Exercise 4-1:

(a)

$$\dot{\underline{x}} = \begin{bmatrix} -2 & 0 \\ 0 & -4 \end{bmatrix} \underline{x} = A\underline{x} \approx \frac{f(x+\Delta t) - f(x)}{\Delta t}$$

$$\Rightarrow x_{k+1} = x_k + \Delta t f(x_k) \quad (\text{if } \dot{x} = Ax)$$

$$\Rightarrow x_{k+1} = (I + \Delta t A) x_k$$

stable when $|\text{eigs}(I + \Delta t A)| < 1$

$$\Rightarrow I + \Delta t A = \begin{bmatrix} 1-2\Delta t & 0 \\ 0 & 1-4\Delta t \end{bmatrix}$$

$\Rightarrow$ compute $\det(A - \lambda I) = 0$

$$\Rightarrow \begin{bmatrix} 1-2\Delta t-\lambda & 0 \\ 0 & 1-4\Delta t-\lambda \end{bmatrix} = 0$$

$$\Rightarrow \lambda = 1-2\Delta t,\ 1-4\Delta t$$

$$\Rightarrow \begin{cases} -1 < 1-2\Delta t < 1 \\ -1 < 1-4\Delta t < 1 \end{cases} \Rightarrow \begin{cases} -2 < -2\Delta t < 0 \\ -2 < -4\Delta t < 0 \end{cases}$$

$$\Rightarrow \begin{cases} 1 > \Delta t > 0 \\ 0.5 > \Delta t > 0 \end{cases}$$

$$\boxed{\text{ustable when } \Delta t > 1}$$ #

(b)

$$\frac{x_{k+1} - x_k}{\Delta t} \approx f(x_{k+1}) = \dot{x} = Ax$$

$$\Rightarrow \quad x_{k+1} = x_k + \Delta t\, f(x_{k+1})$$

$$\Rightarrow \quad x_{k+1} = (I - A\Delta t)^{-1} x_k$$

stable when $\left| \text{eigs}(I - \Delta t A)^{-1} \right| < 1$

$$\Rightarrow \quad I - \Delta t A = \begin{bmatrix} 1+2\Delta t & 0 \\ 0 & 1+4\Delta t \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1+2\Delta t & 0 \\ 0 & 1+4\Delta t \end{bmatrix}^{-1} = \frac{1}{(1+2\Delta t)(1+4\Delta t)} \begin{bmatrix} 1+4\Delta t & 0 \\ 0 & 1+2\Delta t \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{1}{1+2\Delta t} & 0 \\ 0 & \frac{1}{1+4\Delta t} \end{bmatrix}$$

$$\Rightarrow \quad \lambda = \frac{1}{1+2\Delta t}, \quad \frac{1}{1+4\Delta t}$$

$$\Rightarrow \begin{cases} -1 < \dfrac{1}{1+2\Delta t} < 1 \\ -1 < \dfrac{1}{1+4\Delta t} < 1 \end{cases} \Rightarrow \begin{cases} 1+2\Delta t > 1 \\ 1+4\Delta t > 1 \end{cases}$$

$$\Rightarrow \begin{cases} \Delta t > 0, \quad \Delta t < -1 \end{cases}$$

$$\boxed{\text{simulation will always be stable}}$$

Exercise 4-2

$\dot{\underline{x}} = f(\underline{x})$, $\sigma = 10$, $\rho = 28$, $\beta = 8/3$

$$\begin{bmatrix} \dot{x} = \sigma(y-x) \\ \dot{y} = x(\rho - z) - y \\ \dot{z} = xy - \beta z \end{bmatrix} \Rightarrow \begin{bmatrix} f_1(x, y) \\ f_2(x, y, z) \\ f_3(x, y, z) \end{bmatrix} = \begin{bmatrix} \sigma(y-x) \\ x(\rho - z) - y \\ xy - \beta z \end{bmatrix}$$

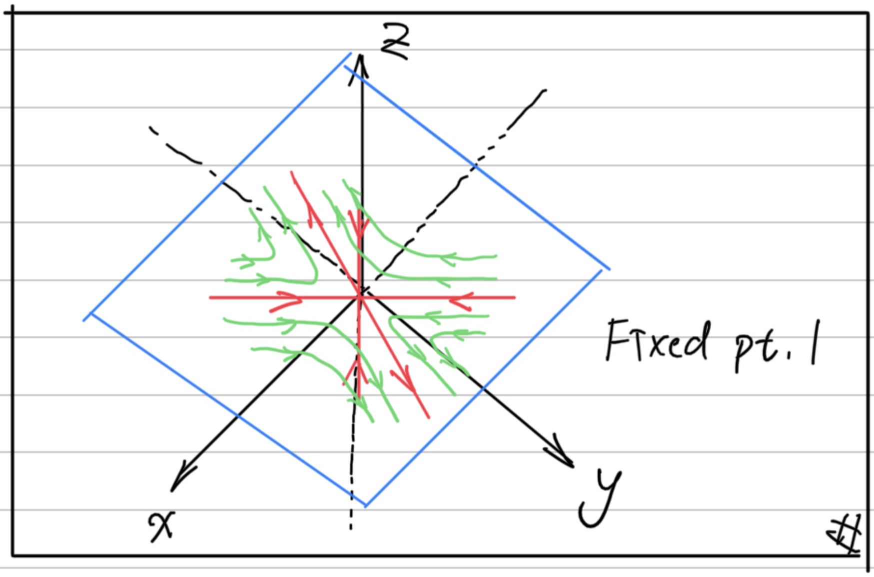
$$\Rightarrow \frac{Df}{Dx} = \begin{bmatrix} -\sigma & \sigma & 0 \\ (\rho - z) & -1 & -x \\ y & x & -\beta \end{bmatrix}, \quad \bar{\underline{x}} = \begin{bmatrix} (0, 0, 0) \\ (\sqrt{72}, \sqrt{72}, 27) \\ (-\sqrt{72}, -\sqrt{72}, 27) \end{bmatrix}$$

Fixed point 1.

$$\bar{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \frac{Df}{Dx}(\bar{x}) = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & 0 \\ 0 & 0 & 8/3 \end{bmatrix}$$

$\Rightarrow$ from python, $\boxed{\lambda = -22.83,\ 11.83,\ -2.67 \quad \hookrightarrow \text{unstable saddle}}$ #

$$\begin{cases} \xi_1 = (-0.6, 0.78, 0) \\ \xi_2 = (-0.61, -0.91, 0) \\ \xi_3 = (0, 0, 1) \end{cases}$$



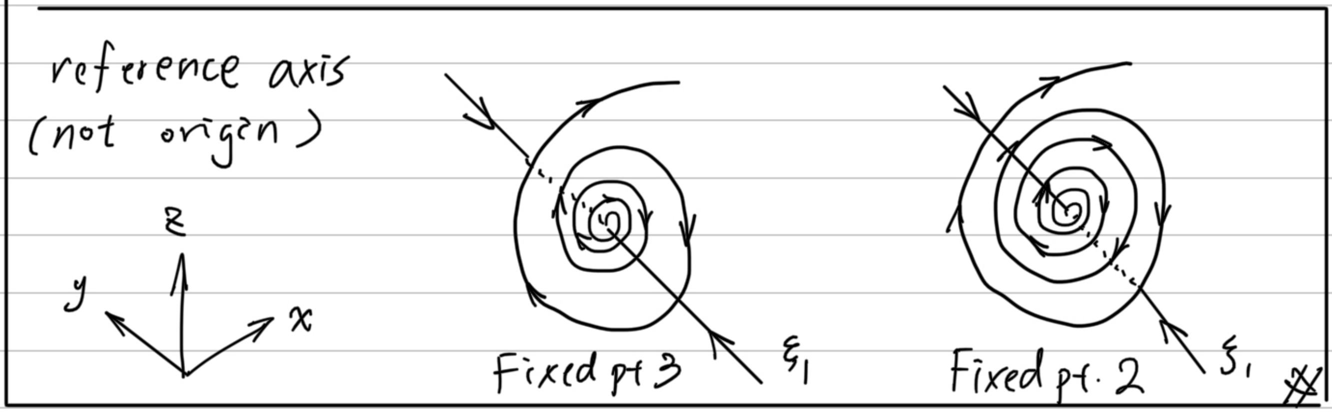
Fixed point 2.

$$\bar{x} = \begin{bmatrix} 6\sqrt{2} \\ 6\sqrt{2} \\ 27 \end{bmatrix} \Rightarrow \boxed{\lambda = -13.85,\ 0.09 + 10.19i,\ 0.09 - 10.19i \quad \hookrightarrow \text{unstable point}}$$ #

Fixed point 3,

$$\bar{x} = \begin{bmatrix} -6\sqrt{2} \\ -6\sqrt{2} \\ 27 \end{bmatrix} \Rightarrow \boxed{\lambda = -13.85,\ 0.09+10.19i,\ 0.09-10.19i}$$

↳ unstable point #



Exercise 4-3

eigenvalues, stability of 3 fixed pt wrt rho 5~50 is at the end of hw with the code ! #

plots is provided at the end of hw with the code #

In the linearized stability analysis, 3 fixed point become unstable when $\rho \geq 25$.

From the plots, when $\rho = 5, 10, 15$,

all points spiral into the center. 2 of the other fix point become ustable stating ρ

Exercise 4-4 taylor expansion

$$\begin{cases} f(t-\Delta t) = f(t) - \Delta t f'(t) + \frac{\Delta t^2}{2!} f''(t) - \frac{\Delta t^3}{3!} f'''(t) \cdots \\ f(t-2\Delta t) = f(t) - 2\Delta t f'(t) + \cancel{\frac{(2\Delta t)^2}{2!} f''(t)} - \frac{(2\Delta t)^3}{3!} f'''(t) + \cdots \end{cases}$$

$$\Rightarrow 4 f(t-\Delta t) = 4f(t) - 4\Delta t f'(t) + \cancel{\frac{4\Delta t^2}{2!} f''(t)} - \frac{4\Delta t^3}{3!} f'''(t) + \cdots$$

$$\Rightarrow 4 f(t-\Delta t) - f(t-2\Delta t) = 3 f(t) - 2\Delta t f'(t) + O(\Delta t^3)$$

$$f'(t) \approx \frac{-4 f(t-\Delta t) + f(t-2\Delta t) + 3f(t)}{2\Delta t} + O(\Delta t^2)$$

$$\boxed{f'(t) = \frac{3f(t) - 4f(t-\Delta t) + f(t-2\Delta t)}{2\Delta t} + O(\Delta t^2) \quad \#}$$

Exercise 4-5

(a)

$$\boxed{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\varepsilon(x_1^2-1)x_2 - x_1 \end{bmatrix} \quad \#} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

(b)

$$\frac{Df}{D\underline{x}} = \begin{bmatrix} 0 & 1 \\ -2\varepsilon x_1 x_2 - 1 & -\varepsilon(x_1^2-1) \end{bmatrix}$$

Exercise 4-5

(a)

$$\boxed{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\varepsilon(x_1^2-1)x_2 - x_1 \end{bmatrix}} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

#

(b)

$$\frac{Df}{D\underline{x}} = \begin{bmatrix} 0 & 1 \\ -2\varepsilon x_1 x_2 - 1 & -\varepsilon(x_1^2-1) \end{bmatrix}$$

$$\bar{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \frac{Df}{D\underline{x}} = \begin{bmatrix} 0 & 1 \\ -1 & \varepsilon \end{bmatrix}$$

$$-\lambda(\varepsilon-\lambda)+1 = 0 \Rightarrow \lambda^2 - \varepsilon\lambda + 1 = 0$$

$$\Rightarrow \lambda = \frac{\varepsilon \pm \sqrt{\varepsilon^2-4}}{2}, \quad \varepsilon > 0$$

$$\boxed{\Rightarrow \begin{cases} \text{two positive real } \lambda \text{ if } \varepsilon > 2, \text{ unstable source} \\ \text{two positive, real with complex conjugate } \lambda \text{ if } 2 > \varepsilon > 0, \text{ unstable spiral} \end{cases}}$$

#

4-2

In [35]:

```
import numpy as np
import math
from numpy import linalg as LA

sigma = 10
beta = 8 / 3
rho = 28

x = 0
y = 0
z = 0
Df_Dx1 = np.array([[-sigma, sigma, 0],[rho-z, -1, -x ],[y, x , -beta]])
print(LA.eig(Df_Dx1))

x = math.sqrt(72)
y = math.sqrt(72)
z = 27
Df_Dx2 = np.array([[-sigma, sigma, 0],[rho-z, -1, -x ],[y, x , -beta]])
print(LA.eig(Df_Dx2))

x = -math.sqrt(72)
y = -math.sqrt(72)
z = 27
Df_Dx3 = np.array([[-sigma, sigma, 0],[rho-z, -1, -x ],[y, x , -beta]])
print(LA.eig(Df_Dx3))
```

```
(array([-22.82772345,  11.82772345,  -2.66666667]), array([[-0.61481679, -0.41
650418,  0.        ],
       [ 0.78866997, -0.9091338 ,  0.        ],
       [ 0.        ,  0.        ,  1.        ]]))
(array([-13.85457791 +0.j        ,   0.09395562+10.19450522j,
         0.09395562-10.19450522j]), array([[ 0.85566502+0.j        , -0.266119
32-0.29501017j,
        -0.26611932+0.29501017j],
       [-0.32982275+0.j        ,  0.03212861-0.56907743j,
         0.03212861+0.56907743j],
       [-0.39881615+0.j        , -0.71921356+0.j        ,
        -0.71921356-0.j        ]]))
(array([-13.85457791 +0.j        ,   0.09395562+10.19450522j,
         0.09395562-10.19450522j]), array([[ 0.85566502+0.j        , -0.266119
32-0.29501017j,
        -0.26611932+0.29501017j],
       [-0.32982275+0.j        ,  0.03212861-0.56907743j,
         0.03212861+0.56907743j],
       [ 0.39881615+0.j        ,  0.71921356+0.j        ,
         0.71921356-0.j        ]]))
```

4-3

In [36]:

```
import numpy as np
import math as m
from numpy import linalg as LA
```

```
sigma = 10
beta = 8 / 3
rho_m = np.arange(5, 55, 5)

for i, rho in enumerate(rho_m):
    for j in range(3):
        fix_pts = [[0, 0, 0],
            [m.sqrt(beta*(rho-1)), m.sqrt(beta*(rho-1)), rho-1],
            [-m.sqrt(beta*(rho-1)), -m.sqrt(beta*(rho-1)), rho-1]]
        print("fix pt",fix_pts[j], "rho ", rho)
        Df_Dx = [[-sigma, sigma, 0],
                 [rho-fix_pts[j][2], -1, -fix_pts[j][0] ],
                 [fix_pts[j][1], fix_pts[j][0], -beta]]
        u, v = LA.eig(Df_Dx)
        print("eigenvalue ", u)
        if all(value < 0 for value in list(u.real)):
            print('stable\n')
        else:
            print("unstable\n")
```

```
fix pt [0, 0, 0] rho  5
eigenvalue  [-13.88152731   2.88152731  -2.66666667]
unstable

fix pt [3.265986323710904, 3.265986323710904, 4] rho  5
eigenvalue  [-11.80921801+0.j          -0.92872433+4.14758424j
  -0.92872433-4.14758424j]
stable

fix pt [-3.265986323710904, -3.265986323710904, 4] rho  5
eigenvalue  [-11.80921801+0.j          -0.92872433+4.14758424j
  -0.92872433-4.14758424j]
stable

fix pt [0, 0, 0] rho  10
eigenvalue  [-16.4658561    5.4658561   -2.66666667]
unstable

fix pt [4.898979485566356, 4.898979485566356, 9] rho  10
eigenvalue  [-12.47567248+0.j          -0.5954971 +6.17416092j
  -0.5954971 -6.17416092j]
stable

fix pt [-4.898979485566356, -4.898979485566356, 9] rho  10
eigenvalue  [-12.47567248+0.j          -0.5954971 +6.17416092j
  -0.5954971 -6.17416092j]
stable

fix pt [0, 0, 0] rho  15
eigenvalue  [-18.54798835   7.54798835  -2.66666667]
unstable

fix pt [6.110100926607786, 6.110100926607786, 14] rho  15
eigenvalue  [-12.96628046+0.j          -0.3501931 +7.58041077j
  -0.3501931 -7.58041077j]
stable

fix pt [-6.110100926607786, -6.110100926607786, 14] rho  15
eigenvalue  [-12.96628046+0.j          -0.3501931 +7.58041077j
  -0.3501931 -7.58041077j]
stable

fix pt [0, 0, 0] rho  20
eigenvalue  [-20.34082208   9.34082208  -2.66666667]
unstable

fix pt [7.118052168020874, 7.118052168020874, 19] rho  20
eigenvalue  [-13.35708312+0.j          -0.15479177+8.70866846j
  -0.15479177-8.70866846j]
stable

fix pt [-7.118052168020874, -7.118052168020874, 19] rho  20
eigenvalue  [-13.35708312+0.j          -0.15479177+8.70866846j
  -0.15479177-8.70866846j]
stable

fix pt [0, 0, 0] rho  25
eigenvalue  [-21.93928222  10.93928222  -2.66666667]
unstable
```

```
fix pt [8.0, 8.0, 24] rho  25
eigenvalue  [-1.36825089e+01+0.j          7.92110753e-03+9.67212654j
  7.92110753e-03-9.67212654j]
unstable

fix pt [-8.0, -8.0, 24] rho  25
eigenvalue  [-1.36825089e+01+0.j          7.92110753e-03+9.67212654j
  7.92110753e-03-9.67212654j]
unstable

fix pt [0, 0, 0] rho  30
eigenvalue  [-23.39553017  12.39553017  -2.66666667]
unstable

fix pt [8.793937305515279, 8.793937305515279, 29] rho  30
eigenvalue  [-13.96140336 +0.j           0.14736835+10.52425233j
   0.14736835-10.52425233j]
unstable

fix pt [-8.793937305515279, -8.793937305515279, 29] rho  30
eigenvalue  [-13.96140336 +0.j           0.14736835+10.52425233j
   0.14736835-10.52425233j]
unstable

fix pt [0, 0, 0] rho  35
eigenvalue  [-24.7418814   13.7418814   -2.66666667]
unstable

fix pt [9.521904571390467, 9.521904571390467, 34] rho  35
eigenvalue  [-14.20531845 +0.j           0.26932589+11.29509556j
   0.26932589-11.29509556j]
unstable

fix pt [-9.521904571390467, -9.521904571390467, 34] rho  35
eigenvalue  [-14.20531845 +0.j           0.26932589+11.29509556j
   0.26932589-11.29509556j]
unstable

fix pt [0, 0, 0] rho  40
eigenvalue  [-26.          15.          -2.66666667]
unstable

fix pt [10.198039027185569, 10.198039027185569, 39] rho  40
eigenvalue  [-14.4218948  +0.j           0.37761407+12.00343958j
   0.37761407-12.00343958j]
unstable

fix pt [-10.198039027185569, -10.198039027185569, 39] rho  40
eigenvalue  [-14.4218948  +0.j           0.37761407+12.00343958j
   0.37761407-12.00343958j]
unstable

fix pt [0, 0, 0] rho  45
eigenvalue  [-27.18524844  16.18524844  -2.66666667]
unstable

fix pt [10.83205120618128, 10.83205120618128, 44] rho  45
eigenvalue  [-14.61647158 +0.j           0.47490246+12.66190866j
   0.47490246-12.66190866j]
unstable
```

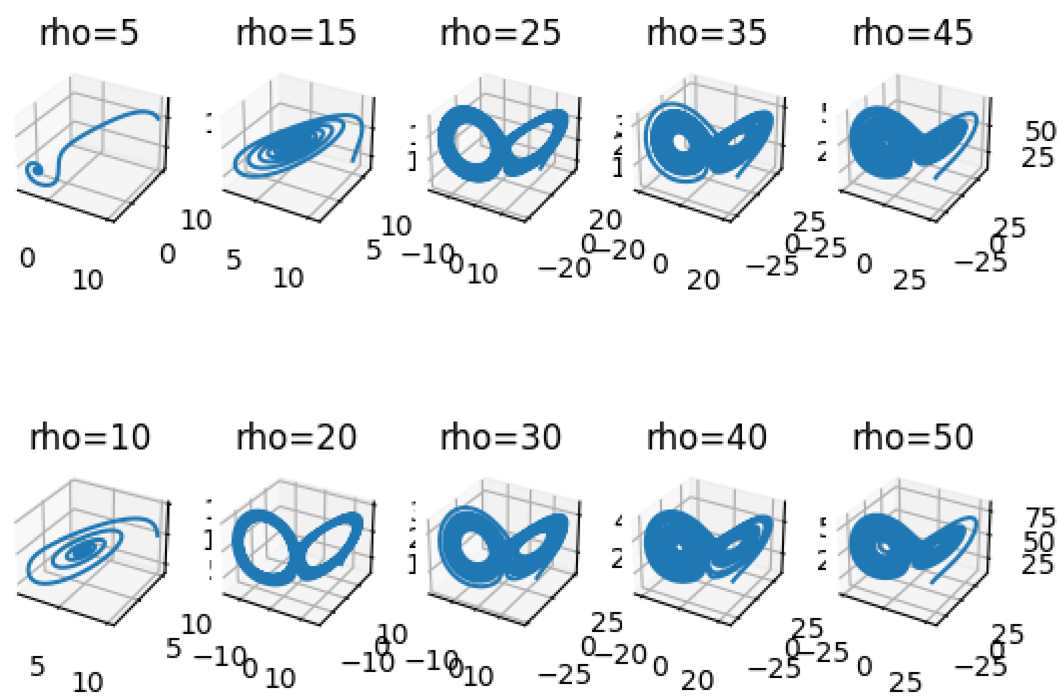
```
fix pt [-10.83205120618128, -10.83205120618128, 44] rho  45
eigenvalue  [-14.61647158 +0.j           0.47490246+12.66190866j
   0.47490246-12.66190866j]
unstable

fix pt [0, 0, 0] rho  50
eigenvalue  [-28.30898946  17.30898946  -2.66666667]
unstable

fix pt [11.430952132988164, 11.430952132988164, 49] rho  50
eigenvalue  [-14.79293838 +0.j           0.56313585+13.27944826j
   0.56313585-13.27944826j]
unstable

fix pt [-11.430952132988164, -11.430952132988164, 49] rho  50
eigenvalue  [-14.79293838 +0.j           0.56313585+13.27944826j
   0.56313585-13.27944826j]
unstable
```

4-3

In [37]:

```python
"""
This part of code is partially referenced from L18_simulateLORENZ.ipynb
Modeified by Henry Chang
"""
import numpy as np
import math
from matplotlib import pyplot as plt
from scipy.integrate import solve_ivp
from mpl_toolkits.mplot3d import Axes3D

def lorenz(t, y):
    # y is a three dimensional state-vector
    dy = [sigma * (y[1] - y[0]),
          y[0] * (rho - y[2]) - y[1],
          y[0] * y[1] - beta * y[2]]
    return np.array(dy)

# Lorenz's parameters (chaotic)
sigma = 10
beta = 8 / 3
rho_m = np.arange(5, 55, 5)

# fixed points
fix_pts = [[0, 0, 0],
           [math.sqrt(72), math.sqrt(72), 27],
           [-math.sqrt(72), -math.sqrt(72), 27]]

# Initial condition
y0 = [10, 10, 10]

# Compute trajectory
dt = 0.01
T = 20
num_time_pts = int(T / dt)
t = np.linspace(0, T, num_time_pts)
```

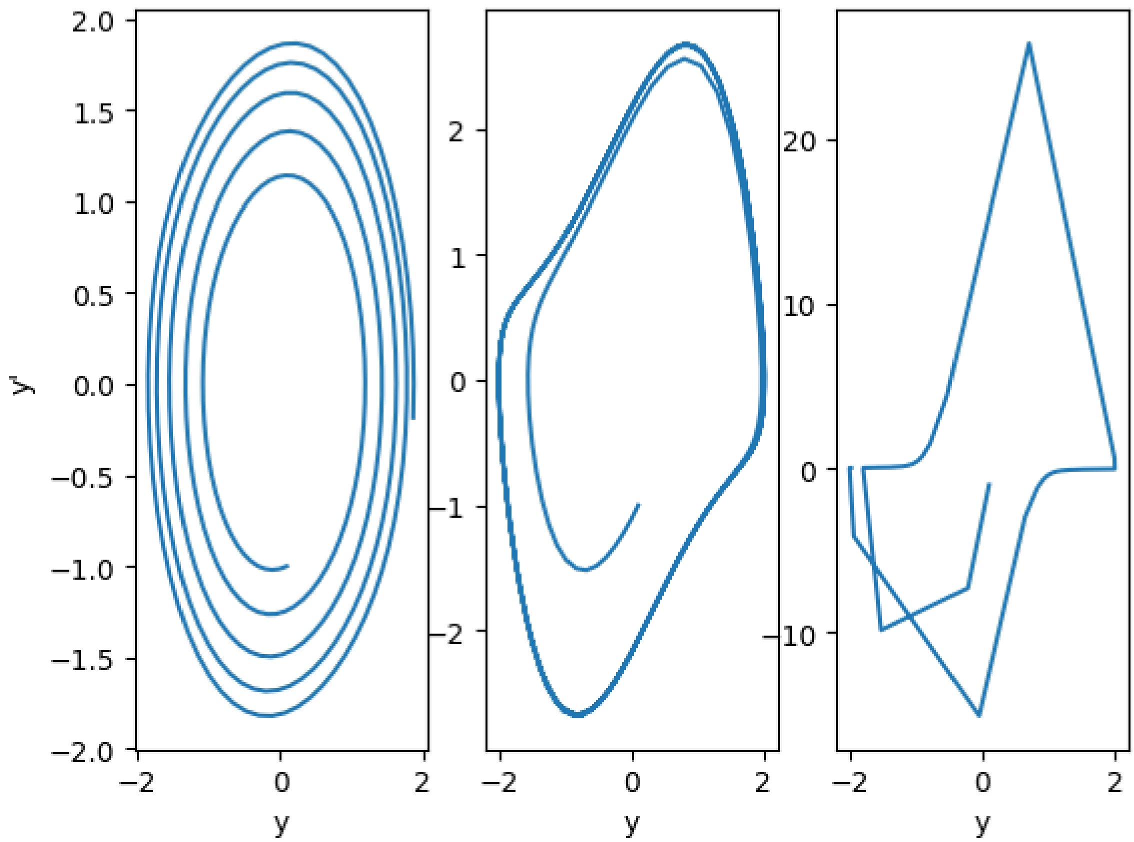
```
fig, axs = plt.subplots(2, 5, subplot_kw={'projection': '3d'})  # make a 3D plo
for i, rho in enumerate(rho_m):
    lorenz_solution = solve_ivp(lorenz, (0, T), y0, t_eval=t)
    t = lorenz_solution.t
    y = lorenz_solution.y.T
    row = (i) % 2
    col = (i) // 2
    axs[row, col].plot(y[:, 0], y[:, 1], y[:, 2])
    axs[row, col].set_title("rho={}".format(rho))
plt.show()

"""
clear large picture at the end
"""
```



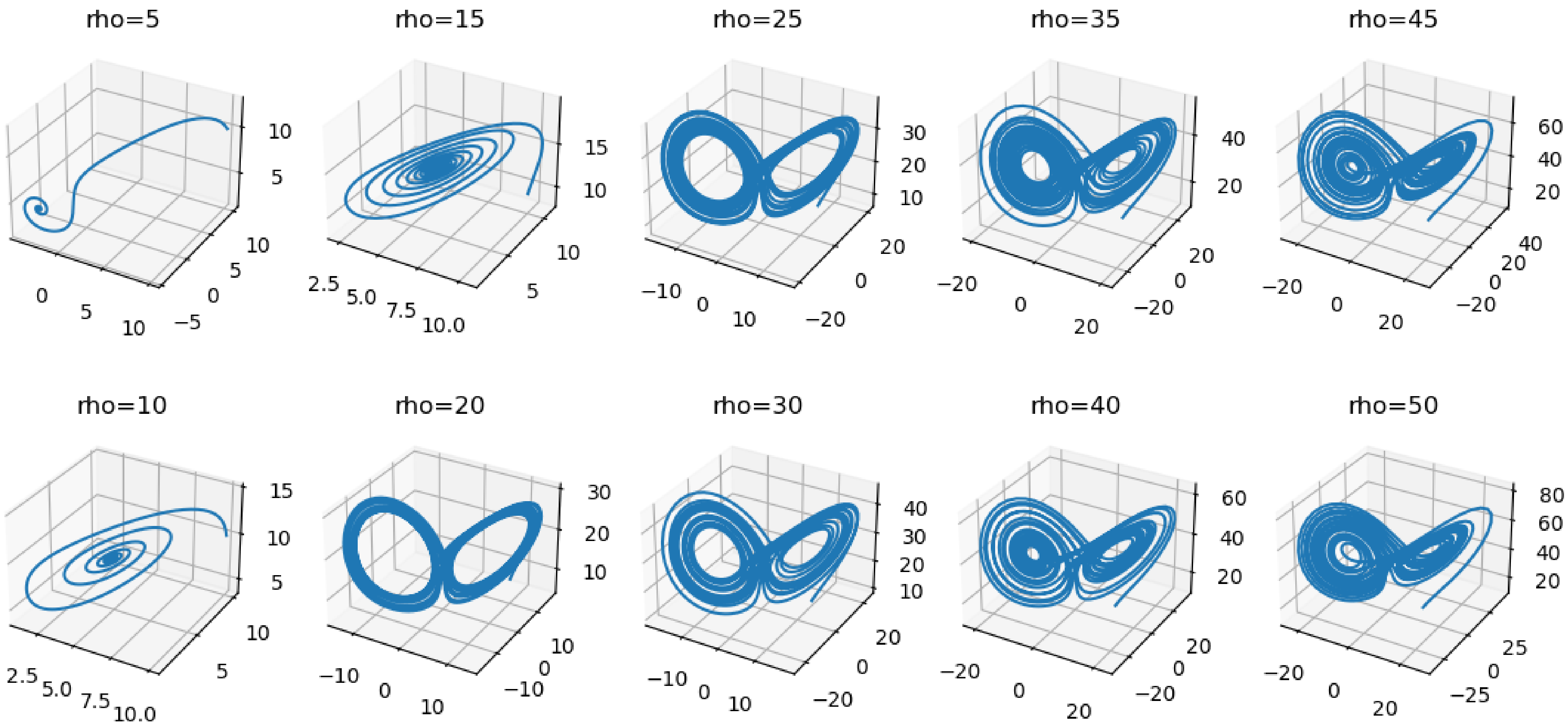
4-5

In [38]:
```
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import solve_ivp

# Parameters
epislons = [0.1, 1, 20]
epislon = epislons[0]

# Initial condition
y0 = [0.1, -1]

# Fixed point
fix_pt = [0, 0]

# Compute trajectory
dt = 0.1
T = 30
num_time_pts = int(T / dt)
```

```
t = np.linspace(0, T, num_time_pts)

def van_der_pol(t, y):
    dy = [y[1], -epislon*(y[0]**2-1)*y[1]-y[0]]
    return np.array(dy)

fig, axs = plt.subplots(1, 3)

for col, epislon in enumerate(epislons):
    solution = solve_ivp(van_der_pol, (0, T), y0, t_eval=t)
    y = solution.y.T
    axs[col].plot(y[:,0], y[:,1])
    axs[col].set_xlabel('y')
    axs[0].set_ylabel("y'")
```



In [ ]:


rho=5
rho=15
rho=25
rho=35
rho=45
rho=10
rho=20
rho=30
rho=40
rho=50