q5

October 15, 2020

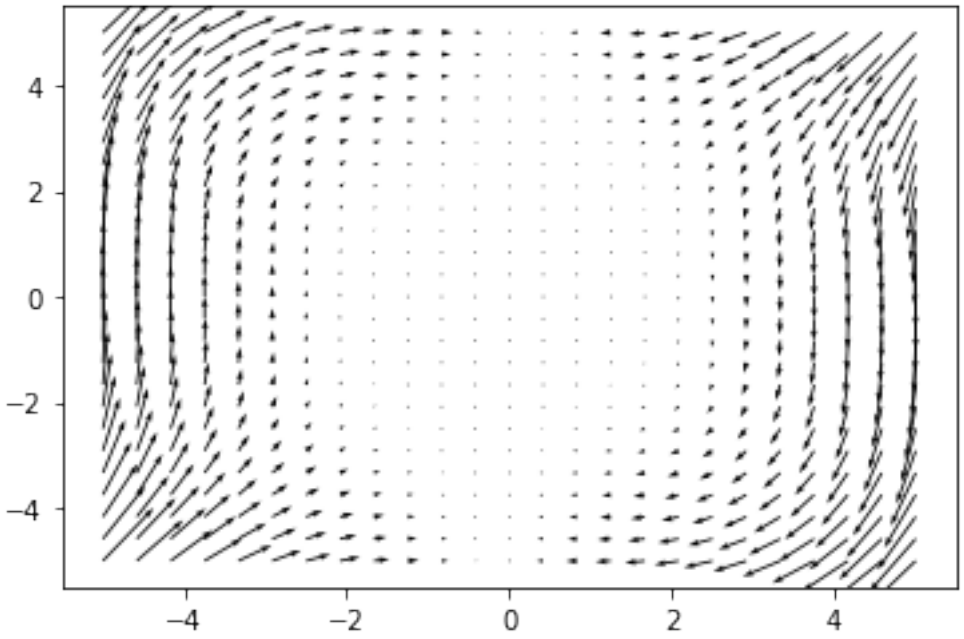
[63]:
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

[50]:
```
def sys(X):
    x1, x2 = X
    return [x2 - x1*np.power(x2,2), -np.power(x1,3)]
def linsys(X):
    x1, x2 = X
    return [-x2, 0]
```

[51]:
```
def phase_portrait2D(f, xmin=-5, xmax=5, ymin=-5, ymax=5, num=25):
    x1 = np.linspace(xmin,xmax, num=num)
    x2 = np.linspace(ymin,ymax, num=num)
    x1, x2 = np.meshgrid(x1,x2)
    u, v = np.zeros(x1.shape), np.zeros(x2.shape)
    Nr, Nc = x1.shape
    for r in range(Nr):
        for c in range(Nc):
            x = x1[r,c]
            y = x2[r,c]
            xd = f([x,y])
            u[r,c] = xd[0]
            v[r,c] = xd[1]
    Q = plt.quiver(x1,x2, u,v)
```

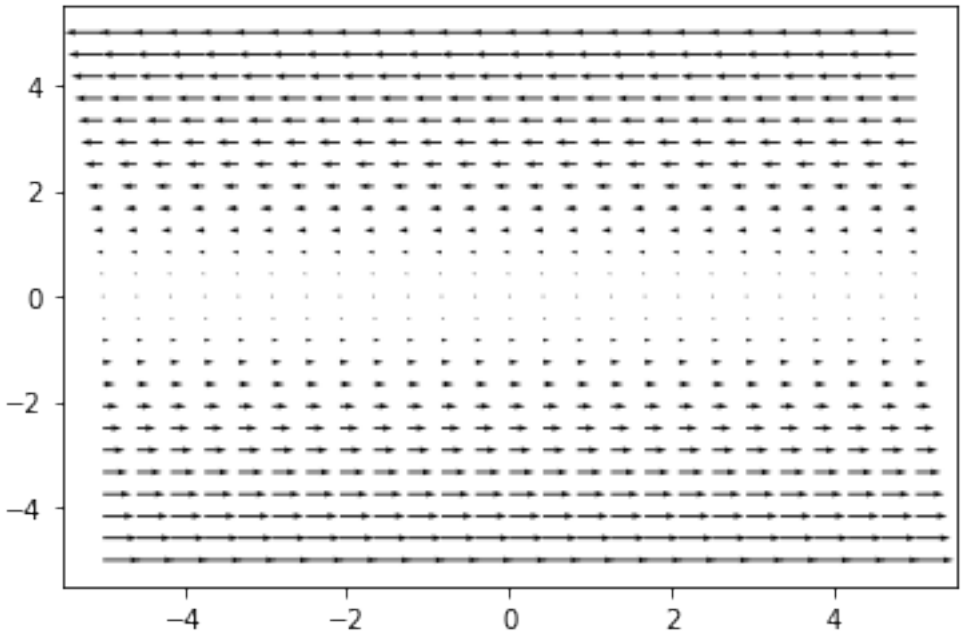
0.1 c.i. Original System Phase Portrait Plot

[52]:
```
phase_portrait2D(sys)
```



0.2 c.ii. Linearized System Phase Portrait Plot

[111]:
```
phase_portrait2D(linsys)
```



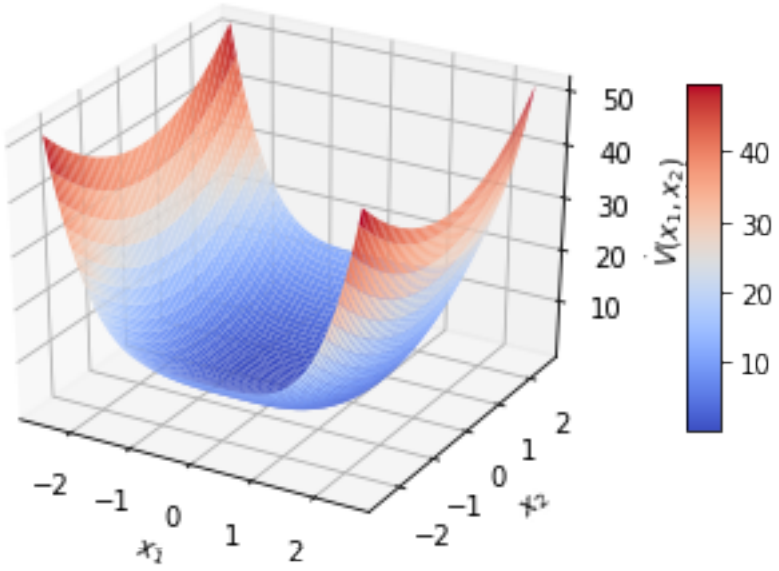
0.3 d. 3D Plot of Lyapunov Function Temporal Derivative

[123]:
```
fig = plt.figure()
ax = fig.gca(projection='3d')

x1 = np.linspace(-2.5,2.5, num=100)
x2 = np.linspace(-2.5,2.5, num=100)
x1, x2 = np.meshgrid(x1,x2)
Vd = x1**4 + 2*x2**2

surf = ax.plot_surface(x1,x2,Vd, cmap=cm.coolwarm, linewidth=2,␣
 ↪antialiased=True)
ax.set_xlabel(r'$x_1$')
ax.set_ylabel(r'$x_2$')
ax.set_zlabel(r'$\dot{V}(x_1,x_2)$')

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.6, aspect=10)
plt.show()
```



[ ]: