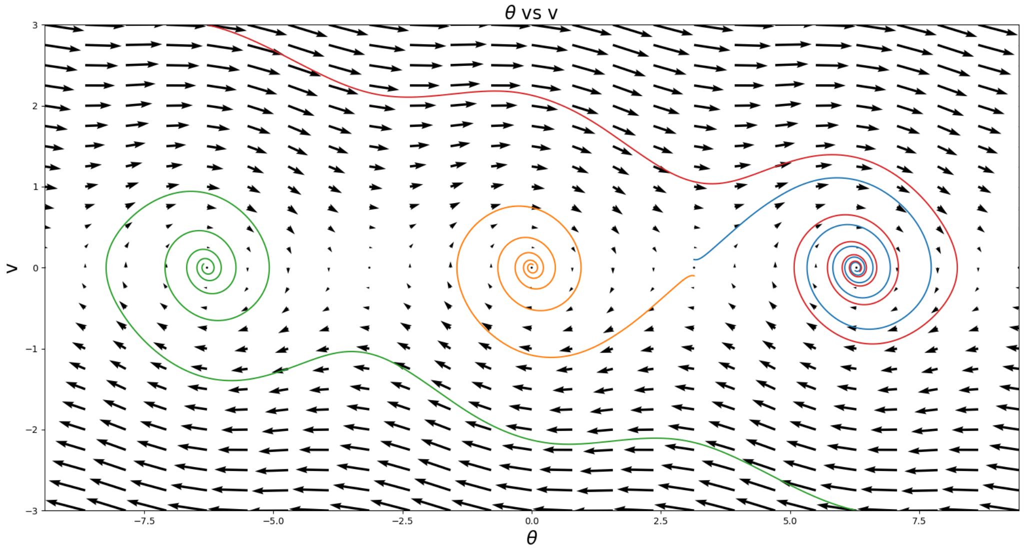
Homework 7 Writeup

Christian Diangco
AMATH 301 B

Problem 1

For each solution, as time increases, the pendulum swings slower and slower until its velocity eventually reaches zero.
Let's say velocity is positive going clockwise. When the initial conditions are (pi, 0.1), the pendulum starts from the top and starts swinging clockwise with a velocity of 0.1. When the initial conditions are (pi, -0.1), the pendulum also starts from the top, but starts swinging counter-clockwise with a velocity of -0.1.
When the initial conditions are (2pi, -3), the pendulum starts from the bottom and starts swinging counter-clockwise with a velocity of -3. When the initial conditions are (-2pi, 3), the pendulum also starts from the bottom, but starts swinging clockwise with a velocity of 3.

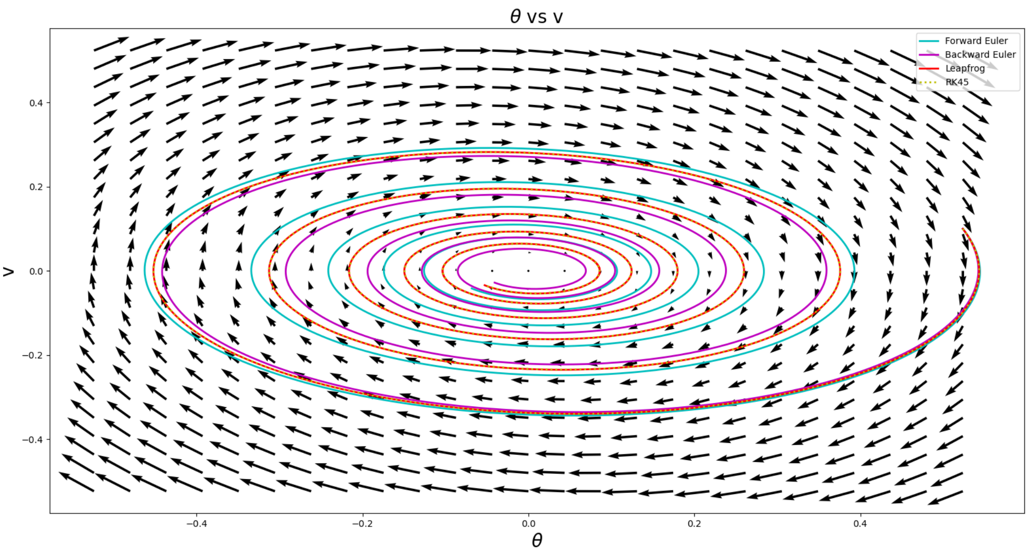



```
theta, v = np.meshgrid(np.linspace(-3*np.pi, 3*np.pi, 25), np.linspace(-3, 3, 25))
g = 9.8
L = 21
o = 0.15
dt = 0.01
tspan = np.arange(0, 50+dt, dt)
dthetadt = lambda theta, v: v
dvdt = lambda theta, v: -g / L * np.sin(theta) - o*v
odefun = lambda t, y: np.array([dthetadt(y[0], y[1]), dvdt(y[0], y[1])])
traj1 = scipy.integrate.solve_ivp(odefun, np.array([tspan[0], tspan[-1]]), np.array([np.pi, 0.1]), t_eval = tspan)
traj2 = scipy.integrate.solve_ivp(odefun, np.array([tspan[0], tspan[-1]]), np.array([np.pi, -0.1]), t_eval = tspan)
traj3 = scipy.integrate.solve_ivp(odefun, np.array([tspan[0], tspan[-1]]), np.array([2*np.pi, -3]), t_eval = tspan)
traj4 = scipy.integrate.solve_ivp(odefun, np.array([tspan[0], tspan[-1]]), np.array([-2*np.pi, 3]), t_eval = tspan)

fig, ax = plt.subplots()
ax.quiver(theta, v, dthetadt(theta, v), dvdt(theta, v))
ax.set_xlabel("$\\theta$", fontsize=20)
ax.set_ylabel("v", fontsize=20)
ax.set_title("$\\theta$ vs v", fontsize=20)
ax.plot(traj1.y[0,:], traj1.y[1,:])
ax.plot(traj2.y[0,:], traj2.y[1,:])
ax.plot(traj3.y[0,:], traj3.y[1,:])
ax.plot(traj4.y[0,:], traj4.y[1,:])
ax.set_xlim(-3*np.pi, 3*np.pi)
ax.set_ylim(-3, 3)
```

Problem 2

The most accurate method is **the Leapfrog Method**. We can see this from the plot because the red line (Leapfrog) follows the yellow dashed line (RK45) the closest.
The error **increases** over time. From the plot we can see that with each pass over the equilibrium position, the lines get further and further apart from the RK45 line, which we are assuming to be the exact solution. The only one that doesn't appear to move away is the Leapfrog line, but that could be due to being too zoomed out.




```
theta, v = np.meshgrid(np.linspace(-np.pi/6, np.pi/6, 25), np.linspace(-np.pi/6, np.pi/6, 25))
g = 9.8
L = 21
o = 0.08
dt = 0.02
theta0 = np.pi / 6
v0 = 0.1
A = np.array([
    [0, 1],
    [-g/L, -o]
])
tspan = np.arange(0, 50+dt, dt)
dthetadt = lambda theta, v: v
dvdt = lambda theta, v: -g / L * theta - o*v

FE_sol = np.zeros([2, len(tspan)])
FE_sol[0,0] = theta0
FE_sol[1,0] = v0
for k in range(len(tspan) - 1):
    FE_sol[:, k+1] = FE_sol[:, k] + dt*A@FE_sol[:, k]

BE_sol = np.zeros([2, len(tspan)])
BE_sol[0,0] = theta0
BE_sol[1,0] = v0
LU, P = scipy.linalg.lu_factor(np.identity(2) - dt*A)
for k in range(len(tspan) - 1):
    BE_sol[:,k+1] = scipy.linalg.lu_solve((LU, P), BE_sol[:,k])

LF_sol = np.zeros([2, len(tspan)])
LF_sol[0,0] = theta0
LF_sol[1,0] = v0
LF_sol[:,1] = LF_sol[:,0] + dt*A@LF_sol[:,0]
for k in range(1, len(tspan) - 1):
    LF_sol[:,k+1] = LF_sol[:,k-1] + 2*dt*A@LF_sol[:,k]

RK = scipy.integrate.solve_ivp(lambda t, v: A@v, np.array([tspan[0], tspan[-1]]),
                                   np.array([theta0, v0]), t_eval=tspan)
```

```
RK_sol = RK.y

fig2, ax2 = plt.subplots()
ax2.quiver(theta, v, dthetadt(theta, v), dvdt(theta, v))
ax2.set_xlabel("$\\theta$", fontsize=20)
ax2.set_ylabel("v", fontsize=20)
ax2.set_title("$\\theta$ vs v", fontsize=20)
ax2.plot(FE_sol[0,:], FE_sol[1,:], "c", linewidth=2, label="Forward Euler")
ax2.plot(BE_sol[0,:], BE_sol[1,:], "m", linewidth=2, label="Backward Euler")
ax2.plot(LF_sol[0,:], LF_sol[1,:], "r", linewidth=2, label="Leapfrog")
ax2.plot(RK_sol[0,:], RK_sol[1,:], "y:", linewidth=2, label="RK45")
ax2.legend()
```