This documentation shows examples of how to use MATLAB for solving and graphing ordinary differential equations. For further information you might want to explore some of the examples provided by the MATLAB demo program or use MATLAB's help options. Type *demo*, then examine the option for *Numerical Analysis (nonlinear) Demonstrations* and the suboption for *Ordinary Differential Equations.*

MATLAB has two ordinary differential equation solvers, *ODE23.M* and *ODE45.M.* These are implementations of the Runge-Kutta-Fehlberg methods for solving vector systems of first order ordinary differential equations. (The Runge-Kutta-Fehlberg method is an explicit, variable step-size, single step numerical solver with the ODE23 standing for the error analysis done with second and third order pairs of formulae, *i.e.*, comparing second and third order Runge-Kutta methods similar to the one presented in class to determine what step-size it takes for a given accuracy.) The ODE45 program uses higher order approximations to solve the differential equation, so it often takes large step-sizes. This makes for poorer graphical displays. Thus, I would highly recommend using the ODE23 program for most problems.

To use ODE23, the user must provide a function, say *func*, then type the following:

```
[x,y] = ode23('func',x0,xf,y0);
```

where the x0 is the initial x value, xf is the final x value, and y0 is the initial y value. (The semicolon prevents having the solution typed to the screen.) To plot the solution one only needs to invoke the MATLAB plotting routine, which is easily done by typing the following:

```
plot(x,y)
```

The MATLAB demo provides more information on how to use MATLAB's plotting function.

Example of a First Order Differential Equation.

Consider the following differential equation:

$$y' = x^2 - y^2.$$

We have developed no techniques for solving this differential equation explicitly, so numerical solutions provide a good option for analyzing the result. To show how to solve and graph the solution, let us consider three different initial conditions and then graph each of these solutions on a single graph using MATLAB. For this example we shall examine the initial conditions $y(0) = -0.5, 0, 1$, and study the solution for $x \in [0, 2]$. (Note: If you attempt to solve this equation with $y(0) \leq -0.7$, then $y(x) \to -\infty$ for $x < 2$.)

We begin by creating an M-file using the editor inside MATLAB. The file has the following form:

```
function yprime = ex1(x,y);
yprime = x^2 - y^2;
```

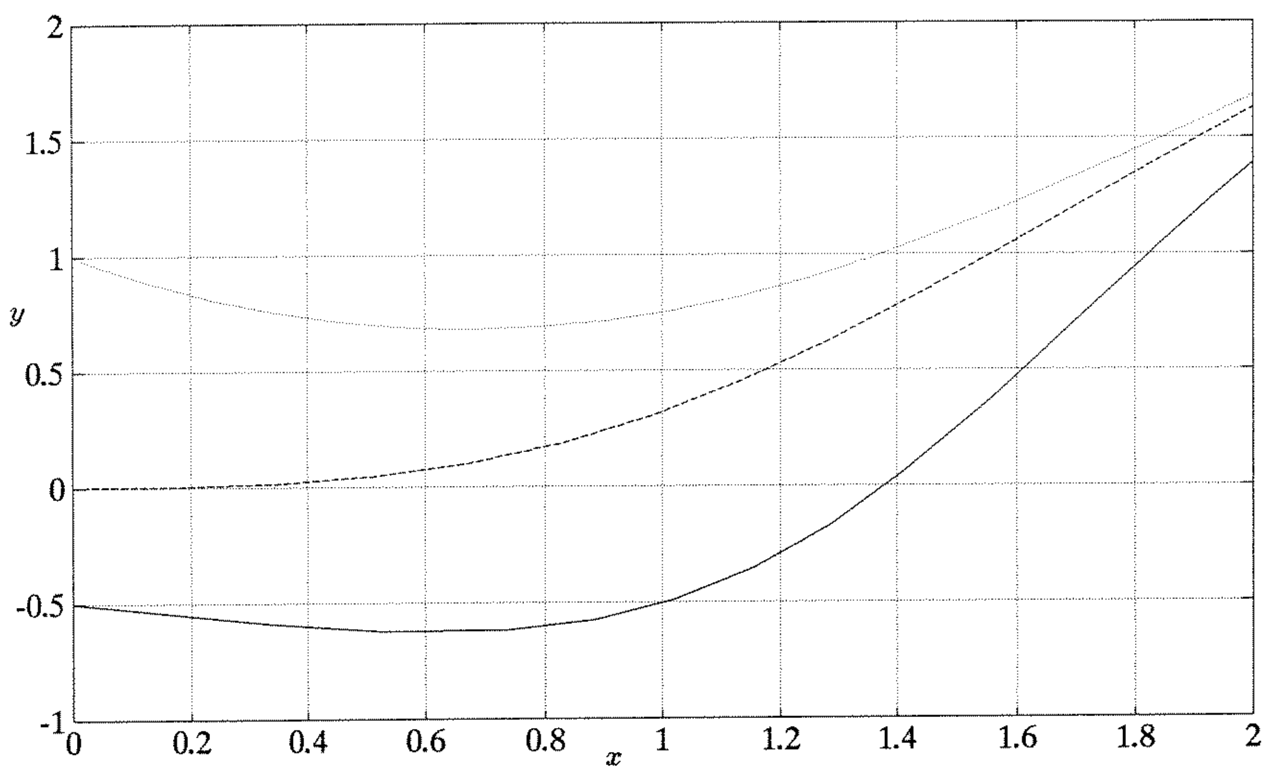
After saving this file as an M-file, we can now use MATLAB to solve our problem. Type the following three lines to solve our three initial value problems:

```
[x1,y1] = ode23('ex1',0,2,-1);
[x2,y2] = ode23('ex1',0,2,0);
[x3,y3] = ode23('ex1',0,2,1);
```

MATLAB will have produced vector pairs [xi, yi], i = 1,2,3, for the three solutions.

To see the graphs of these solutions as shown below simply type the following:

```
plot(x1,y1,x2,y2,x3,y3);grid
```



Example of a Second Order Differential Equation – The Pendulum with Damping.

It can be shown that the following second order differential equation models a damped pendulum:

$$y'' + 0.1y' + \sin(y) = 0.$$

This is another example of a problem that we are unable to solve analytically. In order to study this problem numerically, we first create a system of first order differential equations. Let $y_1 = y$ and $y_2 = y_1'$. With these variables, $y_1(t)$ represents the angular displacement, while $y_2(t)$ represents the angular velocity. As shown in class, the above differential equation can be written:

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= -\sin(y_1) - 0.1y_2. \end{aligned}$$

To study this with MATLAB we first create the following M-file:

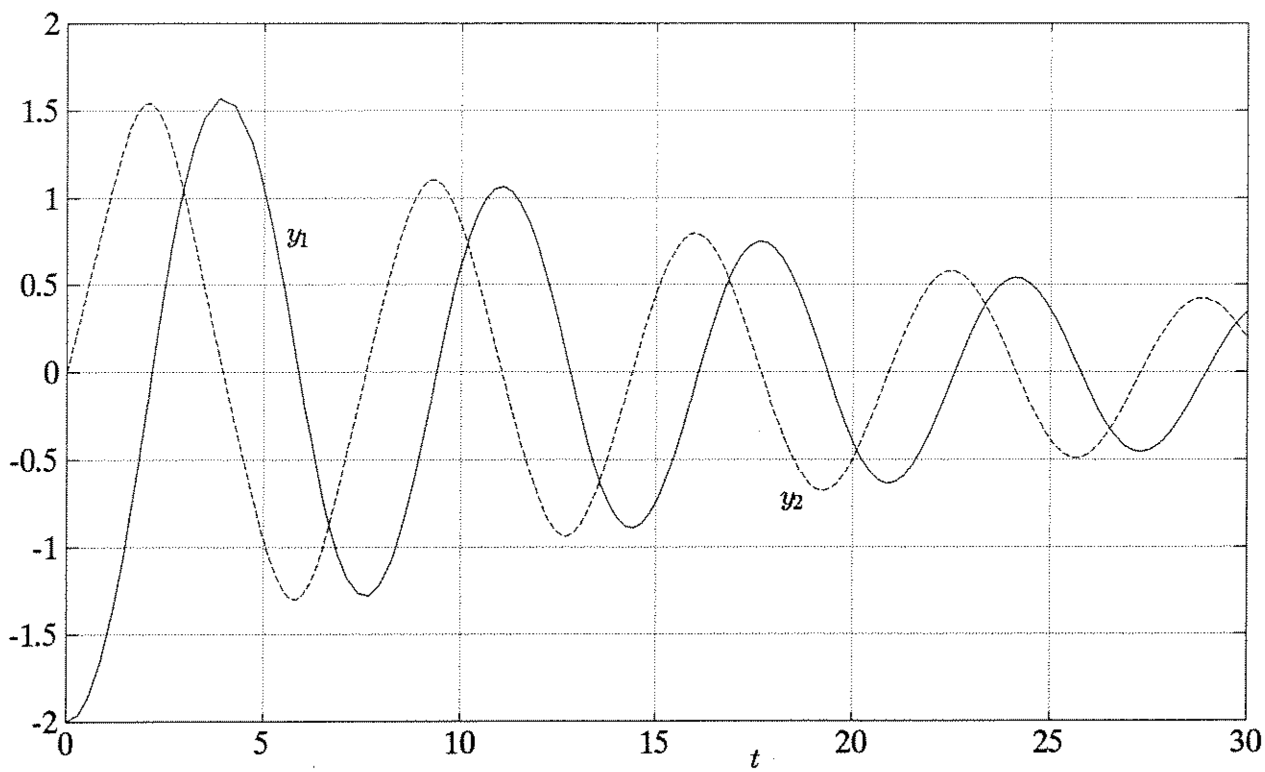
```
function yprime = ex2(t,y);
yprime = [y(2);(-sin(y(1)) - 0.1*y(2))];
```

Notice that we are creating a 2×1 matrix representing the right hand side of the system of equations above.

Now suppose we want to study the following two cases. First, assume that the initial displacement is -2 with no initial velocity. Second, assume that the initial displacement is zero and the initial velocity is 3. To study these cases for $t \in [0, 30]$ with MATLAB we type the following:

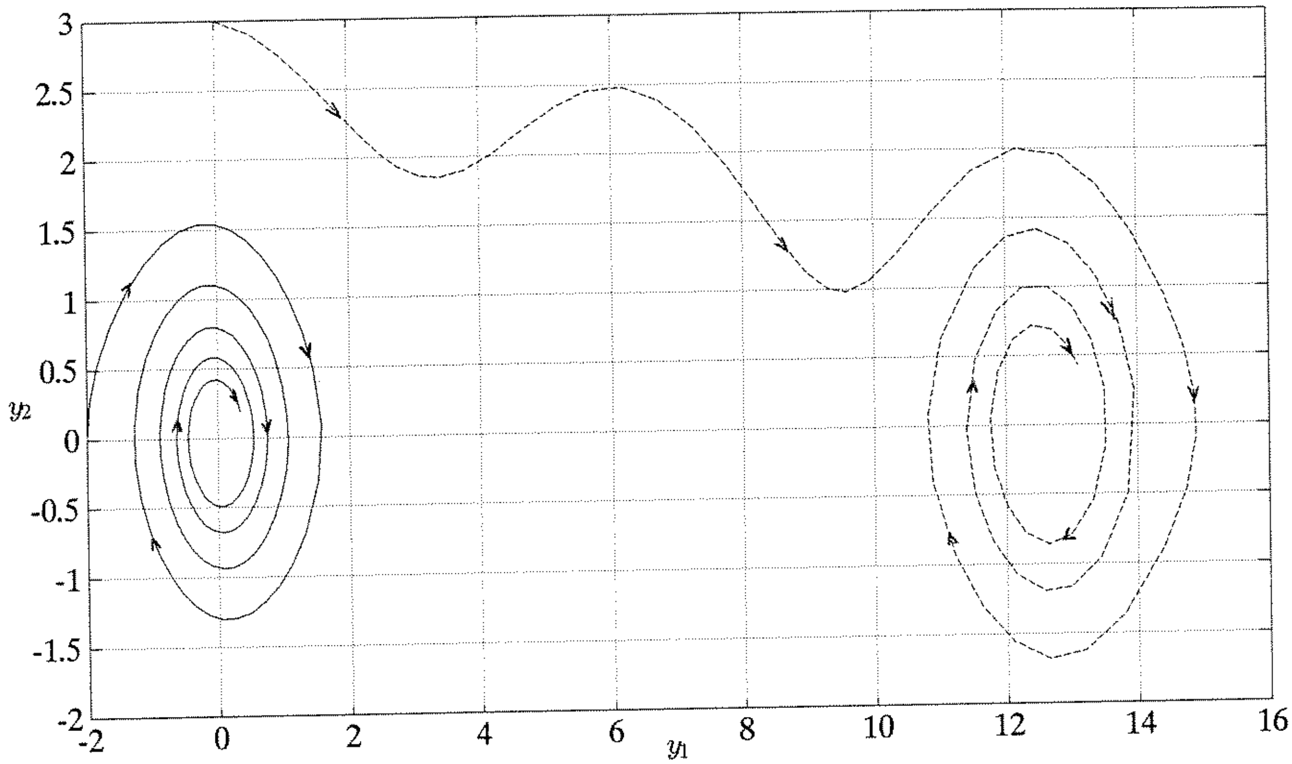
```
[tA,yA] = ode23('ex2',0,30,[-2,0]);
[tB,yB] = ode23('ex2',0,30,[0,3]);
```

The simplest way to view the solution is to have MATLAB plot the solution as a function of time. By typing *plot(tA,yA)* a graph is produced showing both y_1 (displacement) and y_2 (angular velocity) as a function of time for the first solution computed above by the ODE23 routine. This is shown below.



A better way to view the behavior of this system is to draw what are known as phase portraits. In this case the solutions are plotted in the y_1 vs y_2 plane. (Arrows that I've drawn on the plots show the direction of the solution as time increases.) This diagram can be plotted using MATLAB by typing the command:

```
plot(yA(:,1),yA(:,2),yB(:,1),yB(:,2));grid
```



Here the plot routine draws the phase portraits for both of the different initial conditions given above on a single graph.