Exercise 3-1 Consider the ODE for the pendulum from Ex. 2-5 in the previous homework:

$$\ddot{\theta} = -\sin(\theta)$$

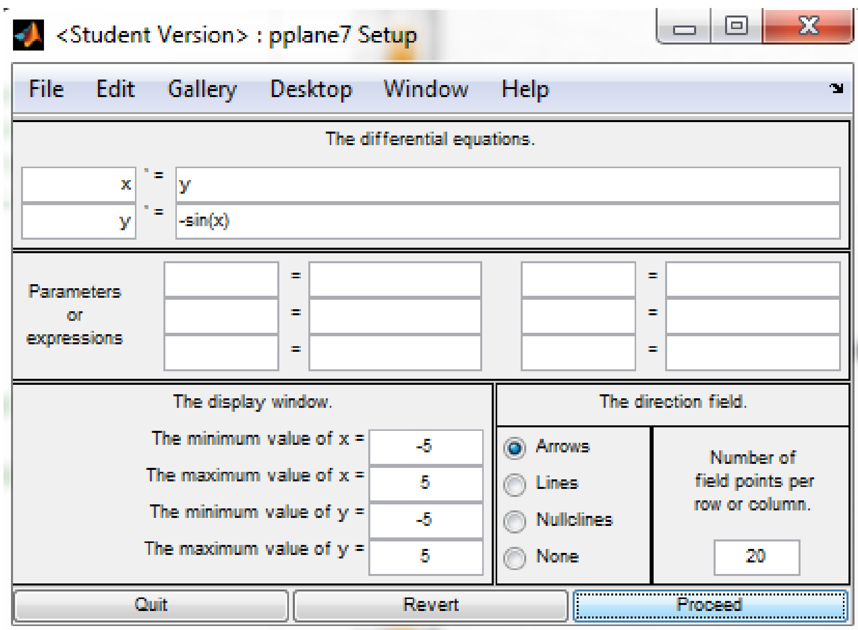
Draw the phase portrait for this ODE (i.e., plot trajectories on the θ vs $\dot{\theta}$ axes).

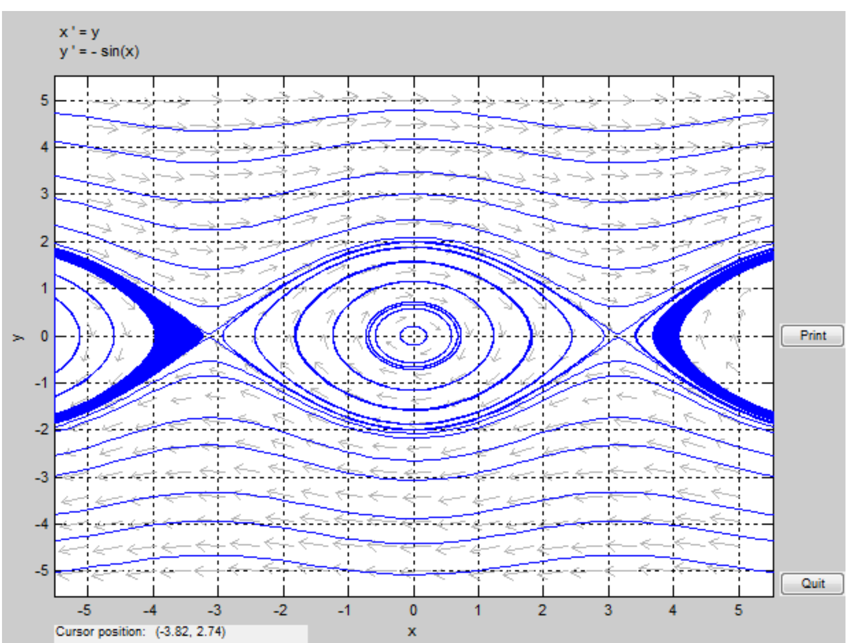
Next, add damping:

$$\ddot{\theta} = -\sin(\theta) - \dot{\theta}$$

Draw the phase portrait for the damped ODE. For both cases, feel free to use `pplane` for help with the phase portraits.

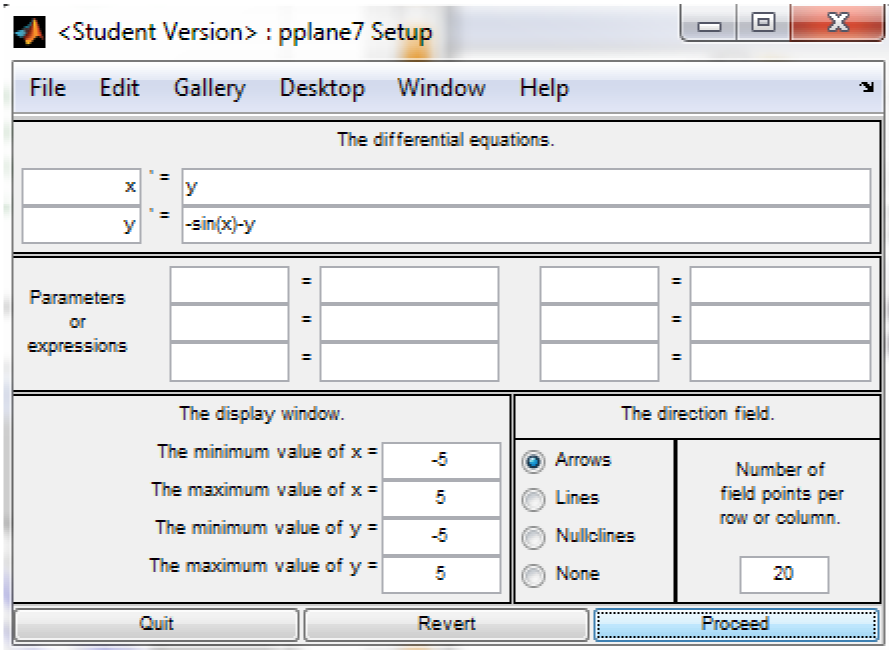
Using pplane in MATLAB for undamped system:

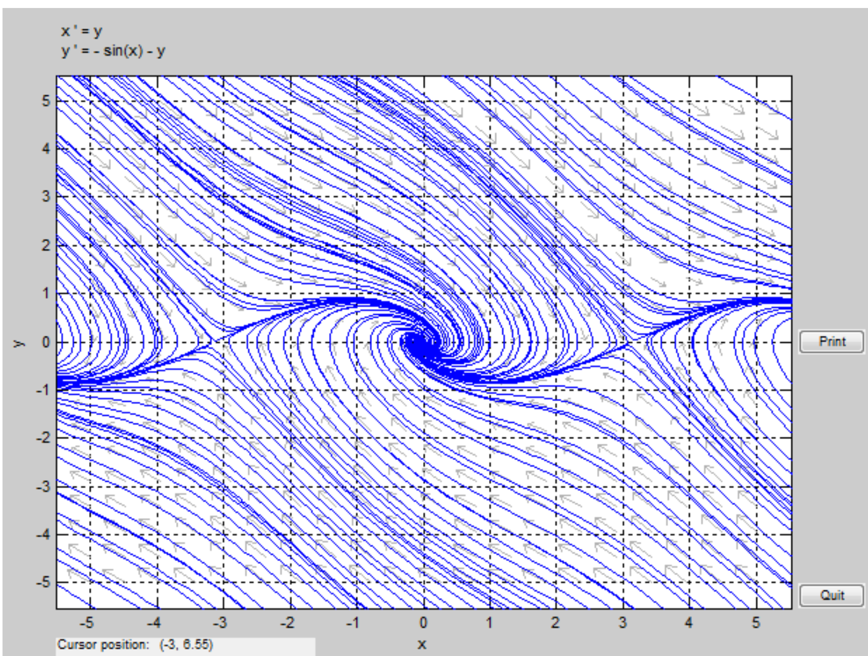




You can easily see the behavior at the two fixed points in the system with a center at (2nπ,0) and saddle points at (nπ,0).

Using pplane for the damped case:





In the damped case you still have the saddle points at (nπ,0), but now there is a spiral point at (2nπ,0).

Exercise 3-2 Consider the differential equation for the pendulum from Ex. 2-5:

$$\ddot{\theta} = -\sin(\theta)$$

Now, we are going to add stabilizing feedback control, assuming that we can measure θ and $\dot{\theta}$:

$$\ddot{\theta} = -\sin(\theta) + \tau$$

where $\tau = -2\dot{\theta} - 2(\theta - \pi)$ is an applied torque at the base of the pendulum.

What is the new stability of the inverted position, when $\theta = \pi$?

So $\ddot{\theta} = -\sin(\theta) + \left(-2\dot{\theta} - 2(\theta - \pi)\right) = -\sin(\theta) - 2\dot{\theta} - 2\theta + 2\pi$

$\dot{\theta} = \omega$

$\omega = -\sin(\theta) - 2\omega - 2\theta + 2\pi$

$\frac{\partial f}{\partial \theta} = \begin{bmatrix} 0 & 1 \\ -\cos(\theta) - 2 & -2 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix}$ $\quad$ *Evaluated at* $(\pi, 0)$ $\quad$ $\frac{\partial f}{\partial \theta} = \begin{bmatrix} 0 & 1 \\ -(-1) - 2 & -2 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix}$

So the A matrix becomes: $\begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}$ *and the eigen values are* $-1, -1$.

Since both of the eigenvalues are real and negative,
we know that the system will be ***STABLE*** *at (π,0).*

Exercise 3-3 Consider the following ODE with very nearly parallel eigenvectors:

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} -0.01 & 0 \\ 1 & -0.011 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}$$

(a) Is the system stable? Please explain why or why not (note that you can do any calculations either by hand or in MATLAB, as long as you explain what you are doing and why).

(b) Compute the solution of this ODE using MATLAB (`ode45`) for the initial condition $\begin{bmatrix} 0.1 \\ 0 \end{bmatrix}$. Plot the trajectory in the x-v plane. Does the solution make sense given your answer to part (a)?

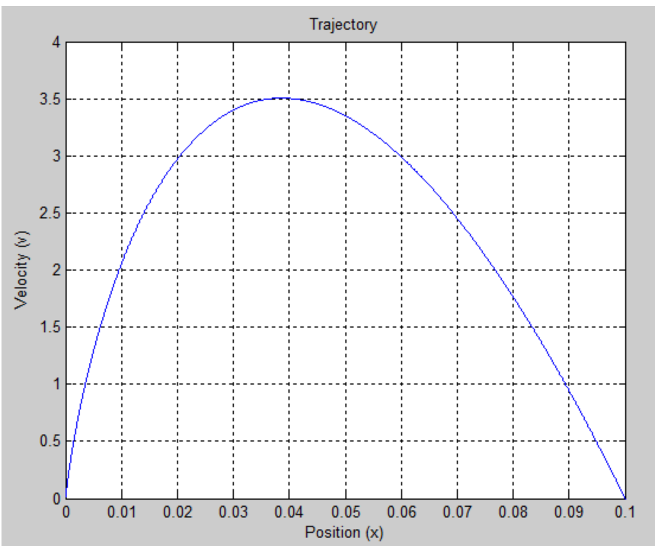
(c) If v is velocity, so kinetic energy is $T = \frac{1}{2}v^2$, plot the kinetic energy look like in time?

This example illustrates the phenomena of **transient energy growth** experienced by stable systems that have nearly parallel eigenvectors corresponding to very similar eigenvalues. These systems are called non-normal (i.e., $AA^T \neq A^TA$). This is common to many systems, and is responsible for the transition from a laminar solution to a turbulent solution in many fluid systems (because the transient energy growth takes the fluid far enough away from the stable fixed point so that nonlinearities become large and develop into turbulence).

a) *Since* $A = \begin{bmatrix} -0.01 & 0 \\ 1 & -0.011 \end{bmatrix}$ *the eigenvalues are* -0.01 *and* -0.011

Since both of the eigenvalues are real and negative, we know that the system will be ***STABLE***.

b)



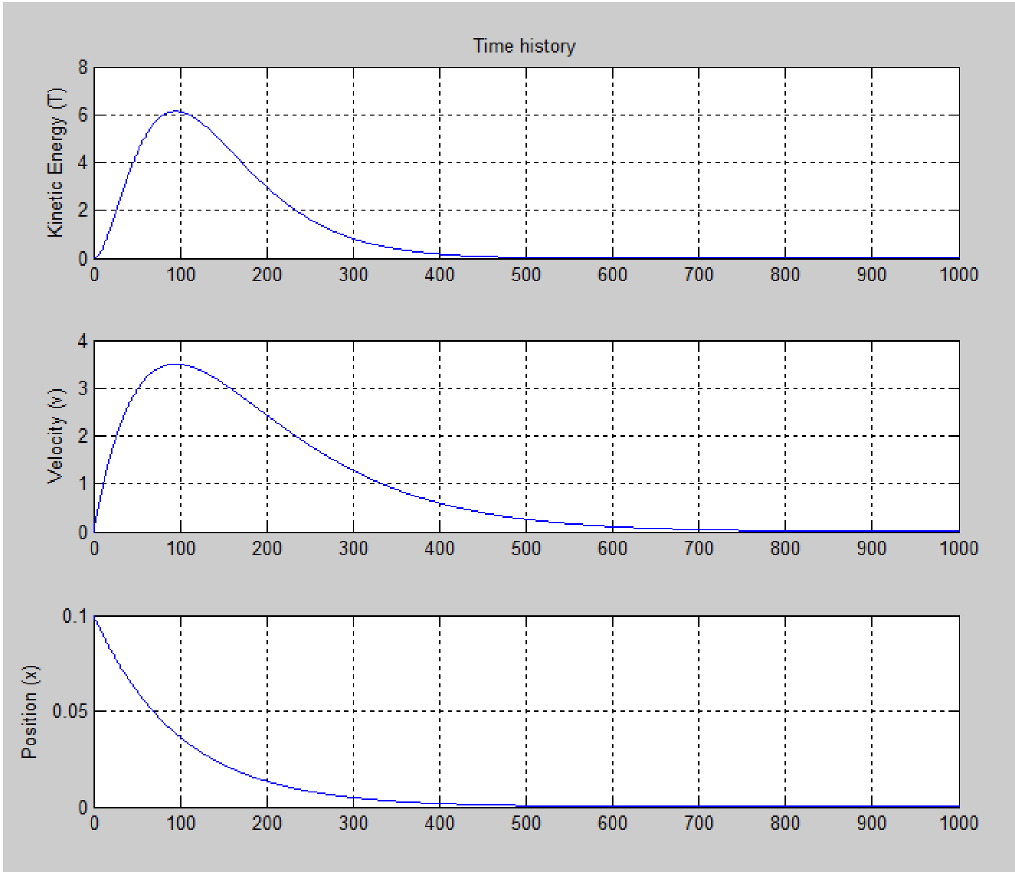

```
time = linspace(0,1000,10001);
A = [-0.01 0; 1 -0.011];
x0 = [0.1 0]';

% Run ode45
[t,x] = ode45(@(t,x) A*x,time,x0);

figure(3), clf
plot(x(:,1),x(:,2))
grid on
xlabel('Position (x)')
ylabel('Velocity (v)')
title('Trajectory')
```

Yes, the solution makes sense given that we found that the function would be stable which matches with the trajectory returning to zero.

c)




```
% Plot K.E vs Time
T = 0.5.*(x(:,2).*x(:,2));
figure(4), clf
subplot(311)
plot(t,T)
grid on
title('Time history')
ylabel('Kinetic Energy (T)')
subplot(312)
plot(t,x(:,2))
grid on
ylabel('Velocity (v)')
subplot(313)
plot(t,x(:,1))
grid on
ylabel('Position (x)')
```

Note: Although the position and velocity graphs help give a better understanding of the system, a single plot of kinetic energy vs. time would have been sufficient for part c.