```
clc
clear
f=@(x)4*x-tan(x);
g1=@(x)x-((4*x-tan(x))/(4-(sec(x))^2));
tol=0.001;
h=0.1;
for i=0:h:2
```

```
    if(f(i)*f(i+h)<0)
        a=i;
        b=i+h;
        c=(a+b)/2;
        break;
    end
end
for i=2:h:5
    if(f(i)*f(i+h)<0)
        a=i;
        b=i+h;
        d=(a+b)/2;
        break;
    end
end
x0=c;
```

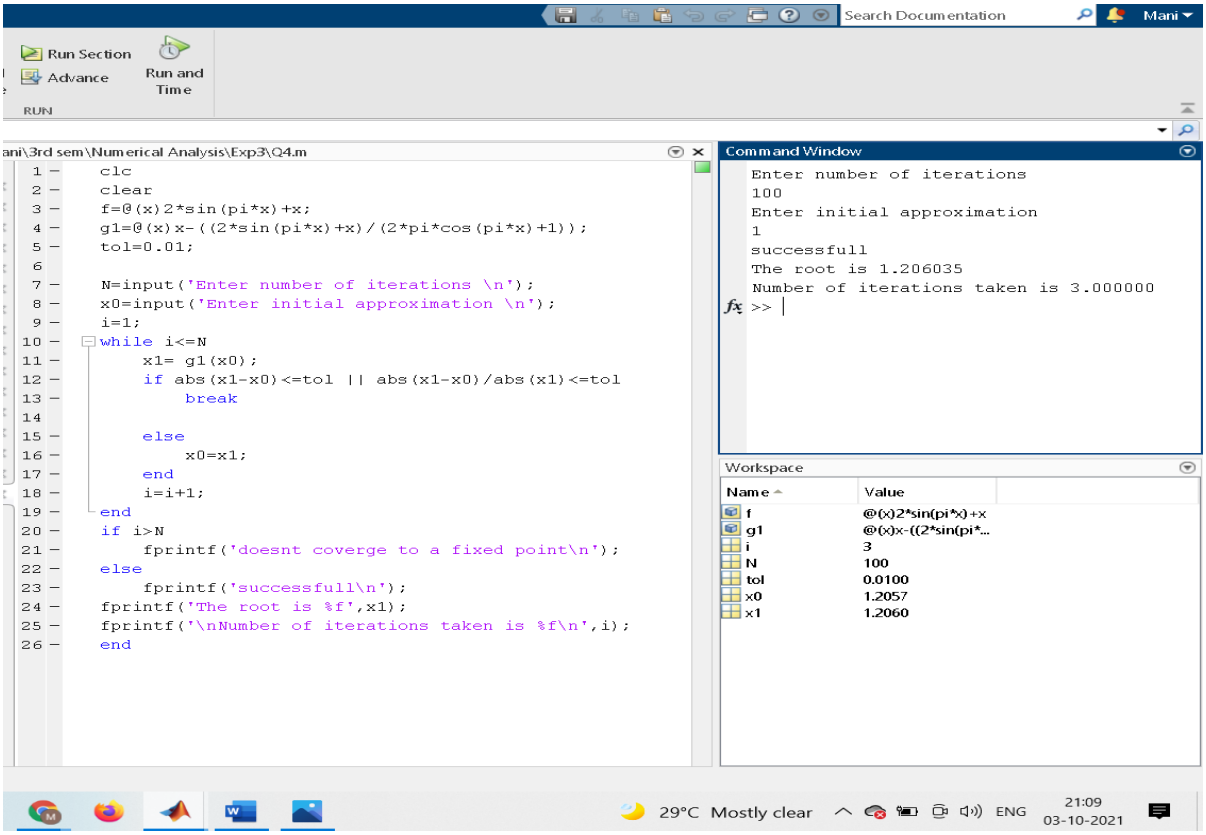
```
for k=1:1:2

N=input('Enter number of iterations \n');
i=1;
fprintf('Initial Approximation(via IVT):
%f\n',x0);
while i<=N
   x1= g1(x0);
   if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
      break

   else
      x0=x1;
   end
   i=i+1;
end
if i>N
```

```
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The %d smallest root is %f',k,x1);
fprintf('\nNumber of iterations taken is %f\n\n',i);
end

x0=d;
end
```

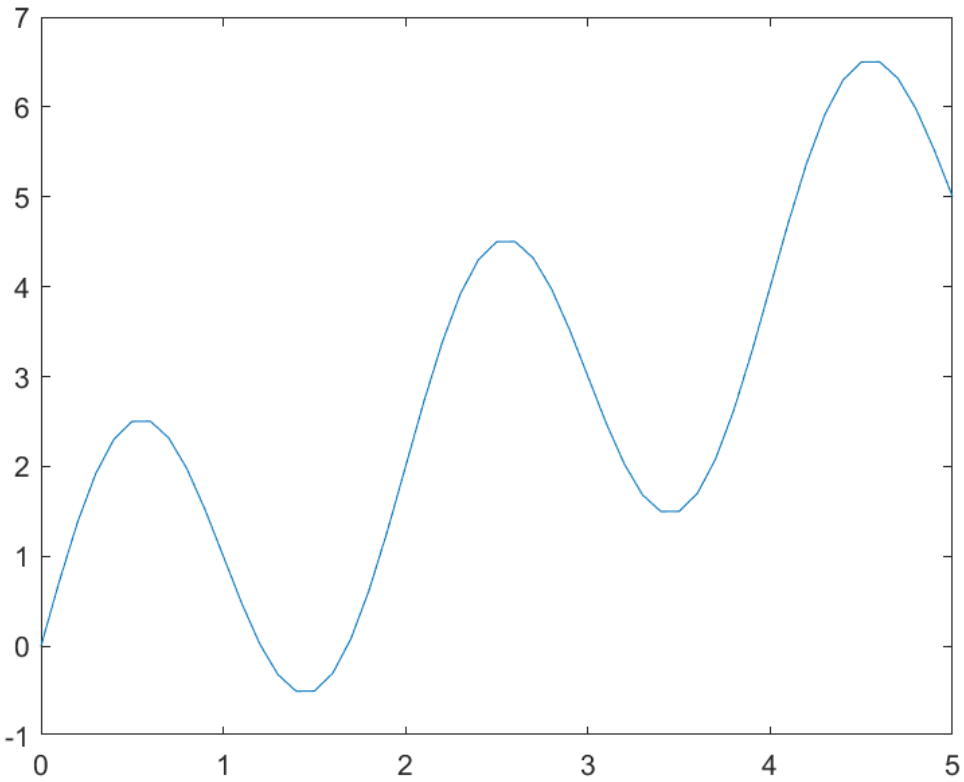
Q3

4. Use a fixed-point iteration method to determine a solution accurate to within 10^{-2} for $2\sin\pi x + x = 0$ on [1,2]. Use initial guess $x_0 = 1$.

Ans3



GRAPH:



CODE:

```
clc
clear
f=@(x)2*sin(pi*x)+x;
g1=@(x)x-((2*sin(pi*x)+x)/(2*pi*cos(pi*x)+1));
tol=0.01;

N=input('Enter number of iterations \n');
x0=input('Enter initial approximation \n');
```

```
i=1;
while i<=N
    x1= g1(x0);
    if abs(x1-x0)<=tol || abs(x1-x0)/abs(x1)<=tol
        break

    else
        x0=x1;
    end
    i=i+1;
end
if i>N
    fprintf('doesnt coverge to a fixed point\n');
else
    fprintf('successfull\n');
fprintf('The root is %f',x1);
```

```
fprintf('\nNumber of iterations taken is
%f\n',i);
end
```