Numerical exercise TFY4305/FY8910 Nonlinear dynamics 2024

Practical information

Your answer should take the form of a report in pdf format and a zip file containing all the code used to produce the results. Collaboration is encouraged, so feel free to discuss the problems with anyone. However, remember that you are to write both your report and code yourself. The report does not need to include details from the assignment, but should be written so that anyone with access to the assignment and your report could reproduce your work.

The problems are split up into two parts: "warm-up" and "challenge". For the warm-up problems corresponding figures similar to the ones you should reproduce are given in the assignment. These problems are meant to give an introduction to the different concepts. The challenge problems do not have given examples, but they are closely linked to the warm-up problems. If things seem too difficult, you should focus on doing well on the warm-up problems, rather than a half-hearted attempt on all problems. The Strogatz book is also a valuable resource for some of the problems.

Feel free to program in any language you like. Together with the assignment on Blackboard there exist an example of how to solve the Lorenz equations by using an implementation of the 4th order Runge-Kutta routine in Python. Feel free to use the example as a basis in your own implementation. I would recommend that you try to write general code as there is a possibility to reuse code to solve similar tasks for different problems and systems.

The project is about programming. There is no points in making the report longer than necessary. Excluding figures, the text should not exceed 5 pages. Less is preferred..

Good luck!

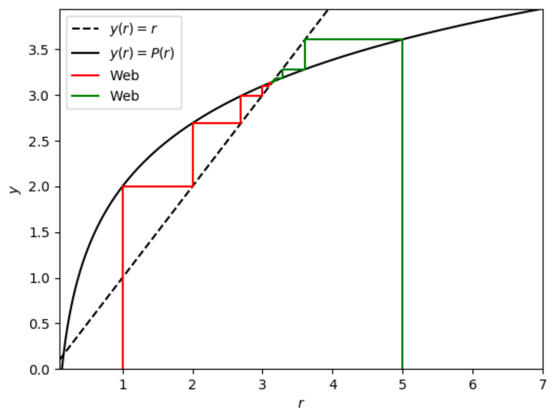


Figure 1: Illustration of the cobweb construction. Starting out at $r_0 = 1$ (the red line), we move vertically until we hit the Poincaré map $P(r)$. Then we move horizontally until the line $y = r$ is intersected. Continuing this process, we see that the cobweb converges towards $r \sim 3$. If we start at $r_0 = 5$ (the green line), we see that the web still converges to the point $r \sim 3$. Hence the system has a stable limit cycle at that point.

Problems

Problem 1 - Poincaré maps, cobweb constructions and stable limit cycles.

Preliminaries: This problem is based upon sections 8.5 and 8.7 in Strogatz' book.

Method: In n dimensions, a Poincaré map $P(\mathbf{x})$ is a mapping from an $n-1$ dimensional surface S to S itself. If we define the vector $\mathbf{x}_k$ as the k-th intersection between some vector field and S, we have

$$\mathbf{x}_{k+1} = P(\mathbf{x}_k). \tag{1}$$

Given a Poincaré map $P(r)$, the cobweb construction can be computed as follows: First, plot a line $y = r$. Then, start at a point $r = r_0$ and move vertically until you hit $P(r)$. From this point, move horizontally until you intersect the line y. Continue this process for as many steps as you like. If the cobweb converges to a point, the system has a stable limit-cycle at that point. The cobweb construction is demonstrated in Figure 1.

Warm-up:

a) Do the cobweb analysis in example 8.7.1 in Strogatz' book numerically, with $\dot{\theta} = 4\pi$ (in the book one uses $\dot{\theta} = 1$).[1] (Hint: instead of evaluating the integral, one may note that this switch changes all factors of 4π to 1). Use initial conditions $r_0 < 1$ and $r_0 > 1$ to show that the system has a stable limit cycle at $r = 1$.

b) Plot the first 30 values of the iterative map starting from $r_0 = 0.1$. By this we mean that we plot the value of r as a function of iterations done in the web (see Figure 2).

Challenge: Consider the vector field given by

$$\dot{r} = A(r - \pi)e^{-\omega t}, \qquad \dot{\theta} = \pi. \tag{2}$$

a) Find the Poincaré map $P(r)$.

b) Just looking at the map $P(r)$, postulate for which values of A and ω the system exhibits a stable limit cycle.

c) Plot the cobweb for a few values of A and ω to "confirm" your hypothesis.

[1] The reason we use this value for the time derivative is because it gives "nicer" plots.

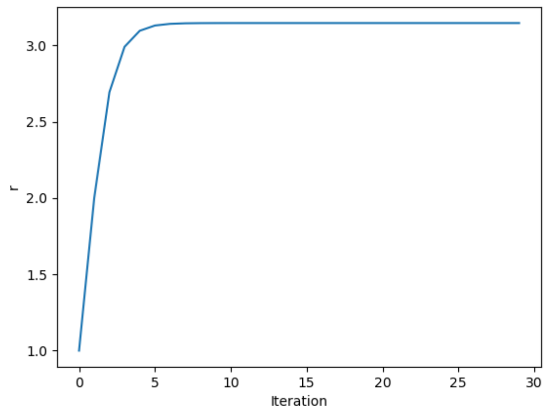


Figure 2: Example of how to plot the iterative map as a function of iterations. This plot corresponds to the red line in figure 1. The plot shows that a particle starting out at $r_0 = 0.1$ ends up rotating in a circle with stable radius $r \sim 3$ if we just wait long enough.

Problem 2 - The Lorenz equations

Preliminaries: This problem is based on chapter 9 in the book.[2]

Method: The Lorenz equations are a set of three coupled differential equations,

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{aligned} \tag{3}$$

where σ, ρ and β are free parameters. In particular, we will consider

$$\begin{aligned} \sigma &= 10, \\ \rho &= 28, \\ \beta &= \frac{8}{3}, \end{aligned} \tag{4}$$

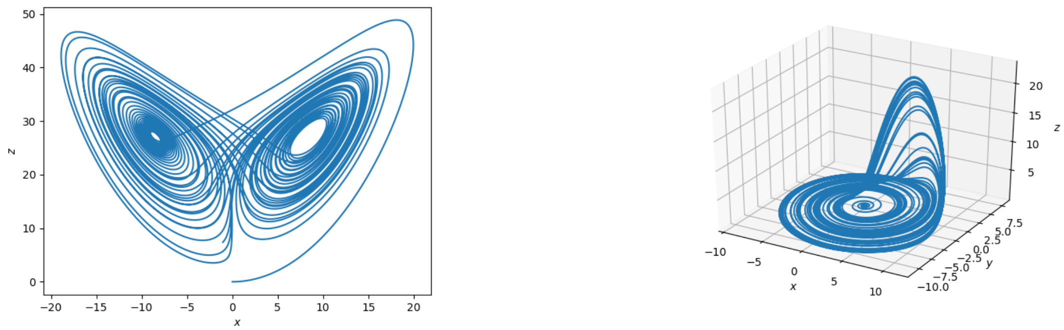
for which the system is found to have some interesting chaotic behaviour. One can solve these equations by numerical integration. An example of this is given on Blackboard where a 4th-order Runge-Kutta routine is used.

Warm-up: Use the initial conditions

$$\begin{aligned} x_0 &= 0.001, \\ y_0 &= 0.001, \\ z_0 &= 0.001, \end{aligned} \tag{5}$$

and solve the Lorenz equations up to the dimensionless time $T = 50$. You can integrate using whatever scheme or integrator you like, however, I encourage you *not* to use the naive $x_{n+1} = x_n + \dot{x}\Delta t$. Plot your solution in the xz-plane. You should comment on your choice for time-step (or tolerance if you use a variable stepsize integrator). If done correctly, the result will resemble Figure 3a.

[2]This does not necessarily mean than one has to read the whole chapter to do the assignment.



(a) Lorentz system

(b) Rössler system

Figure 3: Example plots of the Lorenz and Rössler systems.

Challenge: We will in this task try to subtract some information about the Lorenz equations. These equations make up what is called a strange attractor. We call a closed set A an attractor if

1. Any trajectory that starts in A, stays in A at all times.
2. If sufficiently close, any trajectory will eventually end up in A.
3. There is no proper subset of A that satisfies 1. and 2.

A *strange* attractor is an attractor which is sensitive to initial conditions. What this means is that two trajectories starting very close to each other will diverge and have two completely different futures. In this task we will investigate this property of the Lorenz set.

a) First plot z as a function of time. You will see that even though this plot is chaotic, there might be some order to it.

b) Let z_n be the n-th maxima of z for the Lorenz equations. Integrate the Lorenz equations for a large time $T \sim 10^5$ and plot z_{n+1} as a function of z_n. Use the plot to give an argument for why there cannot exist any stable limit cycles.

Two trajectories starting close to each other should diverge quickly away from one another on the strange attractor. In general, the separation distance $||\delta(t)||$ between the two endpoints $\mathbf{x}_1$ and $\mathbf{x}_0$ on two different trajectories (see Figure 9.3.4 on page 328 in the book) increases exponentially, and can be approximated by

$$||\delta(t)|| \equiv ||\mathbf{x}_1 - \mathbf{x}_0|| \sim \delta_0 \mathrm{e}^{\lambda t}, \tag{6}$$

where the constant λ often is referred to as the Liapunov exponent. The last part of this task considers how we can estimate λ. One way to do this is the following:

i) We need to start our analysis *on* the strange attractor. To enforce this, start from a random point near the origin ($\mathbf{x}_0 \sim 10^{-6}$) and integrate the Lorenz equations out to about $T = 50$. We will call this point $\mathbf{x}_0$.

ii) Create a random point $\mathbf{x}_1$ near $\mathbf{x}_0$ so that $||\mathbf{x}_1 - \mathbf{x}_0|| = 10^{-6}$.

iii) Integrate the Lorenz equations further over a time interval of 30 using both the initial conditions $\mathbf{x}_0$ and $\mathbf{x}_1$. Save the separation distance between the endpoints at each iteration in a vector. Then plot the logarithm of the elements in the vector as a function of dimensionless time. You should observe that the separation distance grows exponentially[3] until some cutoff time T_c where it flattens out. Can you think of why?

iv) Repeat steps i)-iv) a few times and establish roughly at what time T_c the cutoff occurs. You should aim to set T_c so that you are sure not to include the flat region, as we want to do a linear fit to the curve later.

v) Repeat steps i)-iii), now only integrating up to the cutoff time T_c. Use some kind of linear fitting to find the slope of the separation distance plot. This will be your Liapunov exponent λ.

vi) Using different initial conditions, you will get different values for λ. We are interested in some average over a vast number of different Liapunov exponents.

c) Estimate the average Liapunov exponent λ, and plot (in the same figure) all the different separation distances that you used to estimate λ, together with the "line" given by

$$\delta_0 \mathrm{e}^{\lambda t}. \tag{7}$$

(Hint: For small values of dt you may run out of memory if you plot many of the graphs in the same plot. This is because each curve may contain millions of data points. However, you can easily plot only a fraction of these data points and still get a nice curve.)

Problem 3 - The Rössler system

Preliminaries: It might be instructive to read chapter 12.3 for this task.

Warm-up: The Rössler system is defined by

$$\begin{aligned} \dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c). \end{aligned} \tag{8}$$

Use the values

$$a = 0.2, \qquad b = 0.2, \qquad c = 5.7, \tag{9}$$

with initial conditions equal to the ones used for the Lorenz set and integrate the equations. Make a plot of the system in the xz-plane. If you want to, you can also make a 3D-plot, just because the graph is cool. You should note that it might be necessary to integrate for a longer time T when working with the Rössler system compared to the Lorenz system. An example plot is shown in Figure 3b.

Challenge: Do all the steps in task 2 for the Rössler system. It should be possible to reuse most of your code. The task here is to compare the solutions of the Rössler system with the solutions to the Lorenz equations. Possible things to compare may be, but are not limited to,

* The shape of the graph
* The Liapunov exponent
* The maxima z_{n+1} as a function of z_n

Done!

[3] In other words you should see a linear growth in the log-plot.