```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
import sys
import os

# Numerical integration
omega = 1
nu = 1
mu = 1/10
T = 2*np.pi/(omega+nu*mu)
timesteps = int(T*10)
t_array = np.linspace(0,T,timesteps)
dt = T/len(t_array)
xmin = 0
xmax = T

# J11 = 1/10-X20**2-2*X10[0]*X20[0]-3*X10[0]
# J12 = -3*X20**2-2*X10[0]*X20[0]-X10[0]**2-1
# J21 = 1+X20**2+3*X10[0]**2-2*X10[0]*X20[0]
# J22 = 1/10+2*X10[0]*X20[0]-3*X20[0]**2-X10[0]**2

# J = np.array([[J11, J12],\
#               [J21, J22]])

def dynamical_system(IC, t):
    X1 = IC[0]
    X2 = IC[1]
    M11 = IC[2]
    M12 = IC[3]
    M21 = IC[4]
    M22 = IC[5]

    dX1 = (1/10)*X1-X2**3-X1*X2**2-X1**2*X2-X2-X1**3
    dX2 = X1+(1/10)*X2+X1*X2**2+X1**3-X2**3-X1**2*X2

    J11 = 1/10-X2**2-2*X1*X2-3*X1**2
    J12 = -3*X2**2-2*X1*X2-X1**2-1
    J21 = 1+X2**2+3*X1**2-2*X1*X2
    J22 = 1/10+2*X1*X2-3*X2**2-X1**2

    dM11 = J11*M11+J12*M21
    dM12 = J11*M12+J12*M22
    dM21 = J21*M11+J22*M21
    dM22 = J21*M12+J22*M22
    return [dX1, dX2, dM11, dM12, dM21, dM22]

X1 = mu**0.5
X2 = 0
M11 = 1
M12 = 0
M21 = 0
M22 = 1
IC = [X1,X2,M11,M12,M21,M22]
eqs = odeint(dynamical_system, IC, t_array)

X1 = eqs[:,0]
X2 = eqs[:,1]

M11 = eqs[:,2]
M12 = eqs[:,3]
M21 = eqs[:,4]
M22 = eqs[:,5]

M11_last = M11[-1]
M12_last = M12[-1]
M21_last = M21[-1]
M22_last = M22[-1]


print('M11: ', np.round(M11_last,4))
print('M12: ', np.round(M12_last,4))
print('M12: ', M12_last)
print('M21: ', np.round(M21_last,4))
print('M22: ', np.round(M22_last,4))

M = np.array([[M11_last,M12_last],[M21_last,M22_last]])
eig_values, eig_vectors = np.linalg.eig(M)
```

```python
print(eig_values)
stab_exps = np.log(eig_values)/T
print('Stability exponents: ', np.round(stab_exps,4))


# for t in (t_array-1):

#     t = int(t)
# J11 = 1/10-X2[t]**2-2*X1[t]*X2[t]-3*X1[-1]
# J12 = -3*X2[t]**2-2*X1[t]*X2[t]-X1[t]**2-1
# J21 = 1+X2[t]**2+3*X1[t]**2-2*X1[-1]*X2[-1]
# J22 = 1/10+2*X1[t]*X2[t]-3*X2[t]**2-X1[-1]**2

#     J = np.array([[J11, J12],\
#                   [J21, J22]])

#     M = M + np.matmul(J,M)*dt

#     M11[t+1] = M[0,0]
#     M12[t+1] = M[0,1]
#     M21[t+1] = M[1,0]
#     M22[t+1] = M[1,1]

fig, ax = plt.subplots(figsize=(7,7))
ax.plot(t_array, X1, '-', linewidth=2, label='X1')
ax.plot(t_array, X2, '-', linewidth=2, label='X2')
ax.plot(t_array, M11, '-', linewidth=2, label='M11')
ax.plot(t_array, M12, '-', linewidth=2, label='M12')
ax.plot(t_array, M21, '-', linewidth=2, label='M21')
ax.plot(t_array, M22, '-', linewidth=2, label='M22')

ax.set_title('$3.2d$')
ax.set_xlabel('t')
ax.set_ylabel('X')
ax.set_xlim(xmin,xmax)
ax.set_box_aspect(1)

plt.legend(loc="lower right", prop={'size': 8})
plt.savefig('Dynamical systems/DS HW3/3.2/3.2d.png', bbox_inches='tight')
# plt.show()
```