# *First-Order ODE Solver in Matlab*

To solve a first-order ODE (ordinary differential equation) in Matlab, you must first define the ODE in a file. Suppose the ODE you wish to solve is given by the following initial-value problem:

$$y' = y^2 - x, y(0) = -1$$

Go to the File menu in Matlab and select New/M-File. Type the following:

```
% yprime.m
% Returns the value of a 1st-order ODE at point (x,y).
% The specific 1st-order ODE must be hardcoded in this function.
% This function may be called by ode45.
function yp = yprime(x,y)
   yp = y^2-x;
```

The first four lines are comments that are returned by the command "help yprime" in Matlab. The line that begins with the word "function" says that the function named ‘yprime' must be called with two parameters (or values), an x-value and a y-value, and it returns a result named ydot (the derivative defined in the function). The line that begins with "yp" says that to find the value of the first derivative, you need to square the y-value and then subtract the x-value. That result is assigned to yp and returned. Save this file under the name "yprime.m" to the ’work’ folder.

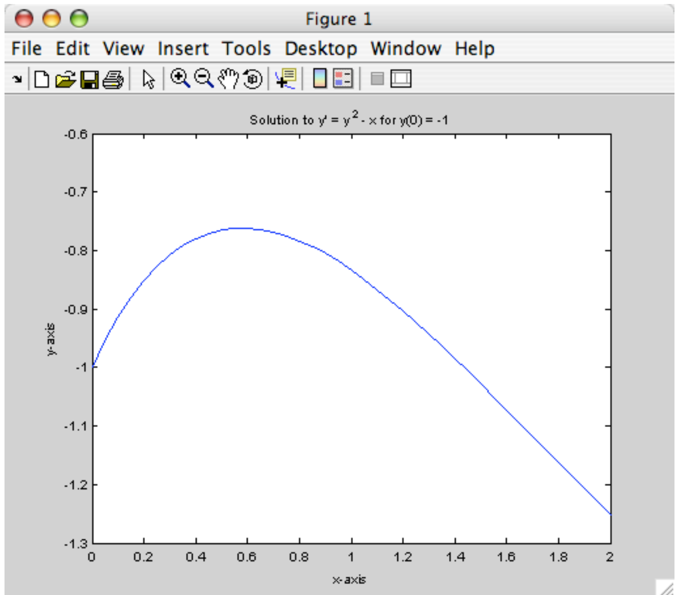
Now that the derivative function is defined, you can use the ode45 function defined in Matlab (based on the Runge-Kutta-4 method) to solve the ODE numerically. Type the following at the command prompt in Matlab:

```
> [x,y]=ode45(‘yprime’,[0,2],-1);
```

This line calculates the solution to the ODE defined in the file ‘yprime.m’ over the range from x=0 to x = 2, with the initial condition y(0)=-1, and places the calculated values in the two vectors x and y. You can graph the solution by typing:

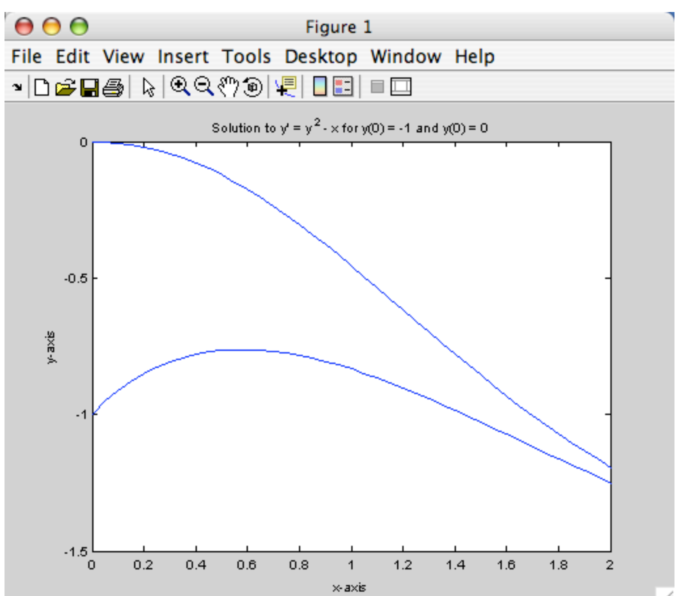
```
> plot(x,y)
> title('Solution to y''=y^2-x for y(0)=-1')
> xlabel('x-axis')
> ylabel('y-axis')
```

## *First-Order ODE Solver in Matlab*




To find the solution for the initial condition y(0) = 0 and graph it, we type:

```
> [x,y]=ode45(‘yprime’,[0,2],0);
> hold
> plot(x,y)
> title('Solutions to y''=y^2-x for y(0)=-1 and y(0)=0')
> xlabel('x-axis')
> ylabel('y-axis')
```



# *First-Order ODE Solver in Matlab*

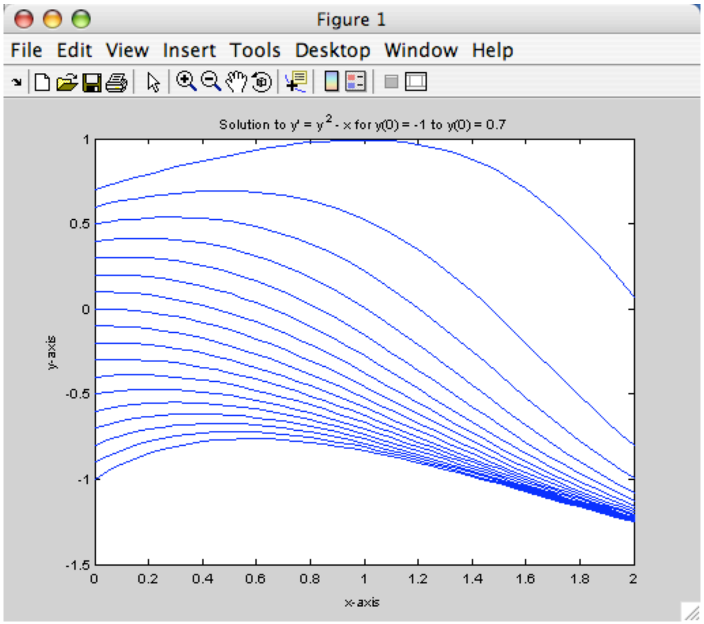
Note that both solutions are plotted on the same axes. If you want to plot several solutions for initial conditions ranging from -1.0, in steps of 0.1, to 0.7, create the following file and save it under the name "odeSolutions.m":

```
% odeSolutions.m
% Calculate and plot solutions to the ode:  y'=y^2-x
% over the interval [0,2] for initial conditions -1.0:0.1:0.7.
[x,y]=ode45('yprime',[0,2],-1);
plot(x,y)
hold on
for init=-0.9:0.1:0.7
    [x,y]=ode45('yprime',[0,2],init);
    plot(x,y)
end
title('Solutions to y''=y^2-x for y(0)=-1 to y(0)=0.7')
xlabel('x')
ylabel('y')
hold off
```

Then type into Matlab:

> odeSolutions

and get




To change the ODE to the following initial-value problem:

$$y' = (y-1)\cos(xy)$$

you need to open the file 'yprime.m' and edit the yprime function. Replace the line (in red) that begins with ydot so that the file reads:

```
% yprime.m
% Returns the value of a 1st-order ODE at point (x,y).
% The specific 1st-order ODE must be hardcoded in this function.
% This function may be called by ode45.
function ydot = yprime(x,y)
   ydot=(y-1)*cos(x*y);
```

Save the file and return to Matlab. Then, to get solutions for initial conditions y(0) = 2:-0.5:-1, modify (changes in red) your odeSolutions.m file to read:

```
% odeSolutions.m
% Calculate and plot solutions to the ode:  y'=(y-1)*cos(x*y)
% over the interval [0,2] for initial conditions 2:-0.5:-1.
[x,y]=ode45('yprime',[0,2],2);
plot(x,y)
hold on
for init=1.5:-0.5:-1
    [x,y]=ode45('yprime',[0,2],init);
    plot(x,y)
end
title('Solutions to y''=(y-1)*cos(x*y) for y(0)=2:-0.5:-1')
xlabel('x')
ylabel('y')
hold off
```