SCIENTIFIC COMPUTING

ASSIGNMENT 4

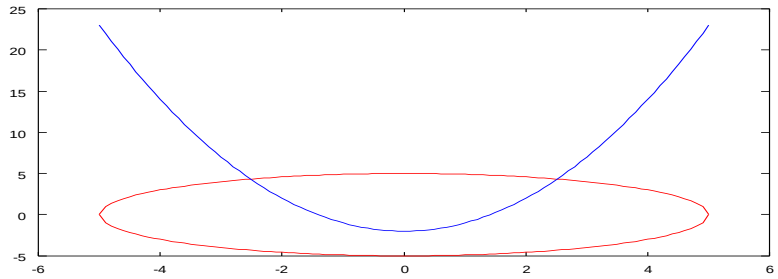
TUSHAR SIRCAR
130123038

1. SOURCE CODE:

```
function v = func1(x)

v = (ones(size(x))*25) - (x.^2);
v = sqrt(v);

plot(x,v,"color","r");
plot(x,-v,"color","r");


end
```

```
function v = func2(x)
v = (x.^2)- (2*ones(size(x)));
plot(x,v);
end
```

```
function v = makePlot()

x = [-5:0.1:5];
hold on;
func1(x);
func2(x);
hold off;

end
```



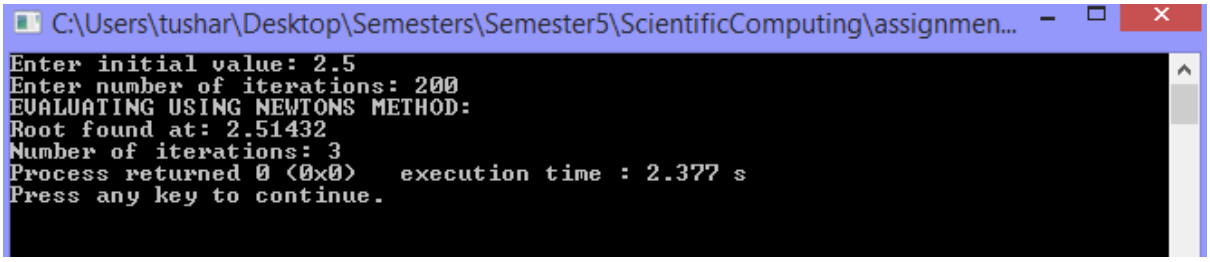
From graph, solution to the two equations is -2.5 and 2.5.
Two root are imaginary!

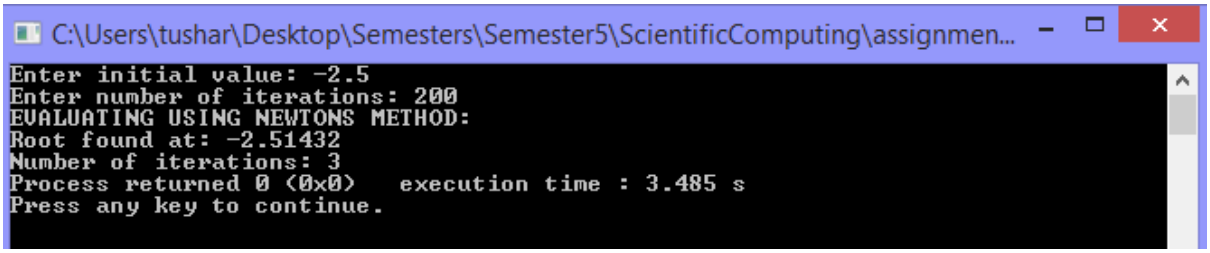
2.

(a) SOURCE CODE

```
1   #include<iostream>
2   #include<cmath>
3   using namespace std;
4
5   double getF(double x)
6   {
7       return (x*x*x*x) - (3*x*x) - 21;
8   }
9
10  double getFDASH(double x)
11  {
12      return (4*x*x*x) - (6*x);
13  }
14
15  int main()
16  {
17      double prevVal;
18      double currentVal;
19      double firstInitial;
20      double secondInitial;
21      int n;
22      double epsilon = 0.0001;
23      bool rootFound = false;
24
25      cout<<"Enter initial value: ";
26      cin>>firstInitial;
27      prevVal = firstInitial;
28
29      cout<<"Enter number of iterations: ";
30      cin>>n;
31
```

```
32      cout<<"EVALUATING USING NEWTONS METHOD:\n";
33      for(int i=0; i<n; i++)
34      {
35          currentVal = prevVal - (getF(prevVal)/getFDASH(prevVal));
36          if(abs(currentVal - prevVal) < epsilon)
37          {
38              cout<<"Root found at: "<<currentVal;
39              cout<<"\nNumber of iterations: "<<i+1;
40              rootFound = true;
41              break;
42          }
43          prevVal = currentVal;
44        //  cout<<prevVal<<endl;
45      }
46
47      if(!rootFound)
48          cout<<"Failed to converge!";
49
50
51
52
53      return 0;
54  }
```





NEWTONS METHOD
Initial Value: 2.5
Root Found: 2.51432
Number Of Iterations: 3

NEWTONS METHOD
Initial Values: -2.5
Root Found: -2.51432
Number Of Iterations: 3

(b). SOURCE CODE

```cpp
#include<iostream>
#include<cmath>
using namespace std;

double getG(double x,double y)
{
    return sin(x+y) - exp(x-y);
}
double getGx(double x,double y)
{
    return cos(x+y) - exp(x-y);
}
double getGy(double x,double y)
{
    return cos(x+y) + exp(x-y);
}
double getH(double x,double y)
{
    return cos(x+6) - (x*x*y*y);
}
double getHx(double x,double y)
{
    return -sin(x+6) - (2*x*y*y);
}
double getHy(double x,double y)
{
    return (-2*x*x*y);
}

int main()
```

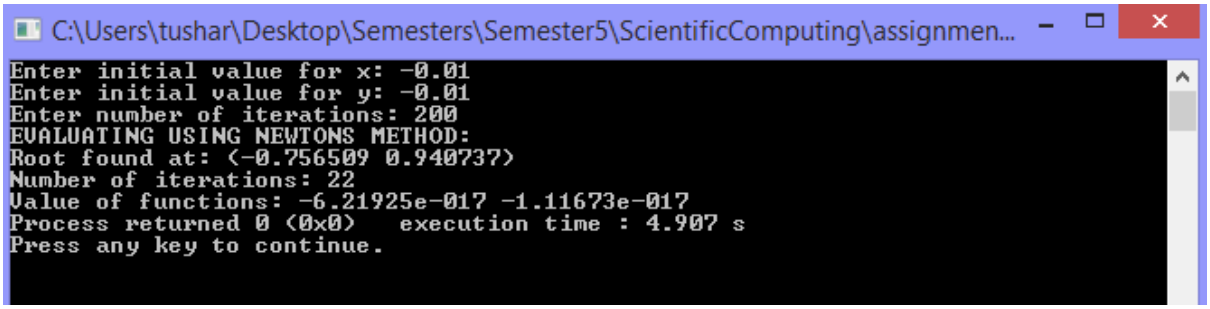
```cpp
#include<iostream>
#include<cmath>
using namespace std;

double getF(double x)
{
    return log((x*x)+1) - (exp(0.4*x)*cos(3.14*x));
}

int signum(double x)
{
    if( x < 0)
        return -1;
    if( x > 0)
        return 1;
    return 0;
}

int main()
{
    double low,high;
    int n;
    double epsilon = 0.000001;
    bool rootFound = false;
    double prevValue;

    cout<<"Enter interval to find ONLY NEGATIVE ZERO: ";
    cin>>low>>high;

    cout<<"Enter number of iterations: ";
    cin>>n;
```

```
int main()
{
    double prevValX,prevValY;
    double currentValX,currentValY;
    double initialX,initialY;
    int n;
    double epsilon = 0.0000001;
    bool rootFound = false;

    cout<<"Enter initial value for x: ";
    cin>>initialX;
    cout<<"Enter initial value for y: ";
    cin>>initialY;
    prevValX = initialX;
    prevValY = initialY;
```

```

cout<<"Enter number of iterations: ";
cin>>n;

cout<<"EVALUATING USING NEWTONS METHOD:\n";

for(int i=0; i<n; i++)
{
    double x = prevValX;
    double y = prevValY;
    currentValX = x - ((getG(x,y)*getHy(x,y) - getH(x,y)*getGy(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
    currentValY = y - ((getH(x,y)*getGx(x,y) - getG(x,y)*getHx(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
    if(abs(currentValX - prevValX) < epsilon && abs(currentValX - prevValX) < epsilon)
    {
        cout<<"Root found at: ("<<currentValX<<" "<<currentValY<<")";
        cout<<"\nNumber of iterations: "<<i+1;
        rootFound = true;
        cout<<endl<<"Value of functions: "<<getG(currentValX,currentValY)<<" "<<getH(currentValX,currentValY);
        break;
    }
    prevValX = currentValX;
    prevValY = currentValY;
   // cout<<prevVal<<endl;
}
```

```
    if(!rootFound)
        cout<<"Failed to converge!";



    return 0;
}
```

Using Cauchy Rimeanns equation and using Newtons Method for complex numbers as given in ques 4, we get the following result:

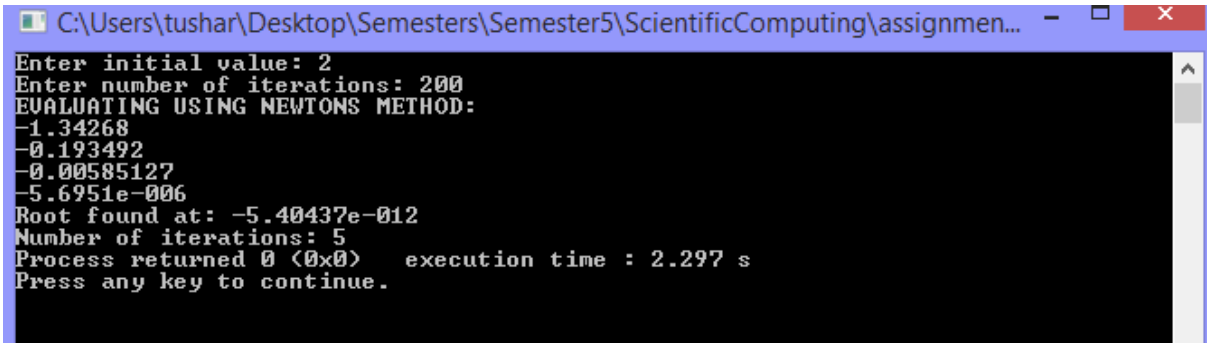


Root Found At: (-0.756509,0.940737) that is -0.756509 + 0.940737i
Number Of Iterations: 22

3. SOURCE CODE

```
1   #include<iostream>
2   #include<cmath>
3   using namespace std;
4
5   double getF(double x)
6   {
7       return 1 - (x/(exp(x)-1));
8   }
9
10  double getFDASH(double x)
11  {
12      return (exp(x)*(x-1) + 1)/((exp(x)-1)*(exp(x)-1));
13  }
14
15  int main()
16  {
17      double prevVal;
18      double currentVal;
19      double firstInitial;
20      double secondInitial;
21      int n;
22      double epsilon = 0.0001;
23      bool rootFound = false;
24
25      cout<<"Enter initial value: ";
26      cin>>firstInitial;
27      prevVal = firstInitial;
28
29      cout<<"Enter number of iterations: ";
30      cin>>n;
```

```
31
32      cout<<"EVALUATING USING NEWTONS METHOD:\n";
33      for(int i=0; i<n; i++)
34      {
35          currentVal = prevVal - (getF(prevVal)/getFDASH(prevVal));
36          if(abs(currentVal - prevVal) < epsilon)
37          {
38              cout<<"Root found at: "<<currentVal;
39              cout<<"\nNumber of iterations: "<<i+1;
40              rootFound = true;
41              break;
42          }
43          prevVal = currentVal;
44          cout<<prevVal<<endl;
45      }
46
47      if(!rootFound)
48          cout<<"Failed to converge!";
49
50
51
52      return 0;
53  }
54
```



Root Found At: -5.40437*exp(-12)

Number Of Iterations: 5

4.

$$z_{n+1} = z_n - \frac{f(z)}{f'(z)}$$
$$x_{n+1} + iy_{n+1} = x_n + iy_n - \frac{g+ih}{g_x+ih_x}$$

Rationalizing the fractional part of the equation

$$x_{n+1} + iy_{n+1} = x_n + iy_n - \frac{(g+ih)(g_x+ih_x)}{(g_x+ih_x)(g_x+ih_x)}$$

Separating the real and imaginary parts

$$x_{n+1} = x_n - \frac{gg_x+hh_x}{g_x^2+h_x^2}$$
$$y_{n+1} = y_n - \frac{-gh_x+hg_x}{g_x^2+h_x^2}$$

Using (a) in the above equation

$$x_{n+1} = x_n - \frac{gh_y-hg_y}{g_xh_y-h_xg_y}$$
$$y_{n+1} = y_n - \frac{hg_x-gh_x}{g_xh_y-h_xg_y}$$

Hence proved

5. (a)

SOURCE CODE

```
#include<iostream>
#include<cmath>
using namespace std;

double getG(double x,double y)
{
    return (x*x*x) - (3*x*y*y) - x - 1;
}
double getGx(double x,double y)
{
    return (3*x*x) - (3*y*y) - 1;
}
double getGy(double x,double y)
{
    return -6*x*y;
}
double getH(double x,double y)
{
    return (-y*y*y) + (3*x*x*y) - y;
}
double getHx(double x,double y)
{
    return (6*x*y);
}
double getHy(double x,double y)
{
    return (-3*y*y) + (3*x*x) - 1;
}

```

```cpp
int main()
{
    double prevValX,prevValY;
    double currentValX,currentValY;
    double initialX,initialY;
    int n;
    double epsilon = 0.0001;
    bool rootFound = false;

    cout<<"Enter initial value for x: ";
    cin>>initialX;
    cout<<"Enter initial value for y: ";
    cin>>initialY;
    prevValX = initialX;
    prevValY = initialY;

    cout<<"Enter number of iterations: ";
    cin>>n;

    cout<<"EVALUATING USING NEWTONS METHOD:\n";

    for(int i=0; i<n; i++)
    {
        double x = prevValX;
        double y = prevValY;
        currentValX = x - ((getG(x,y)*getHy(x,y) - getH(x,y)*getGy(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
        currentValY = y - ((getH(x,y)*getGx(x,y) - getG(x,y)*getHx(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
        if(abs(currentValX - prevValX) < epsilon && abs(currentValX - prevValX) < epsilon)
        {
            if(abs(currentValX - prevValX) < epsilon && abs(currentValX - prevValX) < eps
            {
                cout<<"Root found at: ("<<currentValX<<" "<<currentValY<<")";
                cout<<"\nNumber of iterations: "<<i+1;
                rootFound = true;
                break;
            }
            prevValX = currentValX;
            prevValY = currentValY;
           // cout<<prevVal<<endl;
        }

        if(!rootFound)
            cout<<"Failed to converge!";



        return 0;
    }
```

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...
Enter initial value for x: 1
Enter initial value for y: 1
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
Root found at: (-0.662359 -0.56228)
Number of iterations: 11
Process returned 0 (0x0)   execution time : 2.672 s
Press any key to continue.
```

Root Found At: (-0.662359,-0.56228) that is -0.662359 -0.56228i

Number Of Iterations: 11

(b) SOURCE CODE

```cpp
#include<iostream>
#include<cmath>
using namespace std;

double getG(double x,double y)
{
    return (2*x*x*x) - (6*x*y*y) - (6*x) + (6*y) - 6;
}
double getGx(double x,double y)
{
    return (6*x*x) - (6*y*y) - 6;
}
double getGy(double x,double y)
{
    return (-12*x*y) + 6;
}
double getH(double x,double y)
{
    return (6*x*x*y) - (2*y*y*y) - (6*x) - (6*y) + 6;
}
double getHx(double x,double y)
{
    return (12*x*y) - 6;
}
double getHy(double x,double y)
{
    return (6*x*x) - (6*y*y) - 6;
}
```

```cpp
int main()
{
    double prevValX,prevValY;
    double currentValX,currentValY;
    double initialX,initialY;
    int n;
    double epsilon = 0.0001;
    bool rootFound = false;

    cout<<"Enter initial value for x: ";
    cin>>initialX;
    cout<<"Enter initial value for y: ";
    cin>>initialY;
    prevValX = initialX;
    prevValY = initialY;

    cout<<"Enter number of iterations: ";
    cin>>n;

    cout<<"EVALUATING USING NEWTONS METHOD:\n";

    for(int i=0; i<n; i++)
    {
        double x = prevValX;
        double y = prevValY;
        currentValX = x - ((getG(x,y)*getHy(x,y) - getH(x,y)*getGy(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
        currentValY = y - ((getH(x,y)*getGx(x,y) - getG(x,y)*getHx(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*getHx(x,y)));
```

```
56          currentValY = y - ((getH(x,y)*getGx(x,y) - getG(x,y)*getHx(x,y))/(getGx(x,y)*getHy(x,y) - getGy(x,y)*g
57          if(abs(currentValX - prevValX) < epsilon && abs(currentValX - prevValX) < epsilon)
58          {
59              cout<<"Root found at: ("<<currentValX<<" "<<currentValY<<")";
60              cout<<"\nNumber of iterations: "<<i+1;
61              rootFound = true;
62              break;
63          }
64          prevValX = currentValX;
65          prevValY = currentValY;
66         // cout<<prevVal<<endl;
67      }
68
69      if(!rootFound)
70          cout<<"Failed to converge!";
71
72
73
74
75      return 0;
76  }
77
```

```
C:\Users\tushar\Desktop\Semesters\Semester5\ScientificComputing\assignmen...
Enter initial value for x: 1
Enter initial value for y: 1
Enter number of iterations: 200
EVALUATING USING NEWTONS METHOD:
Root found at: (-0.0768527 0.865702)
Number of iterations: 5
Process returned 0 (0x0)   execution time : 3.516 s
Press any key to continue.
```

Root Found At: (-0.0768527,0.865702) that is -0.0768

Number Of Iterations: 5