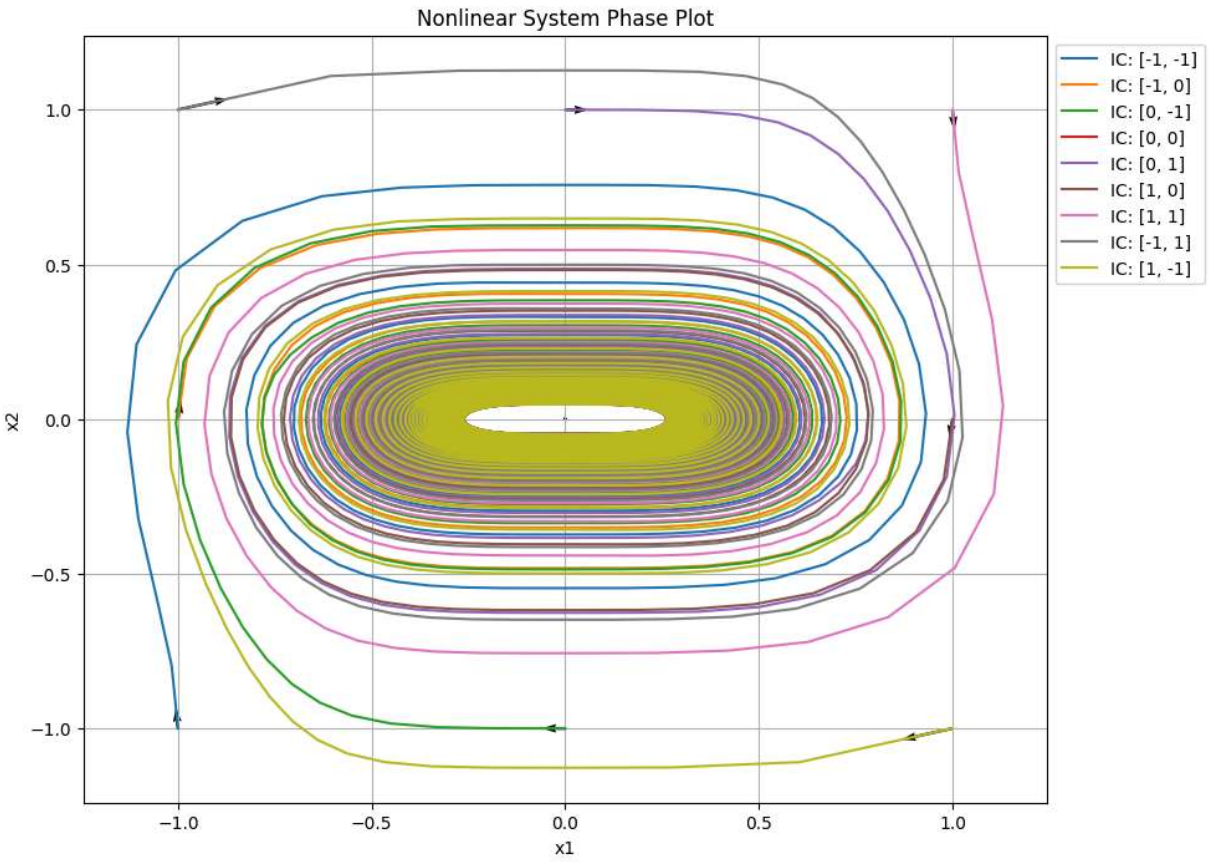
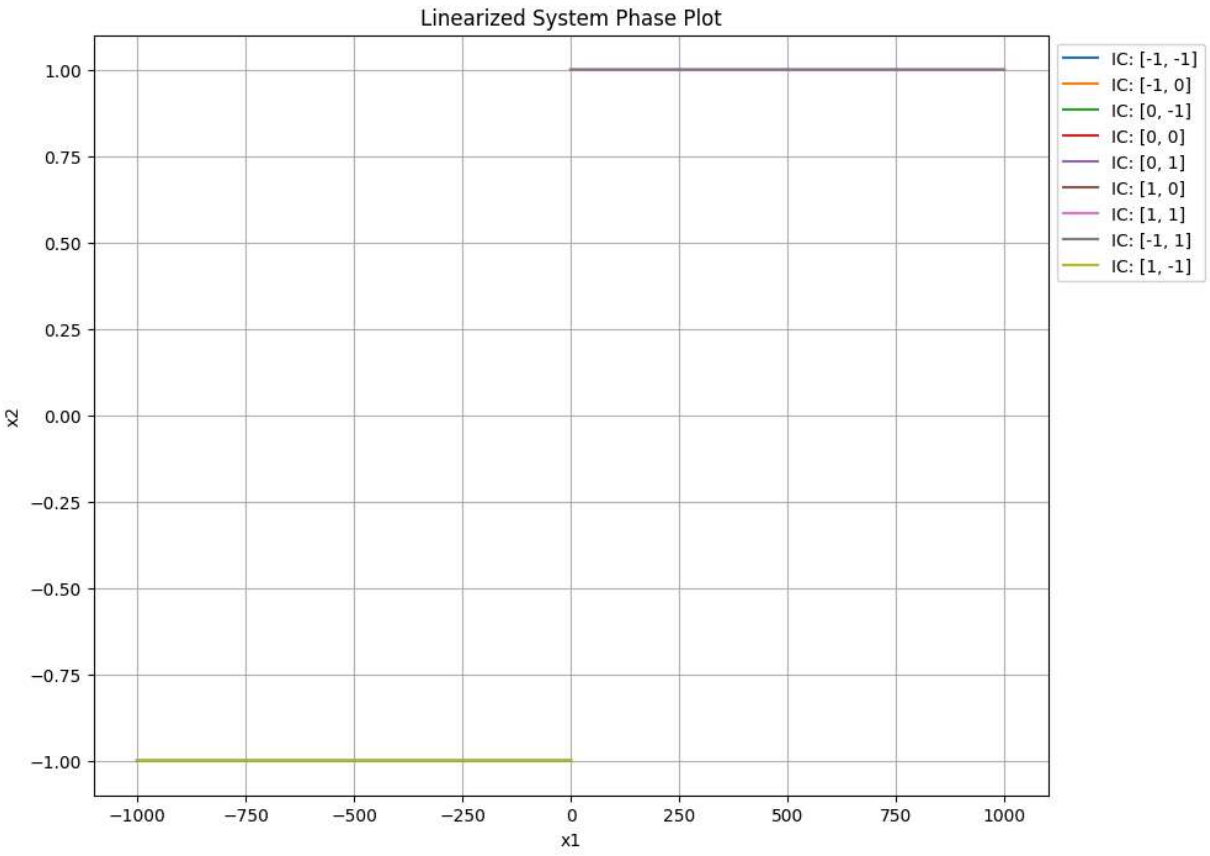
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

def nonlinear_dyn(t, x):
    dx1dt = x[1] - x[0] * (x[1] ** 2)
    dx2dt = - x[0] ** 3

    return [dx1dt, dx2dt]

t_span = (0, 1000)
t = np.linspace(0, 1000, 5001)

init_cond = [[-1, -1], [-1, 0], [0, -1], [0, 0], [0, 1], [1, 0], [1, 1], [-1, 1], [
```

In [2]:
```
plt.figure(figsize = (10, 8))
for ic in init_cond:
    sol = solve_ivp(nonlinear_dyn, t_span, ic, t_eval = t)
    plt.plot(sol.y[0], sol.y[1], label = f'IC: {ic}')
    for i in range(0, len(sol.t) - 1, 500):
        plt.quiver(sol.y[0][i], sol.y[1][i], sol.y[0][i+1] - sol.y[0][i], sol.y[1][

plt.title('Nonlinear System Phase Plot')
plt.xlabel('x1')
plt.ylabel('x2')
plt.grid()
plt.legend(loc = 'best', bbox_to_anchor=(1, 1))
plt.show()
```



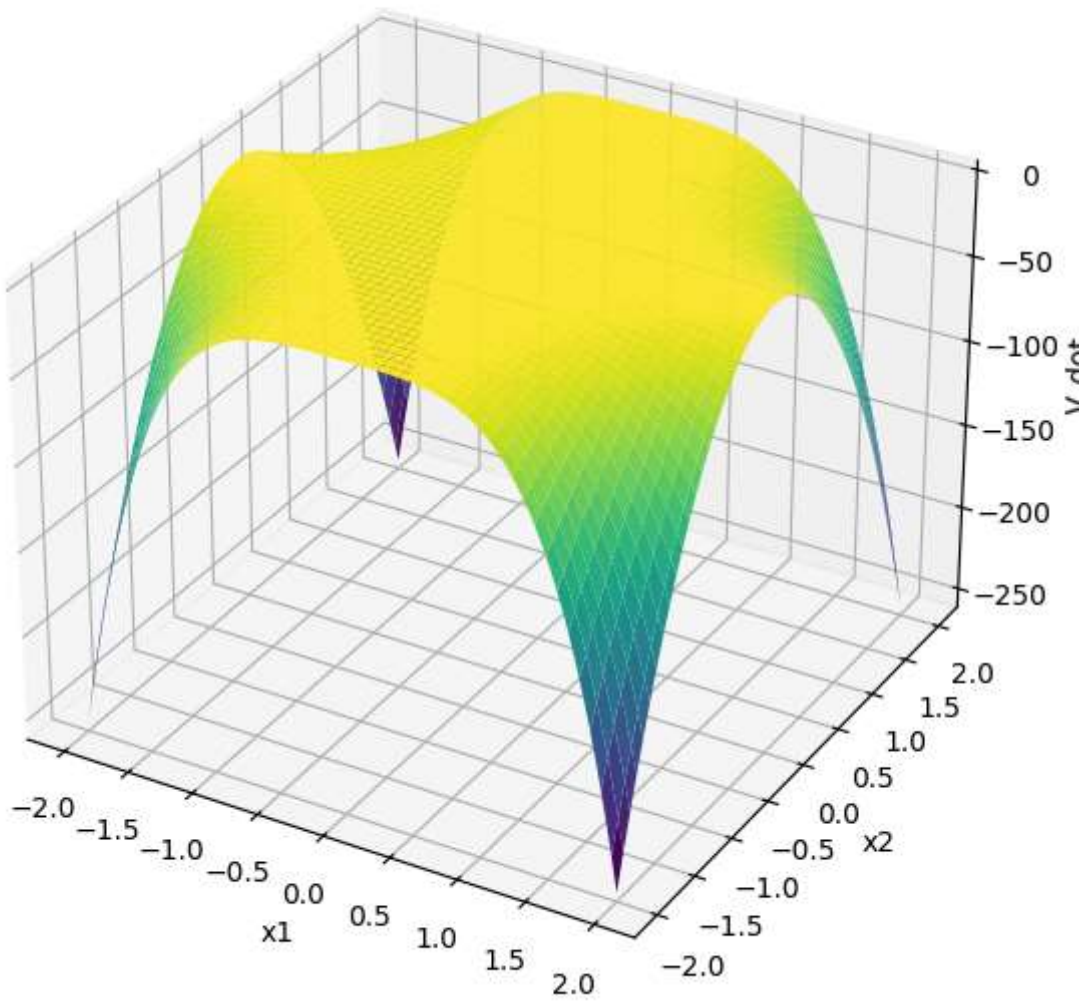
In [3]:
```
A = np.array([[0, 1], [0, 0]])

def linear_dyn(t, x):
    return A @ x
```

In [4]:
```
plt.figure(figsize = (10, 8))
for ic in init_cond:
    sol = solve_ivp(linear_dyn, t_span, ic, t_eval = t)
    plt.plot(sol.y[0], sol.y[1], label=f'IC: {ic}')

plt.title('Linearized System Phase Plot')
plt.xlabel('x1')
plt.ylabel('x2')
plt.grid()
plt.legend(loc = 'best', bbox_to_anchor=(1, 1))
plt.show()
```



In [5]:
```python
from mpl_toolkits.mplot3d import Axes3D

def dV(x1, x2):
    dv = 4 * x1**3 * (x2 - x1 * x2**2) + 4 * x2 * (-x1**3)
    return dv

x1 = np.linspace(-2, 2, 100)
x2 = np.linspace(-2, 2, 100)
X1, X2 = np.meshgrid(x1, x2)

V_dot = dV(X1, X2)
```

In [6]:
```python
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X1, X2, V_dot, cmap='viridis', edgecolor='none')

ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_zlabel('V_dot')
ax.set_title('Variation of V_dot with respect to x1 and x2')

plt.show()
```


Variation of V_dot with respect to x1 and x2
0
−50
−100
−150
−200
−250
V_dot
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
x2