Numerical Solution to Ordinary Differential Equations

We frequently need to solve a differential equation, ie an equation of the form $\frac{dy}{dx} = f(x, y);\ a \le x \le b;\ y(a) = \alpha$. In other words, we are not given the function y but its derivative. This derivative may be only a function of x, but it could be a function of x and y. We are also usually given the value of y at one end of the domain, ie it is what is called an *initial value problem*. Fortunately Matlab provides some extremely powerful routines to solve these problems.

Matlab has several routines, with syntax as below. Note that Matlab assumes the independent variable is t, or time, not x, but of course the syntax is unchanged. The ODE should be checked for stiffness first, and if it is stiff, then another routine should be used.

```
[TOUT,YOUT] = ode23(ODEFUN,TSPAN,Y0)
[TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0)
```

In other words, either of these routines could be used, but generally `ode45` will provide better accuracy. Generally, this should be the first solver you try. There are a number of other solvers that could also be discussed, but we will concentrate on `ode45`. The meaning of the parameters in the above call is as shown in the following table.

Parameter	Discussion
`TOUT`	A vector of time values returned by the solver..
`YOUT`	The y values calculated at each time point in the vector `TOUT`.
`ODEFUN`	The name of the Matlab function file used to calculate the derivative of y at each time step.
`TSPAN`	The initial and final times, ie the limits of the independent variable. If the value of the function is needed at specific time values, then `TSPAN` should contain all these points.
`Y0`	The initial conditions.

Note that y may be a vector, meaning that we may in fact be solving a system of ODE's. In other words, `YOUT`, `Y0` and the output of `ODEFUN` must also be vectors. Our first example will be a single ODE.

Example 1

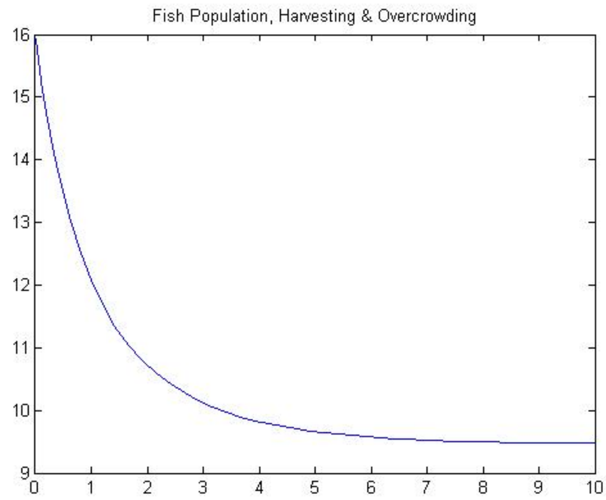
Solve the ODE $\frac{dy}{dt} = y - \frac{y^2}{12} - 2,\ y(0) = 16$. This arises in the modelling of a fish population with overcrowding and harvesting (Borelli, Coleman). Then first of all we need a function M file that returns the derivative $y' = \frac{dy}{dt}$. Enter the following code into Matlab and save it with the filename `Examp1.M`.

```
function deriv=Examp1(t,y);
% Note that t is not explicitly used in this ODE,
% but it is still a required parameter.

deriv=y-y.^2/12-2;
```

Now let us suppose we need to find the population over the period t=0 to 10. Then we shall first solve the problem without specifying the output values of t, and then with time values spaced one month apart. Enter the code below and run it. The figure below results.

```
% Program to call the example Examp1
tspan=[0 10];
y0=16;
[TOUT,YOUT] = ode45('Examp1',tspan,y0);
plot(TOUT, YOUT)
title('Fish Population, Harvesting & Overcrowding')
```



Now let us suppose we need to solve a system of ODEs, perhaps the following, also from Borrelli and Coleman. It is known as the Lotka-Volterra system, a predator-prey system of ODEs that attempts to model the population of a single prey species and a single predator, with harvesting. The prey species is y, the predator species is x. (In the absence of prey, the predator decreases, shown by the term $-x$, and in the absence of the predator, the prey increases, shown by the term y.)

$$\frac{dx}{dt} = -x + \frac{xy}{10} - 0.2x,\ x(0) = 8$$
[1]
$$\frac{dy}{dt} = y - \frac{xy}{5} - 0.2y,\ y(0) = 16$$

[1] For information on the coefficients and why they take the signs they do, consult Borrelli and Coleman, Differential Equations A Modeling Perspective

Again note that there is no explicit dependence on time in this system. We need to redefine x and y to y_1 and y_2. Then the system becomes as follows.

$$\frac{dy_1}{dt} = -y_1 + \frac{y_1 y_2}{10} - 0.2y_1,\ y_1(0) = 8$$

$$\frac{dy_2}{dt} = y_2 - \frac{y_1 y_2}{5} - 0.2y_2,\ y_2(0) = 16$$

The Matlab code to calculate this is as follows.

```
function derivs=Examp2(t,y)
% Matlab code for example 2

y1=-y(1)+y(1)*y(2)/10-0.2*y(1);
y2=y(2)-y(1)*y(2)/5-0.2*y(2);

derivs = [y1; y2];
```

Output is a column vector.

As usual, although t is not explicitly required in the file it is a necessary parameter. To solve this problem with `ODE45`, try the following call.

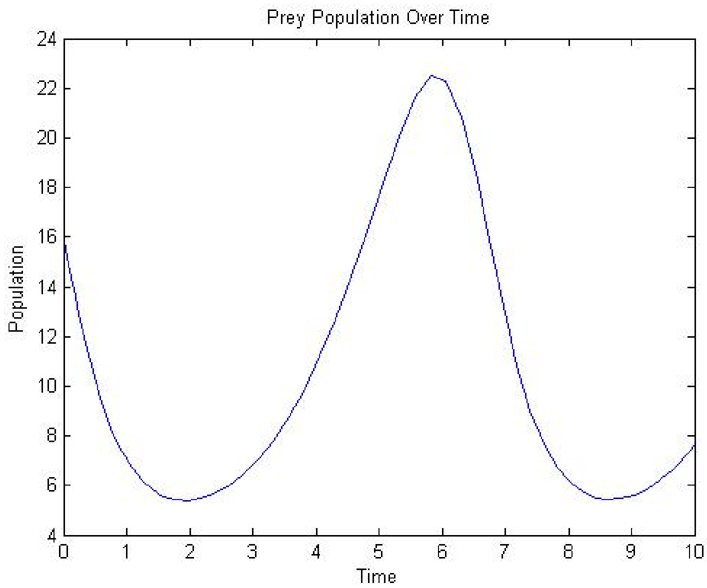
```
% Program to call the example Examp1
tspan=[0 10];
y0=[8;16];;
[TOUT,YOUT] = ode45('Examp2',tspan,y0);

figure(1)
plot(YOUT(:,1), YOUT(:,2))
title('Predator-Prey interaction');
xlabel('Predator Population');
ylabel('Prey Population');

figure(2)
plot(TOUT,YOUT(:,1));
title('Predator Population Over Time');
xlabel('Time');
ylabel('Population');

figure(3)
plot(TOUT,YOUT(:,2));
title('Prey Population Over Time');
xlabel('Time');
ylabel('Population');
```

This code generates three graphs, the first being the predator and prey populations being plotted on the same axes, and the second two being the individual populations plotted against time. One of them is shown below.



The next section, which will be added when time permits, will show a simple example with *t* explicitly appearing on the right hand side, and more complicated examples that do not have a simple formula for the derivatives.