ENGR 232 Lab #3 – Extinction of the Passenger Pigeon

In the previous lab we studied the **logistic equation** which has the general form:

$$\frac{dP}{dt} = rP \cdot \left(1 - \frac{P}{K}\right)$$

Above, the constant r defines the rate of growth, while K is the **carrying capacity**. In this lab, we will study a modification of the logistic equation which includes a **threshold** T. Logistic growth with a threshold is modeled by the nonlinear, autonomous differential equation:

$$\frac{dP}{dt} = -rP \cdot \left(1 - \frac{P}{K}\right) \cdot \left(1 - \frac{P}{T}\right)$$

where $r > 0$ and $0 < T < K$. Notice the additional minus sign.

The extinction of the passenger pigeon, is a tragic and important example of **anthropogenic extinction**. In the seventeenth and eighteenth centuries, billions of these graceful birds were endemic to North America, perhaps somewhere between 3 and 5 billion. The passenger pigeon or wild pigeon (Ectopistes migratorius) derived its common name from the French word "passager", which referred to the vast flocks of birds "passing by" on their migratory routes. It is said these flocks were so vast they often occluded the sun.

For more on the extinction of the passenger pigeon, see this very readable article on Wikipedia:

https://en.wikipedia.org/wiki/Passenger_pigeon

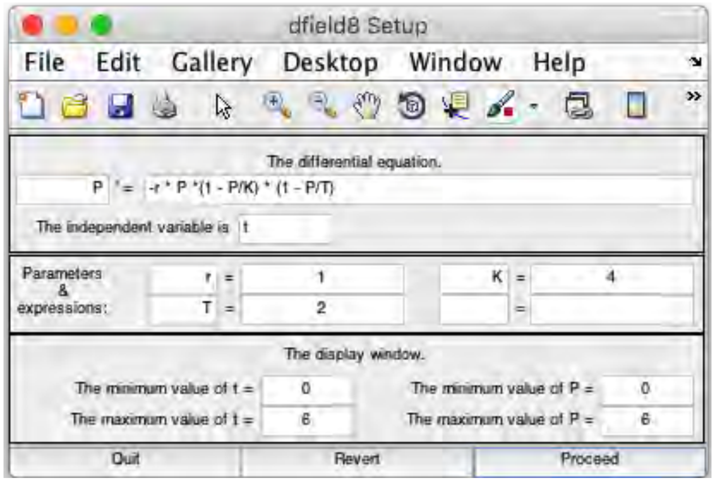
Long harvested sustainably by Native Americans, the arrival of Europeans brought intensified hunting, particularly in the 19^{th} century. "A slow decline between about 1800 and 1870 was followed by a rapid decline between 1870 and 1890. The last confirmed wild bird is thought to have been shot in 1900." – From the Wiki article.

Martha, thought to be the last passenger pigeon, died on September 1, 1914, at the Cincinnati Zoo.

Martha: The Last Passenger Pigeon
(Public Domain)

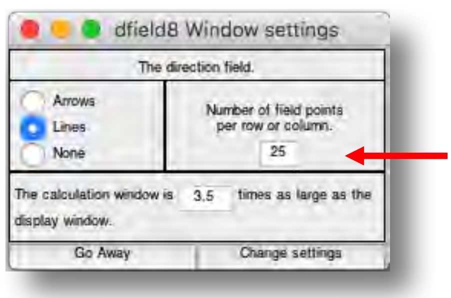
Parameters: We will work in units of **billions**. To explore how the anthropogenic extinction of the passenger pigeon can by modeled by a logistic model including threshold, let's first define some reasonable numbers.
Let the rate of growth be $r = 1$. Let the carrying capacity be $K = 4$ and let the threshold be $T = 2$.
(Recall the units are in billions of birds.)

Questions 1 and 2: Run **dfield8**, and enter the differential equation (with threshold).
a. In the "Parameters & expressions" area, define $r = 1$, $K = 4$ and $T = 2$. (units are in billions of birds)
Denote the dependent variable by P and the independent variable by t. Enter the differential equation.



b. Adjust the display window so both axes go from 0 to 6.

c. Under **Options**, select **Window settings.** and specify exactly 25 field points per row and column. This will guarantee the tick marks are perfectly flat at the three critical points.



d. The sample plot below only shows solutions that are **above** the carrying capacity of $K = 4$ billion birds. Perhaps there had been an abundance of food for the previous few years, but now the available food has returned to a more normal amount. The curves clearly show the **autonomous** nature of this DE, with each solution curve shifted **horizontally** to the next.

Using the **Keyboard Input** option, solution curves were found (in both directions) from the initial points:

$$P(0) = 5, \quad P(1) = 5, \; P(2) = 5, \; P(3) = 5, \; P(4) = 5 \; and \; P(5) = 5$$

Notice all these curves have exactly the same shape, and each is merely shifted horizontally to the next as expected. Also, the three critical points at $P = 0, 2, 4$ are shown as red horizontal lines.

Create a similar plot using **dfield8.m**

In addition to the curves through the initial points:

$$P(0) = 5, \quad P(1) = 5, \; P(2) = 5, \; P(3) = 5, \; P(4) = 5 \; and \; P(5) = 5$$

add curves in the middle region satisfying:

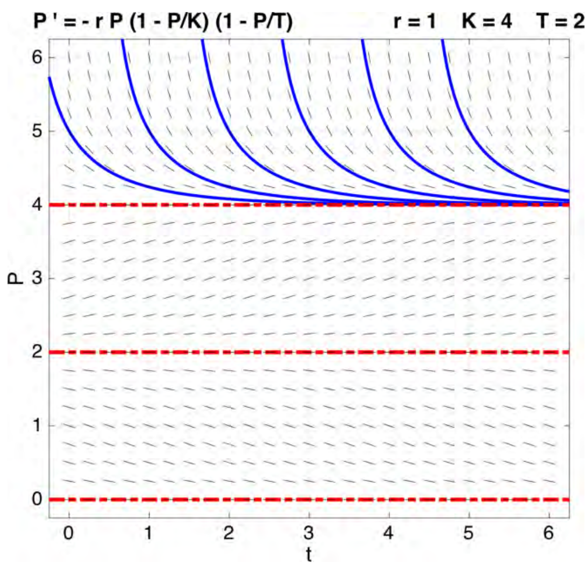
$$P(0) = 3, \quad P(1) = 3, \; P(2) = 3, \; P(3) = 3, \; P(4) = 3 \; and \; P(5) = 3$$

and also add curves in the lower region satisfying:

$$P(0) = 1, \quad P(1) = 1, \; P(2) = 1, \; P(3) = 1, \; P(4) = 1 \; and \; P(5) = 1$$

Use the **Property Editor** to color curves in the top region **blue**, **green** in the middle and **red** in the bottom. Increase each line thickness to 3.

Questions 1 and 2: Replace the plot below with your completed plot. Show top solutions in blue, middle solutions in green and bottom solutions in red. Also include the three equilibrium solutions.



Your **blue** curves in this region $P > K$ represent overpopulation.

Your **green** curves in this region $T < P < K$ represent a healthy population with ample food.

Your **red** curves in this region (below threshold) represent the extinction of the passenger pigeon.

Questions 3-5: Build your very own custom direction field plotter in three easy steps!

Now that you have explored the solutions **qualitatively** using **dfield8.m**, it's time to "look under the hood" so to speak. Let's try to make our own "**dfield8**". We'll need to plot the direction field and solution curves given a number of initial conditions. For the solution curves, we'll use **ode45**. In the remaining portion of this lab, we'll try to duplicate the graph you just obtained using **dfield8** from scratch – and even add a few bells and whistles!

The code below shows how you can create a direction field for any first-order differential equation.
In practice, you will need to tweak the settings to obtain a polished look. I'll give the code in three blocks.
Each block should remind you of settings that are given when you first open **dfield8**.

Block 1: Define your differential equation of the form: $\frac{dP}{dt} = f(t, P)$
(It doesn't matter what you name the variables, as long as you are consistent.)

```
% BLOCK 1: Extinction of the Passenger Pigeon.
% Enter the parameters for the logistic growth model with threshold.
clear, clc
r = 1;  K = 4;  T = 2;
f = @(t,P) -r * P .* (1 - P/K) .* (1 - P/T)
```

Notice we have defined the slope field using an **anonymous function**. We also wrote it as a function of both t and P, even though this particular example is autonomous. This will allow you to reuse the code later when a new DE is not autonomous. Just change the parameters and the anonymous function for the DE's slope function $f(t, P)$.

Block 2: Setting up your plotting window.
Much of the code below is just to polish the appearance of the resulting graph.
Adjust the parameters as desired.

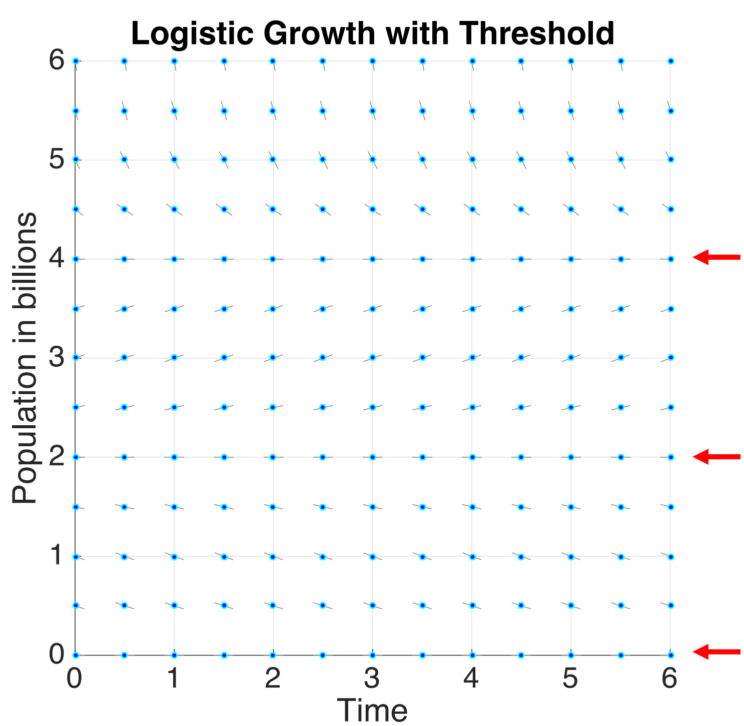
```
% BLOCK 2: Set the plot window dimensions here.
tmin = 0; tmax= 6;  ymin = 0;  ymax = 6;
figure % Pop up a new figure
axis([tmin tmax ymin ymax ])
axis square
grid on; hold on
set(gca, 'FontSize', 20)
title('Logistic Growth with Threshold')
xlabel('Time'); ylabel('Population in billions')
```

Block 3: Graph your Direction Field as a grid of tick marks with slope defined by the RHS of $\frac{dP}{dt} = f(t, P)$
Focus on the **nested for loops** which is the heart of where the tick marks for the direction field are drawn.

```
% BLOCK 3: Add the slope field here
radius = 0.1                  % Control the size of the tick marks
spacing_horizontal = 0.5      % Control their horizontal spacing.
spacing_vertical = 0.5        % Control their vertical spacing.
my_color = [0.5, 0.5, 0.5]    % Control their color.
optional_dots = 1             % Control whether a dot is placed. Use 0 or 1

for tp =tmin: spacing_horizontal: tmax
    for yp = ymin: spacing_vertical: ymax
        slope = f(tp,yp); theta = atan(slope);
        dt = radius*cos(theta);  dy = radius*sin(theta);
        % Plot the tick marks!
        plot([tp - dt , tp + dt ], [yp - dy , yp + dy] , 'Color', my_color)

        plot( tp, yp, 'cyano', 'MarkerSize', 4,  'MarkerFaceColor', 'blue')

    end
    pause(0.2)
end
```

Place this in an **if** block, to make the dots optional.

When you have all three blocks working in your script for today's lab, run the code to see a direction field similar to this. Notice each column of tick marks is the same, because this equation is autonomous!



There is no harvesting in this plot. See Q4 and Q5.

At the critical values, the tick marks are horizontal.

Question 3: Add an **if** statement inside the inner for loop, to control whether the small dots visible at the center of each tick mark are drawn. The user will be able to control whether the dots are shown by setting the variable **`optional_dots`** to 0 for NO and 1 for YES.
Tip: For help on the syntax for **if** statements type: >> help if
Don't forget to indent the body of the if block, and to use the **end** keyword to leave the if block.

Question 3: Optional Dots
Paste in **<u>just</u>** your complete **if** block here, showing the header, indented body and the **end** keyword.

Questions 4 & 5: Population with Harvesting
Draw the direction field for this new DE which models the population under **<u>harvesting</u>**.
The birds are harvested (killed for food) at a continuous rate of $h = 0.25$ billion birds per year.

$$\frac{dP}{dt} = -rP \cdot \left(1 - \frac{P}{K}\right) \cdot \left(1 - \frac{P}{T}\right) - h$$

The constants r, K and T are unchanged. Based solely on the new direction field with harvesting, estimate the two modified, positive critical values.
Hint: Just look for where the tick marks appear horizontal. You can decrease their spacing (and size), then "embiggen" to get a better estimate. Is the lower extinction region bigger or smaller with harvesting??

Question 4. With harvesting at $h = 0.25$ the former critical value of 4 drops to about:	____
Question 5. With harvesting at $h = 0.25$ the former critical value of 2 rises up to:	____

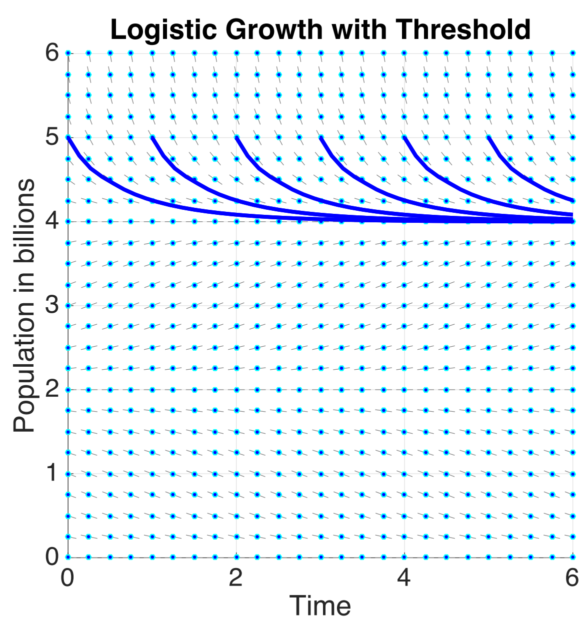
Alert: Be sure to set the harvesting back to zero before proceeding.
Questions 6 – 10 Will be graded based on your submitted graph.

Set the harvesting back to zero and draw the direction field again by running the three blocks of code. We will now add solutions on top of your direction field using **ode45**.

Now add the next block of code to draw some solution curves in blue in the top strip where the population is **above** the carrying capacity.

```
% BLOCK 4: Population is above carrying capacity.
% Plot solutions in the top strip.
tStart = 0;  tEnd = 6;    yStart = 5  % Initial value. Rerun with values 3 and 1.
[t_out, y_out] = ode45(f, [tStart tEnd], yStart);

both_directions = 1        % Change to 0 to draw in the forward direction only.
show_initial_point = 1     % Change to 0 to suppress drawing the initial points.
line_color ='blue'         % Rerun with green for 3 and red for 1.
dot_color ='cyan'          % Rerun with green for 3 and red for 1.

% Since the DE is autonomous, the curves just shift horizontally.
% Initial conditions y(k) = yStart. Forward only.
for k = 0 : 5
    plot(t_out+k, y_out, line_color, 'LineWidth', 3)
end
```

You should be able to reproduce this graph at this stage.



Question 6: You can see the solution curves are only plotting **<u>forward</u>** in time from the initial points $y(k) = 5$, where k = 0, 1, 2, 3, 4, 5, resulting in six solution curves. Notice how each is just shifted **<u>horizontally</u>**!

Add an **if** statement containing a **for** loop, to extend all six solutions backwards in time, **<u>if</u>** the variable **both_directions** is set to 1. You will obtain credit by submitting the final plot.

The key statement you need to get the solutions backwards in time is:

```
[t_out, y_out] = ode45(f, tStart :-0.1: -tEnd, yStart);
```

Place it inside your **if** statement, <u>before</u> the **for** loop.

Solving backwards in time.

* Ignore the warning you may see from ode45 here. The error is happening "off screen".

Question 7: Notice the control parameter in this statement: `show_initial_point = 1`

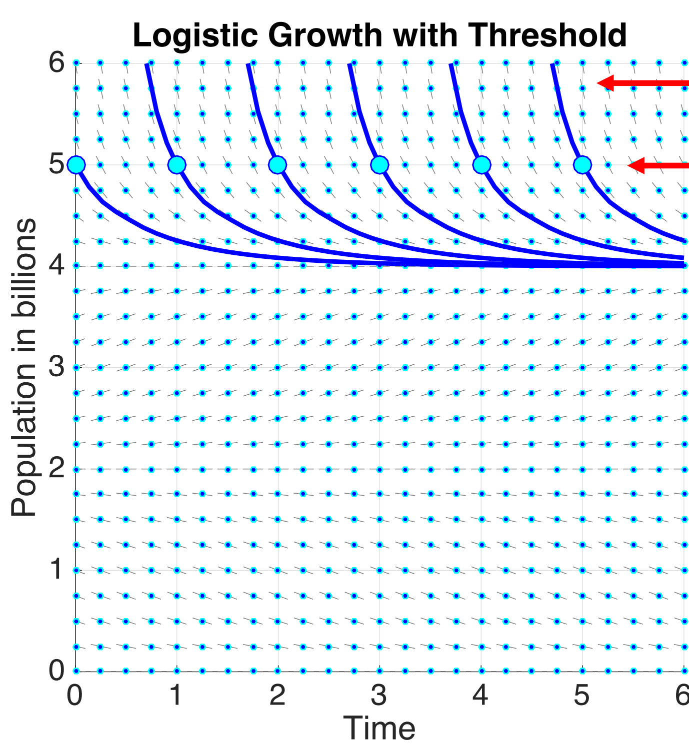
The desire is to show each initial point with a marker if this is set to 1.
Recall the initial points are: $y(k) = 5$, where k = 0, 1, 2, 3, 4, 5,
Add an **if** statement containing a **for** loop, to show all six initial points, **<u>if</u>** the variable `show_initial_point` is set to 1. You will obtain credit by submitting the final plot.

Drop this line of code inside your for loop which should let k go from 0 to 5.
It will draw a circle at the specified size and color at each initial point.

```
plot(k, yStart, 'bo', 'MarkerSize', 12, 'MarkerFaceColor', dot_color) %Markers for IC.
```

The initial points: k = 0 : 5

At this stage, you should be able to reproduce this plot.



Solutions extend backwards in time.

Initial conditions are marked by dots or small circles.

Question 8: Rerun your code for block 4, but for solutions starting in the middle region at `yStart=3`.

Change both the line color and the dot color to **green**. This represents a healthy population with adequate food.

Question 9: Rerun your code for block 4, but for solutions starting in the lower region at `yStart=1`.

Change both the line color and the dot color to **red**. This represents a doomed population dwindling inexorably to **extinction**. Sadly, this is what WE did to the passenger pigeon and many, many other species too numerous to count.

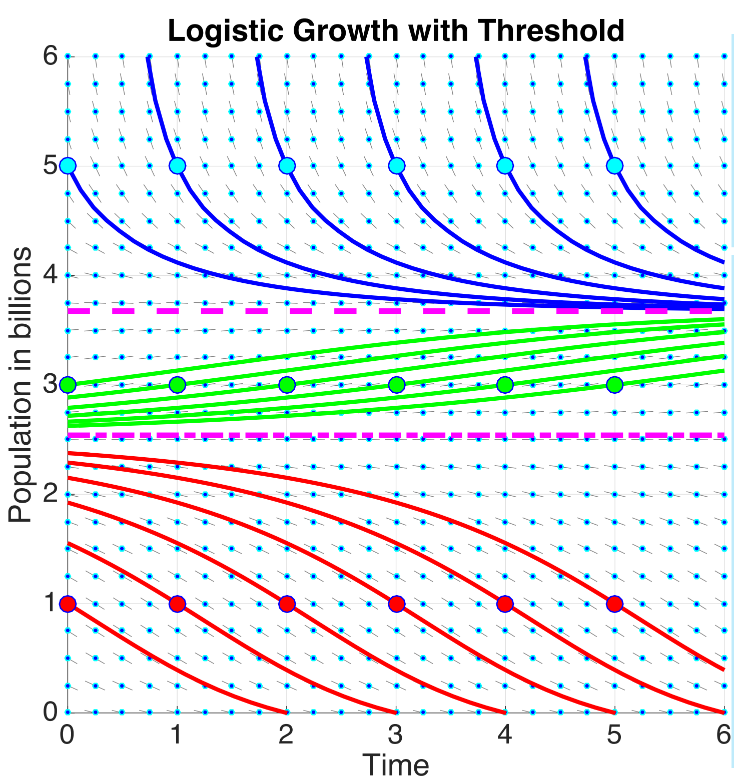
Question 10: Draw in each of the **three** equilibrium solutions as **magenta** **horizontal** lines.

Use a **dotted** line for $P(t) = 0$, use a **dash-dot** line for $P(t) = T$, and use a **dashed** line for $P(t) = K$.

Here's some code for the one at zero.

```
% Equilibrium solution at 0 using dotted line
plot([tmin tmax],[0,0],  'magenta:', 'LineWidth', 4)
```

Questions 6-10: Replace this graph with your completed figure. Your figure will appear similar, but also different because this sample plot includes harvesting with $h = 0.25$ Note how the critical values are not at 2 and 4 in the sample. Yours however will be!



This sample plot includes harvesting! Your graph will not. So your critical points will be **precisely** at 0, 2 and 4. Notice the middle region has shrunk considerably and the extinction region has expanded.

Grader will award five points as follows.

All solutions also extend backwards in time.

Dots are placed at each of the initial conditions and are colored by region.

Green solution curves appear to rise from 2 to 4.

Red solution curves fall from 2 to 0.
The passenger pigeons are going extinct!!

There are three horizontal lines in magenta.

Done? Place your answers in the Answer Template and Submit before the deadline.

Be sure all questions are answered, then submit your answers for this lab as a single PDF.
Submission must be a single PDF file! Only one submission is allowed.