MATLAB.6
Numerical Methods First consider the ode

y'=f(x,y)= 2*x*y^2
y(0)=0.1.

We analyze this using the Euler,Improved Euler and Runge-Kutta methods.
First we make a M-file for the right hand side.

```
*******************************************
function w=f(x,y)
w=2*x*y^2;
*******************************************
```

Next we make M-files for each of the three methods.

```
********************************************
    function [X,Y]=eu(x,y,xf,n)
h=(xf-x)/n;
X=x;Y=y;
for i=1:n
        y=y+h*f(x,y);
        x=x+h;
        X=[X;x];
        Y=[Y;y];
end
********************************************
        function [X,Y]=imeul(x,y,xf,n)
h=(xf-x)/n;
X=x;Y=y;
for i=1:n
        k1=f(x,y);
        k2=f(x+h,y+h*k1);
        y=y+h*(k1+k2)/2;
        x=x+h;
        X=[X;x];
        Y=[Y;y];
end
*********************************************
function [X,Y]=rk(x,y,xf,n)
h=(xf-x)/n;
X=x;Y=y;
for i=1:n
        k1=f(x,y);
        k2=f(x+h/2,y+h*k1/2);
        k3=f(x+h/2,y+h*k2/2);
        k4=f(x+h,y+h*k3);
        y=y+h*(k1+2*k2+2*k3+k4)/6;
        x=x+h;
        X=[X;x];
        Y=[Y;y];
end
*********************************************
```

Here (x,y) are the initial values, xf is the final x-value and n is the number of partitions.
[X,Y] is the (n+1) x 2 matrix representing the computed nodes.
To plot all three you go to the command window and enter

****

```
[z,w]=eu(0,0.1,3,20);
[s,t]=imeul(0,0.1,3,20);
[u,v]=rk(0,0.1,3,20);
plot(z,w,s,t,'--',u,v,'o')
```

****

Euler is given by a solid curve,Improved Euler by '--'s and Runge-Kutta by 'o's.

ASSIGNMENT 6 :

1. Let f(x,y) be as above.Type in the M-files for f.m and the three approximations.
Plot their graphs for n=20 and xf=3 .

The actual solution is y(x)=1/(10-x^2).
To find the distance between y(x) and the Euler approximation on the interval [0,3] for a given value of n type :

****

```
x=0:3/n:3;
y=1./(10-x.^2);
y=y';
% Here we have transposed y from a row vector to a column vector.
[z,w]=eu(0,0.1,3,n);
max(abs(y-w))
```

****

Here
abs(y-w)=[abs(y(1)-w(1)),..,abs(y(n+1)-w(n+1))]'
and max(abs(y-w)) is the maximum of the n+1 components.

The theory predicts that

(*) max(abs(y-w))<C*(3/n)

for some constant C and each n>1.

2. Set C1(n) = max(abs(y-w)) / (3/n).
Compute C1(n) for n=100,n=200,..,n=800.
Does C1(n) grow as n gets large or tend to level off? Is this consistent with (*) ?

Set C2(n)=max(abs(y-t)) / (3/n)^2 for the Improved Euler method.
Compute C2(n) for the same values of n.

What should C3(n) be for the Runge-Kutta method? Compute C3(n).