Lesson 10: Mass-Spring, Resonance and ode45

10.1 Applied Problem.

Trucks and cars have springs and shock absorbers to make a comfortable and safe ride. Without good shock absorbers, the truck or car will continue to bounce upon hitting an object. In fact, if a series of objects are hit and the shock absorbers are not very good, then the amplitude of the bounce could even increase!

The objective of this lesson is to study a second order differential equation, which is a simple model of this important topic. This will be converted into a system of first order differential equations so that the Matlab command ode45 can be used to study the solutions for different choices of spring and damping constants.

10.2 Differential Equation Model.

A coupled mass-spring system with damping and external force is modeled by a single second order differential equation for the displacement from the equilibrium position, \( y(t) \),

\[
m y'' + c y' + k y = f(t) \text{ and } y(0) \text{ and } y'(0) \text{ are given.}
\]

The velocity is defined to be \( y' \). This equation is derived from Newton's law of motion, which requires the sum of the forces to be equal to the mass times the acceleration. The forces are \( -c y' \) for damping, \( -k y \) for small deformations of the spring, and \( f(t) \) for external forces. In our problem \( f(t) \) could be a complicated expression of trig functions, and the exact solution will be difficult to find. Therefore, we are forced into considering higher order numerical methods with variable step sizes.
The above second order differential equation can be equivalently written as a coupled system of two differential equations for $y_1(t) = y(t)$ and $y_2(t) = y'(t)$. By definition the derivative of $y_1$ must be $y_2$. The derivative of $y_2$ is $y''$, which we can solve for via the second order differential equation. Thus, the equivalent coupled system is

\begin{align*}
y_1' &= y_2 \quad \text{with } y_1(0) = y(0) = \text{the initial position and} \\
y_2' &= (f(t) - c y_2 - k y_1)/m \quad \text{with } y_2(0) = y'(0) = \text{the initial velocity.}
\end{align*}

Matlab's ode45 can be used to solve such systems.

### 10.3 Method of Solution.

Consider the special case where there is no damping, $c = 0$, the external force is a trig function, $f(t) = F_0 \cos(\omega t)$, and the initial position and velocity are both zero. The reader should verify that the solution is

\[ y(t) = \frac{F_0}{(k - m \omega^2)} (-\cos ((k/m)^{1/2}t) + \cos(\omega t)). \]

If the frequency of the forcing term, $\omega$, approaches the natural frequency of the coupled spring mass, $(k/m)^{1/2}$, then we can expect some resonance. In order to be more precise, note $y(t)$ as a function of $\omega$ is of the form $\infty/\infty$ as $\omega$ approaches $(k/m)^{1/2}$. By using L'Hopital's rule we can show $y(t)$ approaches, as $\omega$ approaches $(k/m)^{1/2}$,

\[ Y(t) = \frac{F_0}{(2(km)^{1/2})} t \sin((k/m)^{1/2} t). \]

This is also the solution of the spring-mass equation with $f(t) = F_0 \cos((k/m)^{1/2} t)$. Thus the amplitude must increase until the system "breaks" or the model is no longer valid.

We want to be able to approximate the solution for more general cases. What happens if there is a damping term, $-c y$, where $c$ is small? Or, what happens for more complicated forcing terms?
10.4 Matlab Implementation.

The single second order differential equation must be converted, as above, to a system of differential equations for the position, denoted by the symbol $y(1)$, and for the velocity, denoted by the symbol $y(2)$. The m-files for this system are called `ypms.m` and `ms.m`. In the calculation we used $m = 1$, $c = 0$, $k = 1$, $f(t) = 1 \cos(1 \cdot t)$ with initial position and velocity set equal to 0. In subsequent calculations we let $c = .1$ and 1. where the third line in the `ypms.m` must be changed to, for $c = .1$,

$$ypms(2) = \cos(1 \cdot t) - y(1) - .1 \cdot y(2);$$

Note the amplitude is bounded for small $c$, but the amplitude increases as $c$ get smaller. The amplitude of the resonating solution is $F_0 / (2(km)^{1/2}) \cdot t = 1/2 \cdot t$.

```matlab
function ypms = ypms(t,y)
    ypms(1) = y(2);
    ypms(2) = cos(1*t) - y(1);
    ypms = [ypms(1) ypms(2)]';

%your name, your student number, lesson number
clear;
t0 = 0;
tf = 100;
y0 = [0 0];
[t y] = ode45('ypms', [t0 tf],y0);
plot(t,y(:,1))
title('your name, your student number, lesson number')
xlabel('time')
ylabel('displacement')
%plot(y(:,1),y(:,2));
```
Figure: \( m = 1, c = 0.0 \) and \( k = 1 \)

Figure: \( m = 1, c = 0.1 \) and \( k = 1 \)
Another set of interesting experiments is to keep $m = 1$, $c = 0.0$, $k = 1.0$ and to vary $\omega$ in the external forcing term. In calculations below we have used $\omega = .9$, 1.1 and 1.2. Note how the frequencies and amplitudes have changed.

Figure: $m = 1$, $c = 1.$ and $k = 1$
Figure: $m = 1, c = 0., k = 1$ and $\omega = 0.9$

Figure: $m = 1, c = 0., k = 1$ and $\omega = 1.1$
Figure: $m = 1, c = 0., k = 1$ and $\omega = 1.2$

10.6 Additional Calculations.

Consider the mass-spring system with damping where $f(t) = 2\cos(t)$, $m = 1.0$, $c = 0.1$ and variable $k$.

(a). State the second order differential equation and the equivalent system of differential equations.

(b). Modify the ypms.m and ms.m files.

(c). Execute the ms.m file using $k = 1.0$.

(d). Repeat (c) with $k = 1.0 - .S$ and $k = 1.0 + .S$.

(e). Compare the three solutions. What happens as the $k$ varies from 1.0?