```
%Create a function Ch2Example.m and enter the following commands
%function dy= Ch2Example(x,y)
%
%The original ode is dy/dx=1/(x^2+y^2)
%
%dy = 1/(x^2 + y^2);
%end of function Ch2Example.m
function dy=Ch2Example(x,y)
dy = 1/(x.^2 + y.^2);
end

%Create a function Euler.m and enter the following commands
%function [xout,yout]=Euler(fname,xvals,y0,h)
%This code implements the Euler Method for numerically
%solving y'=f(x,y)
%
%fname=the function f for the ODE
%xvals = vector that contains the initial x0 and xf
%y0 = initial y
%h = stepsize
```

```
function [xout,yout]=Euler(fname,xvals,y0,h)
x0=xvals(1); xf=xvals(2);
x=x0; y=y0;
steps=(xf-x0)/h;
xout=zeros(steps+1, 1);% allocates space for xout
yout=zeros(steps+1, 1);% allocates space for yout
xout(1)=x; yout(1)=y;
for j=1:steps
    f=feval(fname,x,y);
    x=x+h;
    y=y+h*f;
    xout(j+1)=x;
    yout(j+1)=y;
end
end
%end of function Euler.m

%Create a function RK4.m and enter the following commands
%function [xout,yout]=RK4(fname,xvals,y0,h)
%This code implements the 4th order Runge--Kutta method for
%numerically solving the ODE y'=f(x,y)
%
%fname = the function f for the equation.
%x0 =initial x
%n = number of steps to be taken.
%y0 =initial y
%h =stepsize
function [xout,yout]=RK4(fname,xvals,y0,h)
```

```
x0=xvals(1); xf=xvals(2);
x=x0; y=y0;
steps=round((xf-x0)/h);
f=feval(fname,x,y);
xout=zeros(steps+1, 1);% allocates space for xout
yout=zeros(steps+1, length(f));% allocates space for yout
y=y'; %The ' is needed to match syntax of ode45 in higher dim
xout(1,1)=x; yout(1,:)=y;
for i=1:steps
    k1 = h*f;
    k2 = h*feval(fname,x+(h/2),y+(k1/2));
    k3 = h*feval(fname,x+(h/2),y+(k2/2));
    k4 = h*feval(fname,x+h,y+k3);
    ynext = y +(k1 + 2*k2 + 2*k3 + k4)/6;
    xnext = x+h;
    f = feval(fname,xnext,ynext);
    xout(i,:)=xnext;
    yout(i,:)=ynext'; %we again need the '
    x=xnext;
    y=ynext;
end
end
%end of function RK4.m

%Create a function Ch2NumExample.m and enter the following commands
%function dy= Ch2NumExample(x,y)
%
%The original ode is dy/dx=x*y
%
function dy= Ch2NumExample(x,y)
%dy = x*y;
dy=3*x*exp(-y)
end
%end of function Ch2NumExample.m

function dy= Ch5bNumExample(x,y)
dy=sin(x.^2);
end

function dy= Ch5dNumExample(x,y)
dy=1-y.^2;
end

function dy= Ch9NumExample(x,y)
%dy = x*y;
dy=y+x.^2;
end
```