```python
# 24-677 Linear Control Systems
# Homework 5 Exercise 5
# Ryan Wu (weihuanw)

import matplotlib.pyplot as plt
from scipy.integrate import odeint
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

# define non linear state space function
def stateSpace(x, t):
    d_dot = [x[1] - x[0] * (x[1] * x[1]), -x[0] * x[0] * x[0]]
    return d_dot

# # define linear state space function
# def stateSpace(x, t):
#     A = np.array([[0, 1], [0, 0]])
#     return np.dot(A, x)

# grid setup
x0 = np.linspace(-1, 1, 30)
x1 = np.linspace(-1, 1, 30)
X0, X1 = np.meshgrid(x0, x1)

dX0 = np.zeros(X0.shape)
dX1 = np.zeros(X1.shape)

shape1, shape2 = X1.shape

# looping through each index
for indexShape1 in range(shape1):
    for indexShape2 in range(shape2):
        dxdt = stateSpace([X0[indexShape1, indexShape2], X1[indexShape1, indexShape2]], 0)
        dX0[indexShape1, indexShape2] = dxdt[0]
        dX1[indexShape1, indexShape2] = dxdt[1]

# phase trajectory lines
initialState = np.array([0, 0])
simulationStep = np.linspace(0, 2, 200)
```

```python
finalState = odeint(stateSpace, initialState, simulationStep)



# define three dimension function
def threeDimension(x1_3d, x2_3d):
    v_dot = -4 * x1_3d**4 * x2_3d**2
    return v_dot

x1_3d = np.linspace(-2, 2, 100)
x2_3d = np.linspace(-2, 2, 100)

x1_3d, x2_3d = np.meshgrid(x1_3d, x2_3d)
v_dot = threeDimension(x1_3d, x2_3d)

# plot and figure features (Phase Portraits)
plt.figure(figsize=(10, 8))
plt.quiver(X0, X1, dX0, dX1, color='g')
plt.plot(0, 0, marker='o', color='r')
plt.plot(finalState[:, 0], finalState[:, 1])
plt.xlim(-1, 1)
plt.ylim(-1, 1)
plt.title('Non Linear Phase Portrait Plot', fontsize=20)
# plt.title('Linear Phase Portrait Plot', fontsize=20) # for linear case
plt.xlabel('$x_{1}$', fontsize=14)
plt.ylabel('$x_{2}$', fontsize=14)
plt.savefig('NonlinerPhasePortraitPlot.png')
# plt.savefig('linearPhasePortraitPlot.png') # for linear case
plt.show()

# plot and figure features (3 Dimensional)
fig = plt.figure(figsize=(10, 8))
d_plot = fig.add_subplot(111, projection='3d')
d_plot.plot_surface(x1_3d, x2_3d, v_dot, cmap='viridis')
d_plot.set_xlabel('$x_{1}$')
d_plot.set_ylabel('$x_{2}$')
```

```
d_plot.set_zlabel('V*')
d_plot.set_title('3D Variation Plot')
plt.savefig('3D plot.png')
plt.show()

```