Parm Johal

V00787710

CSC349a Assignment #4

#1.)

a)

```
function [root] = Newton(x0, eps, imax, f, fp)
%this function approximates a root of a
%function using the Newton-Raphson method
    i = 1;
    fprintf ( ' iteration approximation: \n');
    while i <= imax
        root = x0 - f(x0)/fp(x0);
        fprintf ( ' %6.0f %18.8f \n', i, root );
        if (abs(1-x0/root) < eps)
            return;
        end
        i = i + 1;
        x0 = root;
    end
    fprintf ( ' failed to converge in %g iterations\n', imax );
end
```

b)

```
function [y] = fQ1(x)
%Colebrook equation
y = 1/sqrt(x) +
2.0*log10(0.0000015/(3.7*0.005)+2.51/(13743*sqrt(x)));
end
```

```
function [y] = fpQ1(x)
%Derivative of Colebrook equation
y = -1/(2*x^1.5) + 2.0/(log(10)*0.0000015/(3.7*0.005)) * -
2.51/(13743 *x^1.5);
end
```

Parm Johal

V00787710

CSC349a Assignment #4

c)

```
>> Newton(0.008,1e-08,20,@fQ1,@fpQ1)
 iteration approximation:
    1      0.00969941
    2      0.01153771
    3      0.01346050
    4      0.01540391
    5      0.01730235
    6      0.01909668
    7      0.02074066
    8      0.02220447
    9      0.02347497
   10      0.02455331
   11      0.02545122
   12      0.02618700
   13      0.02678200
   14      0.02725800
   15      0.02763550
   16      0.02793284
   17      0.02816575
   18      0.02834743
   19      0.02848866
   20      0.02859816
```

Parm Johal

V00787710

CSC349a Assignment #4

failed to converge in 20 iterations

ans =

0.0286

d)

```
>> Newton(0.08,1e-08,20,@fQ1,@fpQ1)
 iteration approximation:
     1        0.05474651
     2        0.04507123
     3        0.03984327
     4        0.03662650
     5        0.03450185
     6        0.03303529
     7        0.03199278
     8        0.03123641
     9        0.03067956
    10        0.03026520
    11        0.02995444
    12        0.02972000
    13        0.02954235
    14        0.02940729
```

Parm Johal

V00787710

CSC349a Assignment #4

15	0.02930435
16	0.02922574
17	0.02916561
18	0.02911958
19	0.02908430
20	0.02905725

failed to converge in 20 iterations

ans =

0.0291

e) In both cases of upper and lower ends of the range seem to be too far from the true root, which means that the true zero root should be in between 0.0286 and 0.0291. Because of a slow convergence, it will take more than 20 iterations for the method to converge to the true root completely.

#2)

a)

$f(x) = x^m – R$; Using the iterative formula:

$x_{i+1} = x_i – f(x_i)/f`(x_i)$

$= x_i – (x_i^m – R)/(m*x_i^{m-1})$

$= (m*x_i^m – x_i^m + R)/(m*x_i^{m-1})$

$= ((m - 1)*x_i^m + R)/(m*x_i^{m-1})$

$=$ $\mathbf{((m-1) * x_i + R/x_i^{m-1})/m}$

Parm Johal

V00787710

CSC349a Assignment #4

b)

```
function [root] = mth_root(m, R, x0, eps, imax)
    i = 1;
    fprintf ( ' iteration approximation: \n');
    while i <= imax
        root = ((m-1)*x0 + R/x0^(m-1))/m;
        fprintf ( ' %6.0f %18.8f \n', i, root );
        if (abs(1-x0/root) < eps)
            return;
        end
        i = i + 1;
        x0 = root;
    end
    fprintf ( ' failed to converge in %g iterations\n', imax );
end
```

c)

```
>> mth_root(3,2*pi,1,10^-12,20)

 iteration approximation:
     1        2.76106177
     2        2.11543802
     3        1.87830511
     4        1.84584775
     5        1.84527033
     6        1.84527015
     7        1.84527015


ans =

   1.8453
```

Parm Johal

V00787710

CSC349a Assignment #4

#3)

a)

```
function [p] = PolyEval(n, a, y, x)
    p = a(1);
    for i=2:n+1
        horn = a(i);
        for j=1:i-1
            horn = horn*(x+y(j));
        end
        p = p + horn;
        fprintf ( ' %6.0f %18.8f \n', i-1, p );
    end
end
```

b)

>> PolyEval(4, [-1,0,2.33,-1.2,2.2] , [-1,1,-2,-2], 1.234)

```
     1       -1.00000000
     2        0.21802148
     3        0.69853880
     4        1.37334528


ans =

   1.3733
```