21-124 MODELING WITH DIFFERENTIAL EQUATIONS

LECTURE 6: SECOND ORDER EQUATIONS AND SYSTEMS
EXPERIMENTS AND DISCUSSION

This week we will be using `pplane6` and MATLAB's numerical solvers to study a mass-spring-dashpot system. The values we shall use are: $m = 3$ kg, $k = 36$ N/m and $c = 9$ Ns/m. The resulting second order equation, in the case of free motion, is

$$y'' + 3y' + 12y = 0.$$

1. Convert this second order equation to a first order system. Enter the system into `pplane6` and view some solutions to the system.
   (a) Use the "keyboard input" option from the "solutions" menu to plot the solution with initial condition $y(0) = 5$, $y'(0) = v(0) = 0$. You can use the options under the "Graph" menu to look at other presentations of the solution. Take a look at the "y vs. t" graph and the "3D" graph.
   (b) Now we will look at the numerical solution generated by the routine `rk4`. First, you must write a function M-file for the system. I have named my M-file `osc.m`, and I will assume you have done the same.
   (c) Use `rk4` to compute the solution with initial conditions $[y, v] = [5, 0]$ on the interval $[0, 4]$. Use the `plot` command to display a graph of y as a function of t ("y vs. t"). Then create a graph of the solution curve in the phase plane ("v vs. y").
   (d) You can also create three dimensional plots using MATLAB. To do so, you need the command `plot3`. This works just like the `plot` command, but it accepts three vectors as arguments. Make a 3D graph of your solution. In what order must the arguments be to get the same orientation as the 3D graph produced by `dfield6`?

You may have noticed that if you try to plot a solution while you have the ***pplane6*** *windows open, the graph will show up in one of those windows. You can prevent this using the* ***figure*** *command.* MATLAB *numbers its figure windows. The untitled windows show their figure number in the title bar. Entering* ***figure(1)*** *will bring that window to the front, and the next plot command will be sent to that window. To create a new window, choose a number that has not yet been used, say 12, and enter* ***figure(12)****.*

2. We will now apply an external force equal to $10\sin(7t)$ to our system. Since the system is no longer autonomous (i.e. time-independent), `pplane6` may no longer be used.
   (a) Modify your file `osc.m` to include an appropriate forcing term.
   (b) Again, display the solution with initial conditions $y(0) = 5$, $y'(0) = v(0) = 0$. Show the same three views you looked at before, "y vs. t", "v vs. y" and "3D". What differences do you notice?

3. Now we will remove the dashpot, and hence the forcing term, from our model. We will examine the effect that different forcing frequencies have on our equation.
   (a) Find the maximum amplitude (over the interval $[0, 5]$) of the solution with initial condition $[y, v] = [0, 0]$.
   (b) For each forcing frequency, ω, there is a corresponding maximum amplitude, M_ω. We will make a script file that will plot M_ω versus ω. First, modify your function M-file so it reads as follows:

```
function Yprime=osc(t,Y)
global A;
Yprime(1)=Y(2);
Yprime(2)=-12*Y(1)+10*sin(A*t);
```

   The important changes are the line `global A;`, and the change in the forcing term. Normally, the variables in a function M-file are *local* in that they are unaffected by any commands you enter in the MATLAB window. The line `global A;` creates a variable that can be changed in the MATLAB window.
   In the MATLAB window you must also enter `global A`. (It must be initialized in both places.) Then you can conveniently change the forcing frequency.
   (c) Now create a script M-file with the following commands:

```
omega_vect=[];amp_vect=[];
for A=.5:.2:5
[t,y]=rk4('osc',[0,5],[0,0],.1);
M=max(abs(y(:,1)));
omega_vect=[omega_vect,A];
amp_vect=[amp_vect,M];
end
omega_vect
amp_vect
```

We have used a `for`-loop here. The structure for a `for`-loop is `for i=v;statement_1;...;statement_n;end`. The variable `i` takes the value of each element of the vector `v`, one at a time.

(d) Use `plot(omega_vect,amp_vect)` to create a graph of the maximum amplitude as a function of the forcing frequency (a frequency response graph).
What do you notice? Where does this curve reach it's maximum?

4. Let's repeat this with a different forcing function. Download the M-file `force.m` from the 21-124 website.
   (a) Create a graph of the function `force`. Let `s=0:.1:18;` and `z=force(t)`, then `plot(s,z)`.
   (b) In the file `osc.m` change the forcing function from `10*sin(A*t)` to `10*force(A*t)`. Run the script file again to create a frequency response graph.
   What happens?