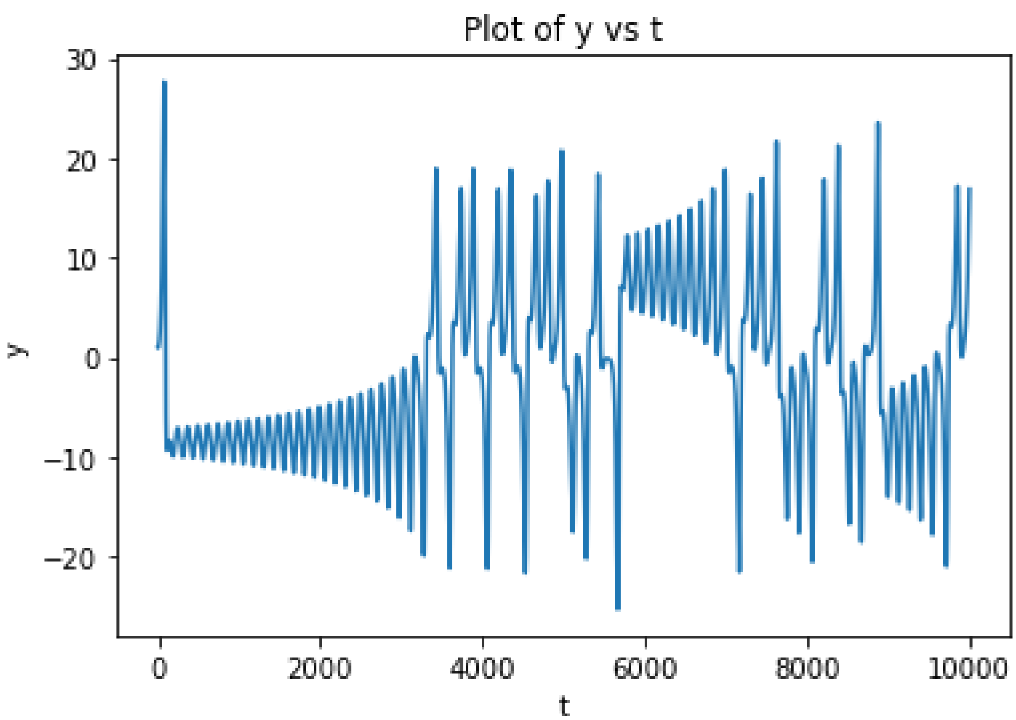
Exercise 3.2: The Lorenz equations

One of the most celebrated sets of differential equations in physics is the Lorenz equations:

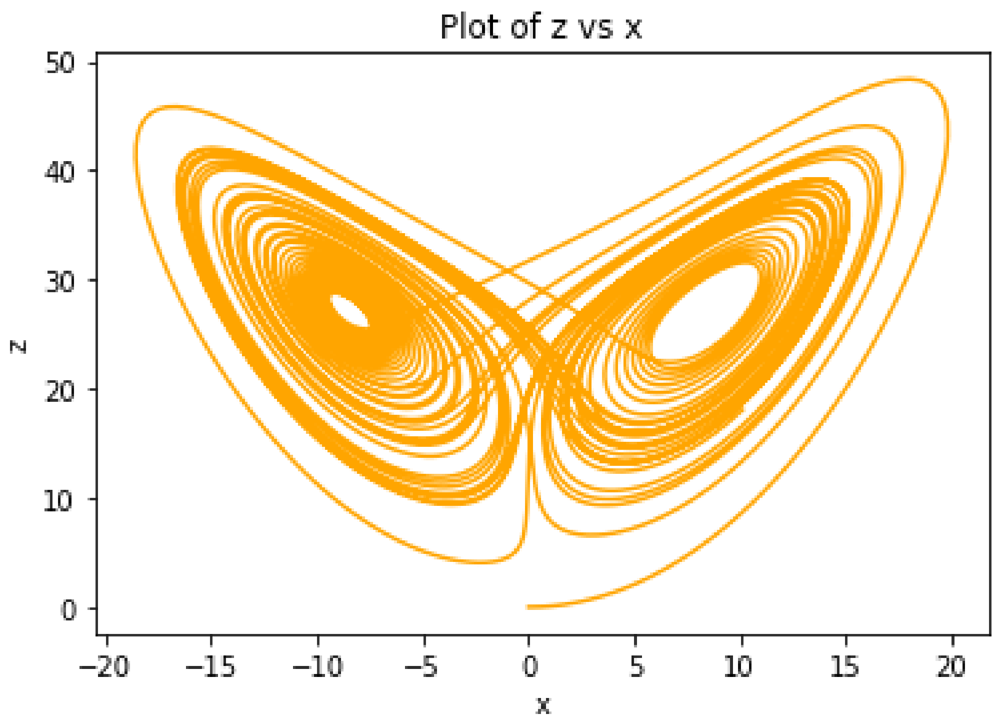
Given:

$$\frac{dx}{dt} = \sigma(y - x),$$
$$\frac{dy}{dt} = rx - y - xz,$$
$$\frac{dz}{dt} = xy - bz$$

where σ, r, and b are constants. These equations were first studied by Edward Lorenz in 1963, who derived them from a simplified model of weather patterns. The reason for their fame is that they were one of the first incontrovertible examples of deterministic chaos, the occurrence of apparently random motion even though there is no randomness built into the equations. Part(a):



Part(b):



Python code:

```
# -*- coding: utf-8 -*-
"""
@author: JH_Song
"""

from numpy import linspace
from matplotlib import pyplot as plt
from scipy.integrate import odeint

#initial conditions
initial_state = [0., 1., 0.]

#constant
sigma = 10.
r = 28
b = 8./3.

def f(state, t):
    [x,y,z] = state
    dxdt = sigma*(y - x)
    dydt = r*x - y - x*z
    dzdt = x*y - b*z
    return [dxdt, dydt, dzdt]

#set time steps
t_0 = 0   #start time
t_f = 50  #end time
N = 10000  #Number of steps generated by linspace()
tpoints = linspace(t_0, t_f, N)

result = odeint(f, initial_state, tpoints)
x = result[:,0]
```

```python
y = result[:,1]
z = result[:,2]

# Part(a)
plt.plot(y)
plt.xlabel('t')
plt.ylabel('y')
plt.title('Plot of y vs t')
plt.show()

# Part(b)
plt.plot(x, z, color='orange')
plt.xlabel('x')
plt.ylabel('z')
plt.title('Plot of z vs x')
plt.show()
```