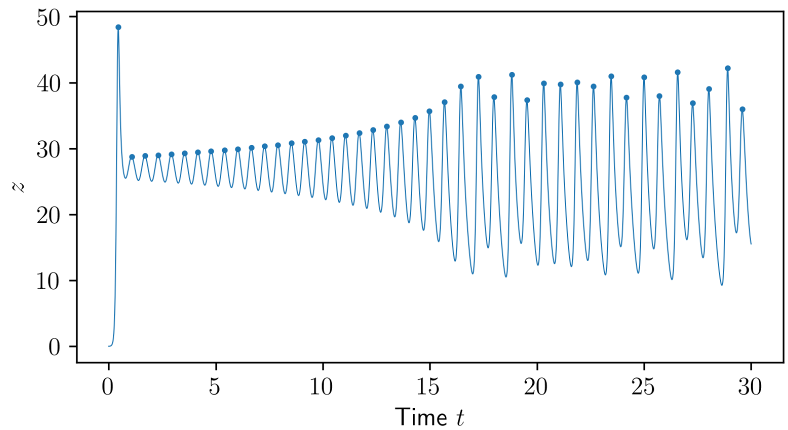
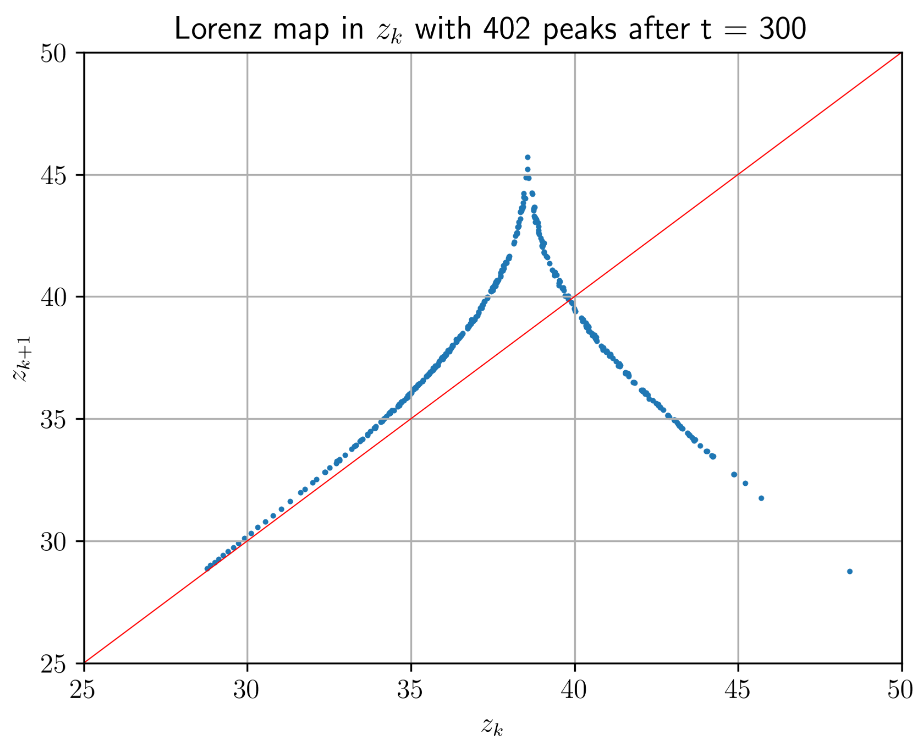
1. 'Almost Closed Orbits' in the Lorenz equations. The Lorenz equations

$$\dot{x} = \sigma(y - x) \tag{1a}$$
$$\dot{y} = rx - y - xz \tag{1b}$$
$$\dot{x} = xy - bz, \tag{1c}$$

to which we will add the initial condition $x(0) = 0, y(0) = 1, z(0) = 0$, admit chaotic solutions for the parameters $\sigma = 10, r = 28, b = 8/3$. Lorenz found that the z coordinate exhibits a repetitive 'peak-trough' pattern, as shown below.



Placing the peak values in a sequence $z_0, z_1, z_2, ...$, and plotting z_n against z_{n+1}, he came up with what is now known as the 'Lorenz map', shown below computed up to $t = 300$.



In this problem, we are interested in the point where the map appears to intersect the line $z_{n+1} = z_n$ with a negative slope.

(a) Compute a numerical solution of the Lorenz equations forward in time from $t = 0$ to $t = 300$, and store the solution using a vector of length (give or take one or two) 30,000 units.

(b) Using a function such as `scipy.signal.find_peaks` or MATLAB's `findpeaks`, determine the sequence of localized peak-values z_k. Your sequence should be approximately 400 entries long. Then, use this to reproduce the lorenz map as shown above.

(c) Find out which of the 400 or so entries in your sequence of z-peaks most nearly lies on the straight line $z_{n+1} = z_n$. None of the points in your sequence will lie exactly on it, but you should be able to quantitively determine which of the members of the sequence is closest to lying on the red line shown above. Hint: you may want to use the Python functions `numpy.argmin` and `numpy.diff`, or their equivalent in other programming languages. You have answered this part when you have found both a value of z_k and the index in the sequence of z-peaks where that value was found. This index can be anywhere between 1 and 400 or so.

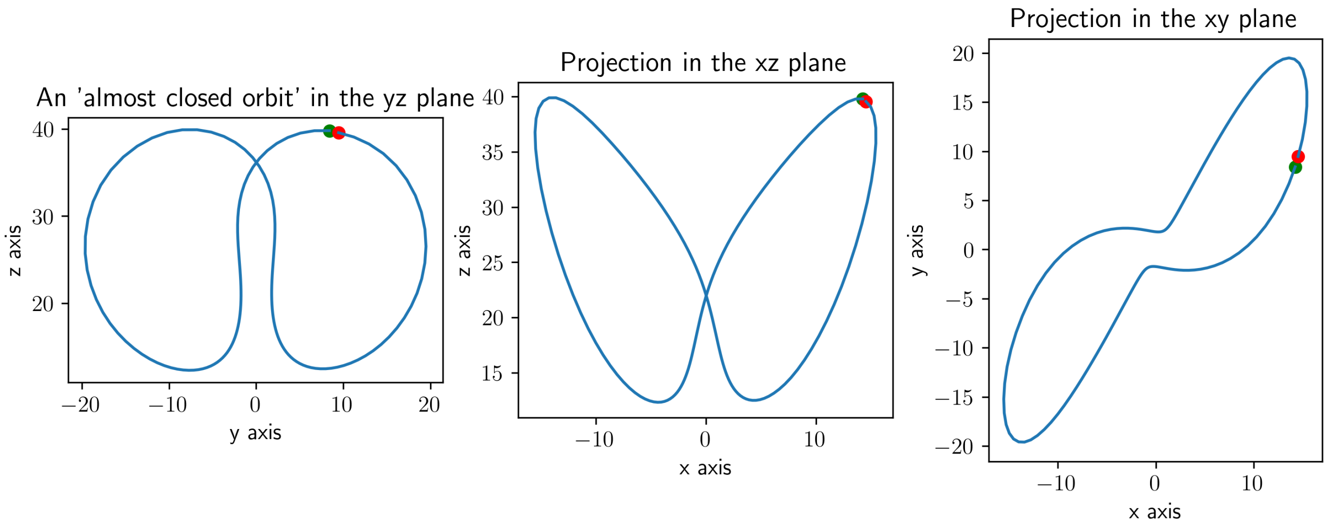
(d) Your numerical solution of the Lorenz equations has given you a time-series for $z(t)$, where both z and t are approximately 30,000 unit-long arrays. The answer you found to question 1c ultimately came from this time-series; find out the index of the time-series to which the answer to question 1c belongs. You have answered this part when you have found the (integer) value of the index, which can be anywhere between 1 and 30,000 or so, and the value of time (a real number between 0 and 300) that this index represents.

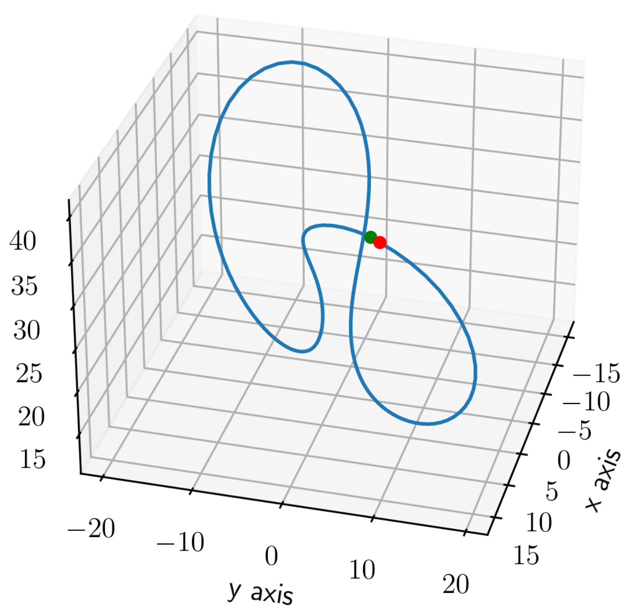
(e) Using your answer to question 1d, plot a subset of the numerical solution to Lorenz's equations starting from the point in time when the 'special' peak you found in 1c was reached. Your plots should start from the time you found in 1d, and should continue for approximately one "period".
Now, the Lorenz equations with these parameters do not show periodic behavior, so there is strictly speaking no 'period', i.e., there is no finite time after which a point in (x, y, z) space will come back to exactly the same point it was at at an earlier time. However, we can think of an *approximate* period by finding — using trial and error and numerics — the time it takes for a point in phase space to almost return to the point it started from. You have answered this part when you have approximately reproduced the figures shown below, and have reported the times (e.g., "from $t = 37.2$ to $t = 39.1$") over which your plots were made.

In Python, the following code can generate a solution of Lorenz's equations.

```python
import numpy as np
from scipy.integrate import solve_ivp

def lorenz_rhs(t,state):
    # Break up state variable into 3:
    x,y,z = state
    s = 10.0
    r = 28
    b = 8/3

    dxdt = s*(y-x)
    dydt = r*x - y - x*z
    dzdt = x*y - b*z

    return [dxdt,dydt,dzdt]

# Initial condition
y0 = [0.0, 1.0, 0.0]

# Time span for the solution (start time, end time)
t_span = (0, 300)

# Plot the solution at the following times:
tvals = np.linspace(0,t_span[1],30000)

# Solve the ODE
solution = solve_ivp(lorenz_rhs, t_span, y0, t_eval=tvals)
```

The figures you are expected to generate for 1e. Note that the green dot represents the start of this 'almost periodic orbit', and the red dot is the end.





2. Maps. Consider the map given by

$$x_{n+1} = r\sin(\pi x_n), \tag{2}$$

where $x \in [0, 1]$ and r is a parameter that can vary from 0 to 1.

(a) For small values of r, $x = 0$ is a stable fixed point. If r is increased beyond a certain value, the origin loses stability and becomes unstable. Write down an equation in r that, when solved, gives the precise value of r at which this transition happens.

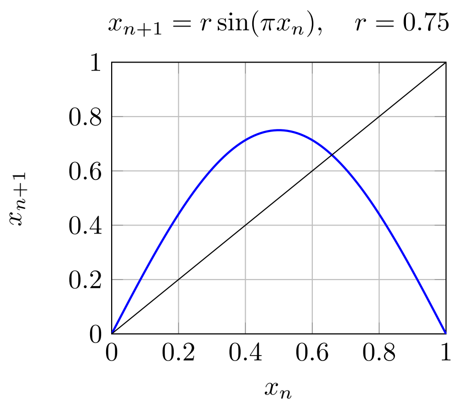
(b) For values of r slightly greater than the critical value you found in 2a, there are two fixed points in the map: $x = 0$ and one other value, let us call it x_a. Write down an equation that can be solved to find x_a in terms of r. You may or may not be able to solve this equation explicitly; for this part, you just need to write down the equation.

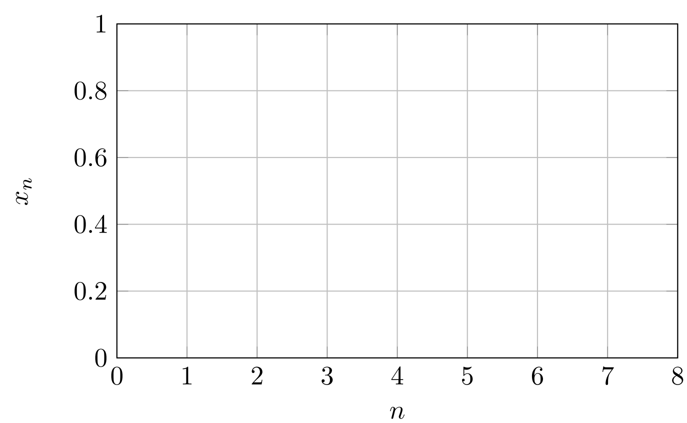
(c) Illustrate the solution to 2b for a single value of r graphically, i.e., sketch a graph and label where on the graph lies the solution to the equation you wrote down for 2b.

(d) Plot x_a as a function of r. On the horizontal axis, your graph should start from the value that you found in 2a, and should extend to $r = 1$. You may wish to do this by first writing a function (e.g., in Python) that gives the value of x_a given a value of r.

(e) Evaluate the stability of the fixed point x_a depending on r. Whether or not x_a depends on the value of r. Do this graphically by making a plot of $f'(x_a(r))$ against r. Once you do this, it will be possible to visually identify the value of r at which the stability of the fixed point x_a changes. You should also be able to write a program that finds the precise value of r at which this transition occurs; don't just eyeball it! You have answered this question once you have programmatically produced the desired plot, and have given the critical value of r to three decimal places.

(f) Consider the map (2) when $r = 0.75$. Draw a precise cobweb diagram on the given axes (or you can make your own copy) for this case, starting from $x = 0.9$ and iterating the map forward for about eight steps. What type of pattern emerges in the cobweb diagram? Sketch the time evolution of x_n vs. n on the given axes as well.





(g) Plot both the map $f(x_n)$ and the second-iterate map $f(f(x_n))$ together on the same axes along with the diagonal line $x_{n+1} = x_n$ for $r = 0.75$. On these axes, locate the structure that you found in 2f.

(h) For what value of r does this map have a period-4 orbit? You may need to perform some trial and error to decide this. For the value of r that you choose, plot the map $f(x_n)$, second-iterate map $f^2(x_n)$, and fourth iterate map $f^4(x_n)$ together, and show that the period-4 orbit corresponds to intersections of the fourth iterate map with the line $x_{n+1} = x_n$.

(i) Is there a value of r for which the map (2) exhibits chaos? Give an example of a value of r for which the map is chaotic, and illustrate this using (1) a cobweb diagram at that value of r, and (2) an orbit diagram for $0 < r < 1$.