AMATH301_Homework6_writeup

February 25, 2023

1 Homework 6 writeup solutions

1.1 Name: Aqua Karaman

1.2 Problem 1

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize
import scipy.integrate

##################### Coding Problem 3 ##########################
# Define the parameters we are going to use:
g = 9.8
L = 11
sigma = 0.12

## Part a
# We now have two ODEs. Define them below as anonymous functions!
dtheta = lambda v: v
dv = lambda theta, v: -g/L * np.sin(theta) - sigma * v

# And initial conditions. This is just like Problem 2 above.
theta0 = -np.pi/8
v0 = -0.1

## Part b
# Test your anonymous function.
odefun = lambda t, p: np.array([p[1],
                                -g/L * np.sin(p[0]) - sigma * p[1]])

A12 = odefun(1, [2, 3])
print(A12)

## Part c
# Solve!
t_span = np.arange(0.,50.+0.01,0.01)
sol = scipy.integrate.solve_ivp(odefun, [0, 50], [theta0, v0], t_eval=t_span)
```

```
print(sol)
A13 = sol.y
print(A13)
```

```
[ 3.         -1.17010134]
  message: 'The solver successfully reached the end of the integration
interval.'
     nfev: 320
     njev: 0
      nlu: 0
      sol: None
   status: 0
  success: True
        t: array([0.000e+00, 1.000e-02, 2.000e-02, …, 4.998e+01, 4.999e+01,
       5.000e+01])
 t_events: None
        y: array([[-0.39269908, -0.39368143, -0.39462844, …,  0.01846549,
         0.01853613,  0.01860504],
       [-0.1       , -0.09646869, -0.09293368, …,  0.00715122,
         0.00697793,  0.00680422]])
 y_events: None
[[-0.39269908 -0.39368143 -0.39462844 …  0.01846549  0.01853613
   0.01860504]
 [-0.1        -0.09646869 -0.09293368 …  0.00715122  0.00697793
   0.00680422]]
```

1.2.1 Part a - Create a meshgrid

You don't need to go through all of these individual parts if you want, it's just to help you organize your work.

[ ]:
```
# Define the theta values
theta_vals = sol.y[0]
# Define the v values
v_vals = sol.y[1]
# Create the mesh
theta_span = np.linspace(-3*np.pi, 3*np.pi, 25)
v_span = np.linspace(-3, 3, 25)
T, V = np.meshgrid(theta_span, v_span)
```

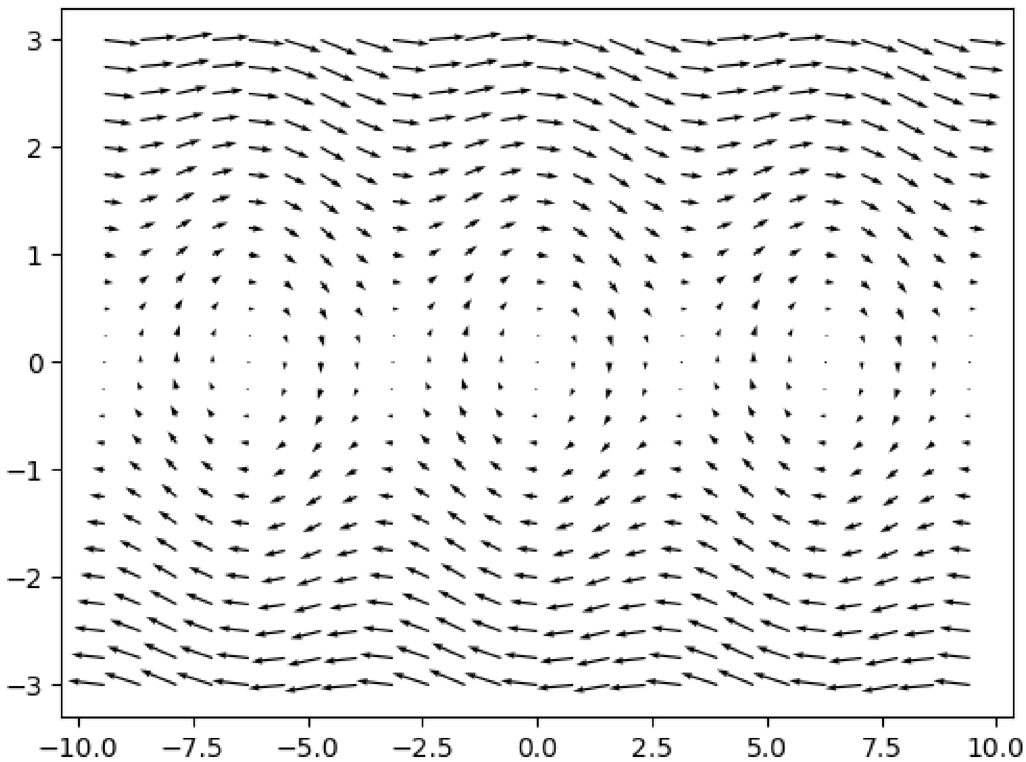
1.2.2 Part b - Create a quiver plot

[ ]:
```
fig, ax = plt.subplots()

dt = 0.01

ax.quiver(T, V, V, dv(T, V))
```

```
[ ]: <matplotlib.quiver.Quiver at 0x216f6f72940>
```



1.2.3 Part c - Label the axes. Here you should have the complete quiver plot.

```
[ ]: ax.set_xlabel('$ (t)$')
     ax.set_ylabel('$v(t)$')
     ax.set_title("Trajectories and vector field of  $\dfrac{dv}{dt} =␣
      ↪-\dfrac{g}{L}\sin{ (t)} -  v(t)$")
```

```
[ ]: Text(0.5, 1.0, 'Trajectories and vector field of  $\\dfrac{dv}{dt} =
     -\\dfrac{g}{L}\\sin{ (t)} -  v(t)$')
```

1.2.4 Part d - Include trajectories.

```
[ ]: t_span = np.arange(0.,50.+0.01,0.01)
     print(t_span)
     sol1 = scipy.integrate.solve_ivp(odefun, [0,50], [np.pi, 0.1], t_eval=t_span)
     sol2 = scipy.integrate.solve_ivp(odefun, [0,50], [np.pi, -0.1], t_eval=t_span)
     sol3 = scipy.integrate.solve_ivp(odefun, [0,50], [2*np.pi, -3], t_eval=t_span)
     sol4 = scipy.integrate.solve_ivp(odefun, [0,50], [-2 * np.pi, 3], t_eval=t_span)

     ax.plot(sol1.y[0], sol1.y[1], '-b')
```

```
ax.plot(sol2.y[0], sol2.y[1], '-r')
ax.plot(sol3.y[0], sol3.y[1], '-k')
ax.plot(sol4.y[0], sol4.y[1], '-m')
```

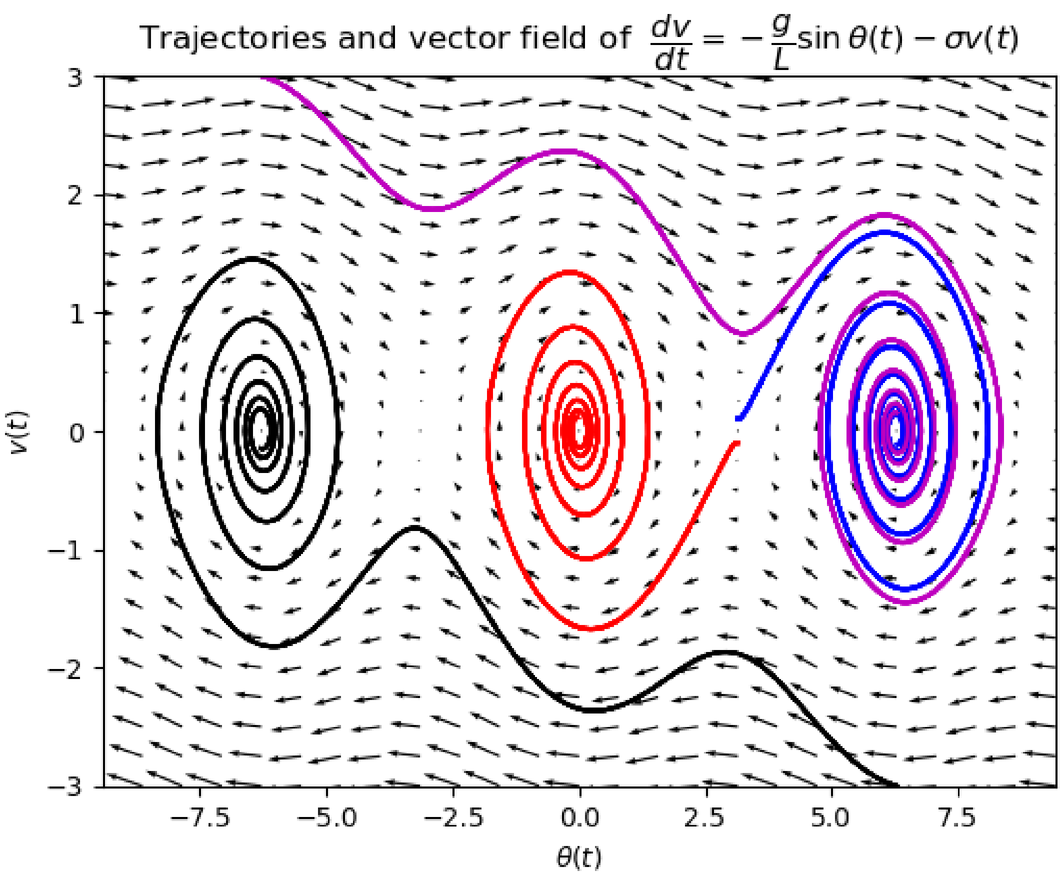
```
[0.000e+00 1.000e-02 2.000e-02 … 4.998e+01 4.999e+01 5.000e+01]
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x216f743ffd0>]
```

1.2.5 Part e - axis display

[ ]:
```
# If you name your axis "ax", then this
# scales the axes: (uncomment both lines)
ax.set_xlim([-3*np.pi, 3*np.pi])
ax.set_ylim([-3, 3])
fig
```

[ ]:



1.2.6 Part f - Discussion

Part (i) - long-term behavior In terms of the pendulum, the physical scenario achieved as $t \to \infty$ is oscillation—in other words, as $t \to \infty$, the pendulum begins to fluctuate between (on an open interval) -2π and @2 \$. This is due to our simulation of friction/air resistance that slows the pendulum down over time.

Part (ii) - Comparing two solutions with $\theta_0 = \pi$. The plots of trajectories I and II shown in the graph are horizontally and vertically mirrored, horizontally translated images of each other; that is, the hypothetical pendula these trajectories represent experience the same exact motion, just in the opposite direction. Trajectory I (blue) starts spinning off clockwise, whereas Trajectory II (red) begins counterclockwise.

Part (iii) - Comparing two solutions with equal and opposite θ_0 and v_0. Similarly to I and II, trajectories III and IV are horiz. and vert. mirrors of each other, around the point (0, 0) specifically. The reason these graphs have the trailing curve on their sides is due to the speed of the hypothetical pendula they represent—the velocities of the pendula are at first fast enough to keep going around their centers; eventually, however, the simulated friction slows them to a point where their centrifugal force isn't stronger than the force of gravity on them, and they are pulled down and begin to oscillate under their centers instead.