Week 5 - Direction Fields and Phase Portraits

MAT330: Differential Equations

Student Name JoAnn Olney

Date 6/2/25

Introduction:

Systems of differential equations can be analyzed and their solutions found through the use of direction fields, eigenvectors, and phase portraits.

In this MATLAB assignment, you'll learn how to create these various plots to aid in understanding the behavior of the solution to a system of differential equations.

Review the code and comments provided in the "Examples" section below, and then use this information to complete the problems listed in the "Problems" section.

Make sure to run your code so all relevant computations/results are displayed, delete the "Introduction" and "Examples" sections, and then export your work as a PDF file for submission (your submission only needs to contain the "Problems" section that you completed).

Examples:

```
% Example 1 Code - Eigenvectors
% All of the example problems provided below work with the same system of
% differential equations:
%
% x1' = 5x1-3x2
% x2' = 3x1-4x2

% The coefficient matrix of the system can be written as
A = [5 -3; 3 -4];

% Computations by hand yield the eigenvectors [2.6180; 1] and [1; 2.6180].
% This can be confirmed by using the MATLAB function eig() to compute the
% eigenvectors. The eigenvectors are stored as columns of the matrix V below.
[V,~] = eig(A);

% By default, MATLAB normalizes eigenvectors to unit length.
% Often, we  scale the vectors to have a value of 1 in a coordinate so the
% quantities are easier to work with. Changing all values in the vector by
% a common factor doesn't change the direction, which is the only important
```

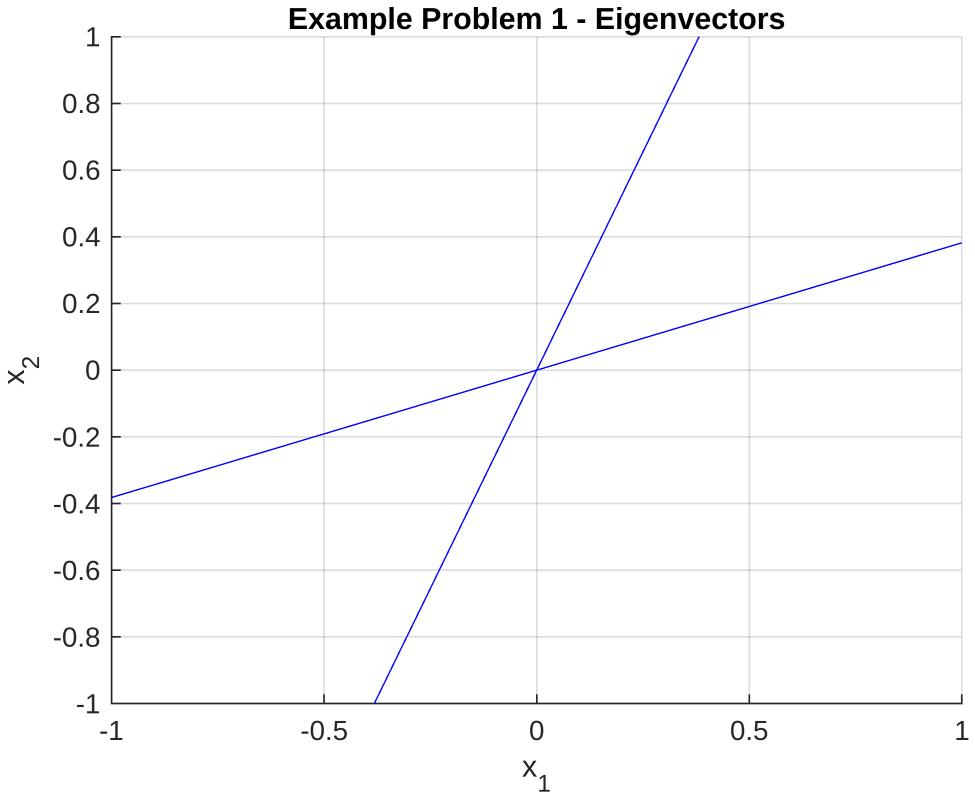
```
% aspect of the eigenvector. The code below gets these scaled vectors where
% we're forcing the smallest entry to a value of 1.
v1 = V(:,1)./min(V(:,1))
```

```
v1 = 2×1
    2.6180
    1.0000
```

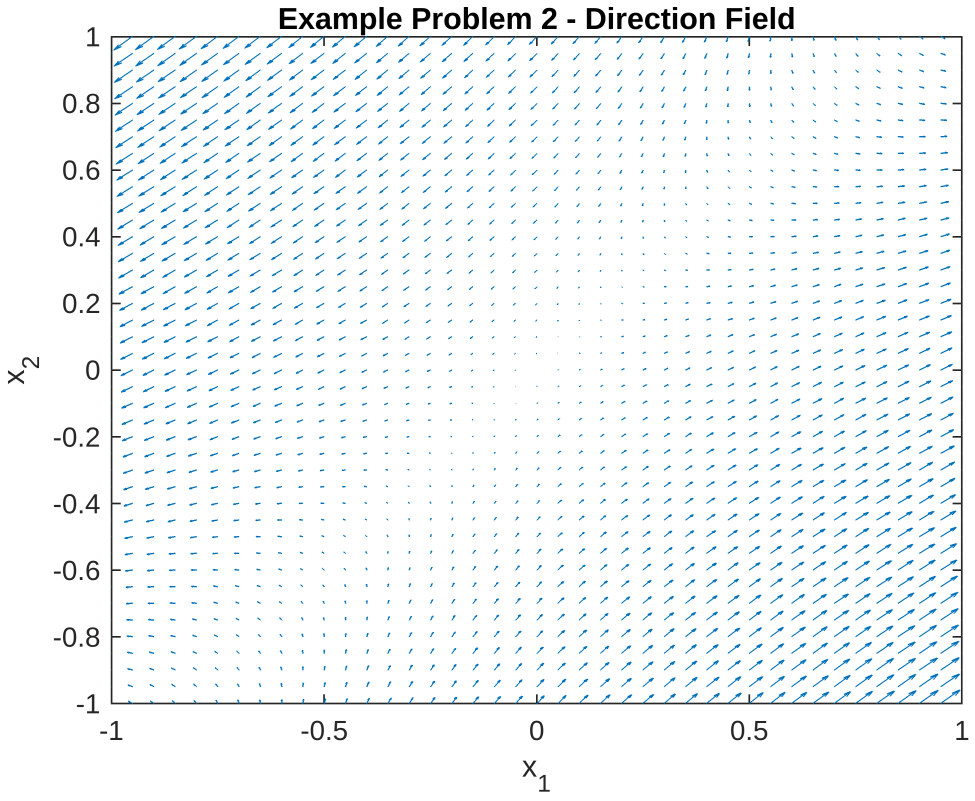
```
v2 = V(:,2)./min(V(:,2))
```

```
v2 = 2×1
    1.0000
    2.6180
```

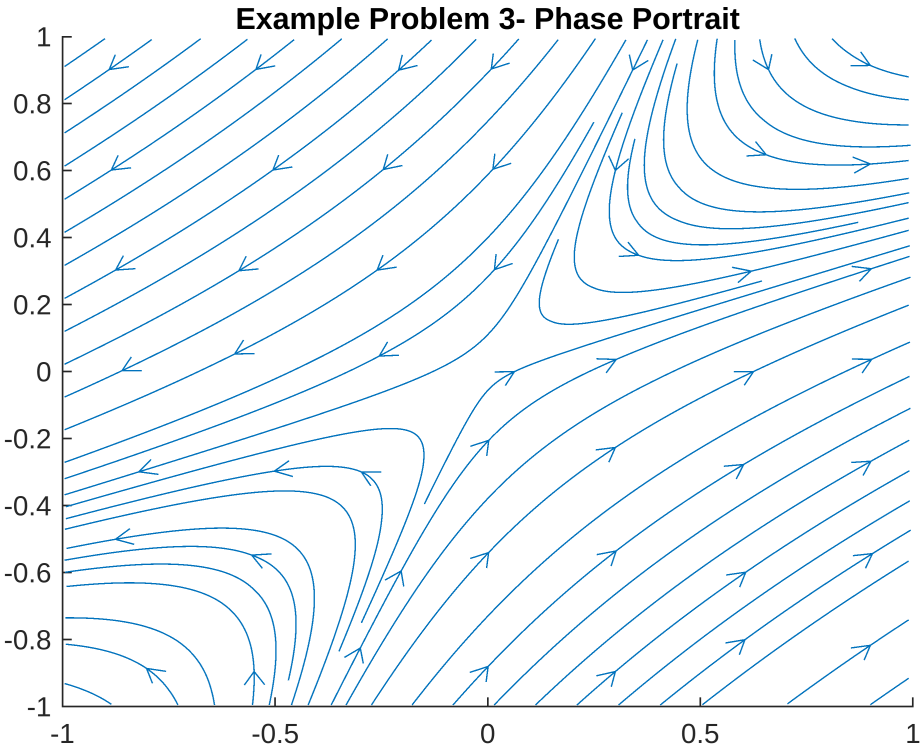
```
% To plot the eigenvectors, we need to have vectors for the x1/x2
% coordinates to plot against
x1Vec = -1:0.05:1;
x2Vec = -1:0.05:1;

% Compute the eigenvectors (Note: these are the v1/v2 relationships found
above)
vec1 = (1/2.6180)*x1Vec; %x2 = (1/2.6180)*x1 for eigenvector 1
vec2 = (2.6180)*x1Vec;  %x2 = 2.6180*x1 for eigenvector 2

% Plot the eigenvectors
figure;
hold on;
plot(x1Vec,vec1,'b');
plot(x1Vec,vec2,'b');
xlabel('x_{1}');
ylabel('x_{2}');
grid on;
title('Example Problem 1 - Eigenvectors');

% put the axes on the same limits
xlim([-1 1]);
ylim([-1 1]);
```




```
% Example 2 Code - Direction Field

% The meshgrid() command is useful for plotting two-dimensional surfaces
% evaluated on a grid of points
[x1, x2] = meshgrid(x1Vec,x2Vec);

% Compute the direction field at the grid of points
x1dot = 5*x1 - 3*x2;
x2dot = 3*x1 - 4*x2;

% The quiver command can be used to plot the direction field
figure;
quiver(x1,x2,x1dot, x2dot);
xlabel('x_{1}');
ylabel('x_{2}');
title('Example Problem 2 - Direction Field');
xlim([-1 1]);
ylim([-1 1]);
```




```
% Example 3 Code - Phase Portrait

% The phase portrait contains similar information as the vector field, it
% is just drawn as a set of contours, as opposed to individual arrows. If
% the vector field is already computed, the streamslice() function can be
% used to plot the phase portrait.

figure;
streamslice(x1Vec,x2Vec,x1dot,x2dot);
title('Example Problem 3- Phase Portrait');
```



Problems:

Problem 1: Consider the system of differential equations $x_1' = -5x_1 + x_2$ and $x_2' = 4x_1 - 2x_2$. Use MATLAB to plot the eigenvectors and direction field of this system on a single plot. Make sure to label both axes and title your figure. Generate your plots for $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$ and set both x_1 and x_2 axes to limits of [-1 1].

```
% Problem 1 Code Here
%coefficient matrix
A = [-5 1; 4 -2];
%eigenvectors
[V,D] = eig(A);

v1 = V(:,1)./min(V(:,1))
```

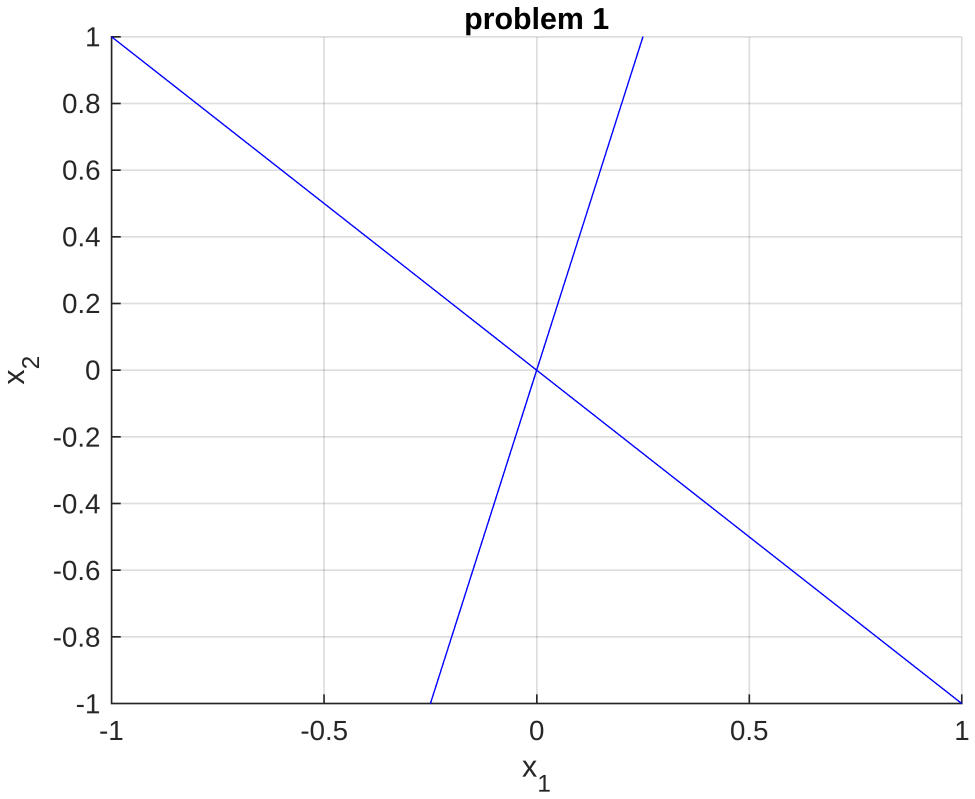
```
v1 = 2×1
     1
    -1
```

```
v2 = V(:,2)./min(V(:,2))
```

```
v2 = 2×1
    0.2500
    1.0000
```

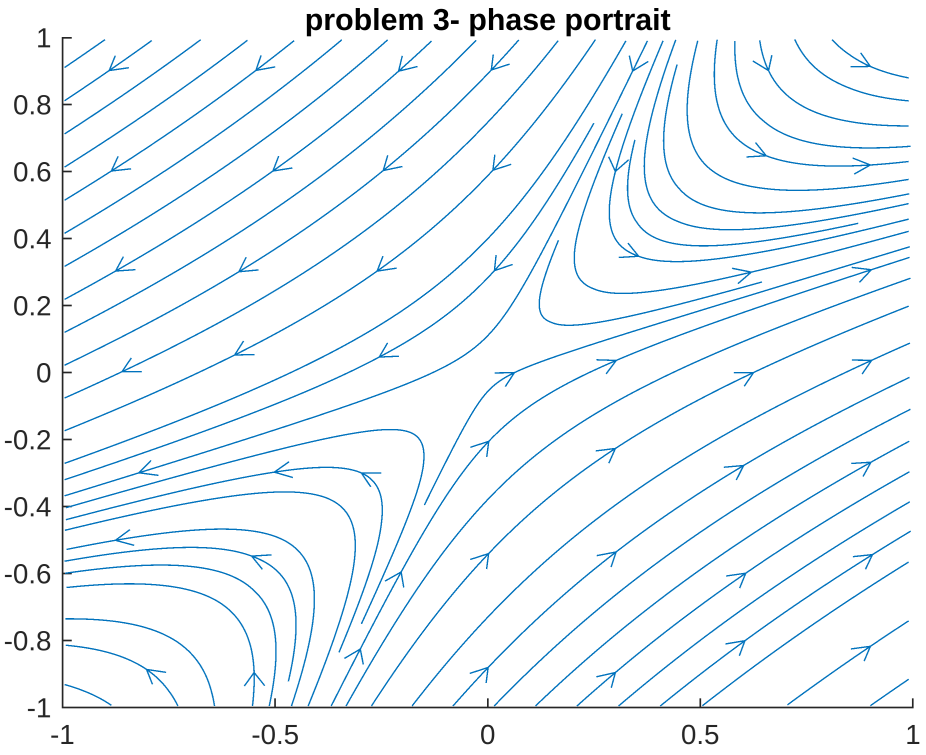
```
%%coordinates to plot
```

```
x1Vec = -1:0.05:1;
x2Vec = -1:0.05:1;

%compute the eigenvectors
vec1 = (-1)*x1Vec;
vec2 = (4)*x1Vec;

%plot the eigenvetors
%figure
figure;
%hold on
hold on;
plot(x1Vec,vec1,'b');
plot(x1Vec,vec2,'b');
%label axes and title
xlabel('x_{1}');
ylabel('x_{2}');
%grid on
grid on;
title('problem 1');

%put the axis on the sam limits
xlim([-1 1]);
ylim([-1 1]);
```



Problem 2: Consider the same system of differential equations in Problem 1. Use MATLAB to plot the phase portrait of this system of differential equations. Make sure to label both axes and title your figure. Use the same x1-axis and x2-axis limits as in Problem 1.

```
% Problem 2 Code Here
%figure
figure;
%same x1-axis & x2-axis as problem 1
streamslice(x1Vec,x2Vec,x1dot,x2dot);
%title
title('problem 3- phase portrait');
```



Problem 3: Analyze the phase portrait from Problem 2 for the initial conditions of $x_1(0) = 0.5$ and $x_2(0) = 0.3$. As $t \to \infty$ to what values do $x_1(t)$ and $x_2(t)$ converge to?

For the initial condition $x_1(0) = 0.5, x_2(0) = 0.3$:

as t $\longrightarrow \infty$, both $x_1(t) \longrightarrow 0$ and $x_2(t) \longrightarrow 0$

This is consistent with the phase portrait above showing the trajectories curving toward and around the origin.