Exercise 10: Simple Harmonic Motion and Pendulums
solving differential equations

Many of the equations we meet in physics involve derivatives and hence are differential equations. An important example is Newton's second law which is a second order differential equation since it involves acceleration (the second time derivative of displacement. MATLAB is equipped with several routines to solve differential equations.

The damped Driven Pendulum:

The angular displacement θ(t) in such a pendulum satisfies the second order differential equation:

$$\frac{d^2}{dt^2}\theta(t) + 2k\,\frac{d}{dt}\theta(t) + \sin(\theta(t)) = f\,\cos(\omega\,t) \qquad \text{Eq. 1}$$

where k is called the damping ratio, f is the amplitude and ω the angular frequency of the forcing term. For simplicity we have set g/l=1 in the equation above, where g is the gravitational acceleration and l the length of the pendulum.

For small amplitude motion we can replace sin(θ) by θ to obtain the equation for damped forced simple harmonic motion:

$$\frac{d^2}{dt^2}\theta(t) + 2k\,\frac{d}{dt}\theta(t) + \theta(t) = f\,\cos(\omega\,t).$$

In MATLAB we can solve such an equations by using the `ode45` routine, which is invoked by the command `ode45(@function,t,u0)`, where `function` defines the right side of the differential equation you would like to solve (see example below), t a vector specifying the interval of integration, and u0 a vector of initial conditions.

Many more differential equation solvers are available in MATLAB, and you can find a description of these in the User's Guide → Mathematics → Calculus. Most of the information you need to know to solve this exercise is however presented in the example below.

As an example, we solve the case of undamped unforced simple harmonic motion.

$$\frac{d^2}{dt^2}\theta(t) + \theta(t) = 0$$

with the initial conditions θ(0)=0.1, θ'(0)=0.

MATLAB Ordinary Differential Equation (ODE) solvers accept only first-order differential equations. To use the solvers with higher-order ODEs, you must rewrite each equation as an equivalent system of first-order differential equations, in this case:

$$\frac{d}{dt}\theta(t) = \theta'(t)$$

$$\frac{d}{dt}\theta'(t) = -\theta(t)$$

Then we need to define a function handle to describe the system of two first-order differential equations in θ and θ'. We do this in a separate function file, which we call e.g. UnUnfSH.m:

```
% Function handle for ODE of an unforced, undamped, simple
% harmonic oscillator

function du = UnUnfSH(t,u)

    theta = u(1);
    theta_prime=u(2);

    du = zeros(2,1);

    % first equation in theta
    du(1) = theta_prime;

    % second equation
    du(2) = - theta;
end
```

Then we define a vector of initial conditions `u0=[0.1 0]`, and a vector specifying the interval of integration `t = 0:0.1:100;`

Now all we need to do is run the `ode45` command, which will give as output the solution of the differential equation in the specified t domain (i.e. θ(t) and θ'(t)).

`[T, SOL] = ode45(@UnUnfSH, t, u0);`

We can then plot θ(t) by selecting the first column in the solution array using e.g.
`plot(T,SOL(:,1))`, and obtain the following figure:
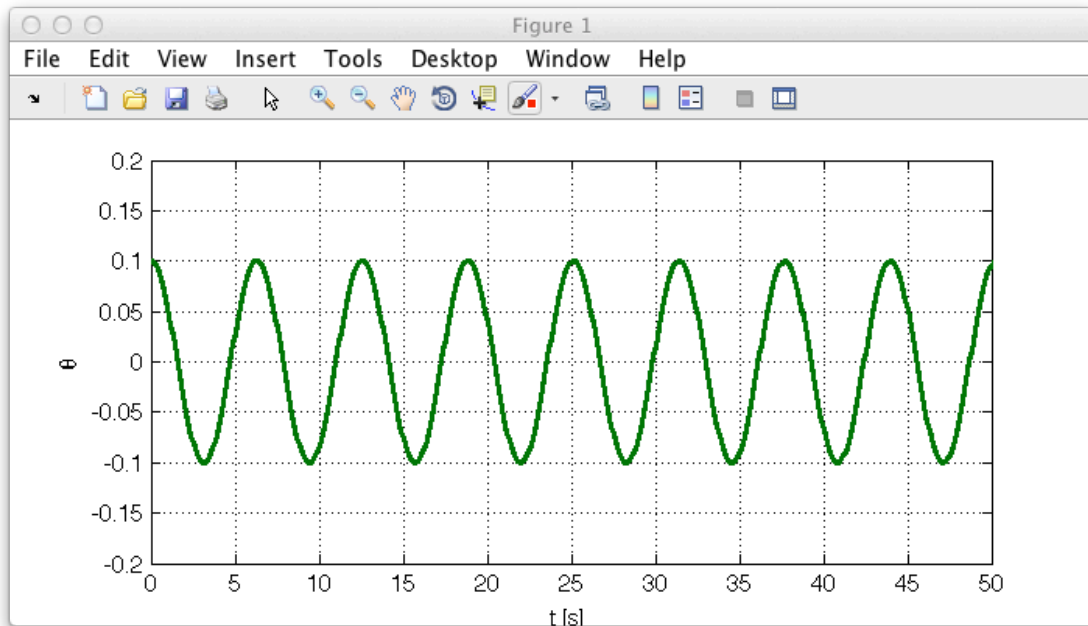
Try changing initial conditions and domain of integration to familiarise with the command.

You should now use the `ode45` command to investigate **damped driven simple harmonic** motion in the following situations:

A. damping but no forcing term, show both under-, over- and critical damping (k < 1, k > 1, k=1)

B. forcing in each damping case away from resonance (choose f=0.1 and ω=1.2).

C. the damped forced **pendulum** (Eq. 1) with k=0.05, f=1.5, and ω=1.1 with initial conditions θ(0)=0.1, θ'(0)=0 and θ(0)=0.101, θ'(0)=0, both plotted on the same graph. Note the effect on the solution of a small change in the initial conditions. This extreme sensitivity to initial conditions is the signature of chaos.

When the exercise is complete save your work and show it to a demonstrator.