Assignment 1

Group Number 32

Software used: **Matlab**

Sanit Prashant Bhatkar (173109003)
Omkar Pawar(173106001)

1. Bisection Method

Code	Output

```
fprintf('\n\nBisection Method\n\n');
string=input('Function f(x): ','s');
fun=inline(string);
a=input('Start interval value a = ');
b=input('End interval value b = ');
c=(a+b)/2.0;
e(1)= b-c;
array(1)=c;
i=1;
if fun(a)*fun(b)>0
    disp('wrong interval')

else
while b-c >= 10^-6

    if fun(a)*fun(c)<0
           b=c;
    else
           a=c;
    end
    i=i+1;
    c=(a+b)/2;
    e(i)= b-c;
    array(i)=c;
end
end
Root= c
Iteration=i
array=array(1:i);
e=e(1:i);
figure
subplot(2,1,1)
plot(e,'r-o');
title('Error vs Iterations')
grid on;
xlim([0 i+1]);
ylim([-0.1 e(1)+0.2]);
subplot(2,1,2)
plot(array,'-o')
title('Root vs Iterations')
grid on;
xlim([0 i+1]);
```

a) For x-cos(x)

Output:

Bisection Method

Function f(x): x-cos(x)
Start interval value a = 0
End interval value b = 1

Root =

0.739085197448730

Iteration =

20

b) For x^6-x-1

Output:

Bisection Method

Function f(x): x^6-x-1
Start interval value a = 1
End interval value b = 2

Root =

1.1347

Iteration =

20

	Root	Number of Iterations
a) x-cos(x)	0.739085197448730	20
b) x^6-x-1	1.134724617004395	20

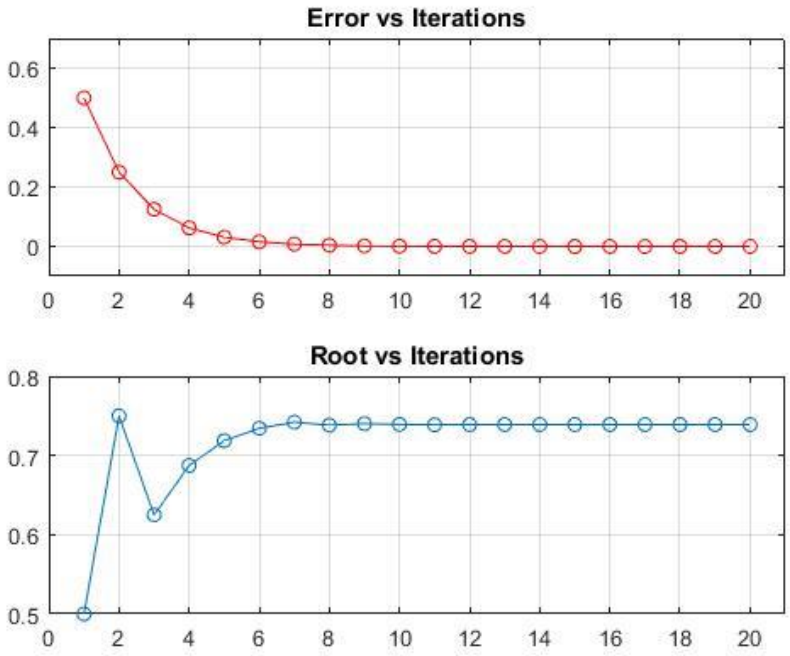


Fig (1): Bisection method convergence graph for f(x) =x-cos(x)

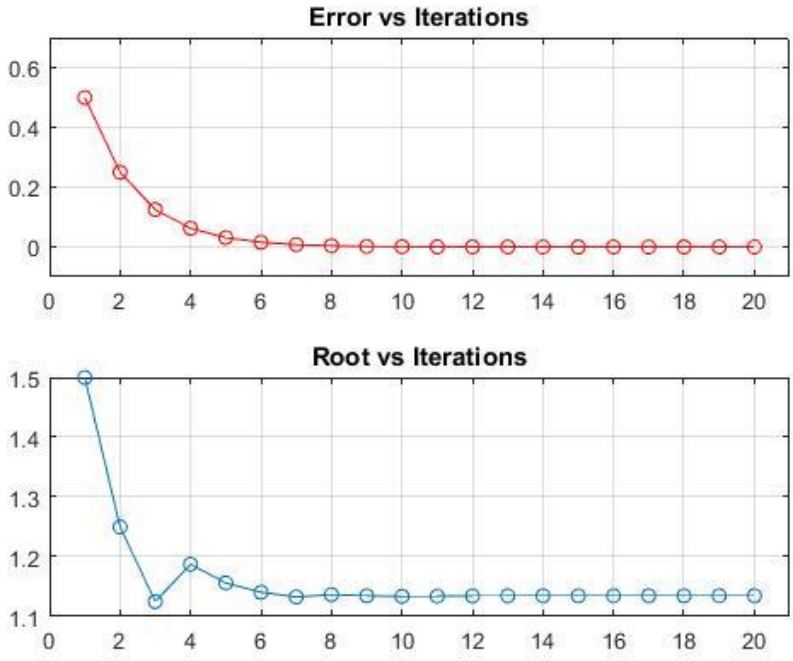


Fig (2): Bisection method convergence graph for f(x) =x^6-x-1

2. Newton Method

Code

```
fprintf('\n\n\nNewton Raphson Method\n');
string1=input('\n\nEnter f(x): ','s');
f=inline(string1);
string2=input('Enter df(x): ','s');
df=inline(string2);
xk(1)=input('Enter initial guess: ');
array(1)=xk(1);
it=0;
e=10^-6;

for i=2:1000
   xk(i) = xk(i-1) - (f(xk(i-1))/df(xk(i-1)));
    it=it+1;
    array(i)=xk(i);
    err(i)=abs((xk(i)-xk(i-1))/xk(i));
    if abs((xk(i)-xk(i-1))/xk(i))*100<e
        root= xk(i)
        Iterations =it
        break
    end
end
array=array(1:it);
err=err(2:it+1);
figure
subplot(2,1,1)
plot(err,'r-o');
title('Relative error vs Iterations')
xlim([0 it+1]);
ylim([-0.1 1.2]);
grid on;
subplot(2,1,2)
plot(array,'-o')
title('Root vs Iterations')
grid on;
xlim([0 it+1]);
```

Output

a) For x-cos(x)

Output:

```
Newton Raphson Method


Enter f(x): x-cos(x)
Enter df(x): 1+sin(x)
Enter initial guess: 0.5

root =

   0.739085133215161


Iterations =

     4
```

b) For x^6-x-1

Output:

```
Newton Raphson Method


Enter f(x): x^6-x-1
Enter df(x): 6*x^5-1
Enter initial guess: 1.5

root =

   1.134724138401520


Iterations =

     6
```

	Root	Number of Iterations
a) x-cos(x)	0.739085133215161	4
b) x^6-x-1	1.134724138401520	6

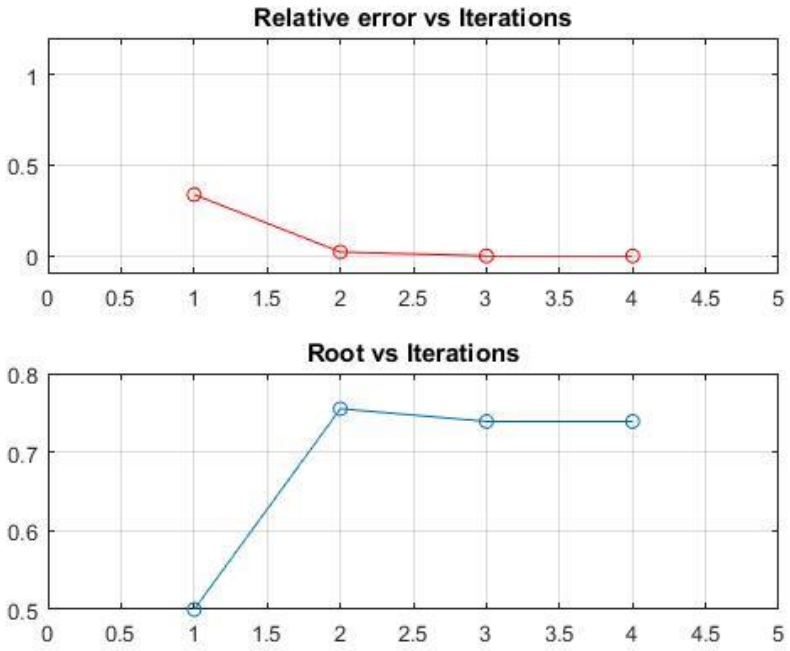


Fig (3): Newton method convergence graph for f(x) =x-cos(x)

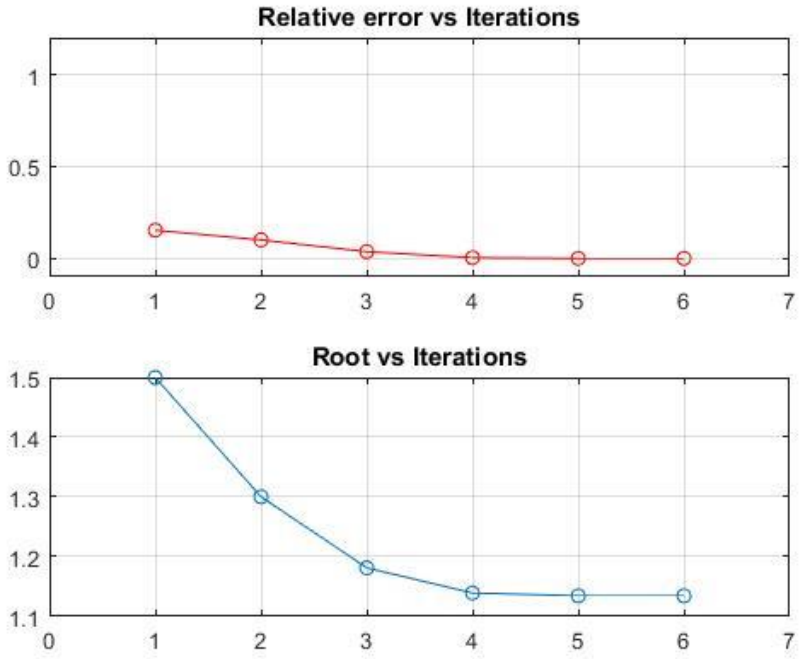
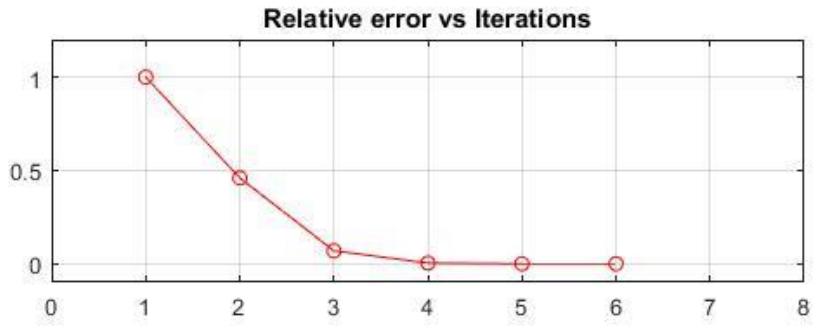


Fig (4): Newton method convergence graph for f(x) =x^6-x-1

3. Secant Method

Code	Output
`` fprintf('\n\nSecant Method\n\n'); string=input('Function f(x): ','s'); f=inline(string); xk(1)=input('Enter start of interval: '); array(1)=xk(1); xk(2)=input('Enter end interval: '); array(2)=xk(2); err(2)=abs((xk(2)-xk(1))/xk(2)); e=10^-6; it=0; for i=3:1000 xk(i) = xk(i-1) - (f(xk(i-1)))*((xk(i-1) - xk(i-2))/(f(xk(i-1)) - f(xk(i-2)))); array(i)=xk(i); err(i)=abs((xk(i)-xk(i-1))/xk(i)); it=it+1; if abs((xk(i)-xk(i-1))/xk(i))*100<e root= xk(i) Iterations =it-1 break end end array=array(1:it); err=err(2:it+1); figure subplot(2,1,1) plot(err,'r-o'); title('Relative error vs Iterations') xlim([0 it+1]); ylim([-0.1 1.2]); grid on; subplot(2,1,2) plot(array,'-o') title('Root vs Iterations') grid on; xlim([0 it+1]); ```	**a)** For x-cos(x) **Output:** Secant Method Function f(x): x-cos(x) Enter start of interval: 0 Enter end interval: 1 root = 0.739085133215161 Iterations = 5 **c)** For x^6-x-1 **Output:** Secant Method Function f(x): x^6-x-1 Enter start of interval: 1 Enter end interval: 2 root = 1.134724138401519 Iterations = 8

	Root	Number of Iterations
a) x-cos(x)	0.739085133215161	5
b) x^6-x-1	1.134724138401519	8



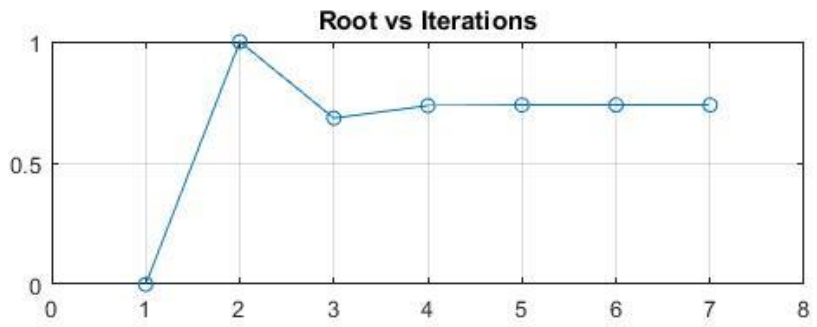
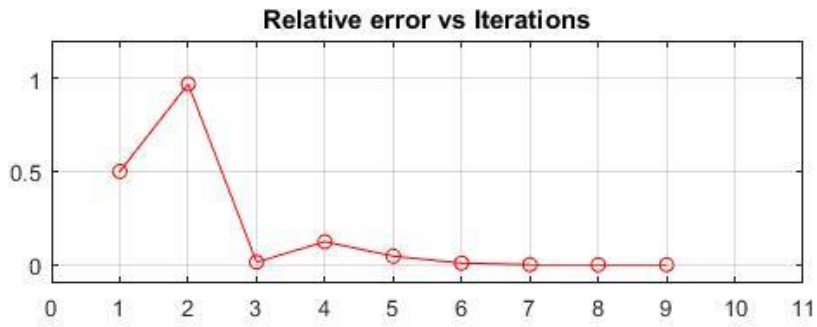


Fig (5): Secant method convergence graph for f(x) =x-cos(x)



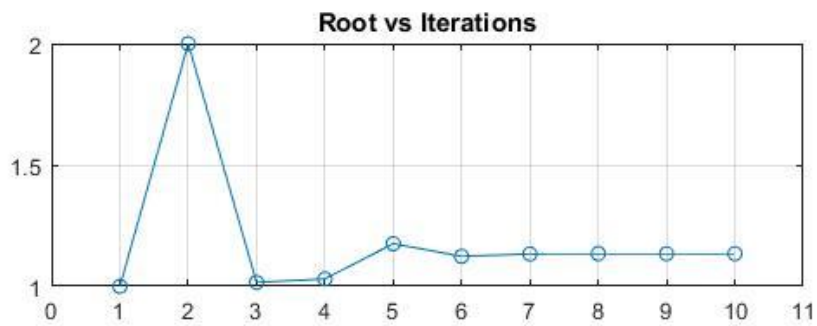


Fig (6): Secant method convergence graph for f(x) =x^6-x-1

Conclusion:

F(x)= x-cos(x)	Root	Iterations
Bisection method	0.739085197448730	20
Newton Raphson method	0.739085133215161	4
Secant method	0.739085133215161	5

F(x)= x^6-x-1	Root	Iterations
Bisection method	0.739085197448730	20
Newton Raphson method	0.739085133215161	6
Secant method	0.739085133215161	8

Number of iterations for convergence

Bisection method > Secant method > Newton Raphson method