
Table of Contents

Autonomous Systems and Stability: Linear Systems	1
Linear Systems	3
a. Two distinct non-zero, real eigenvalues of opposite sign	4
b. Two distinct non-zero, real eigenvalues of the same sign	5
c. Two complex conjugate purely imaginary eigenvalues	6
d. Two complex conjugate eigenvalues with non-zero real part	7
e. A single real eigenvalue whose eigenspace is 1-dimensional	8
f. A single real eigenvalue whose eigenspace is 2-dimensional	9
Singular Coefficient Matrix	11
Singular Coefficient Matrix, cont.	12

Autonomous Systems and Stability: Linear Systems

```
% In this course, we consider 2x2 systems of 1st order, ordinary
% differential equations:

%       x' = f(x, y, t)
%       y' = g(x, y, t).

% In this presentation, we shall specialize to autonomous systems,
% that is, those in which there is no 't' present on the right side of
% either equation.

%       x' = f(x, y)
%       y' = g(x, y).    (*)

% It is convenient to write the two equations as a single vector equation.
% Using the usual Matlab convention for denoting column vectors, let
%       X(t) = [x(t); y(t)],    F = [f; g]
% Then we can write
%       X' = F(X).    (**)

% A solution to such a system is a pair of functions x(t), y(t) that
% satisfy (*) -- or equivalently (**) -- for all t in the domain of
% definition. We shall interpret such a pair as parametric equations for a
% curve in 2-space. The key fact for autonomous systems is that these
% parametric curves never intersect. For suppose X_1(t) and X_2(t) were two
% different solutions to the same autonomous system that met in a point,
% say X_1(t_1) = X_2(t_2), for some t_1 and t_2. If so, consider
%       Y(t) = X_2(t - t_1 + t_2).
% It clearly solves the system of differential equations (**). Moreover,
%       Y(t_1) = X_2(t_2) = X_1(t_1).
% By the Uniqueness Theorem, it must be that
%       Y(t) = X_1(t)
% that is, X_2 is just a reparameterization of X_1.

% Thus the solution curves fill up the plane with non-intersecting curves.
```

```

% We refer to the resulting diagram as the phase portrait for the system.
% Note this is similar to the phase line that we considered for autonomous
% first order equations in a single variable.

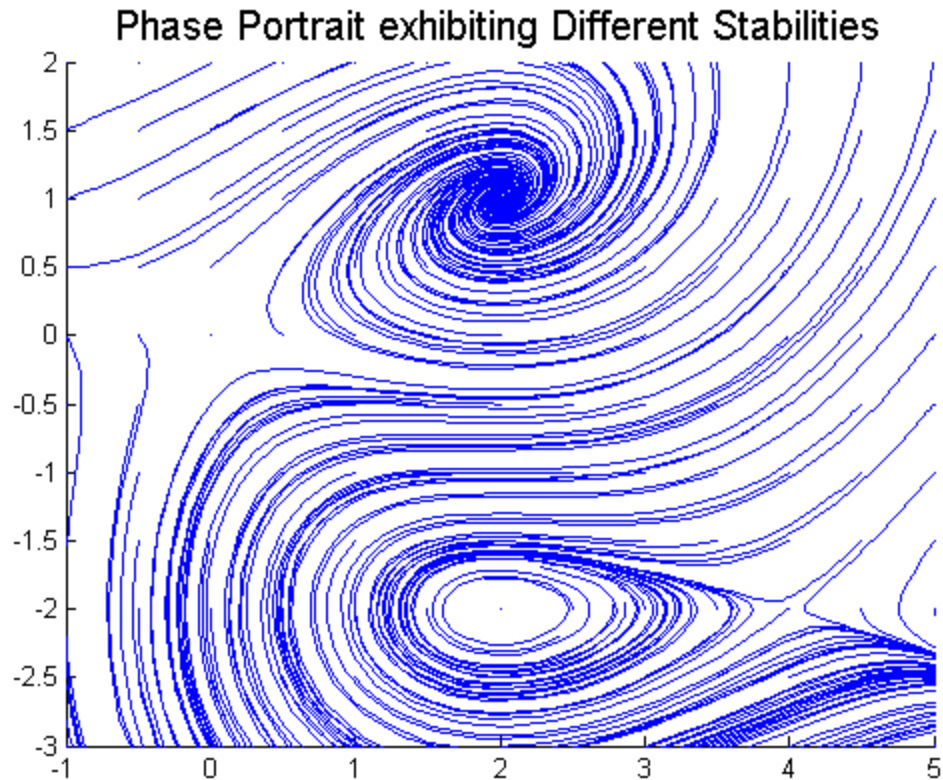
% There are some especially simple solutions to an autonomous system that we
% can always single out. Namely, since the locus of points for which
%     f(x, y) = 0
% is a curve, then the common zeroes of f and g yield constant solutions to
% the system. That is, if f(x_0, y_0) = g(x_0, y_0) = 0, then the curve,
%     x(t) = x_0, y(t) = y_0
% which is of course just a point is a solution to the system.
% We call those points the equilibrium points of the system. It will be
% important to study the stability of equilibrium points; so we say that
% an equilibrium point is:
% Stable -- if solutions that start near the point stay near it;
% Asymptotically Stable -- if solutions that start near the point converge
%     to it as t -> Inf; and
% Unstable -- otherwise.

% Here is an example of a phase portrait with four equilibrium points,
% among which each of the three different stability types is manifested.
% Specifically, there is a stable center, two unstable saddle points and
% an asymptotically stable spiral point. You can inspect the code that
% generates the phase portrait at your leisure.

warning off
figure; hold on
f = @(t, x) [(2 + x(2))*(x(2) - 0.5*x(1)); (2 - x(1))*(x(2) + 0.5*x(1))];
for a = -1:0.5:5
    for b = -3:0.5:2
        [t, xa] = ode45(f, [0 5], [a b]);
        plot(xa(:,1), xa(:,2))
        hold on
        %[t, xa] = ode45(f, [0 -5], [a b]);
        plot(xa(:,1), xa(:,2))
    end
end
axis([-1 5 -3 2])
title ('Phase Portrait exhibiting Different Stabilities', 'FontSize',15)

% You might indulge your curiosity and see what happens in the Command
% Window if you omit the "warning off" command.

```



Linear Systems

Now let us specialize to linear systems. We say that the system is linear if f and g are linear functions of x and y , that is

$$\begin{aligned} \% \quad x' &= ax + by \\ \% \quad y' &= cx + dy. \end{aligned}$$

$\%$ If we write, $A = [a, b; c, d]$, then the system may be written
 $\%$ $X' = AX$.

$\%$ In the Notes, it is shown, using the eigenvalues and eigenvectors of the
 $\%$ matrix $[a, b; c, d]$ how to write down explicit solution formulas for a
 $\%$ linear system. There are several cases depending on the nature of the
 $\%$ eigenvalues. It is my purpose here to illustrate all the cases. I am not
 $\%$ going to supply the vector fields; we'll consider vector fields
 $\%$ for systems in another video dealing with non-linear systems. You also
 $\%$ should realize that for nonlinear systems, one can rarely find formula
 $\%$ solutions and so the phase portraits are generated by numerical methods
 $\%$ (using `ode45` as above). But for linear systems, formula solutions can be
 $\%$ readily be found (see Chapter 14 of *Diff Eqns with Matlab*, by Hunt et al,
 $\%$ for instructions on how to do this in Matlab), and they can be used to
 $\%$ draw all the following phase portraits. However, we shall instead use
 $\%$ Matlab's symbolic solver `dsolve` to achieve the goal.

$\%$ To begin, let us identify the equilibrium points of a linear system. From

```

% formula (**), they are exactly the points X that satisfy
%     AX = 0.
% If A is non-singular, that is, if det(A) is not zero, then only the
% origin X = 0 is an equilibrium point. If det(A) = 0, there are three
% possibilities:
% 1. A = 0. In which case, X' = 0 and every point in the plane is a
%    solution curve. Not a very interesting case.
% 2. A has two distinct eigenvalues, one of which is zero.
% 3. Zero is the only eigenvalue of A, but its eigenspace is
%    1-dimensional.

% We'll draw the phase portraits for cases 2 & 3 after we draw all the
% phase portraits that arise when A is non-singular. Here they are; we
% shall draw a representative phase portrait for each case that arises:

```

a. Two distinct non-zero, real eigenvalues of opposite sign

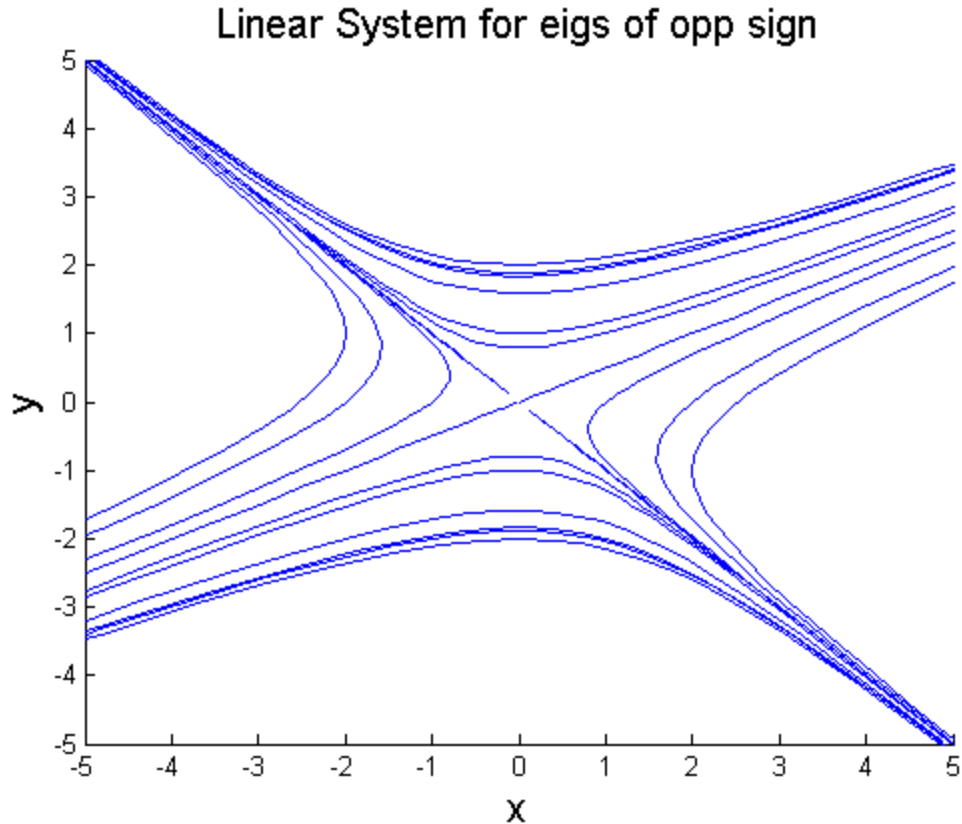
```

% Example:
%     x' = x - 2*y
%     y' = -x.

% Here is the corresponding phase portrait:
clear all
close all
ivp = 'Dx = x - 2*y, Dy = -x, x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for eigs of opp sign', 'FontSize',15)

```



b. Two distinct non-zero, real eigenvalues of the same sign

```

% Example:
%      x' = -3*x + 2*y
%      y' = 2*x - 4*y.

% Here is the corresponding phase portrait:

ivp = 'Dx = -3*x + 2*y, Dy = 2*x - 4*y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

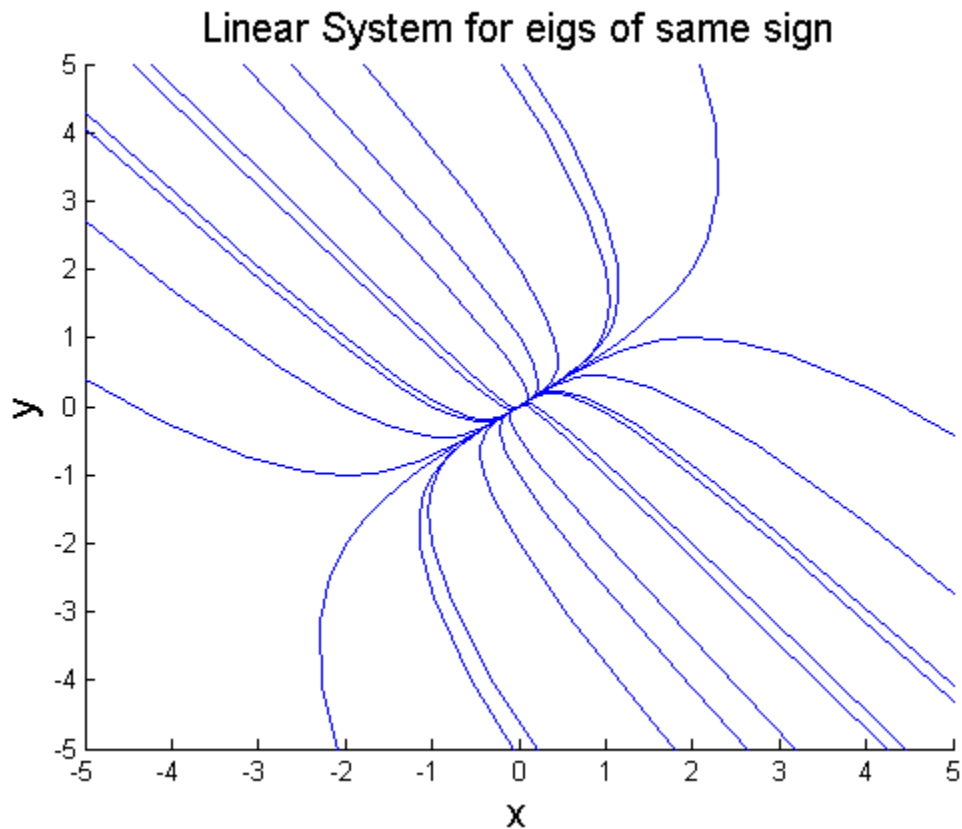
figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'

```

```

ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for eigs of same sign', 'FontSize',15)

```



c. Two complex conjugate purely imaginary eigenvalues

```

% Example:
%      x' = x + 2*y
%      y' = -5*x - y.

% Here is the corresponding phase portrait:

ivp = 'Dx = x + 2*y, Dy = -5*x - y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

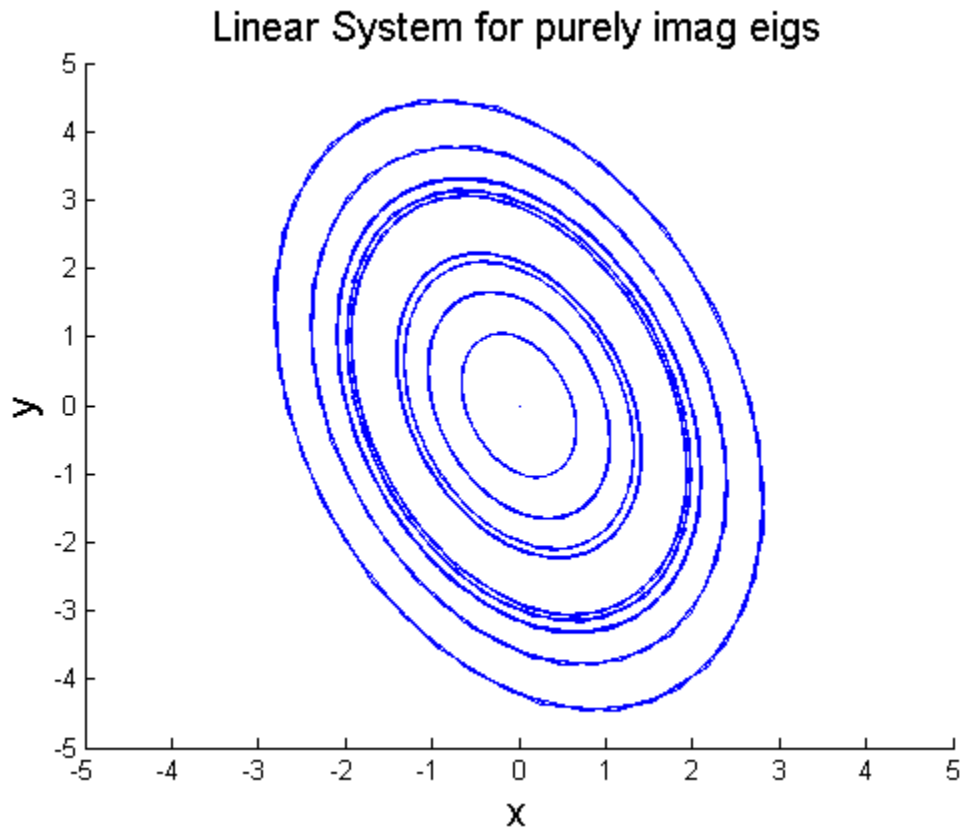
figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end

```

```

    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for purely imag eigs', 'FontSize',15)

```



d. Two complex conjugate eigenvalues with non-zero real part

```

% Example:
%      x' = x + 2*y
%      y' = -5*x - 2*y.

% Here is the corresponding phase portrait:

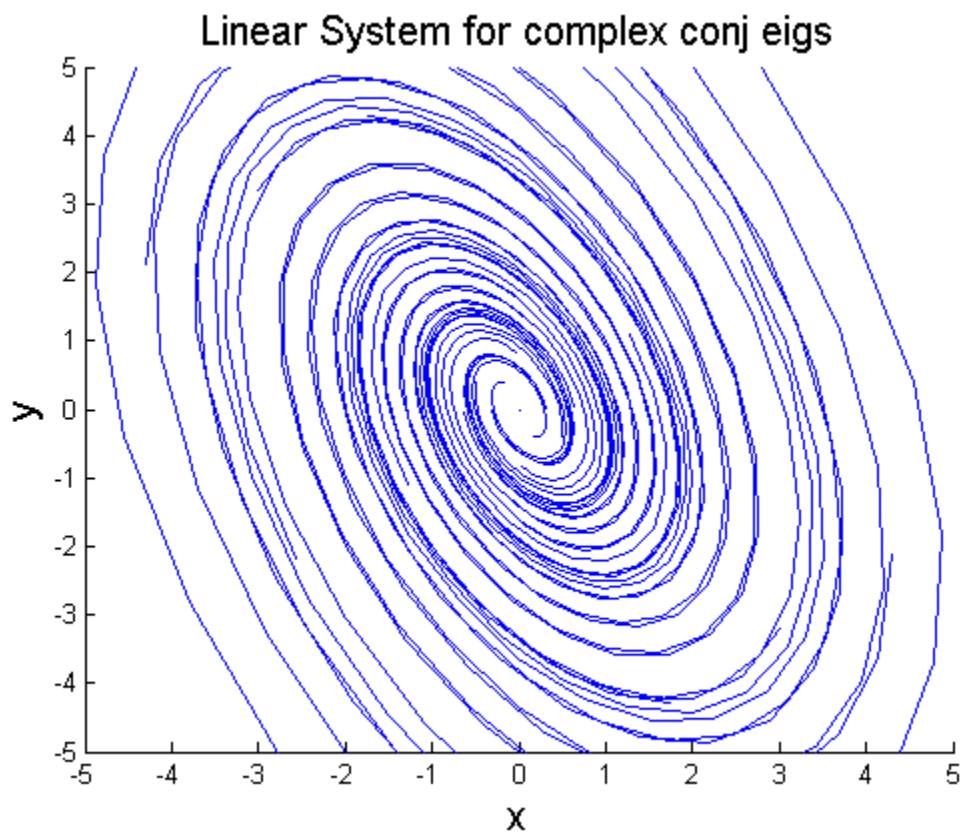
ivp = 'Dx = x + 2*y, Dy = -5*x - 2*y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

```

```

figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for complex conj eigs', 'FontSize',15)

```



e. A single real eigenvalue whose eigenspace is 1-dimensional

```

% Example:
%      x' = x + 3*y
%      y' = -3*x + 7*y.

% Here is the corresponding phase portrait:

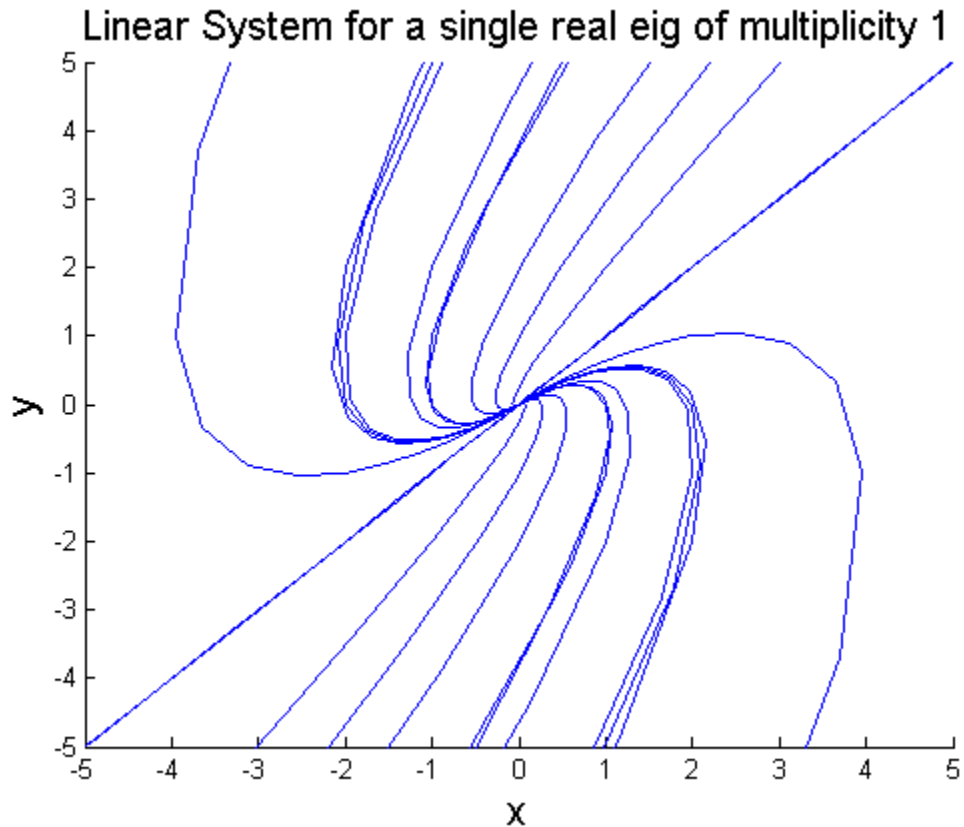
```

```

ivp = 'Dx = x + 3*y, Dy = -3*x + 7*y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for a single real eig of multiplicity 1', 'FontSize',15)

```



f. A single real eigenvalue whose eigenspace is 2-dimensional

```

% Example:
%      x' = 2*x

```

```

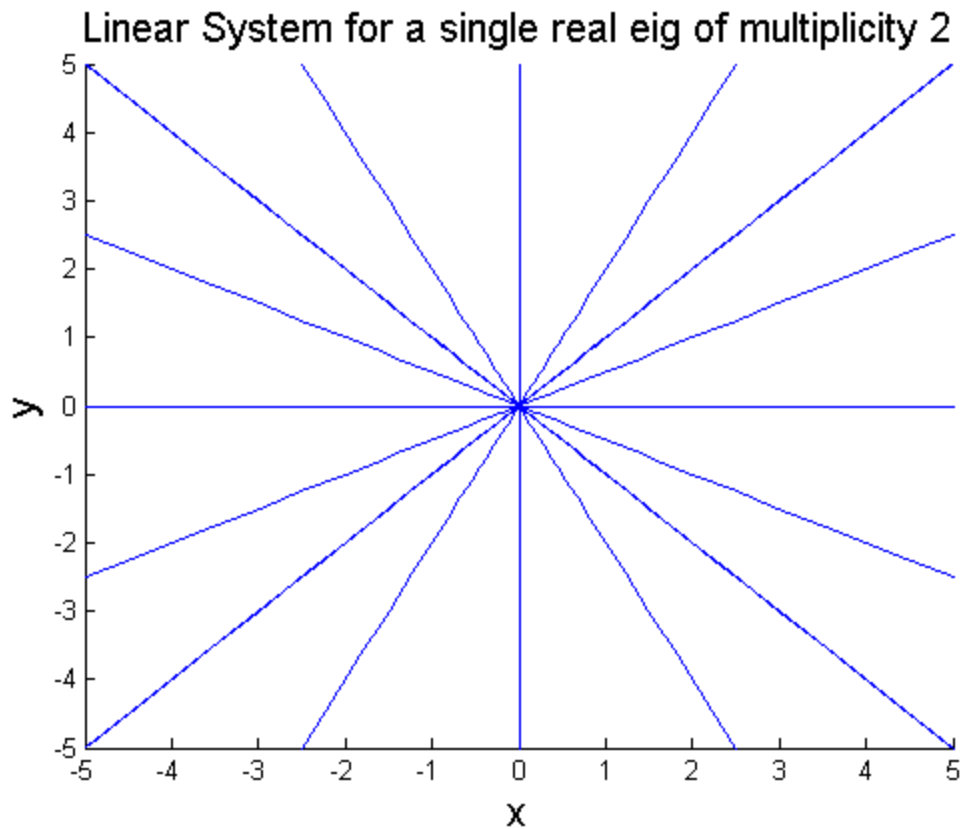
%      y' = 2*y.

% Here is the corresponding phase portrait:

ivp = 'Dx = 2*x, Dy = 2*y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

figure; hold on
t = -2:0.1:2;
for a = -2:2
    for b = -2:2
        plot(xf(t, a, b), yf(t, a, b))
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System for a single real eig of multiplicity 2', 'FontSize',15)

```



Singular Coefficient Matrix

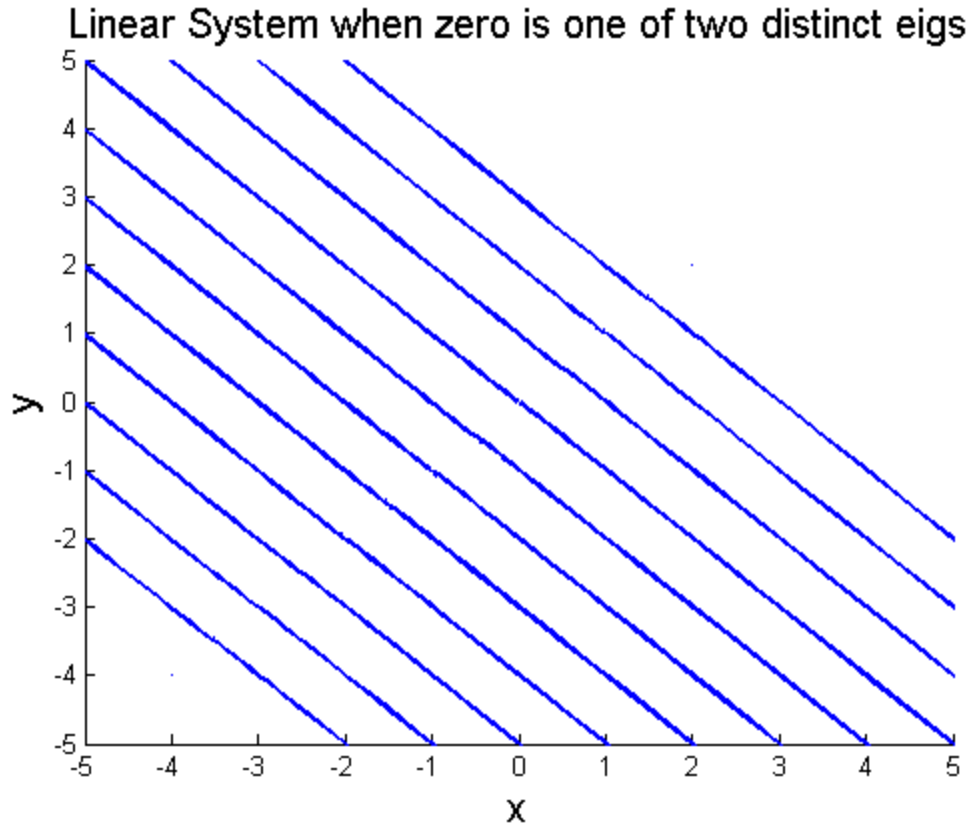
Finally, let's go back and depict the two non-trivial cases that occur when A is singular. First, when A has two distinct real eigenvalues, one of which is zero.

```
% Example:
%      x' = x - y
%      y' = -x + y.

% Here is the corresponding phase portrait:

ivp = 'Dx = x - y, Dy = -x + y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

figure; hold on
t = -2:0.1:2;
for a = -4:2
    for b = -4:2
        plot(xf(t, a, b), yf(t, a, b), 'LineWidth', 2)
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System when zero is one of two distinct eigs', 'FontSize',15)
```



Singular Coefficient Matrix, cont.

And the last case when zero is the only eigenvalue, but it has multiplicity 1.

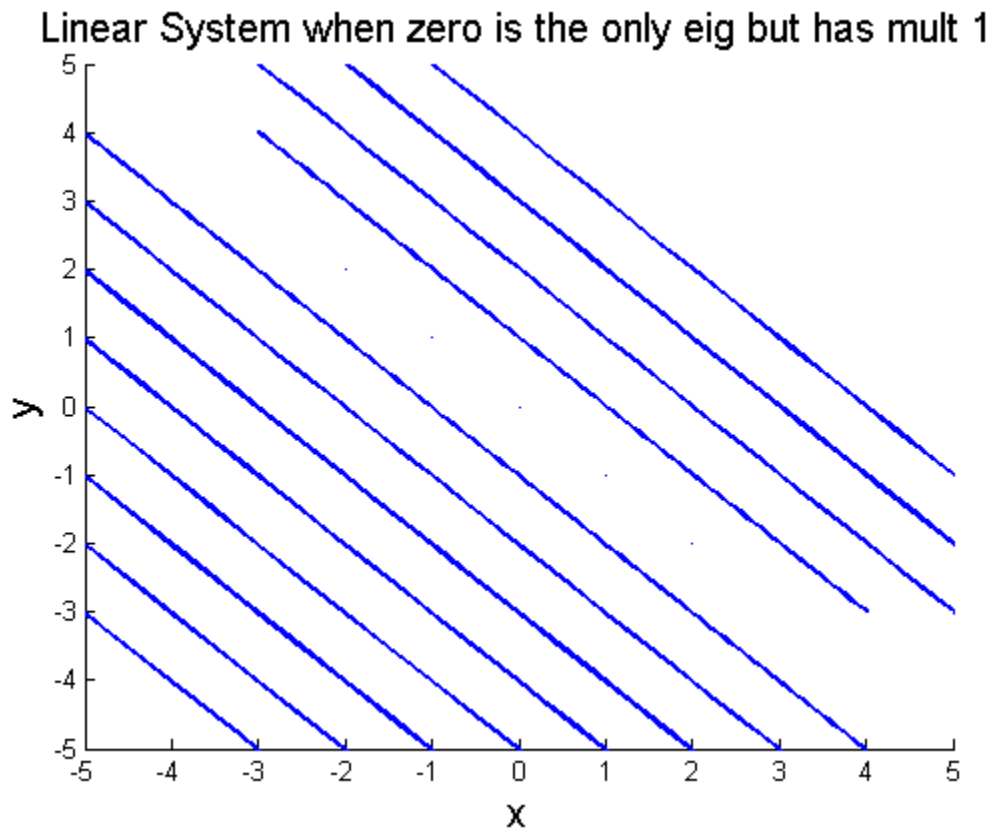
```
% Example:
%      x' = x + y
%      y' = -x - y.

% Here is the corresponding phase portrait:

ivp = 'Dx = x + y, Dy = -x - y , x(0) = a, y(0) = b';
[x, y] = dsolve(ivp, 't');
xf = @(t, a, b) eval(vectorize(x));
yf = @(t, a, b) eval(vectorize(y));

figure; hold on
t = -2:0.1:2;
for a = -4:2
    for b = -4:2
        plot(xf(t, a, b), yf(t, a, b), 'LineWidth', 2)
    end
end
hold off
axis([-5 5 -5 5])
xlabel 'x'
ylabel 'y'
```

```
xlabel ('x', 'FontSize',15)
ylabel ('y', 'FontSize',15)
title ('Linear System when zero is the only eig but has mult 1', 'FontSize',15)
```



The last two pictures look somewhat similar. See if you can explain the difference. Construct the solutions by hand if you need to.

```
% Later in the course we shall see how we may sometimes approximate
% non-linear systems by linear systems. The material here will play a role
% in describing the behavior of nonlinear systems for which we cannot find a
% formula solution.
```

Published with MATLAB® R2013a