1. **MATLAB**

MATLAB commands we use in this lab include the following.

1.1. **disp.** Displays text to the command window. For example,
   ```matlab
   >> disp('This is text sent to the command window')
   ```
   See tic and toc, below.

1.2. **eulermethod_w19.** This isn’t a native MATLAB command; download it from the labs page. This command approximates the solution to a differential equation or system using Euler’s method. It takes as arguments the same arguments as we use with ode45\(^1\), plus a step size to use:
   ```matlab
   >> eulermethod_w19(f_handle, [tmin tmax], init_cond, h);
   ```

1.3. **ode45.** Finds a numerical approximation to a differential equation or system of equations:
   ```matlab
   >> [tsol,xsol] = ode45(f_handle, [tmin tmax], init_cond);
   ```
   Note that we can also specify, instead of just the minimum and maximum \( t \) values, a set of points at which we want to have the solution evaluated. This doesn’t change the numerical calculation; it just means we know what times our \( xsol \) vector will be evaluated at. For example, to ensure that the solution was available at \( t = 0 \), \( t = 0.1 \), \( t = 0.2 \), etc., we could use
   ```matlab
   >> tsol = 0:.1:5;
   >> [tsol,xsol] = ode45(@(t,x) [-x(2); x(1)], tsol, [0; 1]);
   ```
   It is also possible to set options that determine how ode45 behaves; for example, we can set the maximum step size it is allowed to try by setting up an options object and passing that to ode45:
   ```matlab
   >> options = odeset('MaxStep', 1);
   >> [tsol,xsol] = ode45(@(t,x) [x(2); -x(1)], [0 10],...
                        [0;1], options);
   ```

1.4. **ode15s.** This is another numerical solver for differential equations, and takes exactly the same arguments as ode45. It deals well with stiff systems, for which solutions have regions that change very much faster than they do in others.

1.5. **plot.** Plot one vector against another; e.g.,
   ```matlab
   >> plot( tesol,xesol(:,1),'-k', t45sol,x45sol(:,1), '--k' );
   ```

\(^1\)Except that it doesn’t support the addition of options.
2 LAB 4: DISCONTINUOUS FORCING, SIGNALS, AND NUMERICAL METHODS, PART B

1.6. tic. This starts MATLAB's internal timer, so that you can see how long a command runs for; see toc

1.7. toc. This stops MATLAB's internal timer, so that you can see how long a command runs for. For example, to see how long a call to ode45 takes:

```matlab
>> disp('timing for ode45');
>> tic
>> [t,x] = ode45(@(t,x) [1000*x(2); -1000*x(1)], [0 100], [-2;5]);
>> toc
```

2. Background

In this lab we consider a circuit model, which is a second-order, linear, constant-coefficient differential equation. In the prelab we found this to be

\( y'' + 2\gamma y' + \omega_0^2 y = F(t) \)  

The characteristic polynomial of the associated homogeneous equation is \( \lambda^2 + 2\gamma\lambda + \omega_0^2 \), with roots \( \lambda = -\gamma \pm \sqrt{\gamma^2 - \omega_0^2} \). Thus, if \( \gamma \) is a small (relative to \( \omega_0 \)) positive number, the system is underdamped and the solution can be written in the form \( y_c(t) = Re^{-\gamma t} \cos((\sqrt{\omega_0^2 - \gamma^2})t - \phi_0) \) for some \( R \) and \( \phi_0 \). In this lab, we consider forcing functions which are discontinuous.

To solve an equation such as (1) numerically (e.g., with ode45), we rewrite it as a system and the numerical method then uses known data (e.g., the initial conditions and system of equations) to predict the values of the variables \( y \) and \( y' \) at a later time. In Part A we considered Euler's method to see how this works (though it is sufficiently inaccurate that it wouldn't be useful in any production context).

3. Part B

Your entire group of four people is responsible for completing the following and the lab report summarizing your results, which is due at the beginning of Workday 1 for the next Lab.

In the following, we consider the equations \( y'' + y' + 40y = F(t) \) and \( y'' + 40y = F(t) \) for different discontinuous \( F(t) \), with the goals of investigating what the differences are between the response \( y \) when \( F(t) \) is a short impulse and when \( F(t) \) is a delta function, and of seeing what happens when we try to cancel out a response by imposing an impulse.

Unless otherwise stated, let \( I(t) = \frac{1}{a}(u_c(t) - u_{c+a}) \). With the function Impulse.m from the course page, you can define different impulses \( I_j(t) \) with

```matlab
>> I1 = @(t) Impulse(t,c,a1);
>> I2 = @(t) Impulse(t,c,a2);
```

and so on (assuming that \( c \), \( a1 \), and \( a2 \) are already defined, of course).
Exercise 1. Review your Part A work from Exercises 4 for both pairs. Suppose that we consider the problem $y'' + y' + 40y = \delta(t - 1)$. $y(0) = y'(0) = 0$, where $\delta(t)$ is the Dirac delta function at $t = 0$. What will the response look like? Sketch (by hand) what you think it will look like. Then solve the problem numerically with ode15s using $F(t) = l_j(t)$, where the $l_j(t)$ are the impulses similar to those considered in pair 2, exercise 3, having $c = 1$ and $a = 0.5$, $a = 0.25$, $a = 0.05$, and $a = 0.01$. Note how your result confirms (or refutes!) your expectation.

Exercise 2. Now suppose that we have a circuit in which there is an (existing) undesired signal (current). This may have been started by an impulse forcing exercise 2.

To make our analysis easier, let’s take $\gamma = 0$ in (1). Then an initial impulse at the origin is equivalent to solving $y'' + 40y = 0$ with initial conditions $y(0) = 0$, $y'(0) = 1$. Solve this problem numerically with ode45, generating points corresponding to the time vector $t_{sol} = 0:0.01:8$ (see the MATLAB section above to specify the times at which to get solution values). Verify that you get the solution you expect.

Note that the signal has a period; call this $T$ (you should be able to figure out what $T$ is). Now suppose you want to zero out this signal by applying an impulse at $t = T$. What is the magnitude of the impulse you should apply? In what direction? Use ode15s to solve the equation $y'' + 40y = -k l(t)$ with zero initial conditions and $l(t)$ being an impulse starting at $c = T$ having width $a = 0.05$. Pick $k$ so that you will zero out the signal (think about what the impulse does to the solution—in particular, how it is related to the initial condition on $y'(T)$). You will want to have this solution on the same points as the solution you generated with ode45, above. Then plot this solution added to the original signal. Does it behave as you expect? Try smaller and smaller values of $a$ to see if, as your $-kl(t)$ converges to $-k\delta(t - T)$, the cancellation works.

Exercise 3. Finally, let’s revisit Euler’s method and see what happens when we try to solve this problem with that. Generate solutions to the problem $y'' + 40y = F(t), y(0) = 0, y'(0) = 1$ with ode15s and eulermethod.m, with $F(t)$ chosen to be an impulse of width $a = 0.01$ that comes close to canceling out the signal at $t = 1$. Plot the results together to see how they differ. How well does the numerical solutions with Euler’s method do this? Can you explain what you see?

4. Lab Report

Lasers, shmazers. You’ve decided that it’s time to start a new consulting company to make your first million dollars. Because of your talented and charismatic team you’ve immediately been contacted by an electronics company that is having issues with signal cancellation in some of its circuits. They are trying to
cancel unwanted signals (noise) in the circuits by applying an appropriate spike in current to the circuit. They report that they are unable to completely cancel undesired signals as they expect they should, and that their modeling software is providing inconsistent and unreliable predictions of what they should be seeing in the first place.

Write a report that explains how numerical error and the difference between theoretical limiting behavior and the real-world may result in their observed difficulties, following the outline indicated below.

I. Introduction: Briefly introduce the differential equation(s) you are considering, what they model, and how numerical solvers may obtain approximate solutions to them. Describe how the idea of signal cancellation may work, and how that is represented in your models.

II. Body:
   a. Explain how your work in math 216 allows you to do this analysis. In particular, comment on the nature of the unit step and delta functions, and how you are able to solve analytically problems involving those forcing terms.
   b. Use your work from Part A to illustrate some of the issues that confront numerical solvers when working with discontinuous forcing functions. Comment on the differences between different numerical solvers, and why you might pick one over another in different circumstances.
   c. Discuss the limiting behavior of a system with an impulse of decreasing width (and increasing height). Solve the system exactly using a delta function as the forcing and show that the limit of the (numerical) solution as the impulse width decreases agrees with the analytical result you obtain.
   d. Explain how one might choose a signal to cancel another in a circuit. Reflect on how trying to do this with a (finite width) impulse differs from being able to apply a (zero width) delta function forcing.

III. Conclusion: Provide a reasonable explanation for the behaviors that the electronics company is reporting, and some suggestions for how they might remediate to obtain results more to their liking.

References