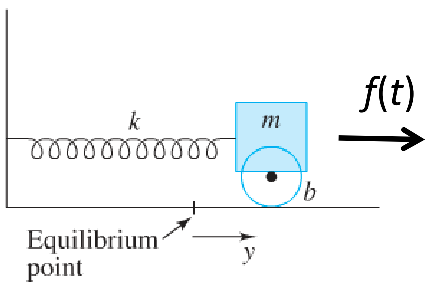
Lab 3: Model of a Spring-Mass-Damper

Consider this Spring-Mass-Damper System (SMD) and its differential equation with forcing function *f*(*t*).



DE: $m \cdot \frac{d^2y}{dt^2} + b \cdot \frac{dy}{dt} + k \cdot y = f(t)$

Initial Conditions: $y(0) = 3,\ y'(0) = -2$

Constants: `m = 9, b = 12, k = 229, f(t) = 0`

Part A: Finding the exact solution

1. The values for the constants *m*, *b* and *k* have been assigned in the code below.

```
%% 1: Constants for the spring-mass-damper system.
% Prepare the workspace.
clc, clear

% First look at the homogeneous case where the forcing term is zero.
% That is, f(t) = 0.
% Constants entered for you as symbols.
m = sym(9); b = sym(12); k = sym(229);

% Find the roots of the characteristic equation:
char_poly = [0, 0, 0]   % <- Fix this stub.
```

a. Fix the above stub for the characteristic polynomial. Notice it is a row vector.

b. Find the roots of the characteristic equation, using the roots command.   >>> help roots

Paste your answers in the box below for credit. Do not use floating point numbers. The imaginary part is an inteter. The real part is a rational number. Use exact expressions.

Question 1: The roots are:

Note the roots are complex conjugates with a negative real part.

2. What is the discriminant $D = b^2 - 4ac \equiv b^2 - 4mk$ of the characteristic equation?

Paste your answer in the box below for credit.

Question 2: The discriminant is *D* = ____

3. Find the <u>exact</u> solution for our IVP (initial value problem) using dsolve.
Initial Conditions: $y(0) = 3,\ y'(0) = -2$. Here's some starter code:

```
%% Question 3: Find the exact solution.
syms y(t)
Dy = diff(y,t); D2y = diff(y,t,t);
m = 9; b = 12; k = 229;
DE = m*D2y + b * Dy + k * y == 0;
```

Now use dsolve to find the exact solution satisfying the initial conditions: $y(0) = 3,\ y'(0) = -2$.
You will need to use two equal signs == inside dsolve to specify each initial condition.
Does the solution have the expected form: $y(t) = e^{\alpha t} \cdot (A\cos\beta t + B\sin\beta t)$ when the roots are complex conjugates $r = \alpha \pm \beta i$?

Paste your answer in the box below for credit.

Question 3: The exact solution is: $\boldsymbol{y(t) =}$

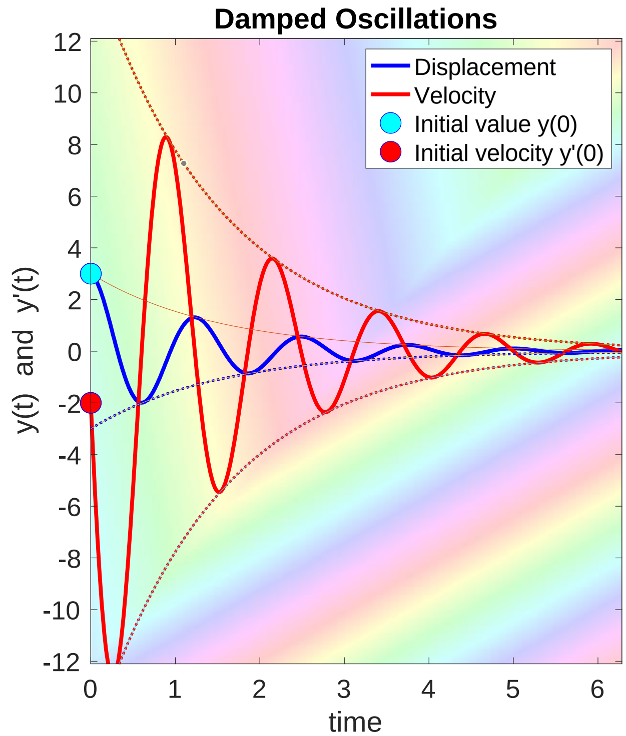
Question 4: Use matlabFunction and diff do define the exact solution $y(t)$ and its derivative $y'(t)$.
Record the exact expression for the derivative $y'(t)$ below.
If you named the solution returned by dsolve as say sol, you can do this as follows.

```
%% Exact solution and its derivative.
y = matlabFunction(sol)
Dy = matlabFunction(diff(y,t))
```

Question 4: The exact solution for the <u>derivative</u> is: $\boldsymbol{y'(t) =}$

Question 5: Plot the exact solution $y(t)$ and its derivative $y'(t)$ over the interval from 0 to 2π.
Plot the exact solution in **blue** with a line thickness of 3, and plot the derivative in **red**. Label both axes, and apply a title such as "Damped Oscillations.

Replace this graph with your completed graph. Your graph <u>must</u> **<u>not</u>** have the colorful background!



GRADER – Envelopes not required!

Part B. State Space Representation

Let us now rewrite the **spring-mass-damper** system in **state space** form.

$$m \cdot \frac{d^2y}{dt^2} + b \cdot \frac{dy}{dt} + k \cdot y = f(t)$$

Instead of the given 2nd-Order DE for the unknown displacement $y(t)$, we can convert to a 1st-order **<u>system</u>** by making the following substitutions. Let $x_1 = y$ and let $x_2 = \frac{dy}{dt}$. Collect these two new variables into a **<u>column</u>** vector, which is called the **state vector**.

$$\vec{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$ **State vector**

Differentiating the state vector, and using the original 2nd-Order DE to eliminate $\frac{dx_2}{dt} = \frac{d^2y}{dt^2}$ we find:

$$\frac{d}{dt}\vec{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{f}{m}(t) - \frac{k}{m}x_1 - \frac{b}{m}x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ f(t)/m \end{bmatrix}$$

Thus, if we define:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \quad \text{and} \quad \vec{\mathbf{b}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The **state-space equations** are:

$$\frac{d}{dt}\vec{\mathbf{x}} = A\,\vec{\mathbf{x}} + \vec{\mathbf{b}}\,\frac{f(t)}{m}$$

6. Write a function in a file named **smd.m** (short for spring-mass-damper) which returns the vector:

xdot = $\frac{d}{dt}\vec{\mathbf{x}}$

Your header will be:

```
function xdot = smd(t,x) % Note, x will be a column vector.

    % The last two lines will be:
    xdot = A*x + b*f/m;
end
```

In between you must:

i. Specify the values for *m*, *b* and *k*. (Use `m = 9, b = 12, k = 229` for all of the lab.)

ii. Set *f* to zero.

iii. Define the 2×2 system matrix $A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}$.

Tip: Use a semi-colon to start a new row in a matrix or column vector.

iv. Define the column vector $\vec{\mathbf{b}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Verify your **smd** function is working by evaluating it. First recall the initial conditions.

$y(0) = 3,\ y'(0) = -2$, so that $x0 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$. Enter the column vector x0, then evaluate:

>> smd(0, x0)

This should equal the derivative of the **state vector** at time zero.

6. Record your answer here for credit. Just write in the column vector returned by: >> smd(0, x0)

Question 6: The value of smd(0, x0) is:

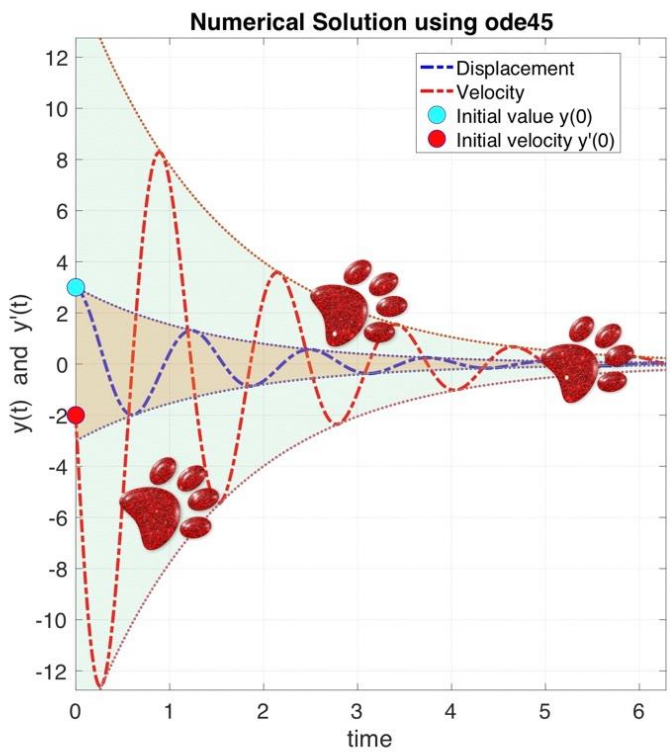
Question 7. Now that **smd.m** is working, you are ready to use ode45 to solve the 1st-order system.
Produce a second simultaneous plot of y and y', but now use ode45. Use a **dash-dot line** for both curves.
Use the initial conditions $y(0) = 3,\ y'(0) = -2$, so that $x0 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$.

Add a **title**: `Numerical Solution using ode45`
Add a **legend** which includes Displacement and Velocity as shown in the sample plot.
Increase the font size to 20.
Sample code to get you started using ode45.

```
x0 = [3; -2]        % Initial condition.
tf = 2*pi;          % Final time.
[t,x] = ode45(@smd, [0,tf], x0);
```

7. Replace this sample plot with your completed graph. Be sure the position is in **blue** and the velocity in **red**. Must use a dash-dot line for both curves. Your work must **not** contain the footprints left by some mysterious ghost in the machine.

Sample




```
GRADER - Envelopes not required!
Shaded areas not required.
```

```
GRADER - Note blue curve
must start at +3.
```

Points 8 – 10: Tiled Plots

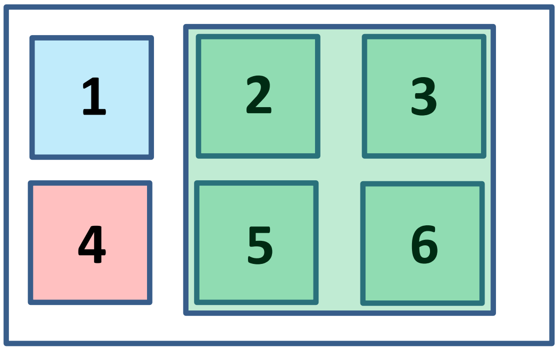
The rest of the points will be earned by submitting a **tiled plot** of the system meeting all the following requirements. Now we will present the above solution in a different fashion using tiled plots to highlight different features. Enter the following command to review how tiled plots work.

```
>> help subplot
```

Consider the command: `subplot(2, 3, 1)`

This will break your figure window into a 2 x 3 grid yielding six sets of smaller axes labelled as follows.

Figure



a. We will plot just the displacement y(t) in tile #1. Use: subplot(2, 3, 1)

b. We will plot just the velocity y'(t) in tile #4. Use: subplot(2, 3, 4)

c. We will create a phase plot using four tiles on the right. Use: subplot(2, 3, [2, 3, 5, 6])

This will reserve the most space for our phase plot which is the star of the show!

In tile #1, plot just the displacement $y(t)$ in blue with a line width of 3. There is no need to recalculate it. The displacement data is in the first column of **x**, which was already computed using ode45. Set the title to 'Displacement' and label both axes. Set the xlabel to `'Time in seconds'` and the `ylabel` to `'y(t)'` Turn the `grid on`.

In tile #4 (bottom left), plot just the velocity $y'(t)$ in red with a line width of 3. There is no need to recalculate it. The velocity data is in the second column of **x**, which was already computed using ode45. Set the title to 'Velocity' and label both axes. Set the xlabel to `'Time in seconds'` and the `ylabel` to `y'(t).` You may need to use the "two-quotes trick" mentioned already. Turn the `grid on`.

Now to earn the last three points.

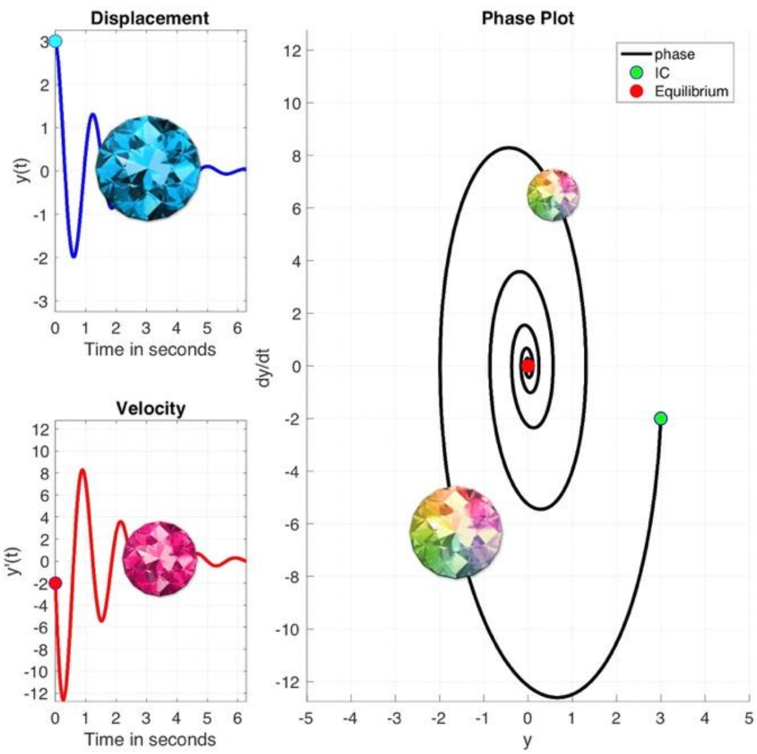
8. Now create the **phase-plot** on the right. It will occupy the entire four cells on the right, and thus use tiles 2, 3, 5 and 6. That is why the sample code uses: subplot(2, 3, [2,3, 5, 6])

The phase plot uses y for the horizontal axis and y' for the vertical axis. Note these are just the first and second columns of x. Plot the phase curve now in **black** (k) using a line width of 3. The xlabel should be set to y and the ylabel to dy/dt. Do not use axes equal for the phase plot. We want to give it the full space available.

9. Turn **hold on** immediately after plotting the phase curve. We will decorate the curve with some salient points. The initial point (in phase space) is $(y(0),\ y'(0)) = (3,-2)$. Show this point now using a **green** circle. The equilibrium point is (0,0). Show the equilibrium point using a **red** circle. Green for start, red for stop.

10. Add a legend to your phase plot similar to that shown in the sample tiled plot below.

Points 8-10: <u>Replace</u> this sample plot with your own tiled plot for points 8-10. Set an appropriate font size.
A portion of the sample has been hidden using shiny baubles. Your graph must **<u>not</u>** contain any baubles.
Your spiral curve will show that the spiral does not quite reach the equilibrium point.



GRADER – Make sure the green dot is at (+3, −2).

Award one point for each of the three tiles if the graph there is complete.

Ready to Submit?

Be sure all ten questions are answered. When your lab is complete, be sure to submit three files:

1. Your **completed Answer Template** as a PDF file
2. A copy of your **MATLAB Live Script**
3. A **PDF** copy of your **MATLAB Live Script** (Save-Export to PDF…)

The due date is the day after your lab section by **11:59pm** to receive full credit. You have one more day, to submit the lab (but with a small penalty), and then the window closes for good and your grade will be zero.