EM314 - NUMERICAL METHODS
ASSIGNMENT - ODE

Hisni Mohammed M.H.

E/15/131

Q1. **a) MATLAB code (Euler.m)**

```
function Y = Euler( f, x0, y0, h, b )
    X = x0:h:b;
    Y(1) = y0;
    fprintf( 'Xn\t\t\tYn\n%f\t%f\n', X(1), Y(1) );
    for i=1:length(X)-1
            Y(i+1) = Y(i) + h*f( X(i), Y(i) );
            fprintf('%f\t%f\n', X(i+1), Y(i+1) );
    end
end
```

b) MATLAB code (ImprovedEuler.m)

```
function Y = ImprovedEuler( f, x0, y0, h, b )
    maxItr = 100;
    X = x0:h:b;
    Y(1) = y0;
    fprintf( 'Xn\t\t\tYn\n%f\t%f\n', X(1), Y(1) );
    for i=1:length(X)-1
            Ydiff = f( X(i), Y(i) );
            Ystar = Y(i) + h*Ydiff;
            Ystardiff = f( X(i+1), Ystar );
            Ydiffavg = (Ystardiff + Ydiff)*.5;
            Yimp = Y(i) +  h*Ydiffavg;
            j=1;
            while( Yimp ~= Ystar && j<maxItr )
                    Ystar = Yimp;
                    Ystardiff = f( X(i+1), Ystar );
                    Ydiffavg = (Ystardiff + Ydiff)*.5;
                    Yimp = Y(i) +  h*Ydiffavg;
                    j = j+1;
            end
            Y(i+1) = Yimp;
            fprintf( '%f\t%f\n', X(i+1), Y(i+1) );
            end
end
```

c) MATLAB code (RungeKutta.m)

```
function Y = RungeKutta( f, x0, y0, h, b)
    X = x0:h:b;
    Y(1) = y0;
    fprintf( 'Xn\t\t\tYn\n%f\t%f\n', X(1), Y(1) );
    for i=1:length(X)-1
            K0 = h*f( X(i), Y(i) );
            K1 = h*f( X(i)+(h/2), Y(i)+ K0/2 );
            K2 = h*f( X(i)+(h/2), Y(i)+ K1/2 );
            K3 = h*f( X(i)+ h, Y(i)+ K2 );
            Y(i+1) = Y(i) + ( K0 + 2*K1 + 2*K2 + K3 )/6;
            fprintf('%f\t%f\n',X(i+1),Y(i+1));
    end
end
```

Q2. MATLAB code (f.m)

```
function res = f( x, y )
        A = 1;
        B = 8;
        Fi = 10;
        Fo = B*sqrt(y);
        res = (Fi - Fo)/A;
end
```

MATLAB code (Q2.m)

```
t0 = 0;
H0 = 0;
b = 1.2;

fprintf('Euler Method with step size h=.2\n');
h = .2;
Y1 = Euler(@f,t0,H0,h,b);
fprintf('\n');
fprintf('Euler Method with step size h=.1\n');
h = .1;
Y2 = Euler(@f,t0,H0,h,b);
fprintf('\n');
fprintf('Improved Euler Method with step size h=.2\n');
h=.2;
Y3 = ImprovedEuler(@f,t0,H0,h,b);
fprintf('\n');
fprintf('4th order Runge-Kutta Method with step size h=.2\n');
h=.2;
Y4 = RungeKutta(@f,t0,H0,h,b);
fprintf('\n');

X1 = 0:.2:b;
X2 = 0:.1:b;
hold on;
plot(X1,Y1,'k');
plot(X2,Y2,'c');
plot(X1,Y3,'g');
plot(X1,Y4,'b');
title('Variation of h(t) with time');
xlabel('Time - t');
ylabel('Height - h(t)');
legend('EulerMethod(h = 0.2)','EulerMethod(h = 0.1)',
        'ImprovedEulerMethod(h = 0.2)','Runge-KuttaMethod(h = 0.2)');
```

Output of the code

Euler Method with step size h=.2

Xn	Yn
0.000000	0.000000
0.200000	2.000000
0.400000	1.737258
0.600000	1.628377
0.800000	1.586651
1.000000	1.571254
1.200000	1.565659

Euler Method with step size h=.1

Xn	Yn
0.000000	0.000000
0.100000	1.000000
0.200000	1.200000
0.300000	1.323644
0.400000	1.403246
0.500000	1.455577
0.600000	1.490398
0.700000	1.513743
0.800000	1.529469
0.900000	1.540095
1.000000	1.547291
1.100000	1.552170
1.200000	1.555481

Improved Euler Method with step size h=.2

Xn	Yn
0.000000	0.000000
0.200000	1.144245
0.400000	1.356678
0.600000	1.458664
0.800000	1.509552
1.000000	1.535363
1.200000	1.548557

4th order Runge-Kutta Method with step size h=.2

Xn	Yn
0.000000	0.000000
0.200000	0.925929
0.400000	1.244274
0.600000	1.398656
0.800000	1.477060
1.000000	1.517670
1.200000	1.538905