2.4

c) $\begin{cases} \dot{x} = ux \\ \dot{y} = sy \end{cases} \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} u & 0 \\ 0 & s \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix}$ $\quad u > 0, \; s < 0, \; \gamma > 0$

$\lambda_1 = s, \; \lambda_2 = u$ $\qquad \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} \gamma \\ 1 \end{bmatrix}$

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = c_1 \underline{V_s} e^{st} + c_2 \underline{V_u} e^{ut}$$

$$\underline{V_s} = A - sI = 0 \Rightarrow \left(\begin{array}{cc|c} u-s & 0 & 0 \\ 0 & s-s & 0 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} u-s & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$$

$$\Rightarrow (u-s)x = 0 \cdot y \Rightarrow x = 0, \; y = 1 \Rightarrow \underline{V_s} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\underline{V_u} = A - uI = 0 \Rightarrow \left(\begin{array}{cc|c} u-u & 0 & 0 \\ 0 & s-u & 0 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & s-u & 0 \end{array}\right)$$

$$\Rightarrow 0 \cdot x = (s-u)y \Rightarrow y = 0, \; x = 1 \Rightarrow \underline{V_u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = c_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{st} + c_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{ut}$$

$$\underline{t=0}: \Rightarrow \begin{bmatrix} \gamma \\ 1 \end{bmatrix} = c_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^0 + c_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^0 = c_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{cases} c_1 = 1 \\ c_2 = \gamma \end{cases}$$

$$\Rightarrow \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{st} + \gamma \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{ut}$$

$$= \begin{bmatrix} 0 + \gamma e^{ut} \\ e^{st} + 0 \end{bmatrix} = \begin{bmatrix} \gamma e^{ut} \\ e^{st} \end{bmatrix} \Rightarrow \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \gamma e^{ut} \\ e^{st} \end{bmatrix}$$

$$x(t_1) = 1 \Rightarrow 1 = \gamma e^{ut_1} \Rightarrow \ln(e^{ut_1}) = \ln\left(\frac{1}{\gamma}\right)$$

$$\Rightarrow ut_1 = \ln\left(\frac{1}{\gamma}\right) \Rightarrow t_1 = \frac{\ln\left(\frac{1}{\gamma}\right)}{u}$$

$$\underline{\text{Ans}}: \; t_1 = \frac{\ln\left(\frac{1}{\gamma}\right)}{u}$$

2.4 d)

$$\begin{cases} \dot{x} = \mu x + y - x^2 & (1) \\ \dot{y} = -x + \mu y + 2x^2 & (2) \end{cases}$$

$$\underset{\sim}{J} = \begin{pmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{pmatrix} = \begin{pmatrix} \mu - 2x & 1 \\ 4x - 1 & \mu \end{pmatrix} \quad (3)$$

F.P. of the saddle node:

$$\dot{x} = \dot{y} = 0 \Rightarrow \begin{matrix} (1) \Rightarrow \\ (2) \Rightarrow \end{matrix} \begin{cases} y = x^2 - \mu x \\ y = \frac{x - 2x^2}{\mu} \end{cases}$$

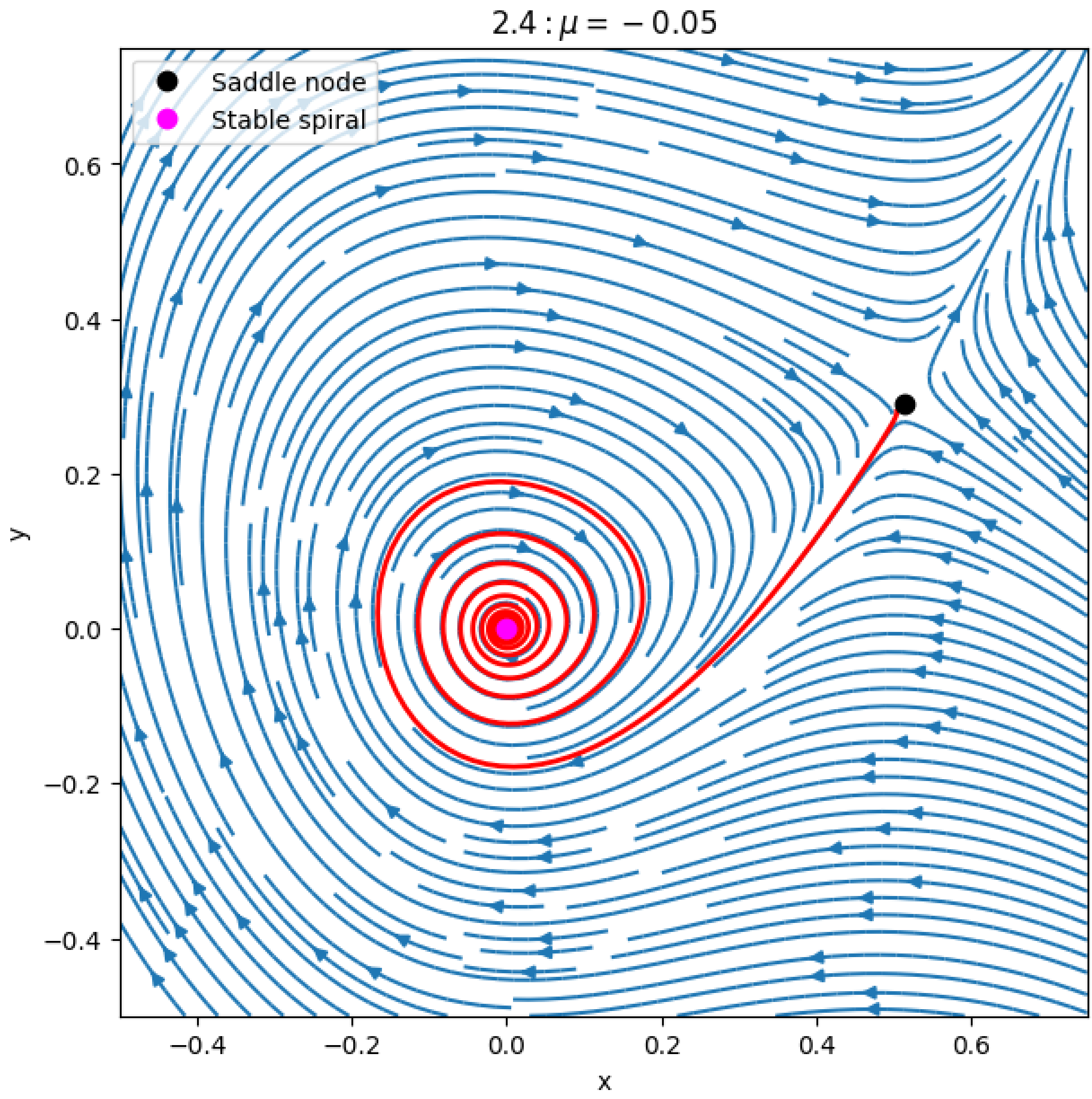
$$\Rightarrow \mu x^2 - \mu^2 x + 2x^2 - x = 0 \Rightarrow x(\underbrace{\mu x - \mu^2 + 2x - 1}_{0}) = 0$$

$$\Rightarrow x(\mu + 2) = \mu^2 + 1 \Rightarrow x = \frac{\mu^2 + 1}{\mu + 2} \quad (4)$$

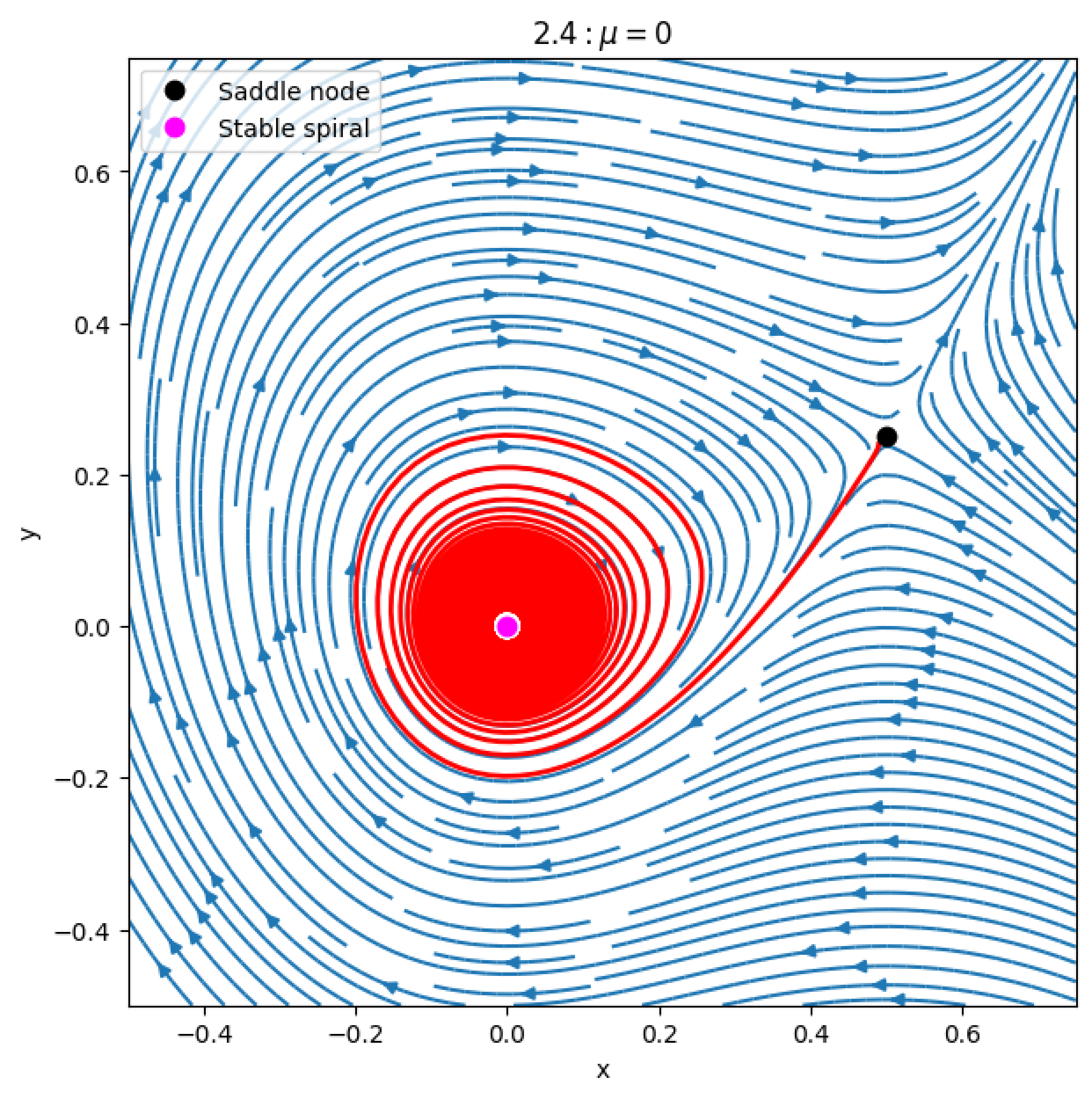
(4) into (3):

$$\underset{\sim}{J} = \begin{pmatrix} \mu - 2\left(\frac{\mu^2+1}{\mu+2}\right) & 1 \\ 4\left(\frac{\mu^2+1}{\mu+2}\right) - 1 & \mu \end{pmatrix}$$

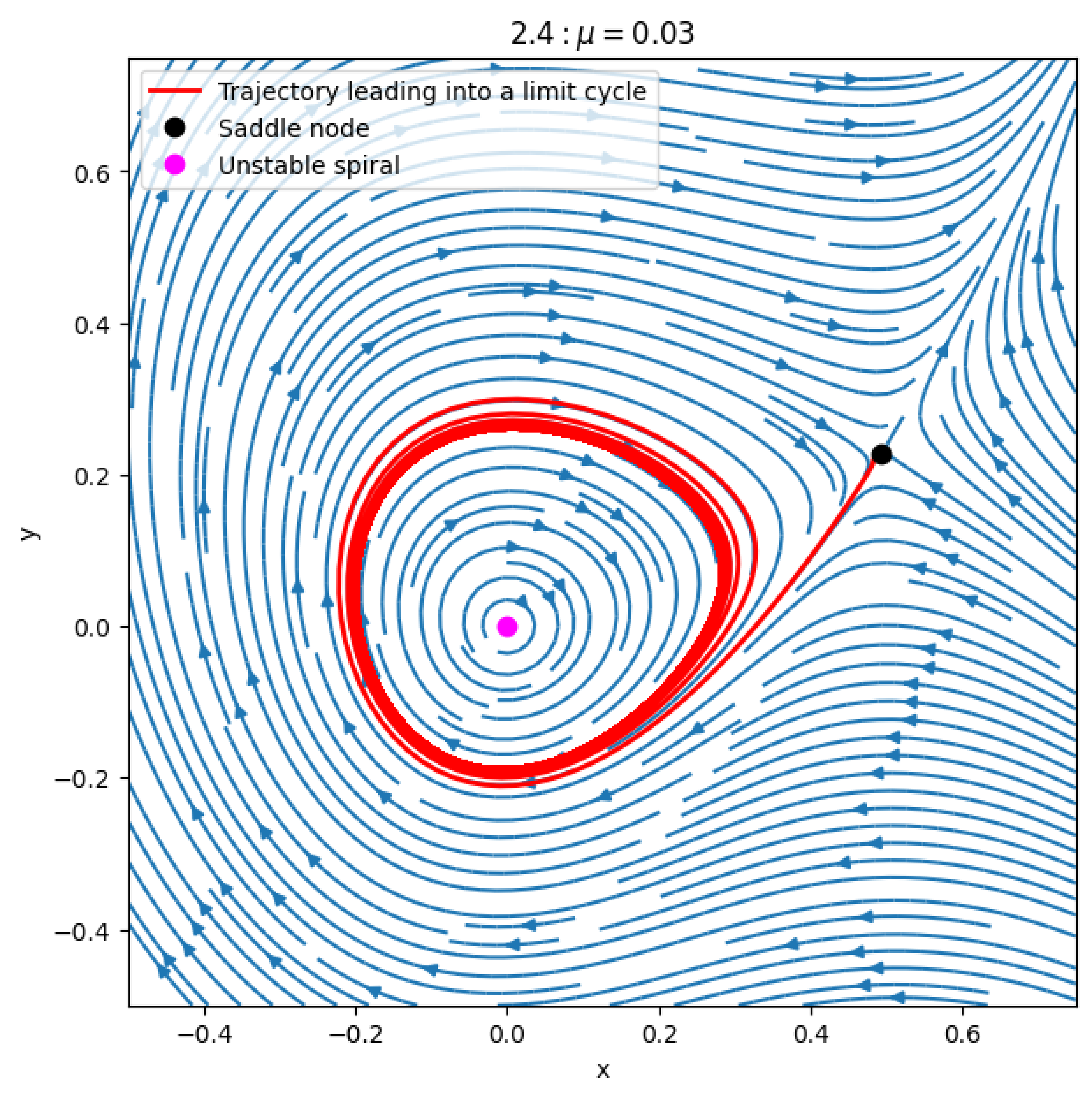
Use mathematica and pick $\mu = \lambda_2 > 0$



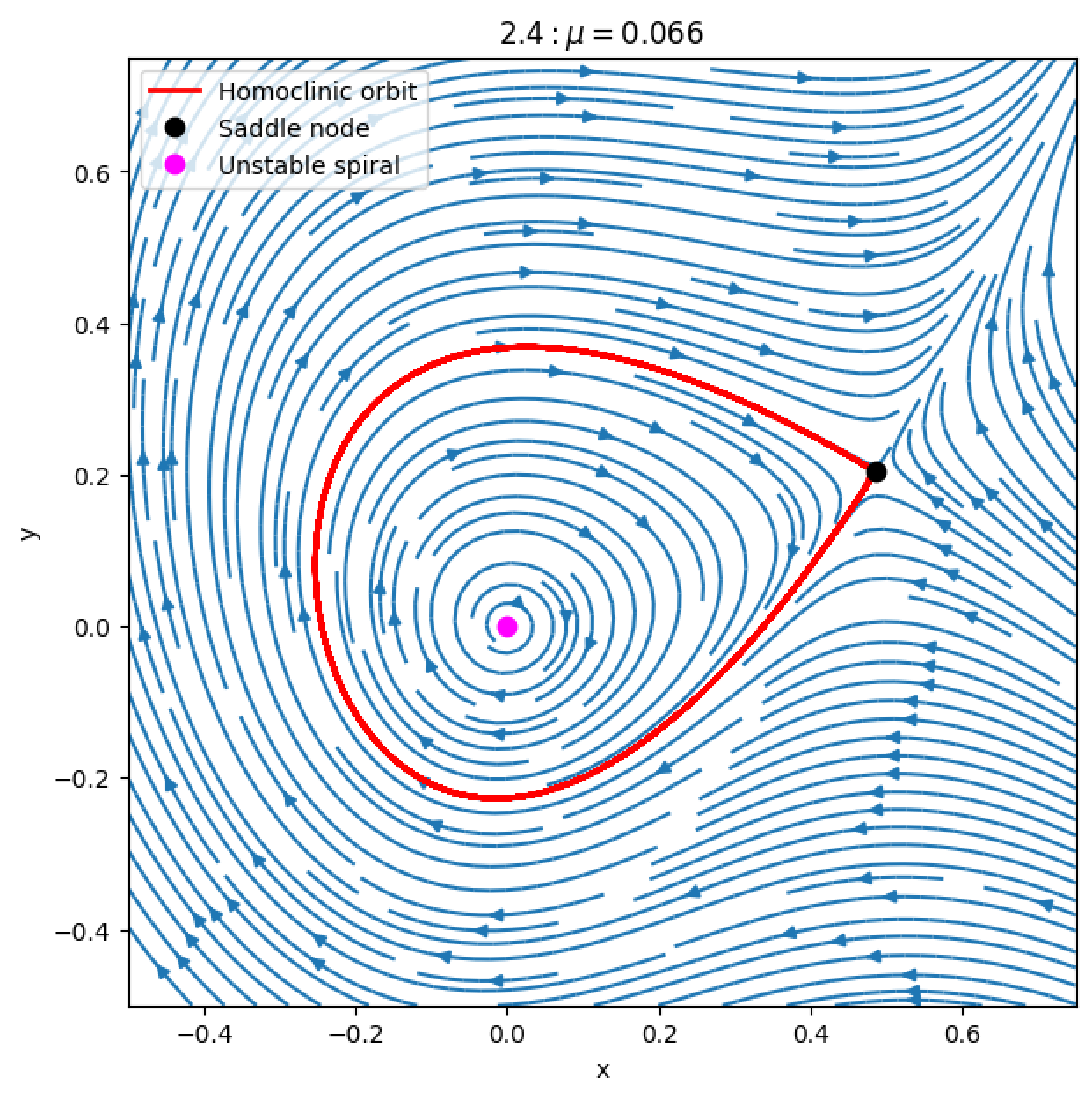
2.4 : μ = − 0.05
Saddle node
Stable spiral
y
x
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
0.6



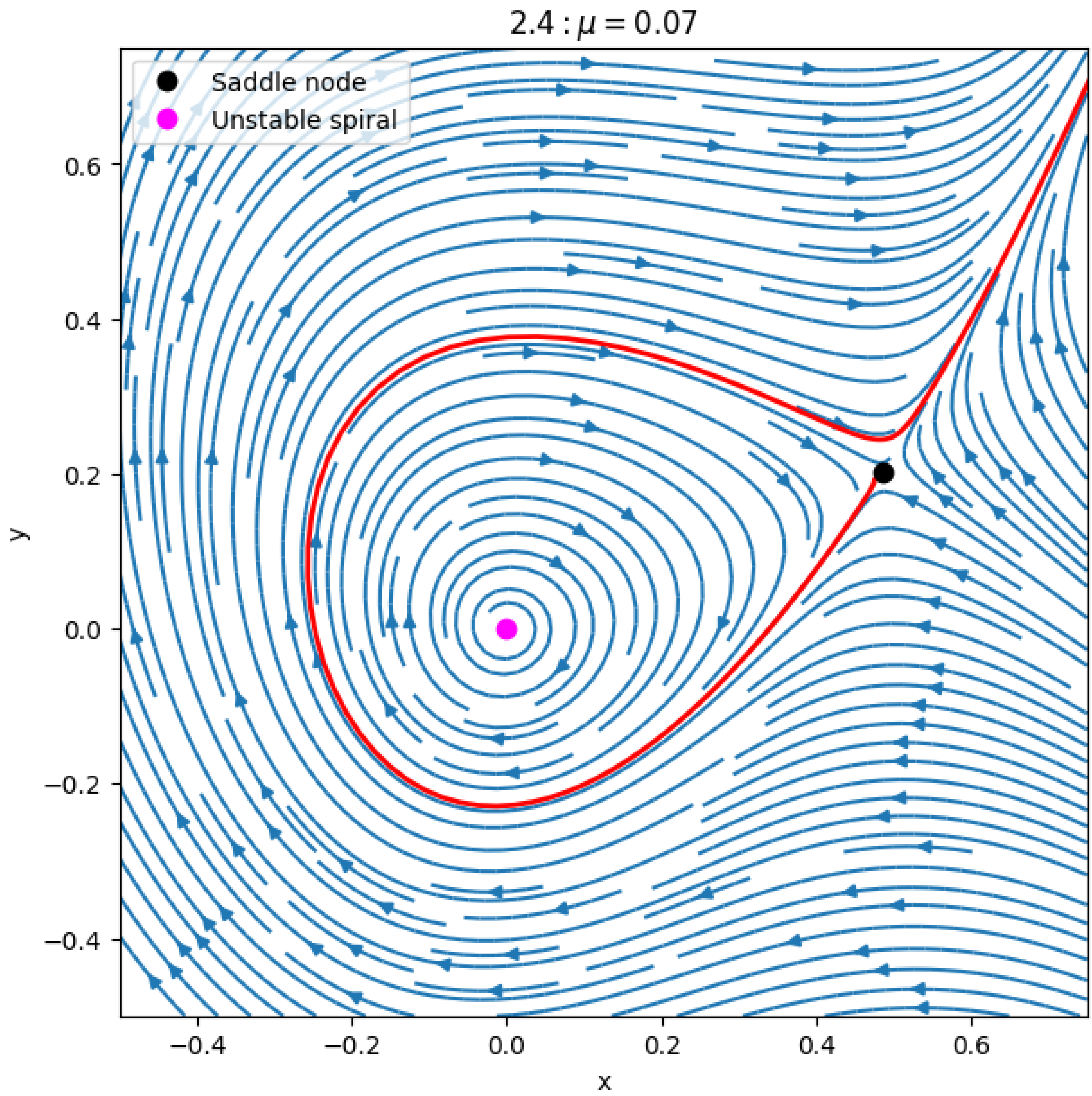
$2.4 : \mu = 0$
Saddle node
Stable spiral
0.6
0.4
0.2
0.0
−0.2
−0.4
y
−0.4
−0.2
0.0
0.2
0.4
0.6
x

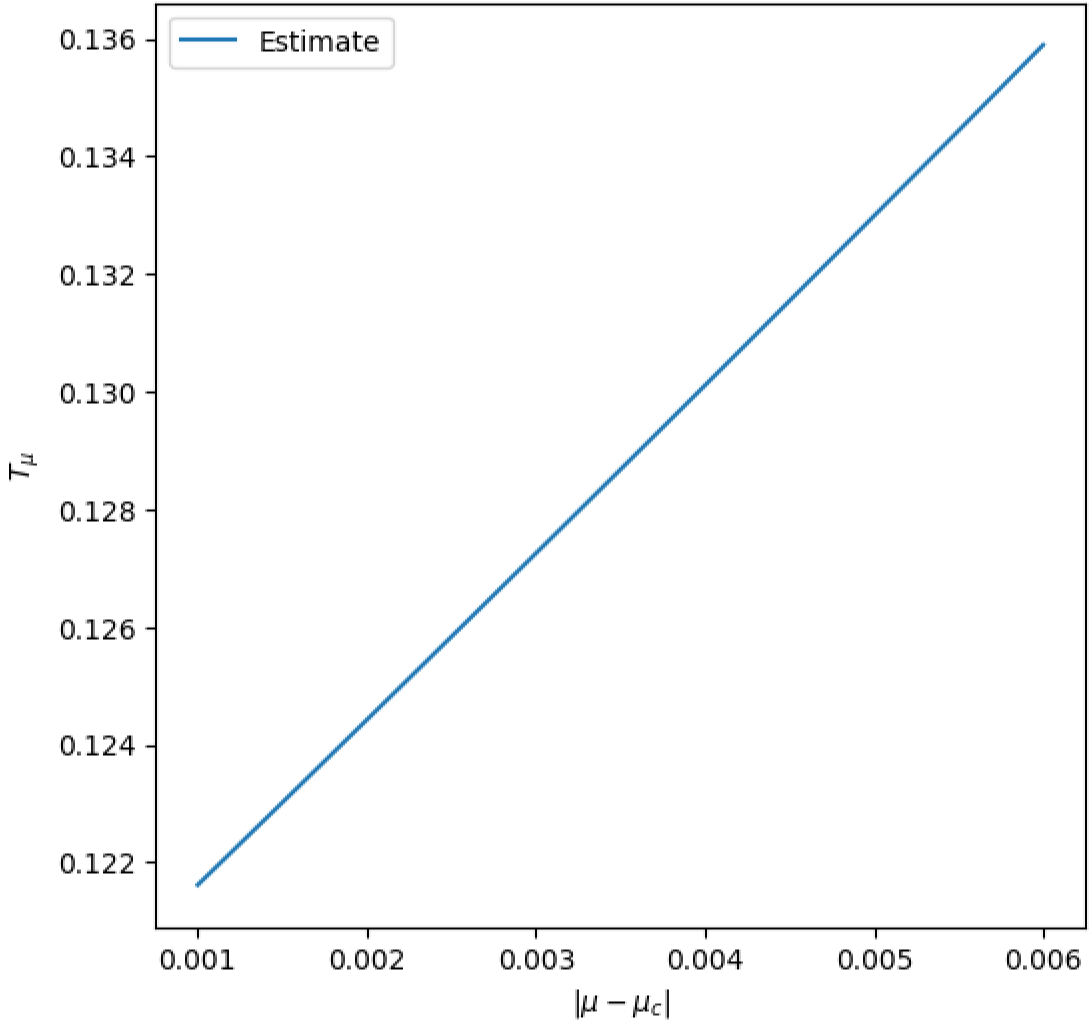


$2.4 : \mu = 0.03$
Trajectory leading into a limit cycle
Saddle node
Unstable spiral
0.6
0.4
0.2
0.0
−0.2
−0.4
y
−0.4
−0.2
0.0
0.2
0.4
0.6
x



$2.4 : \mu = 0.066$
Homoclinic orbit
Saddle node
Unstable spiral
0.6
0.4
0.2
0.0
−0.2
−0.4
y
−0.4
−0.2
0.0
0.2
0.4
0.6
x


$2.4 : \mu = 0.07$
Saddle node
Unstable spiral
0.6
0.4
0.2
0.0
−0.2
−0.4
y
−0.4
−0.2
0.0
0.2
0.4
0.6
x



Estimate
0.136
0.134
0.132
0.130
0.128
0.126
0.124
0.122
T_μ
0.001
0.002
0.003
0.004
0.005
0.006
$|\mu - \mu_c|$

In[109]:=

```
ClearAll["Global`*"];
m = Eigenvalues[{{μ - 2 * (μ^2 + 1)/(μ + 2), 1}, {4 * (μ^2 + 1)/(μ + 2) - 1, μ}}] // MatrixForm;
m // FullSimplify
```

$$\text{MatrixForm}\left[\left\{-\frac{\sqrt{((\mu+4)\,\mu+9)\,\mu^2+5}\;-2\,\mu+1}{\mu+2},\ \frac{\sqrt{(\mu\,(\mu+4)+9)\,\mu^2+5}\;+2\,\mu-1}{\mu+2}\right\}\right]$$

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



xmin = -0.5
ymin = -0.5
xmax = 0.75
ymax = 0.75

# Streamplot
no_points = 100
x_points = np.linspace(xmin, xmax, no_points)
y_points = np.linspace(ymin, ymax, no_points)
X, Y = np.meshgrid(x_points, y_points)

global mu
mu= 0.06599
dxdt_streamplot = mu*X + Y - X**2
dydt_streamplot = -X + mu*Y + 2*X**2

# Numerical integration
T = 100
t = np.linspace(0,T,T*10)
x = np.zeros(T)
y = x.copy()
x2 = x.copy()
y2 = y.copy()
fp = np.array([[(1+(mu**2))/(2+mu), (-(2*mu-1)*(1+mu**2))/((2+mu)**2)], [0,0]])
# x[0] = fp[0,0] - 0.01
# y[0] = fp[0,1] - 0.01
x[0] = 0.01
y[0] = 0.01
# x2[0] = fp[1,0] - 0.005
# y2[0] = fp[1,1] - 0.005

def dynamical_system(xy, t):
    # mu = 0.065
    x = xy[0]
    y = xy[1]
    dxdt_integration = mu*x + y - x**2
    dydt_integration = -x + mu*y + 2*x**2
    return [dxdt_integration, dydt_integration]

x0y0 = [x[0],y[0]]
xy = odeint(dynamical_system, x0y0, t)
ind = np.argmax(xy[:,0])
gamma = ((xy[ind,0] - fp[0,0])**2 + (xy[ind,1] - fp[0,1])**2)**0.5
x = xy[:,0]
y = xy[:,1]
# x0y0 = [x2[0],y2[0]]
# xy2 = odeint(dynamical_system, x0y0, t)
# x2 = xy2[:,0]
# y2 = xy2[:,1]

print(gamma)

```

```
# mu = 0.06591, gamma = 0.004898948029599108
# mu = 0.06592, gamma = 0.004620674420854347
# mu = 0.06593, gamma = 0.0045464915511753004
# mu = 0.06594, gamma = 0.004219778084911094
# mu = 0.06595, gamma = 0.003887937972174111
# mu = 0.06596, gamma = 0.0037182289028526564
# mu = 0.06597, gamma = 0.003512188654658859
# mu = 0.06598, gamma = 0.003105918259005926
# mu = 0.06599, gamma = 0.002830588306209588


# mu = 0.060, gamma = 0.06531584901232429
# mu = 0.061, gamma = 0.059478472832076835
# mu = 0.062, gamma = 0.05033007132854761
# mu = 0.063, gamma = 0.040706178156078014
# mu = 0.064, gamma = 0.03071758583638221
# mu = 0.065, gamma = 0.019671518608965637

# fig, ax = plt.subplots(figsize=(7,7))
# ax.streamplot(X, Y, dxdt_streamplot, dydt_streamplot, density = 2)
# ax.plot(x, y, '-', color='red', linewidth=2)
# ax.plot(x2, y2, '-', color='magenta', linewidth=2)
# ax.plot(fp[0,0],fp[0,1], '.', color='black', markersize=15, label='Saddle node')
# ax.plot(fp[1,0],fp[1,1], '.', color='magenta', markersize=15, label='Unstable spiral')
# ax.set_title('$2.4: \mu = 0.07$')
# ax.set_xlabel('x')
# ax.set_ylabel('y')
# ax.set_xlim(xmin,xmax)
# ax.set_ylim(xmin,xmax)
# ax.set_box_aspect(1)

# plt.legend(loc="upper left")
# plt.savefig('24.a_mu0.07.png', bbox_inches='tight')
# plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
import sys

# 2.4e

# mu = 0.060, gamma = 0.06531584901232429
# mu = 0.061, gamma = 0.059478472832076835
# mu = 0.062, gamma = 0.05033007132854761
# mu = 0.063, gamma = 0.040706178156078014
# mu = 0.064, gamma = 0.03071758583638221
# mu = 0.065, gamma = 0.019671518608965637

# mu = 0.06591, gamma = 0.004898948029599108
# mu = 0.06592, gamma = 0.004620674420854347
# mu = 0.06593, gamma = 0.0045464915511753004
# mu = 0.06594, gamma = 0.004219778084911094
# mu = 0.06595, gamma = 0.003887937972174111
# mu = 0.06596, gamma = 0.0037182289028526564
# mu = 0.06597, gamma = 0.003512188654658859
# mu = 0.06598, gamma = 0.003105918259005926
# mu = 0.06599, gamma = 0.002830588306209588

mu1 = np.array([0.060, 0.061, 0.062, 0.063, 0.064, 0.065])
gammas = np.array([0.06531584901232429, \
        0.059478472832076835, \
        0.05033007132854761, \
        0.040706178156078014, \
        0.03071758583638221, \
        0.019671518608965637])

mu_c = 0.066
mu_array = np.abs(mu1-mu_c)
coef = np.polyfit(np.log(mu_array),np.log(gammas),1)
poly = np.poly1d(coef)
# print(poly)

u = (np.sqrt(((mu1*(mu1+4)+9)*mu1**2+5)) + 2*mu1-1)/(mu1 + 2)
Tmu_estimate = np.log(1/gammas)/u

# fig, ax = plt.subplots(figsize=(7,7))
# ax.plot(np.log(mu_array),np.log(gammas))
# ax.plot(mu_array,gammas)
# ax.set_xscale('log')
# ax.set_yscale('log')
# plt.show()

# Answer: a = 0.6813, A = 0.7669


# 2.4f
a = 0.6813
A = 0.7669

mu2 = np.linspace(0.06,0.065,50)
mu_c = 0.066
mu_abs = np.abs(mu2-mu_c)
gammas = mu2**a + A
u = (np.sqrt(((mu2*(mu2+4)+9)*mu2**2+5)) + 2*mu2-1)/(mu2 + 2)
```

```python
Tmu_theory = np.log(1/gammas)/u

fig, ax = plt.subplots(figsize=(6,6))
ax.plot(mu_abs, Tmu_theory, label='Estimate')
# ax.plot(mu_array, Tmu_estimate, label='Numerical')
ax.set_ylabel('$T_{\mu}$')
ax.set_xlabel('$|\mu-\mu_{c}|$')
# ax.set_xscale('log')
# ax.set_yscale('log')

plt.legend(loc="upper left")
plt.savefig('24f.png', bbox_inches='tight')
plt.show()
```