This page is designed to help primarily with MatLab techniques for answering the questions in Computer Activity 4. This activity is designed to learn a few new features in MatLab that are valuable for ODEs and graphic displays. Specifically, this activity introduces 3D display of simulations, and you should use the *subplot* command to display your 6 related graphs in Part b.

a. This Computer Activity centers around the 2^{nd} order ODE, which is very similar to the damped mass spring problem with a sinusoidal forcing function:

$$\frac{d^2y}{dt^2} + 2c\frac{dy}{dt} + k^2y = F_0\sin(at), \qquad y(0) = 0 \quad \text{and} \quad \dot{y}(0) = 0. \tag{1}$$

The solution of this problem is readily done with the *Method of Undetermined Coefficients*, and a closely related problem is solved on Slides 30-32, including the limiting behavior information.

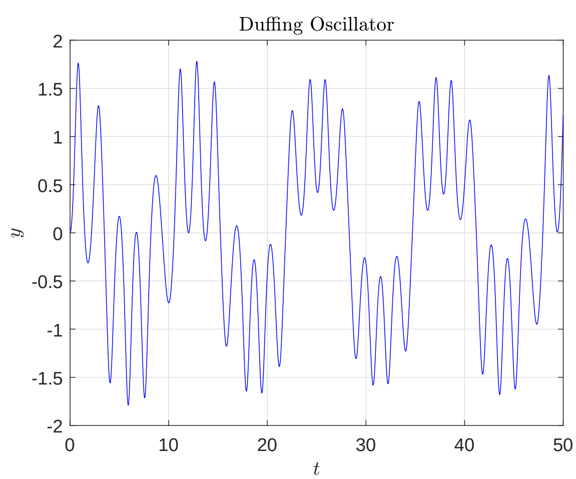
b. Coding the function from Part a should be easy to obtain solutions at any time, t. However, the numerical solution using MatLab's *ode45*, requires first transforming the 2^{nd} order ODE into a system of 1^{st} order ODEs. Below we demonstrate this with a very important classical model, Duffing's equation, which is widely studied for chaotic behavior. It is a forced damped mass spring problem with the spring have a nonlinear stiffness. (You are welcome to explore solutions of this system.) Duffing's equation satisfies:

$$\frac{d^2y}{dt^2} + \delta\frac{dy}{dt} + \alpha y + \beta y^3 = \gamma\cos(\omega t).$$

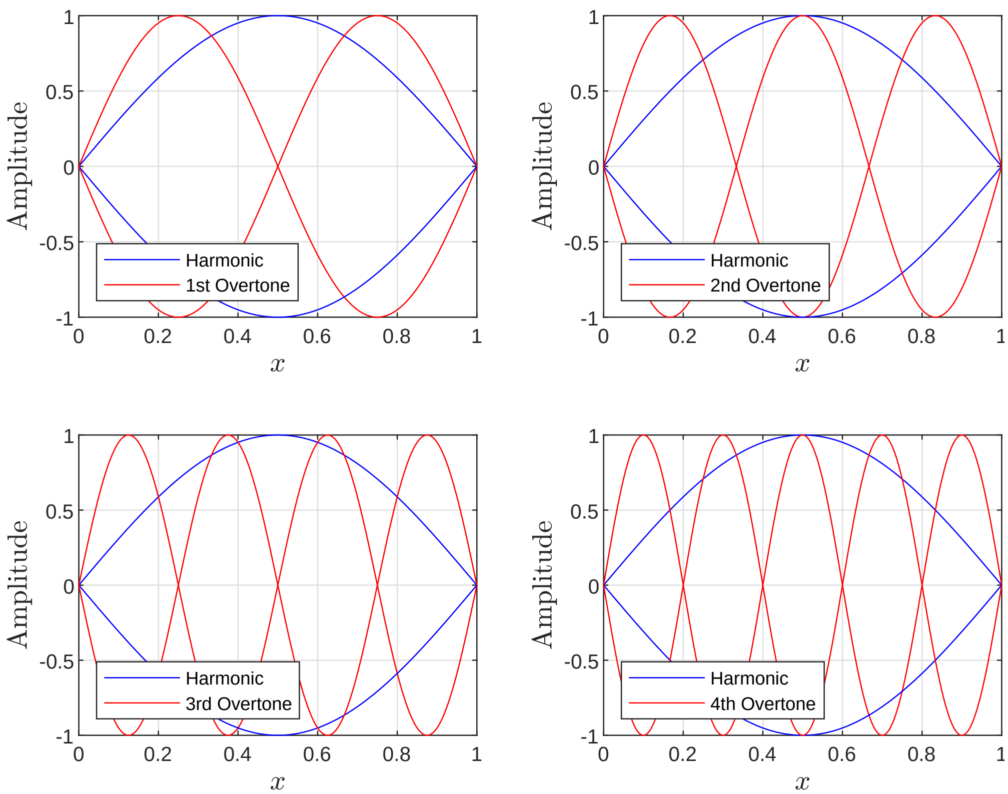
Let $y = x_1$, $\dot{x}_1 = x_2$, so $y'' = \ddot{x}_1 = \dot{x}_2$. It readily follows that:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\alpha x_1 - \beta x_1^3 - \delta x_2 + \gamma\cos(\omega t) \end{pmatrix}.$$

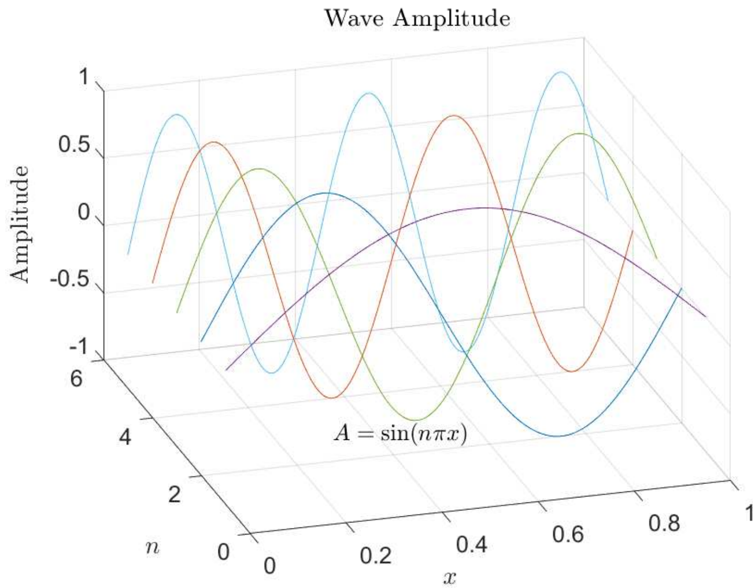
The MatLab fnction for this ODE is available through the hyperlink: duffing.m. This 1^{st} order system of ODEs is numerically solved using MatLab's *ode45*, which is found in the hyperlink to the MatLab script: duffing_plot.m. The result of running the script is the following graph.



The line in the script stating:
t1 = [0:0.002:50];
creates a vector t_1 with elements uniformly spaced by 0.002 from 0 to 50 (or 25001 elements). To obtain the element for $t_1 = 30$, we examine the element 15001, so *t1(15001)* $= 30$ and we find that *y(15001,1)* $= -0.3118002497$, which is visible in the graph. If we want to find the maximum of $y(t)$ for $t \in [0.50]$ from the *ode45 solver*, then we issue the command:
[M,I]=max(y(:,1))
which gives the maximum response of $M = 1.7805791$ and the row index $I = 6426$, where MatLab finds the maximum. We find that $t(6426) = 12.85$ and $y(6426, 1) = 1.7805791$, which does appear to be the maximum in the graph above.
Part b in this Computer Activity has you producing **6** graphs for the test sound waves $a = 2.85$, 3.0, 3.15, 3.3, 3.45, and 3.6 with two graphs on each subplot for $k^2 = 9$ and $k^2 = 12$. It is appropriate to put all these subplots as a single entity, since they are testing very similar phenomenon. Part c has you producing a 3D plot to view maximal response of different hair cells. To demonstrate these graphing techniques in MatLab we examine the displacement of a string fixed at each end (think guitar string). The string has a tone representative of its primary harmonic, but in addition there are overtones that provide a richness to the sound. The displacement of the string is described by $y(x) = \sin(n\pi x)$ for different values of n and $x \in [0, 1]$. The graph below illustrates the amplitude of the string for the primary harmonic and compares it to each of its first four overtones.



A 3D graph showing the waveform of the primary harmonic and its first four overtones is shown below.



The MatLab program to produce these graphs is available through the hyperlink: harmonic_plot.m. This program should provide most of the commands you need for the graphs in this Computer Activity.