UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

NONLINEAR SYSTEMS, CHAOS AND CONTROL IN ENGINEERING

ASSIGMENTS

Author: IVÁN SERMANOUKIAN MOLINA

Professor/s: CRISTINA MASOLLER
ANTONIO JAVIER PONS RIVERO

Bachelor's degree in Aerospace Technologies Engineering

Polytechnic University of Catalonia

April 21, 2021

Contents

1 Exercises for two-dimensional systems

1.1 Exercise 1

Prove the following statement for a linear system of the form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.1}$$

If $\mathbf{c_1} = (x_1 y_1)^T$ and $\mathbf{c_2} = (x_2 y_2)^T$ are solutions of the equation, then $\mathbf{c_3} = \alpha\mathbf{c_1} + \beta\mathbf{c_2}$ is also a solution of the equation.

The same system can be expressed as a system of linear homogeneous differential equations:

$$\dot{x} = ax + by \quad \longrightarrow \quad \dot{x} - ax - by = 0 \tag{1.2}$$
$$\dot{y} = cx + dy \quad \longrightarrow \quad \dot{y} - ax - by = 0 \tag{1.3}$$

Considering that $\mathbf{c_1}$ and $\mathbf{c_2}$ are solutions.

$$\dot{x_1} - ax_1 - by_1 = 0 \quad \text{and} \quad \dot{y_1} - cx_1 - dy_1 = 0 \tag{1.4}$$
$$\dot{x_2} - ax_2 - by_2 = 0 \quad \text{and} \quad \dot{y_2} - cx_2 - dy_2 = 0 \tag{1.5}$$

We have to demonstrate that

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \alpha \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \beta \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{1.6}$$

which can also be expressed as

$$\begin{cases} x_3 & = \alpha x_1 + \beta x_2 \\ y_3 & = \alpha y_1 + \beta y_2 \end{cases} \tag{1.7}$$

Then, if we multiply 1.4 and 1.5 by random constants and sum the results equalised to zero we have

$$k_1 \cdot (\dot{x_1} - ax_1 - by_1) + k_2 \cdot (\dot{x_2} - ax_2 - by_2) = 0 \tag{1.8}$$

which can be reformulated to obtain

$$k_1\dot{x_1} + k_2\dot{x_2} + a(k_1x_1 + k_2x_2) + b(k_1y_1 + k_2y_2) = 0 \tag{1.9}$$

If k_1 and k_2 are substituted by α and β and the equation is rearranged

$$[\alpha x_1 + \beta x_2]' + a(\alpha x_1 + \beta x_2) + b(\alpha y_1 + \beta y_2) = 0 \tag{1.10}$$

Then, if x_3 is added

$$x_3' + a(x_3) + b(y_3) = 0 \tag{1.11}$$

The same method can be followed from 1.6 to 1.12 to obtain

$$[\alpha y_1 + \beta y_2]^{'} + c(\alpha x_1 + \beta x_2) + d(\alpha y_1 + \beta y_2) = 0 \tag{1.12}$$

Which can be transformed adding the y_3 expression to obtain

$$y_3^{'} + c(x_3) + d(y_3) = 0 \tag{1.13}$$

As both equations yield the same result, then for all $\alpha, \beta \in \mathcal{R}$, $\mathbf{c_3}$ is a solution.

It could also be solved vectorially with

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \quad A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{1.14}$$

Then 1.14 is equivalent to

$$\dot{z} = Az \tag{1.15}$$

Assuming that z_1, z_2 are solutions of 1.15, i.e., $\dot{z}_1 = Az_1$ and $\dot{z}_2 = Az_2$. Then for all $\alpha, \beta \in \mathbb{R}, z = \alpha z_1 + \beta z_2$ is also a solution. Indeed,

$$\dot{z} = \alpha \dot{z}_1 + \beta \dot{z}_2 = \alpha A z_1 + \beta A z_2 = A\left(\alpha z_1 + \beta z_2\right) = Az \tag{1.16}$$

1.2 Exercise 2

For the initial conditions $(x_1(0), y_1(0)) = (-1.25, 1.0)$ and $(x_2(0), y_2(0)) = (0.2, 1.5)$, solve numerically (using Matlab) the following systems:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.17}$$

and

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.18}$$

Show, together, the time evolution of variables $x_i(t)$ and $y_i(t)$. One plot for each system. Then, plot, also, $y_i(t)$ as a function $x_i(t)$. Now we are going to consider the evolution in time of a linear combination of the two solutions: Set as an initial condition $(x_3(0), y_3(0)) = (\alpha x_1(0) + \beta x_2(0), \alpha y_1(0) + \beta y_2(0))$ for some α and β of your choice and solve numerically again the time evolution of the systems. Now compare the solutions, $(x_3(t), y_3(t))$ obtained numerically with the combination $(\alpha x_1(t) + \beta x_2(t), \alpha y_1(t) + \beta y_2(t))$ plotting them together in a single graph.

Firstly, we have to take into consideration that equations are written as

$$\begin{cases} \dot{x} = a \cdot x + b \cdot y \\ \dot{y} = c \cdot x + d \cdot y \end{cases}$$

which can also be expressed as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

with the following reduced form

$$\dot{\mathbf{X}} = \mathbf{A} \cdot \mathbf{X}$$

The linear evolution of variables $x_i(t)$ and $y_i(t)$ for the first system with the first initial conditions is plotted in the following graph:

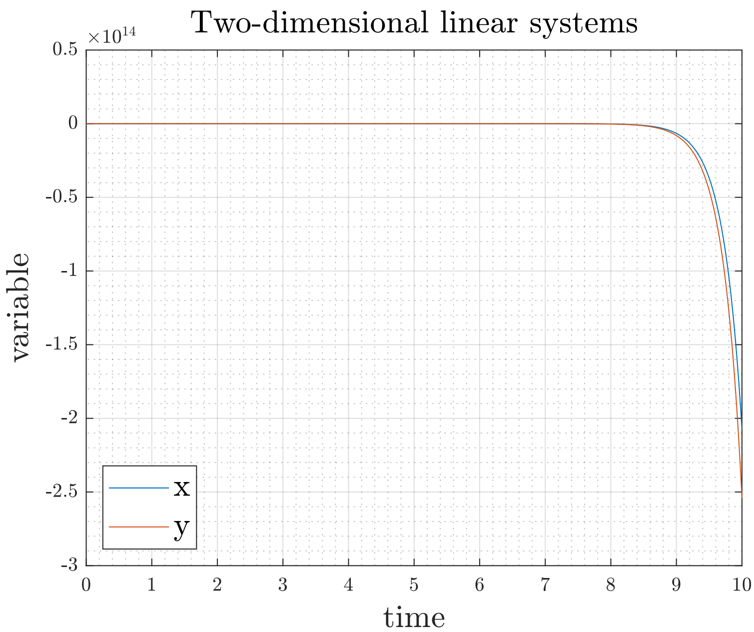

Figure 1 Evolution with time of two-dimensional linear systems

An exponential decrease without oscillations occurs at $t = 8.5$. There is a total decrease of $3 \cdot 10^{14}$ in 1.5 time units. Both x and y decrease similarly. However, y decreases a bit faster maintaining a constant difference with x.

Then, for the first system with the first option of initial conditions, if $y_i(t)$ is plotted as a function of $x_i(t)$ the following results are obtained:

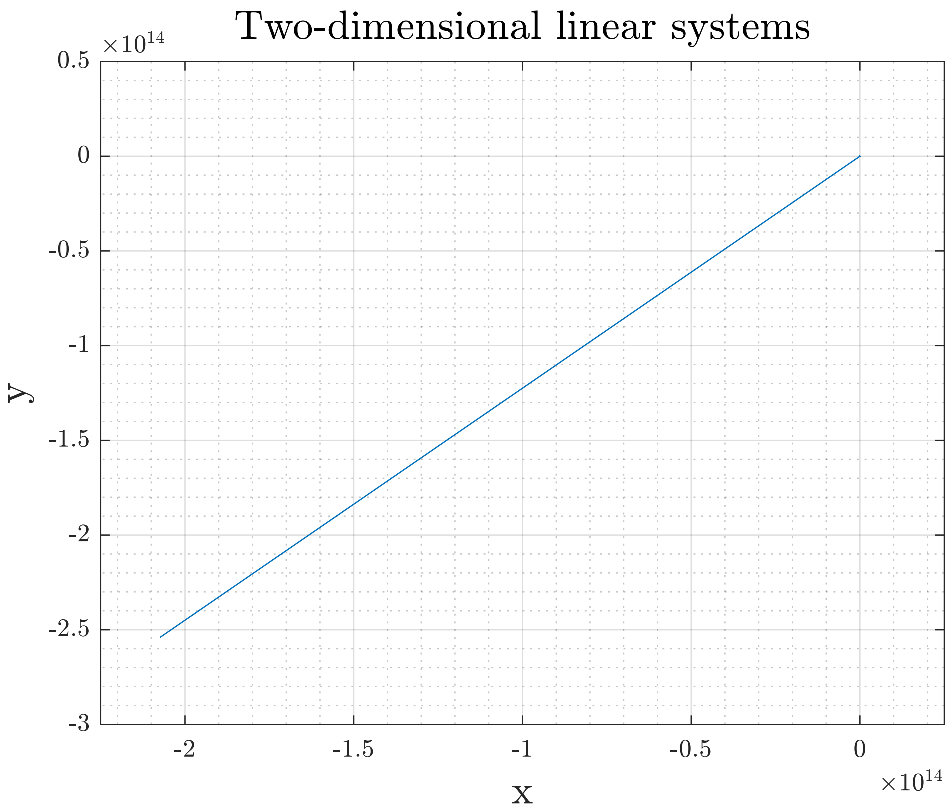

Figure 2 First system phase portrait

Analysing the previous graph for the first system, $x(t)$ and $y(t)$ are linearly dependant, which means that both increase or decrease accordingly. Thus, as the system decreases exponentially without oscillations this result could have been foreseen.

Subsequently, for the first system the second option of initial conditions are considered for the following two graphs, firstly the trajectories and then the phase portrait:

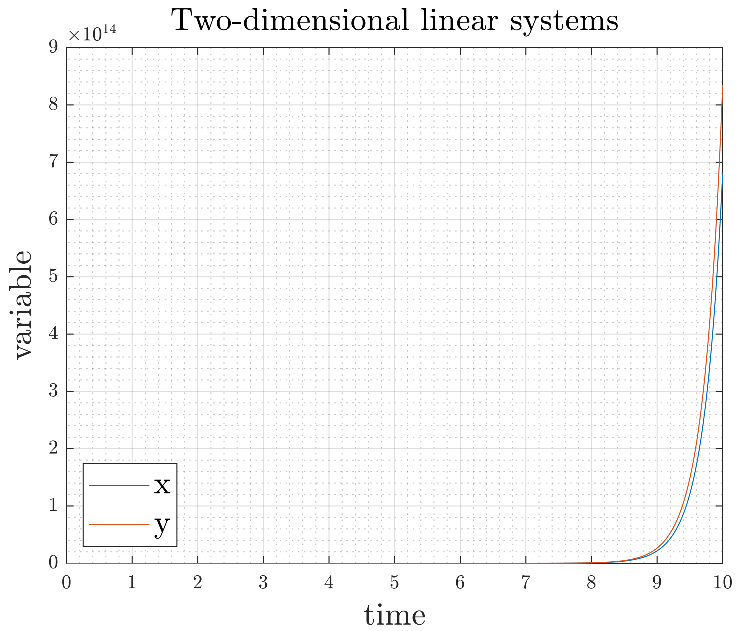

Figure 3 Evolution with time of two-dimensional linear systems

In this case instead of an exponential decrease almost at the end of the time span, an exponential increase occurs at the same time.

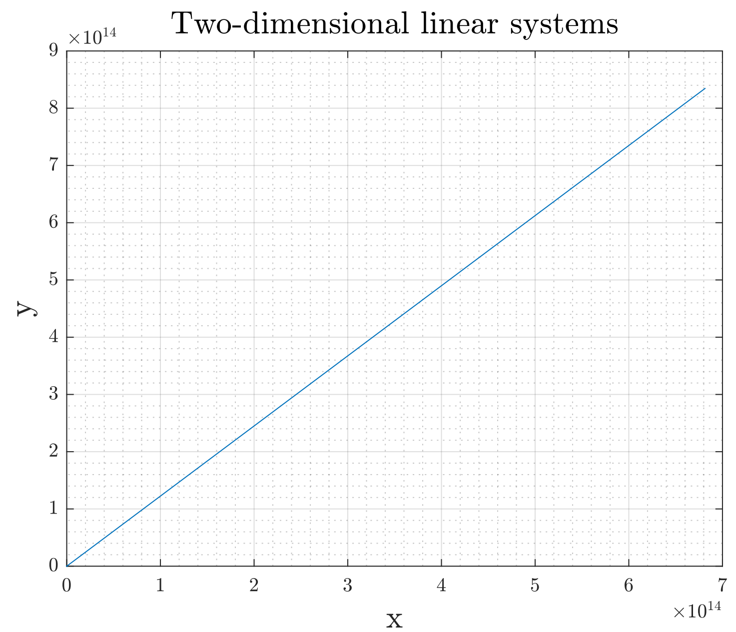

Figure 4 First system phase portrait

If this graph is opposed to the other initial conditions, this case turns up positive really fast while the other is reduced to negative values with a third absolute maximum value.

The linear evolution of variables $x_i(t)$ and $y_i(t)$ for the second system with the first initial conditions is plotted in the following graph:

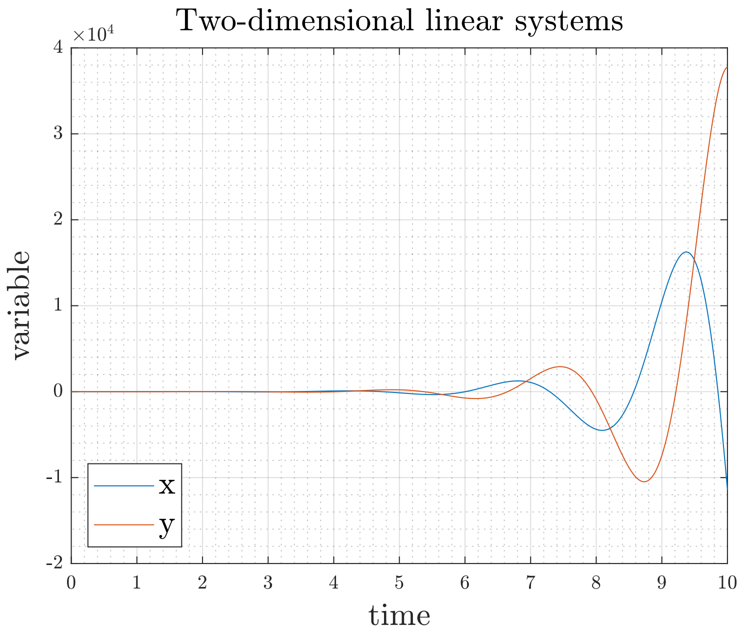

Figure 5 Evolution with time of two-dimensional linear systems

An exponential bidirectional fluctuation with oscillations occurs at $t = 5$. The fluctuation period of x and y are different causing a lagged result in which y amplitude is wider than x for the first iterations.

The phase portrait of the second system with the first initial conditions results in the following chart:

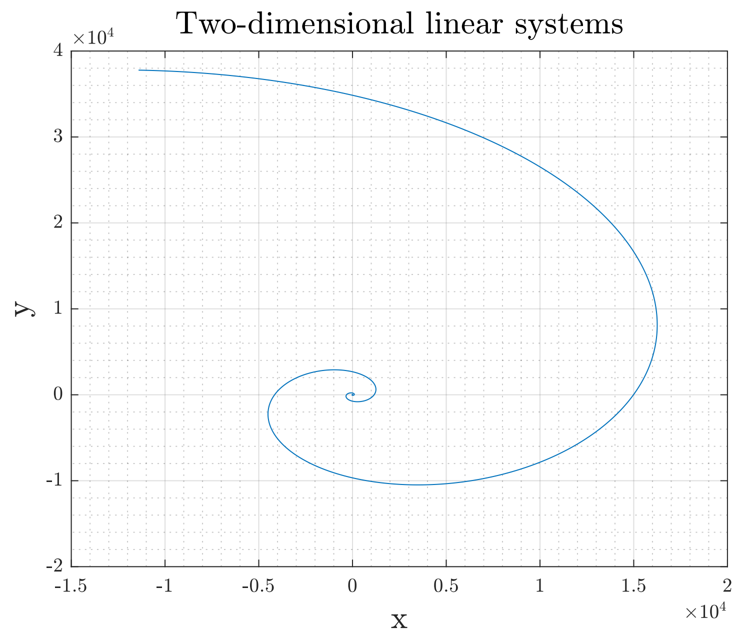

Figure 6 Second system phase portrait

In this case, opposed to the first case, the oscillations alter the linear result to obtain a spiral source, also known as unstable spiral. This graph reminds the reader of a Fibonacci spiral, an approximation of the golden spiral created by drawing circular arcs to connect several areas. The linear evolution of variables

$x_i(t)$ and $y_i(t)$ for the second system with the second initial conditions is plotted in the following graph:

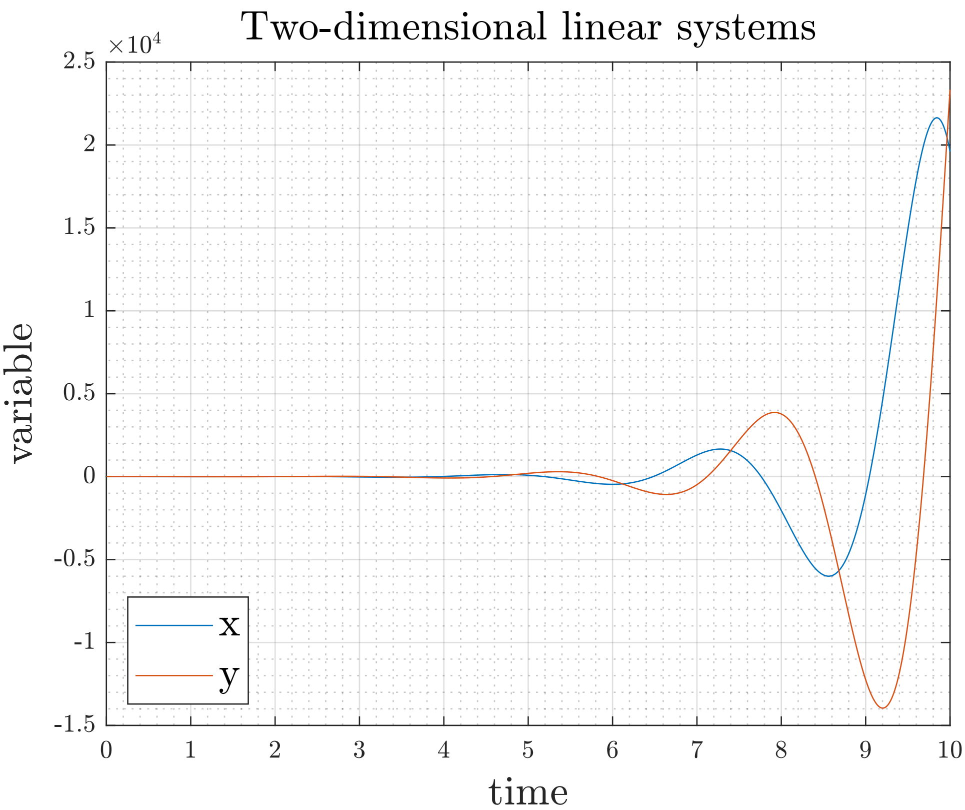

Figure 7 Evolution with time of two-dimensional linear systems

Regarding this graph, the same results as the other initial conditions are obtained with different amplitudes of the oscillations.

Following with phase portraits, the second system with the second initial conditions results in the following chart:

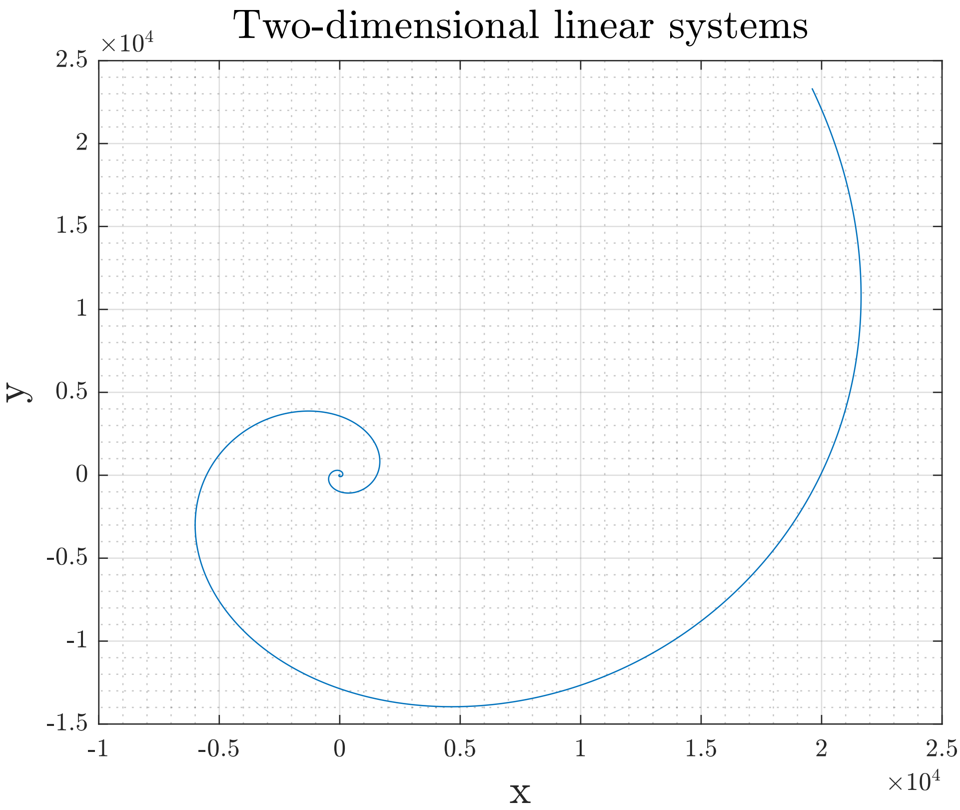

Figure 8 Second system phase portrait

As with the previous oscillations, the same happens with the phase portrait.

These results could have been predicted by looking at the A matrix. The first case cannot produce oscillations because both terms multiplying x and y are always positive and, in that case, they tend to either $+\infty$ or $-\infty$ depending on the initial conditions. However, as the second case has a negative sign, oscillations will be produced throughout the way to infinity.

For the case of a linear combination of the two solutions, the following system is considered:

$$x_3(0) = \alpha x_1(0) + \beta x_2(0) \tag{1.19}$$

$$y_3(0) = \alpha y_1(0) + \beta y_2(0) \tag{1.20}$$

where α and β are random numbers. For this analysis, $\alpha = 1.9$ and $\beta = 14$.

This results in the following set of graphs:

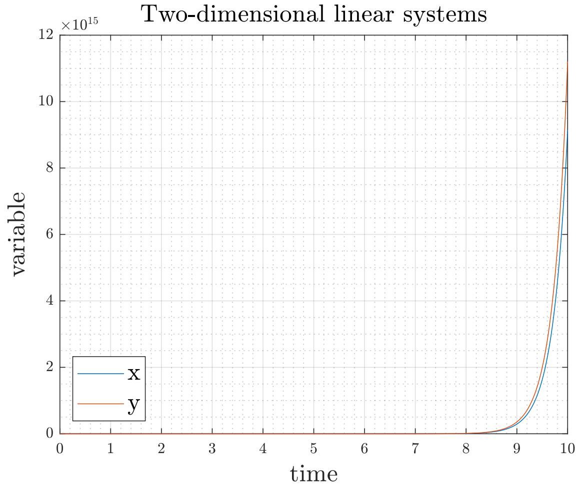

Figure 9 Evolution with time of the first system

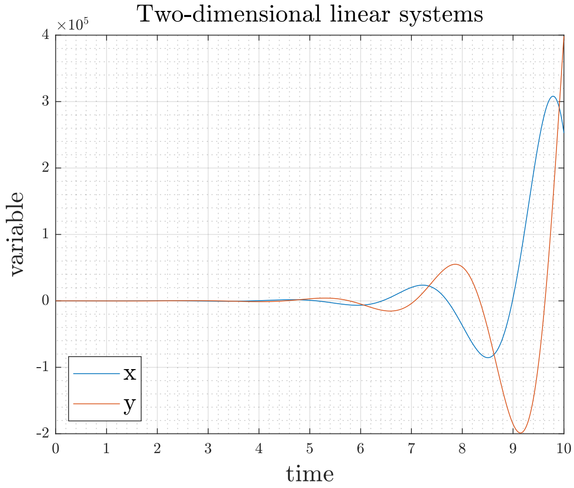

Figure 10 Evolution with time of the second system

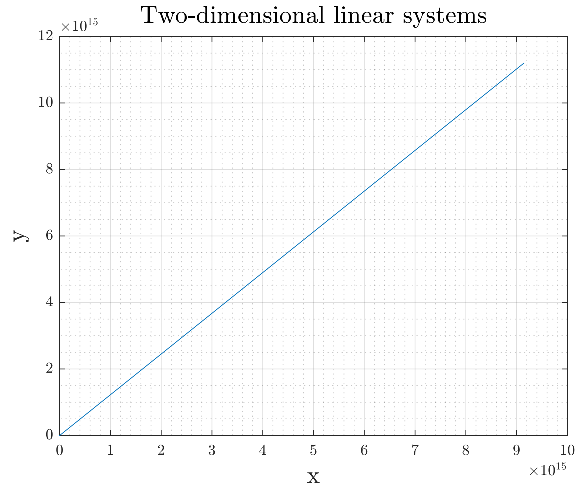

Figure 11 First system phase portrait

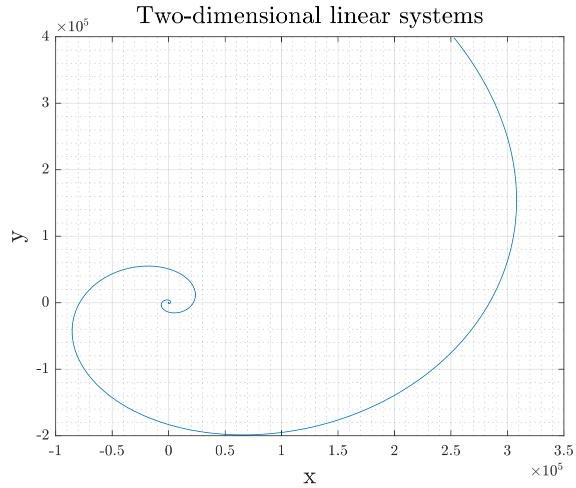

Figure 12 Second system phase portrait

Finally, if it is compared to the system in which $(\alpha x_1(t) + \beta x_2(t), \alpha y_1(t) + \beta y_2(t))$ and plotted together in a single graph:

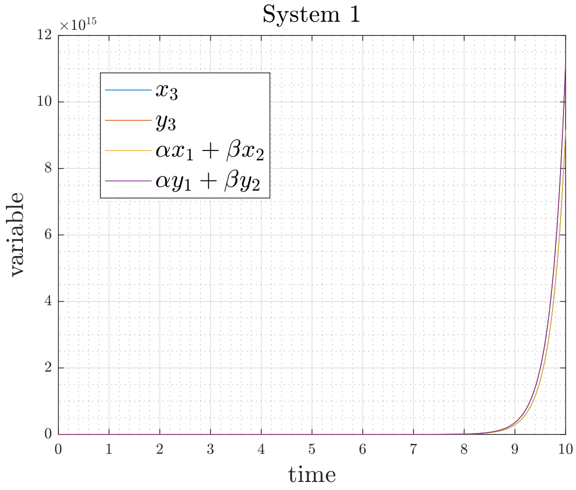

Figure 13 Evolution with time of the first system

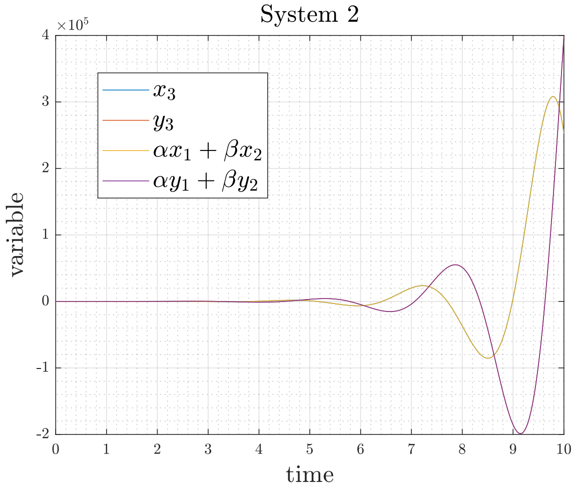

Figure 14 Evolution with time of the second system

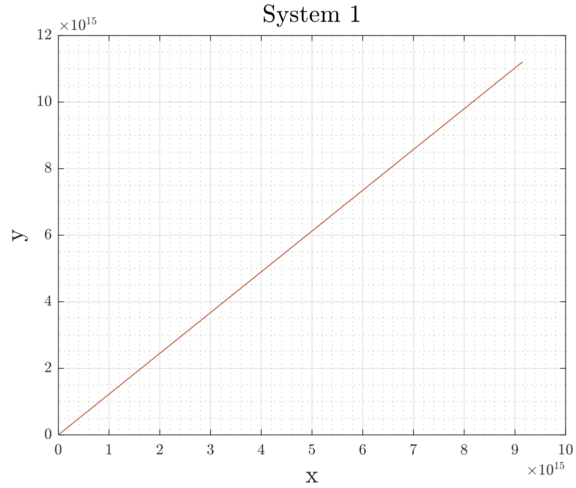

Figure 15 First system phase portrait

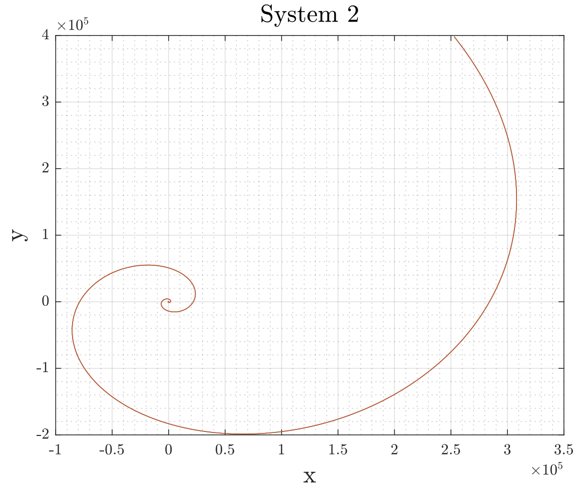

Figure 16 Second system phase portrait

These graphs confirm that the evolution in time of the linear combination of the initial conditions is the same as the linear combination of the solutions.

1.3 Exercise 3

Find all the fixed points of the system

$$\begin{aligned} \dot{x} &= -x + 4x^3 \\ \dot{y} &= -2x \end{aligned} \tag{1.21}$$

and use its linearisation to classify them. Considering different initial conditions, derive, numerically, the phase portrait for the full nonlinear system. Hint: There are (many!!) different fixed points.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} +1 & +1 \\ +4 & -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.22}$$

In order to find the fixed points, 1.22 is equalised to zero

$$\begin{cases} -x + 4x^3 & = 0 \\ -2x & = 0 \end{cases} \tag{1.23}$$

to find the following fixed points

$$(0, A) \qquad \left(\frac{1}{2}, B\right) \qquad \left(-\frac{1}{2}, C\right) \tag{1.24}$$

where A,B,C are values $\in$ R

Then, the Jacobian matrix is

$$\text{J} = \begin{pmatrix} -1 + 12x^2 & 0 \\ -2 & 0 \end{pmatrix} \tag{1.25}$$

with

$$\tau = -1 + 12x^2 \quad \text{and} \quad \triangle = 0 \tag{1.26}$$

which can be evaluated at the Fixed Points to characterise the system.

$$\text{For } (0, A) \text{ we have } \tau = -1 \text{ and } \triangle = 0 \tag{1.27}$$

$$\text{For } \left(\frac{1}{2}, B\right) \text{ we have } \tau = 2 \text{ and } \triangle = 0 \tag{1.28}$$

$$\text{For } \left(-\frac{1}{2}, C\right) \text{ we have } \tau = 2 \text{ and } \triangle = 0 \tag{1.29}$$

Thus, based on the following image:

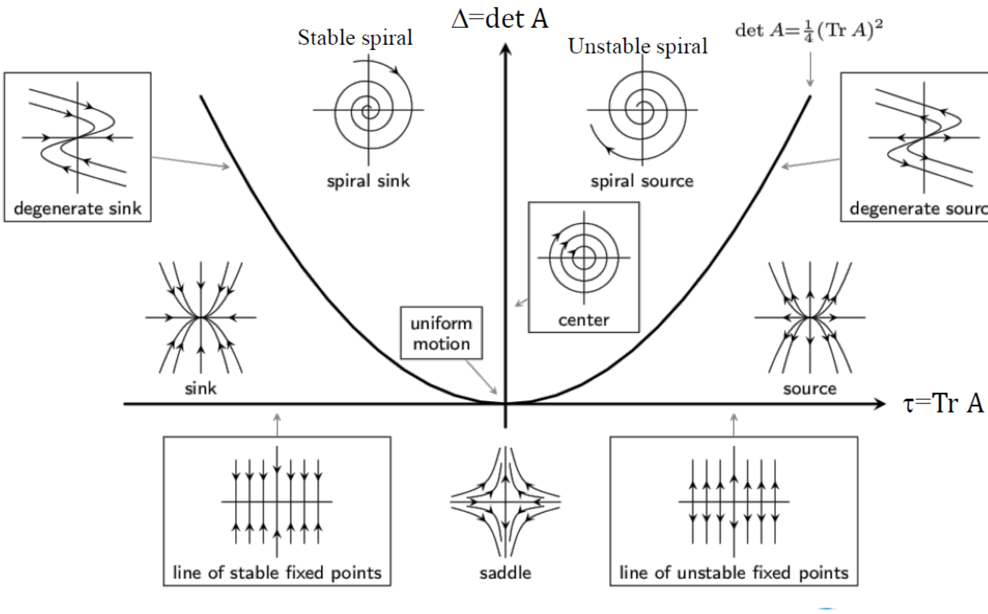

Figure 17 Classification of linear systems. Extracted from [1].

Fixed points can be classified as follows:

$$\text{For } (0, A) \text{ we have a line of stable fixed points} \tag{1.30}$$

$$\text{For } \left(\frac{1}{2}, B\right) \text{ we have a line of unstable fixed points} \tag{1.31}$$

$$\text{For } \left(-\frac{1}{2}, C\right) \text{ we have a line of unstable fixed points} \tag{1.32}$$

If the system equations are plotted for varying initial conditions, the following image is obtained:

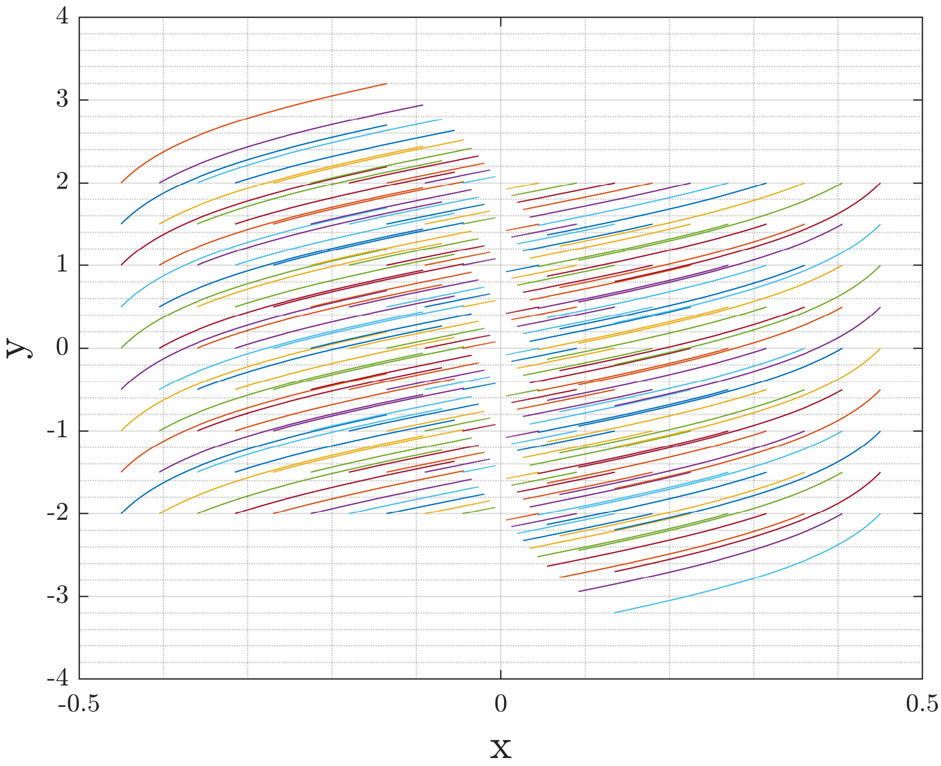

Figure 18 Phase portrait with several initial conditions.

As it can be easily seen, the fixed points are not crossed by any lines. They are lines with constant x and infinite values of y which demarcate the figure.

1.4 Exercise 4

4. Consider the system

$$\begin{aligned}\dot{x} &= x + \exp(-y) \\ \dot{y} &= -y\end{aligned} \tag{1.33}$$

Plot, in phase space, the direction field (using the Matlab function quiver) for this system. On top of it, plot a few trajectories.

The direction field of the previous system is plotted as follows:

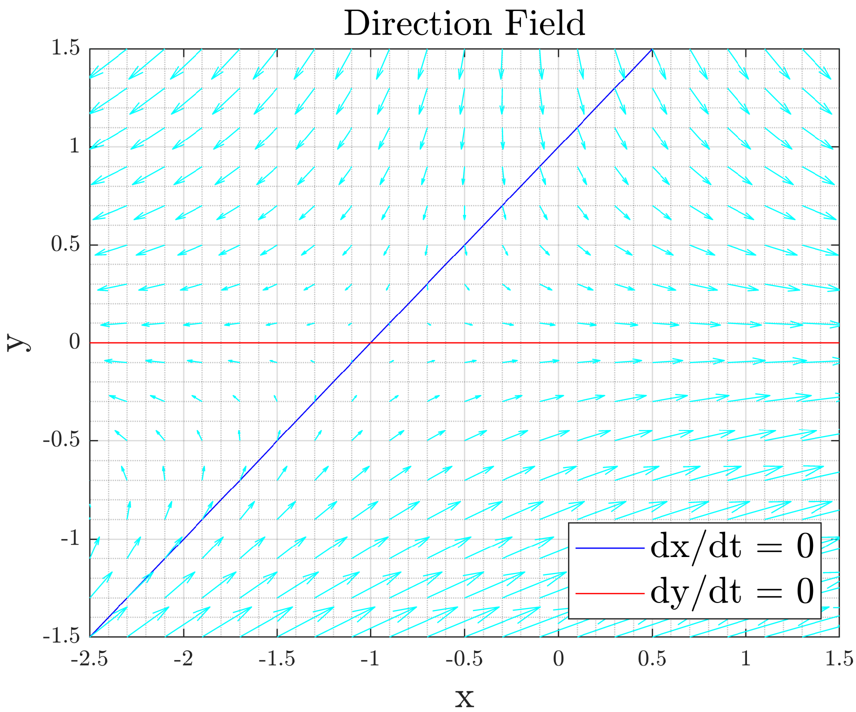

Figure 19 Direction field

The direction lines go along the edge of the null derivatives from both x and y with varying orders of magnitude.

Some trajectories can be obtained from the previous data by plotting x,y dependence with time. It can be seen that non-null values of x always tend to increase while y values tend to decrease and, if both are zero there is no change. Zooming the pictures there is a tendency for the first time unit but from that on x goes to infinity and y to zero.

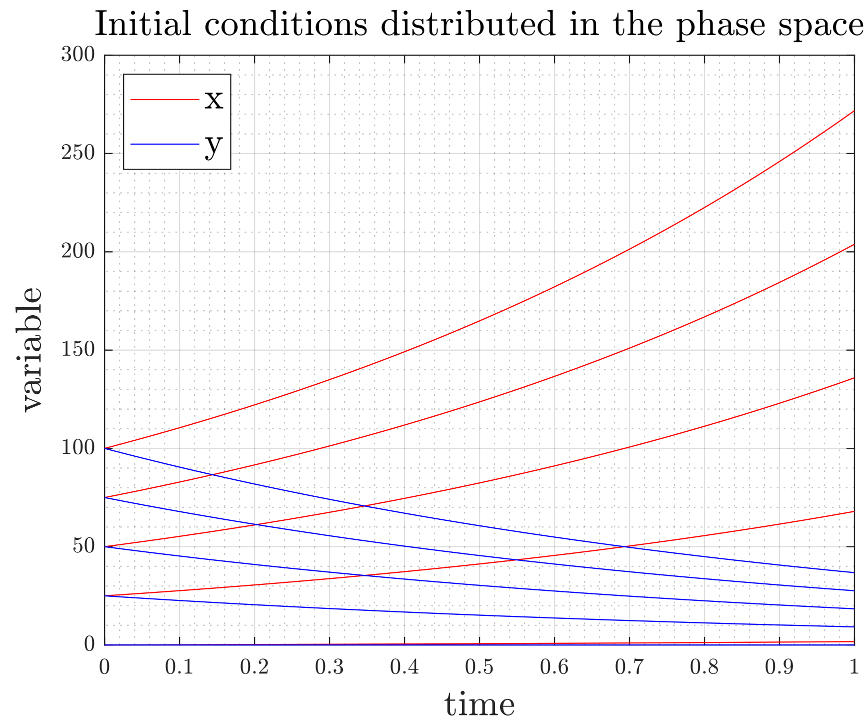

Figure 20 Beginning zoom of variations

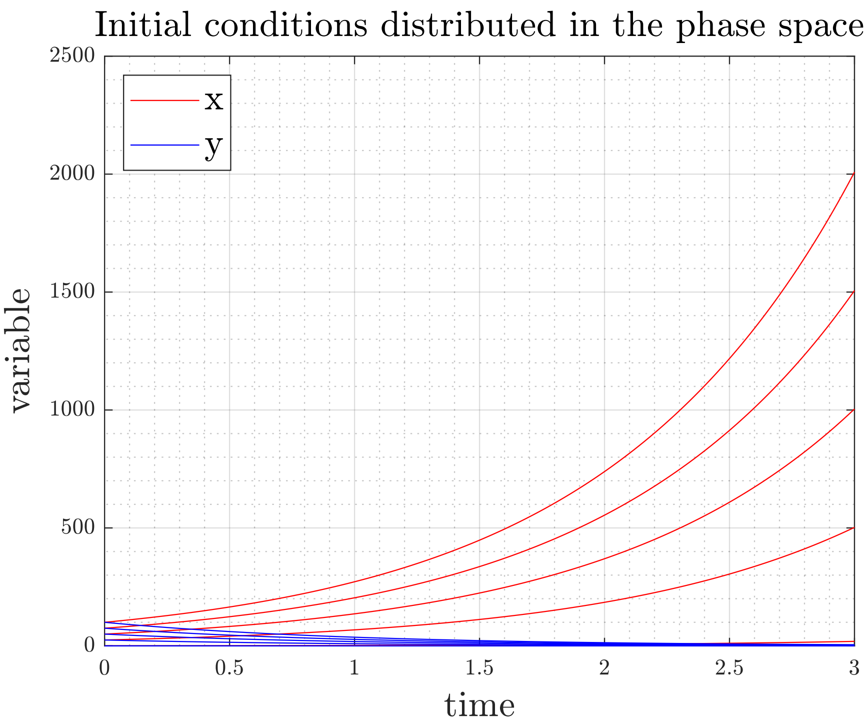

Figure 21 Global vision of variations

2 Exercises for Chaotic systems

2.1 Exercise 1

Let us consider the Lorenz system:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned} \tag{2.1}$$

with $\sigma = 10$ and $\beta = \frac{8}{3}$. Show the evolution in time of variables $x(t), y(t)$ and $z(t)$ for different initial conditions grouped in pairs which are very close (to check if the evolution is similar as time proceeds) for the following ρ values: $21, 24.15, 30$. Show also, x(t) versus z(t) for the same initial conditions. Explain to which dynamical regime corresponds each of the ρ considered.

For $\rho = 21$ the following four graphs have been obtained:

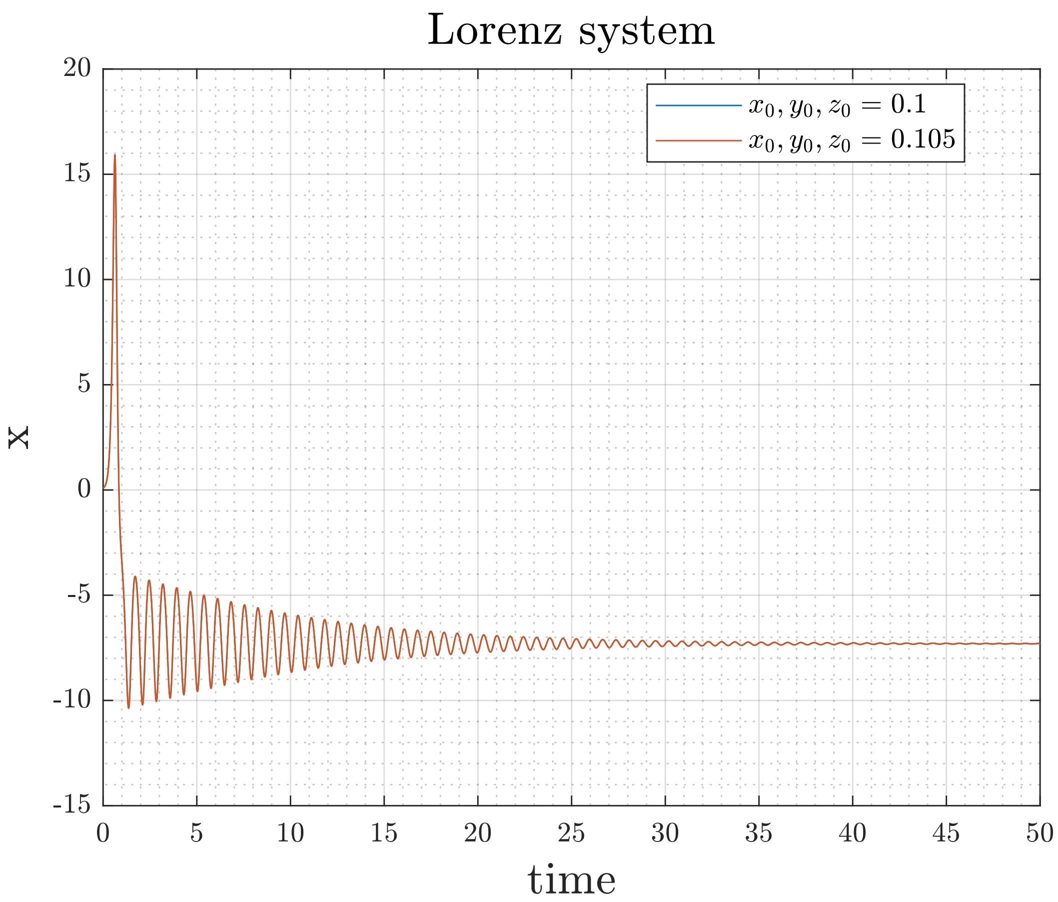

Figure 22 Lorentz system x-t diagram

This graph starts with a pronounced peak and then starts stabilising with several iterations for the first 40 seconds reaching a constant approximate value of $x = 8$.

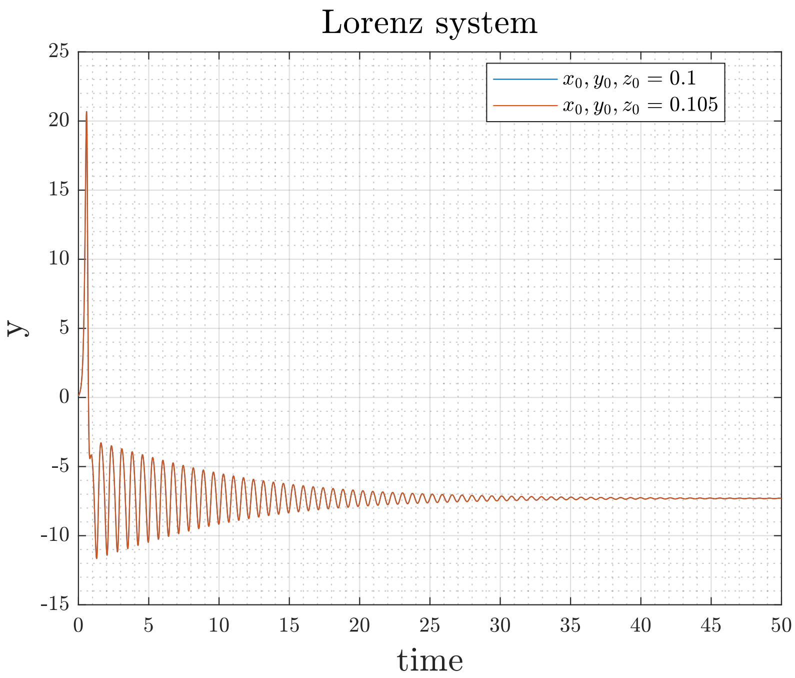

Figure 23 Lorentz system y-t diagram

For y the results are similar to x but the peak is 25% greater at the beginning with almost equal stabilisation amplitude. It also reaches a constant approximate value of $x = 8$.

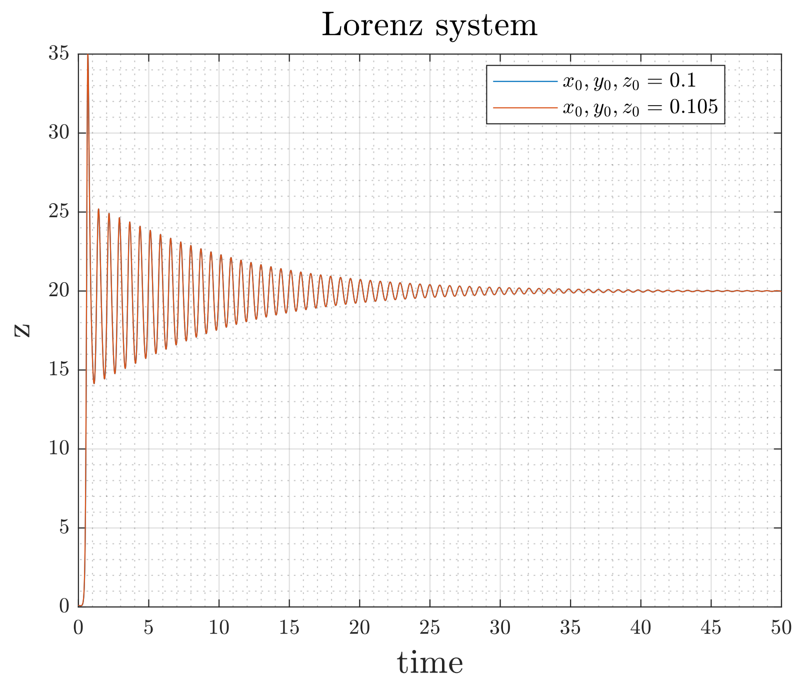

Figure 24 Lorentz system z-t diagram

For z the results have a similar shape but a different analysis. While previous values where negative, for this case the peak is more than 100% bigger than the first and 63% than the second leading to a positive constant value after the stabilising iterations.

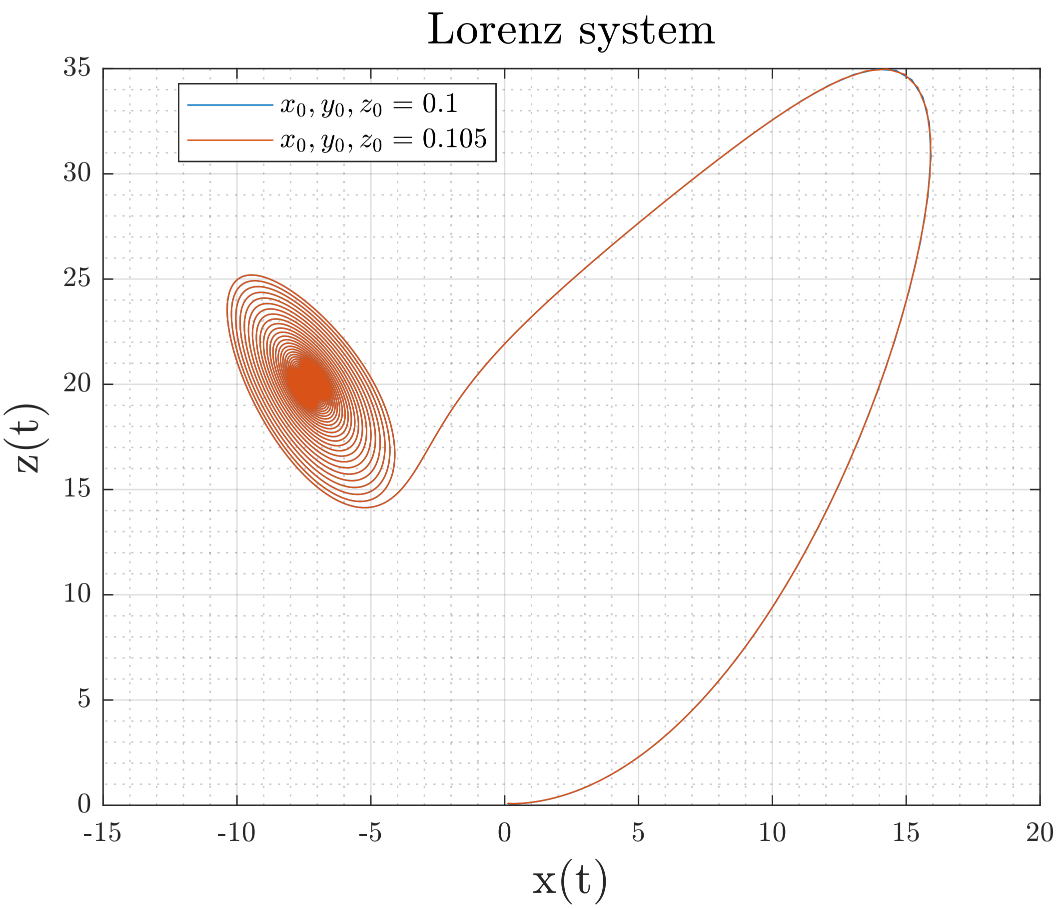

Figure 25 Lorentz system x(t)-z(t) diagram

If the relation between x and z is plotted the system only evolves to one of two fixed point attractors.

For small values of ρ, the system is stable and evolves to one of two fixed point attractors. However, we will see that when ρ is larger than 24.74, the fixed points become repulsors and the trajectory is repelled by them in a very complex way.

For $\rho = 24.15$ the following four graphs have been obtained:

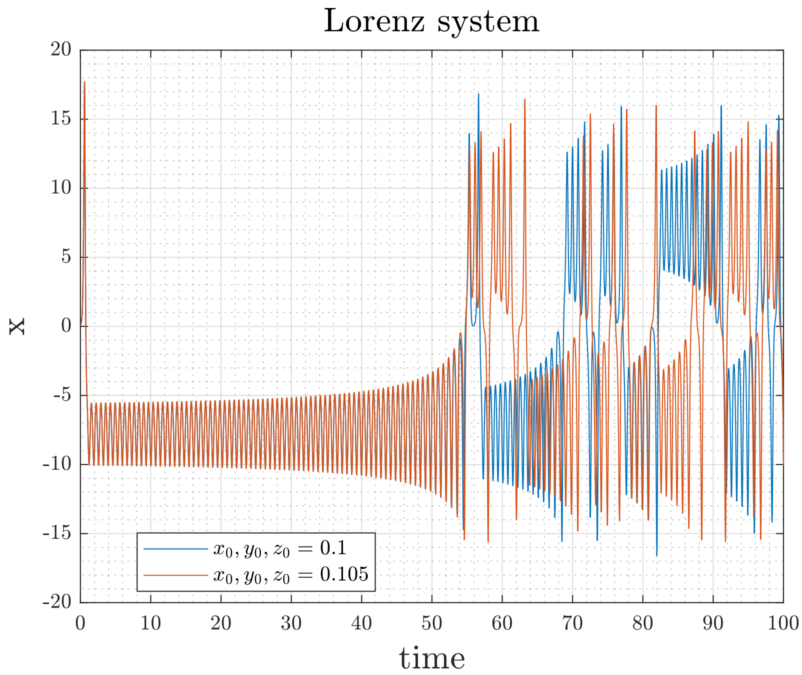

Figure 26 Lorentz system x-t diagram

After an initial peak, x starts with an amplitude of almost 2.5 which keeps increasing for 55 time units up to $x = 7.5$. At this point, the chaotic response starts and for the two cases, even though the values are really close, it is completely different and unpredictable on the long run.

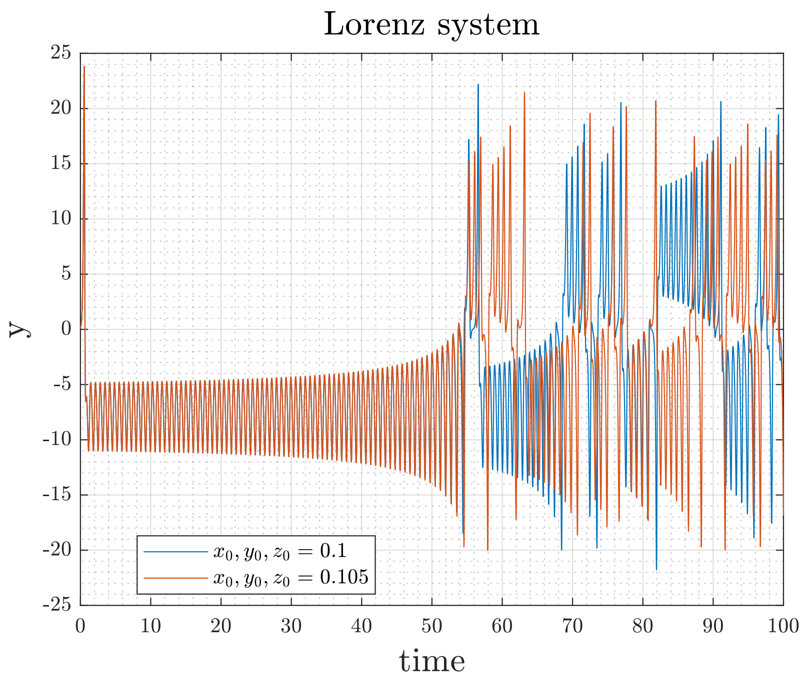

Figure 27 Lorentz system y-t diagram

As seen in the previous case of ρ, y behaves as x.

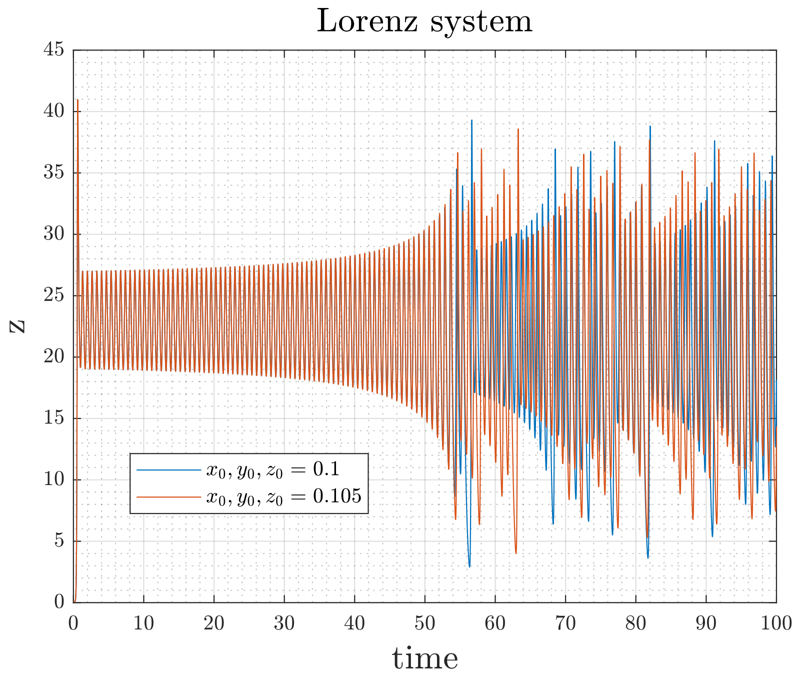

Figure 28 Lorentz system z-t diagram

For z the chaotic response starts at the same time and keeps an increase/decrease loop thoughout time.

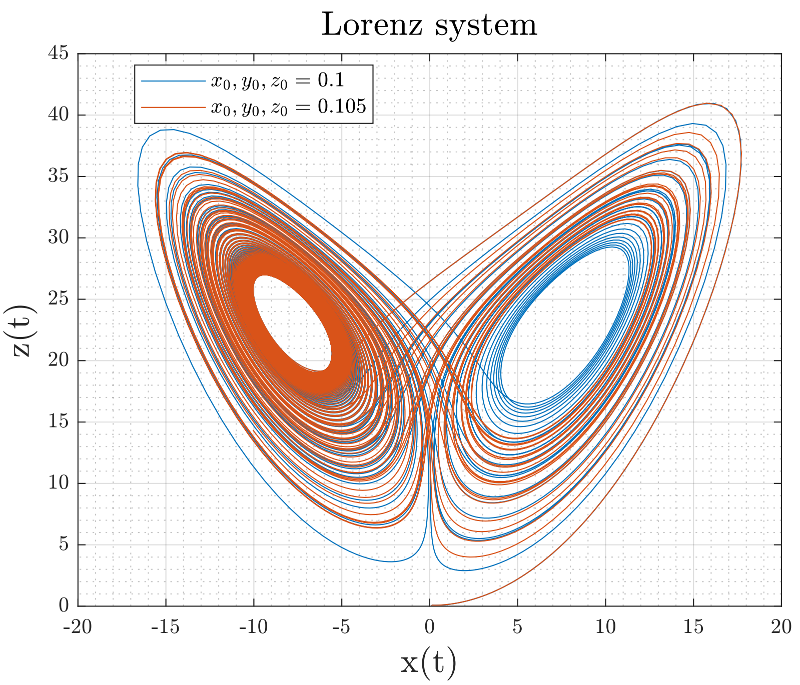

Figure 29 Lorentz system x(t)-z(t) diagram

In this case both fixed point attractors are easily seen.

For $\rho = 30$ the following four graphs have been obtained:

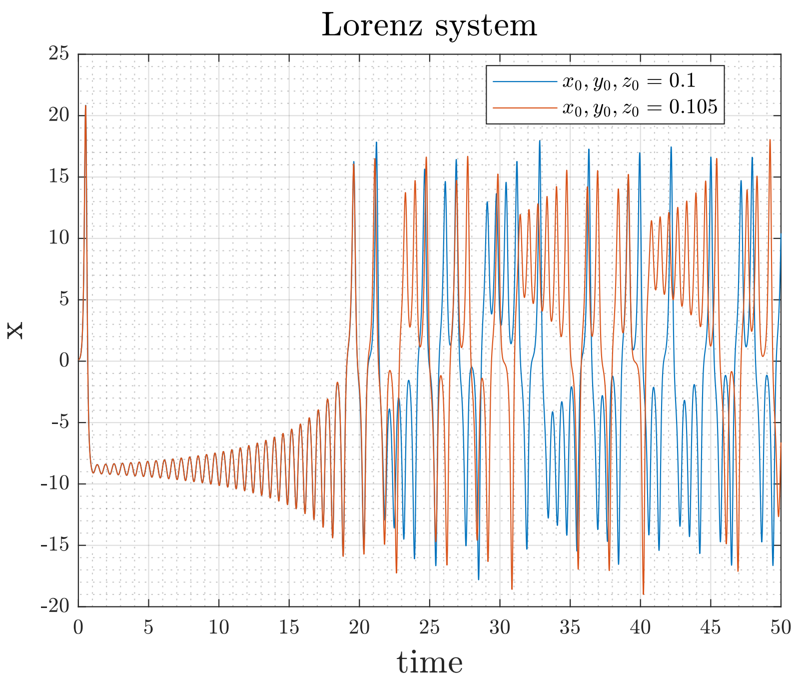

Figure 30 Lorentz system x-t diagram

After an initial transient, the solution settles into an irregular oscillation that never ends so that the motion is aperiodic. For the those different but similar initial cases the results up to the chaotic part are equal but then a slight variation results in completely different results making it impossible to predict in the long run.

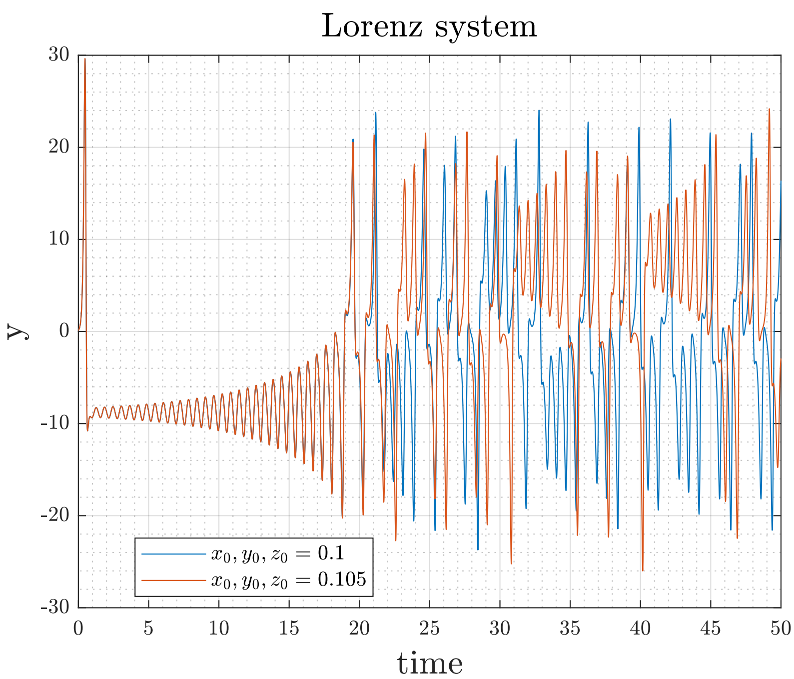

Figure 31 Lorentz system y-t diagram

As seen in the previous cases of ρ, y behaves as x.

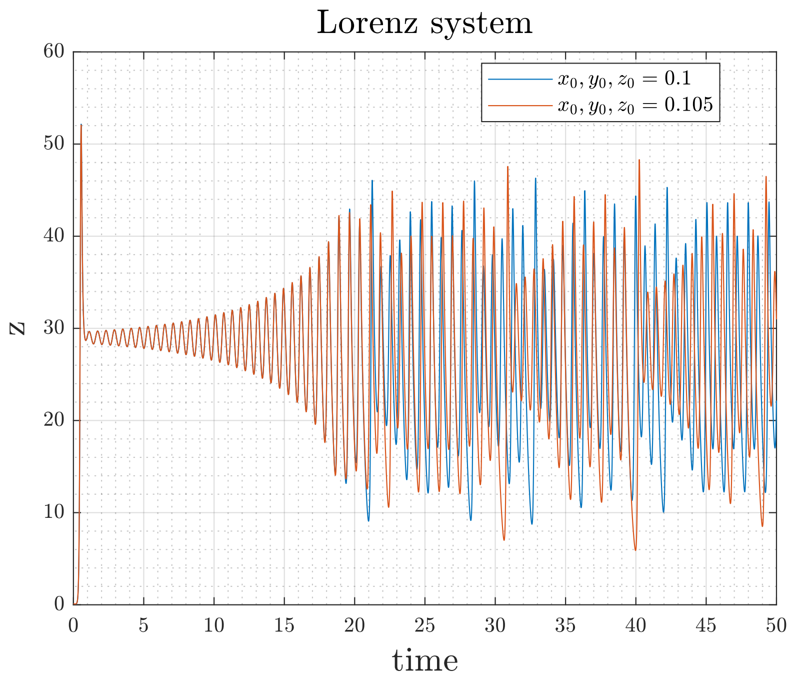

Figure 32 Lorenz system z-t diagram

For z both options end up with a similar chaotic response after the equal start during the first 20 time units.

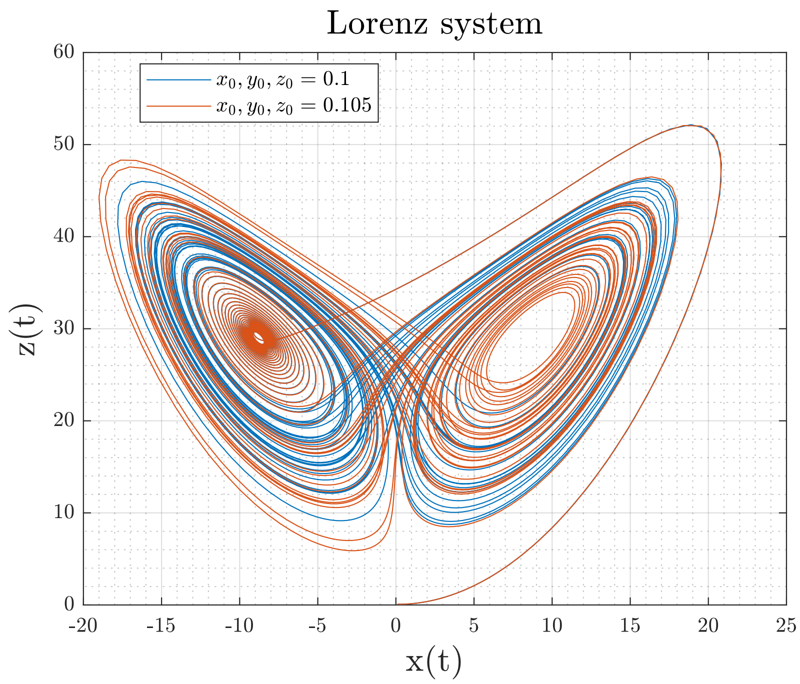

Figure 33 Lorenz system x(t)-z(t) diagram

Finally, as the plot displays a repulsor, the trajectory is repelled in a very complex way from the two points drawing whats is sometimes called a butterfly graph.

2.2 Exercise 2

Using the results of the previous exercise, compute the Maximum Liapunov Exponent obtained from the evolution of the pairs of initial conditions. Explain the relation of the exponent obtained with the predictability of the trajectories obtained from your numerical simulations. Hint: To determine the exponent, use the fitting procedure explained in class selecting accurately the time range used to make the fit. Remember that the exponential amplification is only valid when trajectories are very close.

The Lyapunov exponents are a way of analysing how sensitive a system is to initial conditions. It is a quantity that characterises the rate of separation of infinitesimally close trajectories.

Supposing that $\bar{x}(t)$ is a point in the attractor at time t and that $\bar{x}(t)+\ \delta(t)$ is a close point in phase space where $\|\delta_0\| \ll 1$, then

$$\|\delta(t)\| \sim \|\delta_0\| \cdot \mathrm{e}^{\lambda t}$$

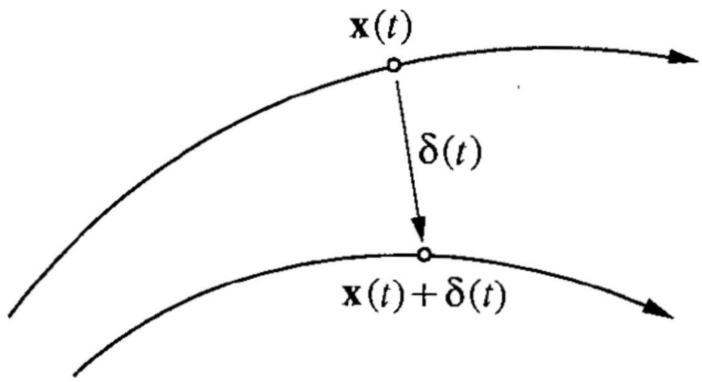

Figure 34 δ calculation. Extracted from [2]

in which λ represents the Lyapunov exponent.

Talking about the predictability, if the system has a positive Lyapunov exponent the trajectories diverge exponentially fast.

In order to compute the maximum Liapunov Exponent, it is plotted the neperian logarithm of $|\delta|$ with time to obtain the slope of the rectilinear zone. Then, there is a time horizon beyond which the prediction is impossible. It has to be taken into consideration that there will always be some error $|\delta_0|$.

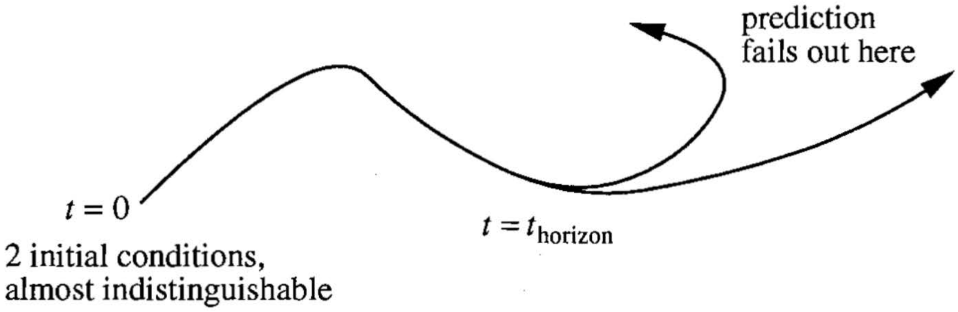

Figure 35 Predictability. Extracted from [2]

After a time t, the discrepancy grows to $|\delta(t)| \sim |\delta(t)|e^{\lambda t}$ with a tolerance error "a". If

$$\|\delta_k(t)\| > \text{a the system is unpredictable and } t_a \sim| \, o\left(\frac{1}{\lambda}\ln\frac{a}{\|\delta_0\|}\right)$$

Very close values have been used to obtain this results from the same initial conditions of exercise 1. For $\rho = 21$ the following four graphs have been obtained:

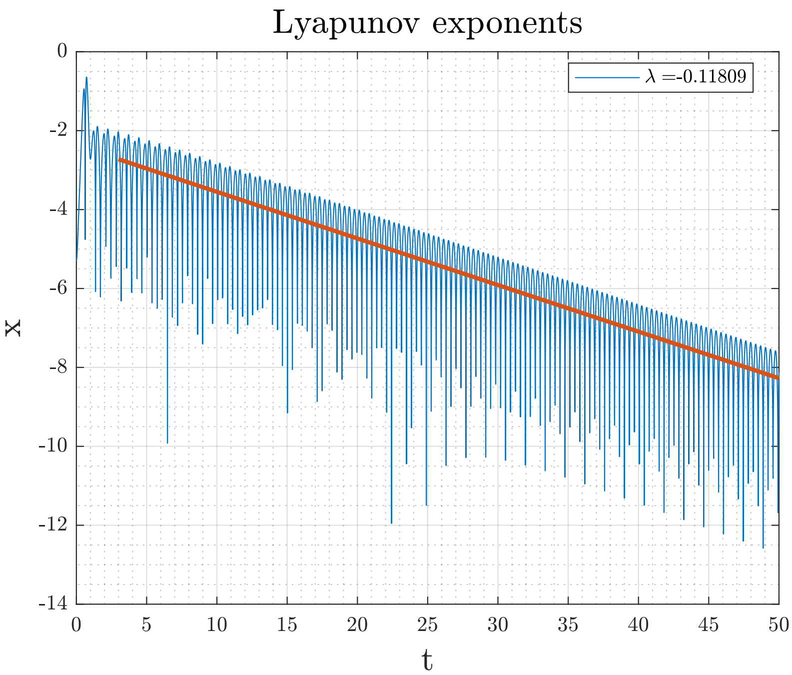

Figure 36 Maximum Lyapunov exponent for x

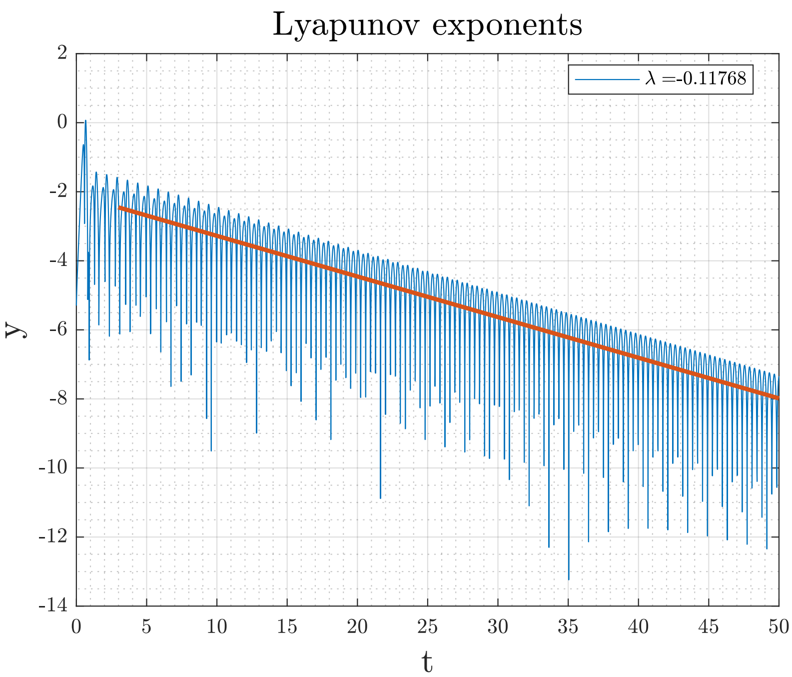

Figure 37 Maximum Lyapunov exponent for y

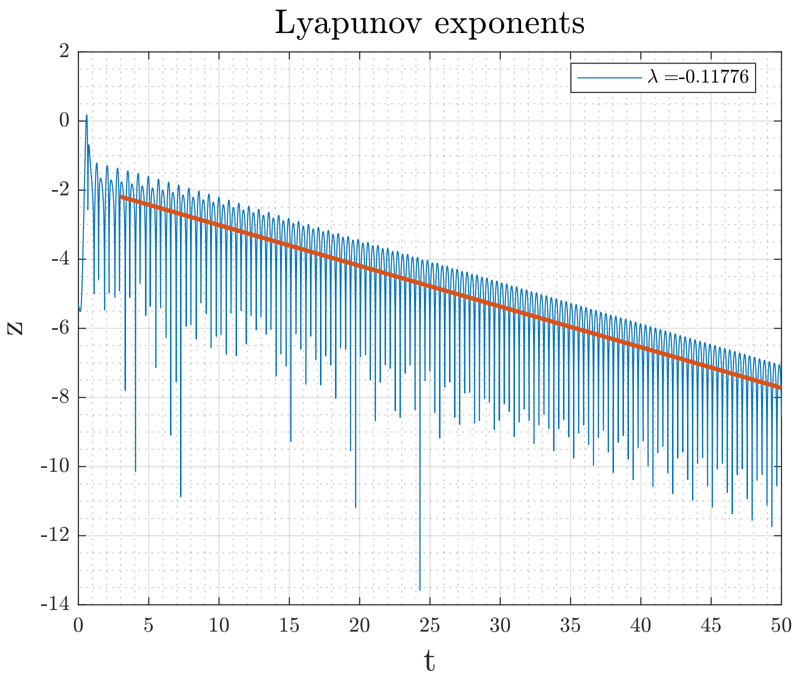

Figure 38 Maximum Lyapunov exponent for z

All lyapunov exponents are negative and within the error margin.

For $\rho = 24.15$ the following four graphs have been obtained:

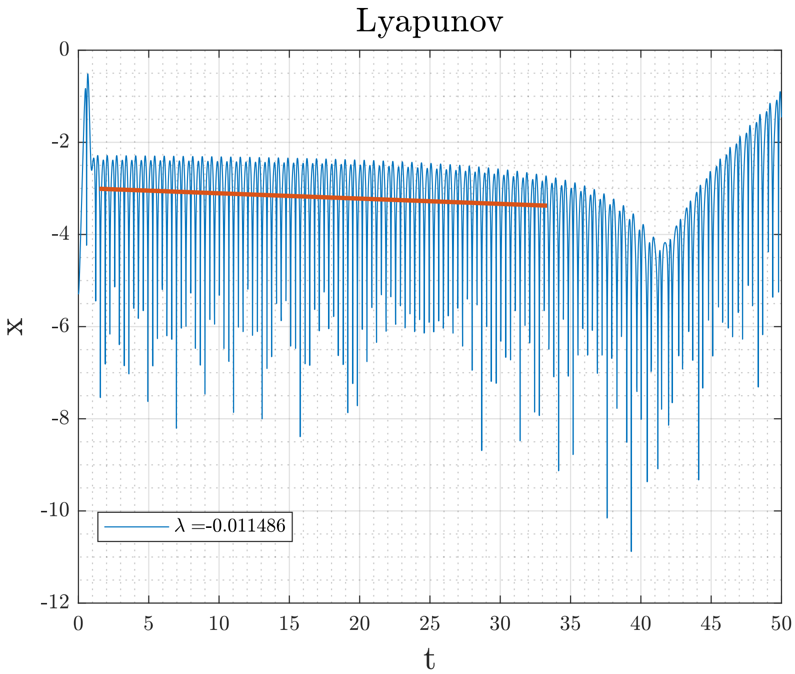

Figure 39 Maximum Lyapunov exponent for x

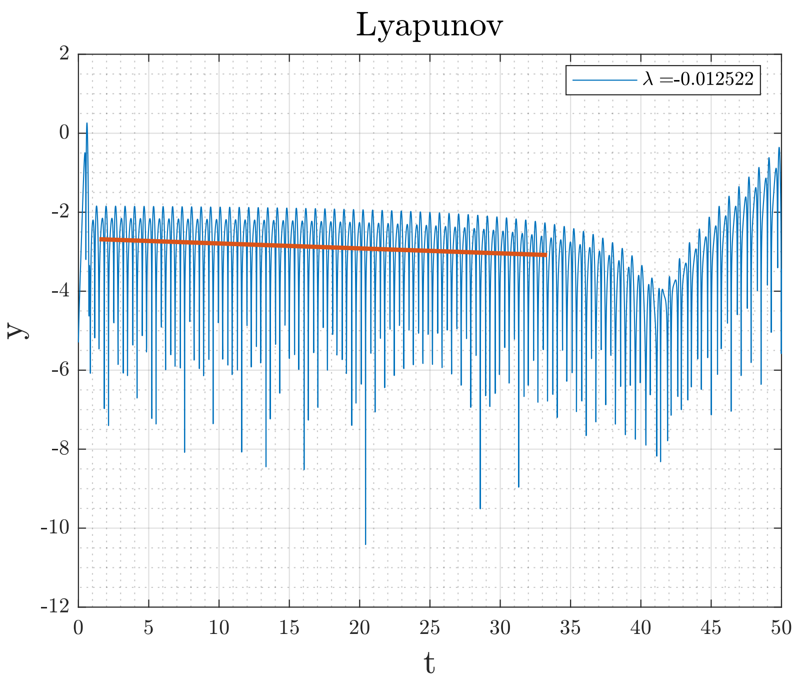

Figure 40 Maximum Lyapunov exponent for y

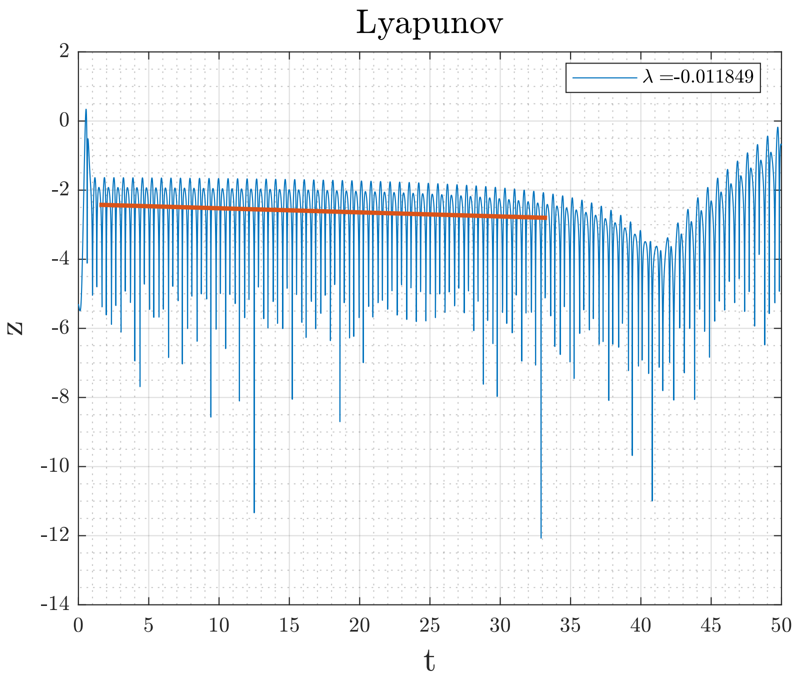

Figure 41 Maximum Lyapunov exponent for z

All lyapunov exponents are almost null and within the error margin.

For $\rho = 30$ the following four graphs have been obtained:

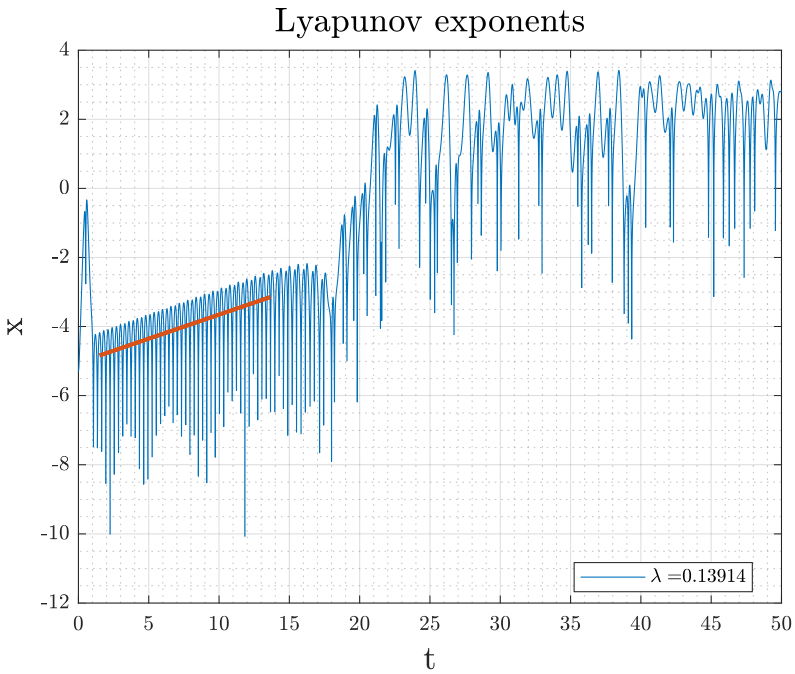

Figure 42 Maximum Lyapunov exponent for x

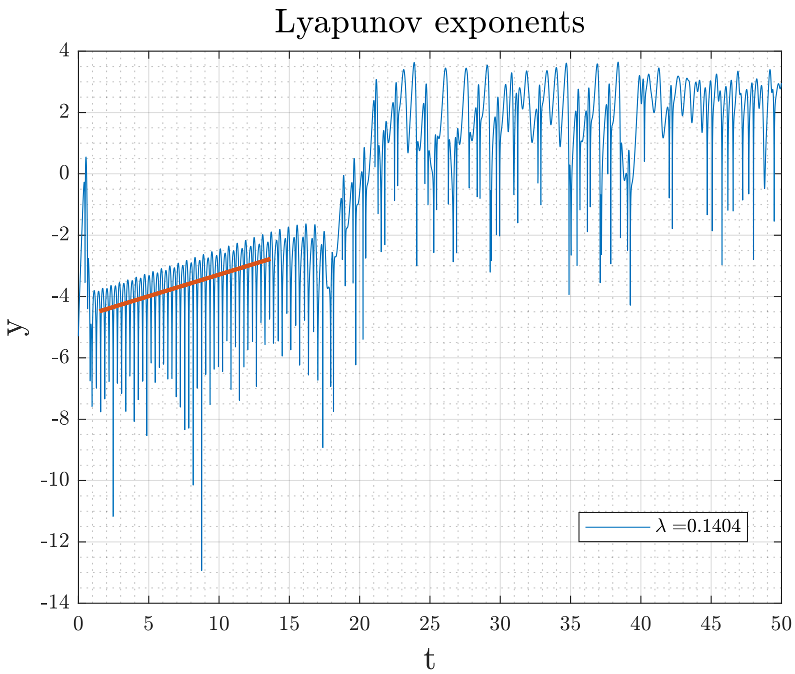

Figure 43 Maximum Lyapunov exponent for y

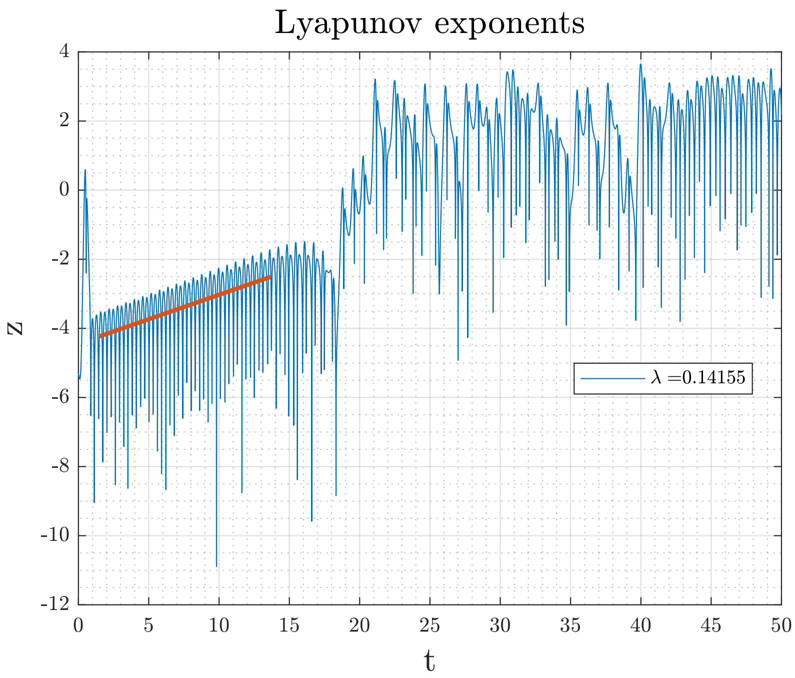

Figure 44 Maximum Lyapunov exponent for z

All Lyapunov exponents are positive and within the error margin. For this case trajectories diverge considerably fast.

2.3 Exercise 3

Consider the Henon Map:

$$\begin{aligned} x_{n+1} &= y_n + 1 - a x_n^2 \\ y_{n+1} &= b x_n \end{aligned} \tag{2.2}$$

Considering $a = 1.4$ and $b = 0.3$, iterate the map for 10.000 successive iterates. Show, all the points in a x_n versus y_n graph. Explain the structure of the graph (discussed in the class) and the mechanism needed to produce it.

Once the Matlab code has iterated the previous system for 10.000 iterations, the following graph is obtained:

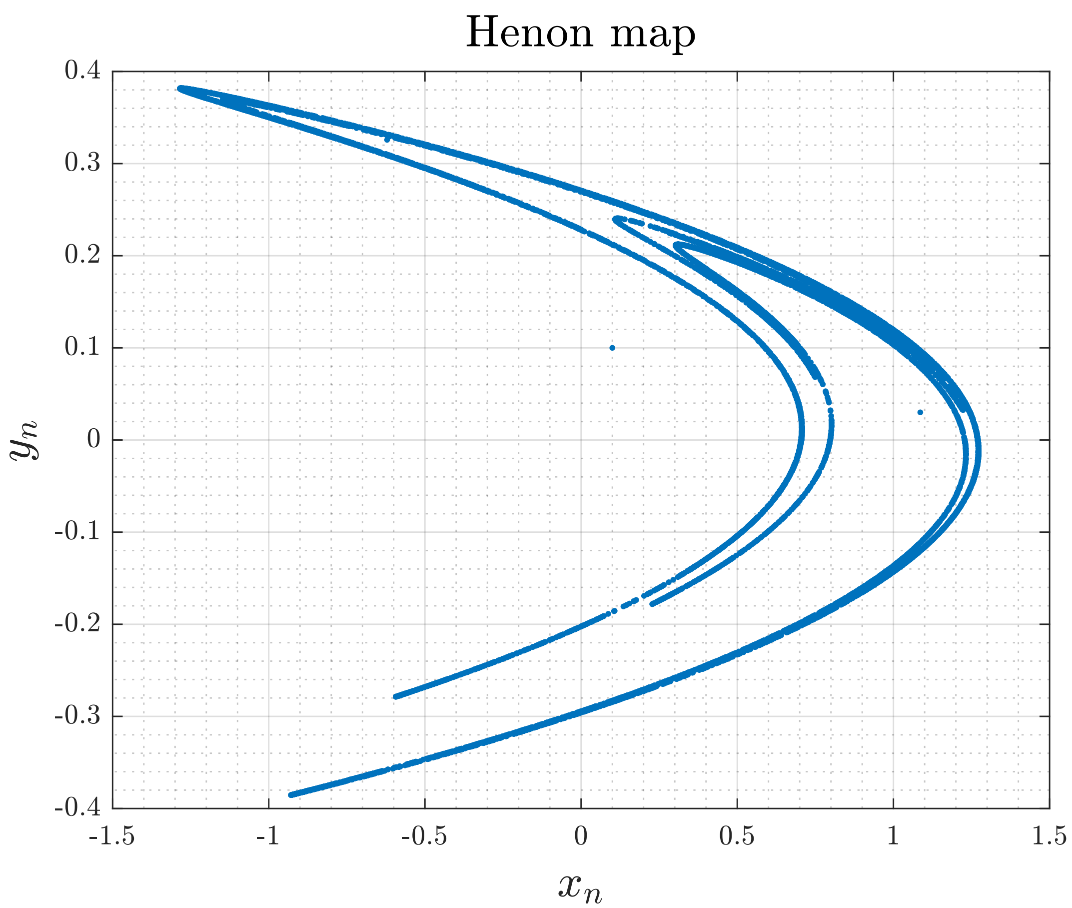

Figure 45 Henon Map with 10.000 iterations

2.3.1 Elementary properties of the Henon Map

The Henon Map captures the essential properties of the Lorentz system [3]. The most important ones are:

1. The Henon map is invertible. As the Lorentz system has a unique trajectory through each point in phase space, it means that is has a unique past.
2. The Henon map is dissipative. Thoughout all the phase space, it contracts areas at the same rate. This property is the analog of constant negative divergence in the Lorenz system.
3. The Henon map has a trapping regions for certain parameter values. There is a region R that gets mapped inside itself. As in the Lorenz system, the strange attractor is enclosed in the trapping region.
4. Some trajectories of the Henon map escape to infinity while all trajectories of the Lorenz system are bounded, meaning that eventually they enter and stay inside a certain large ellipsoid.

2.3.2 Zoom

The attractor is bent like a boomerang and if the lines are zoomed, it is made of many parallel curves:

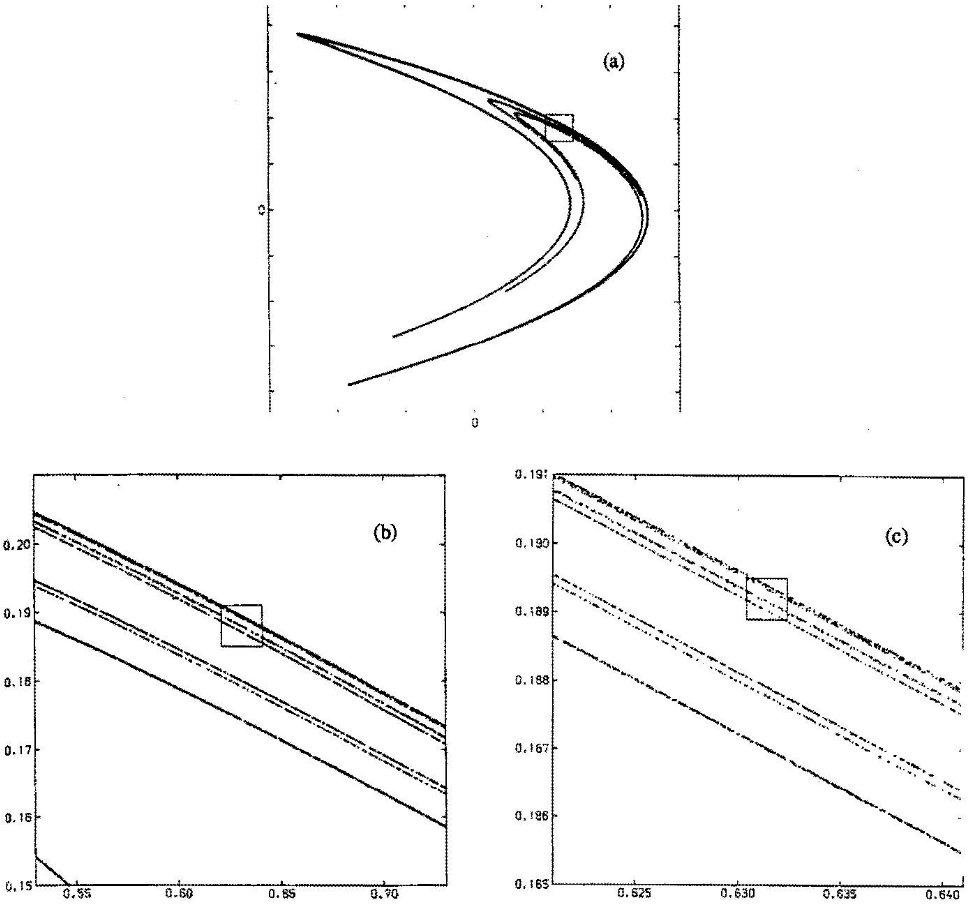

Figure 46 Henon map zoom. Extracted from [3].

2.3.3 Creation of the map

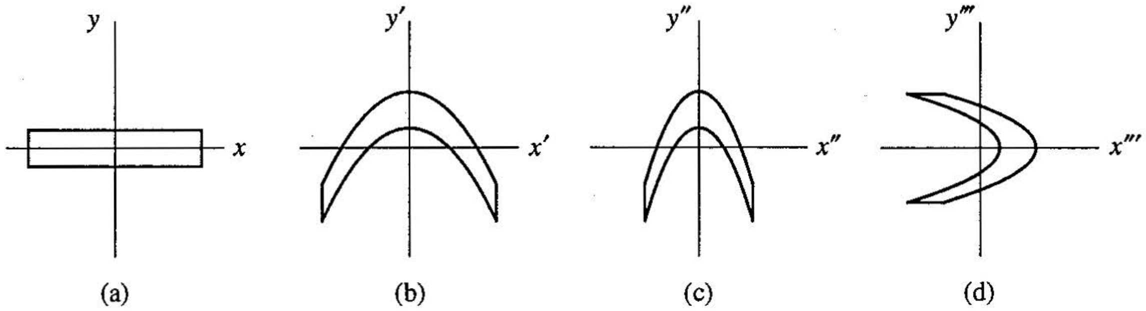

Figure 47 Henon map formation. Extracted from [3].

a) Starting with a rectangular region elongated along the x-axis, the rectangle is stretched and folded by applying the following transformation:

$$T' : x' = x, \quad y' = 1 + y - ax^2 \tag{2.3}$$

(In here,the prime values denote iteration, not differentiation.)

b) The bottom and top of the rectangle get mapped to parabolas where the parameter a controls the folding. Now folding the region even more by contracting Figure b) along the x -axis:

$$T'' : x'' = bx', \quad y'' = y' \tag{2.4}$$

where $-1 < b < 1$,producing c). Then, coming back to the orientation along the x -axis by reflecting across the line $y = x$ as in d):

$$T''' : x''' = y'', \quad y''' = x'' \tag{2.5}$$

Finally, the composite transformation $T = T'''T''T'$ yields the Hénon mapping , where the notation (x_n, y_n) is used for (x, y) and (x_{n+1}, y_{n+1}) for (x''', y''').

2.4 Exercise 4

The Rössler system is defined by the following equations:

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c). \end{aligned} \tag{2.6}$$

Consider this system for $b = 2, c = 4$ and a increasing slowly from 0 to 0.4 in small steps. For each a, plot the evolution in time for one of the variables and its Power Spectrum. Comment the results.

The first plot relates x results with time and the a variable:

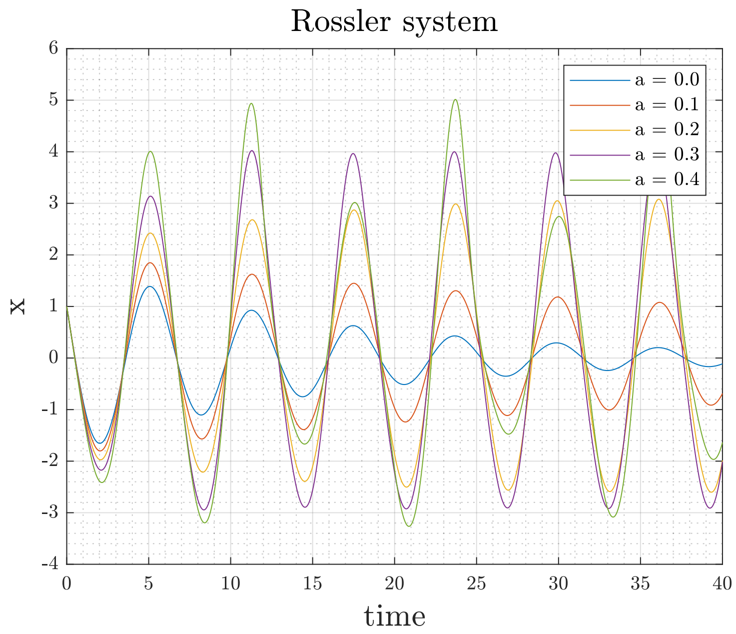

Figure 48 Rössler system x-t

Starting from the same initial conditions, the a parameters changes the results completely. With $a = 0$ the Rössler system tends to 0 in less than 40 time units as a damped system. For values greater than 0 but little enough this tendency is maintained which a much longer period span. From $a = 0.2$ onwards, instead of decrease slowly values increase to a certain amplitude and are maintained in time. Then, for values of $a = 0.4$ or higher there is a double amplitude iteration in which the amplitude is repeated every two iterations with two constant values being reached.

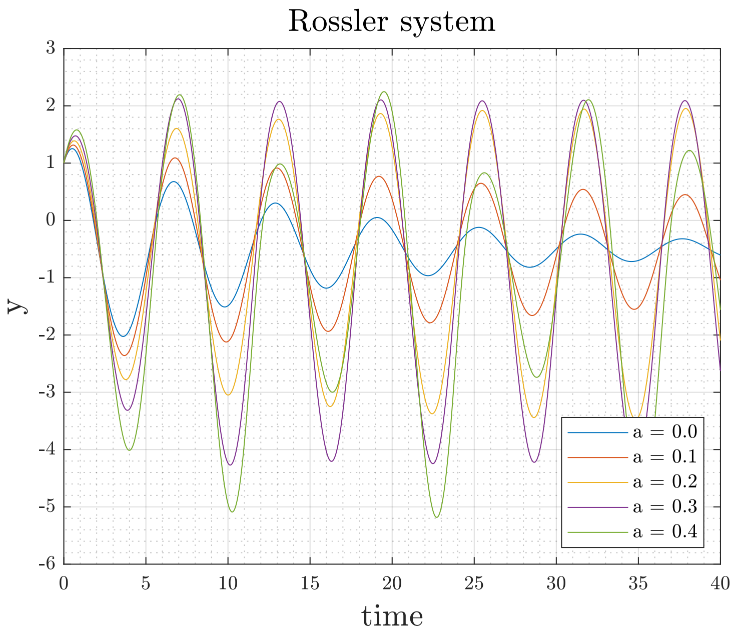

Figure 49 Rössler system y-t

For y the analysis yields the same results encountered for x.

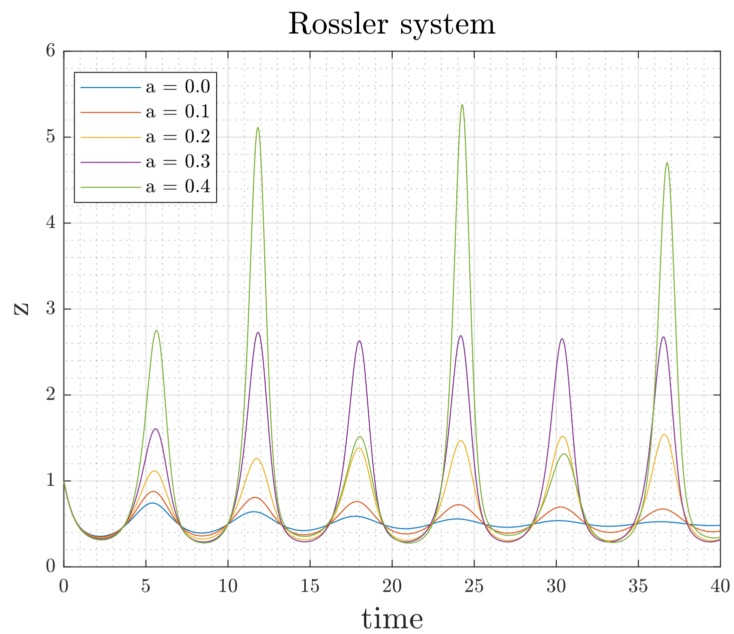

Figure 50 Rössler system z-t

However, z presents a different attitude to its predecessors. While values up to $a = 0.3$ keep the previously defined method, values higher than that start oscillating with different amplitudes that do not repeat in the time span analysed.

If the Rössler system is plotted for x,y and z with all a variables the following result is obtained:

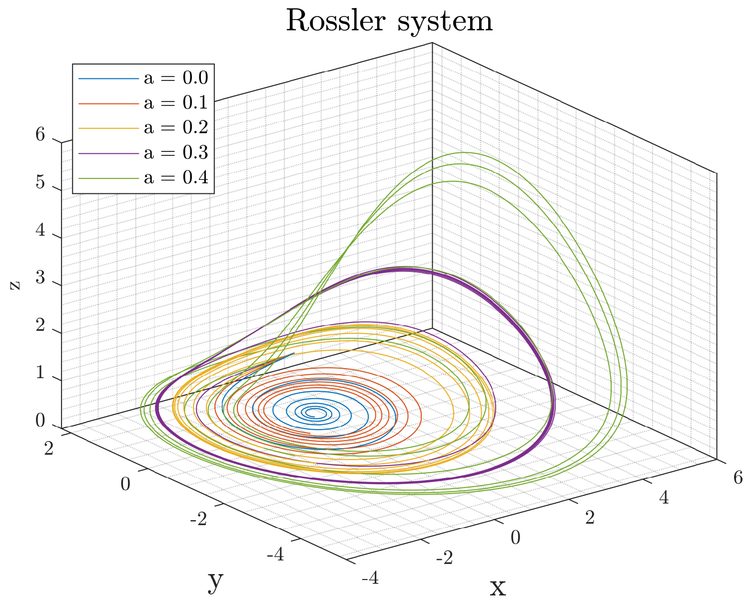

Figure 51 Rössler system x,y,z

Values of a up to 0.2 are almost completely in a single plane while the more a increases above this threshold the more it changes in the third dimension.

The power spectrum $S_{xx}(f)$ of a time series $x(t)$ describes the distribution of power into frequency components composing that signal. According to Fourier analysis, any physical signal can be decomposed into a number of discrete frequencies, or a spectrum of frequencies over a continuous range.

The power spectrum is obtained for each axis:

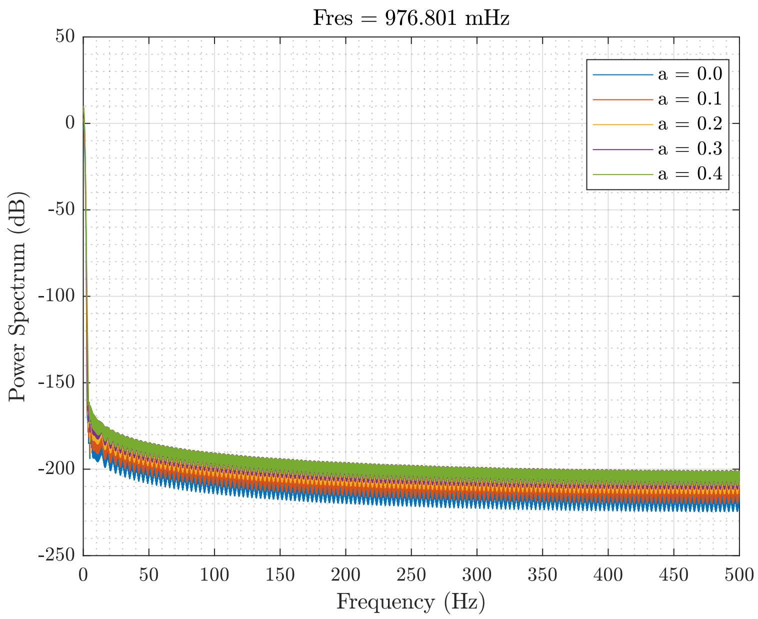

Figure 52 Power Spectrum of x

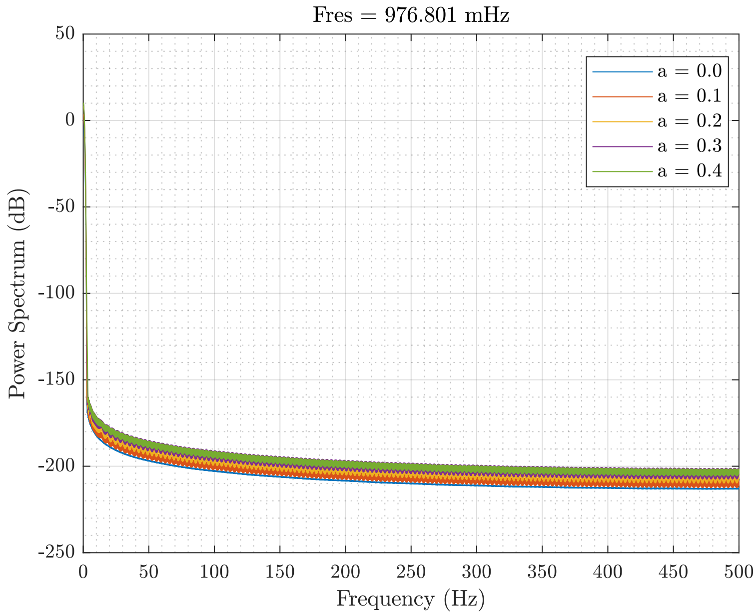

Figure 53 Power Spectrum of y

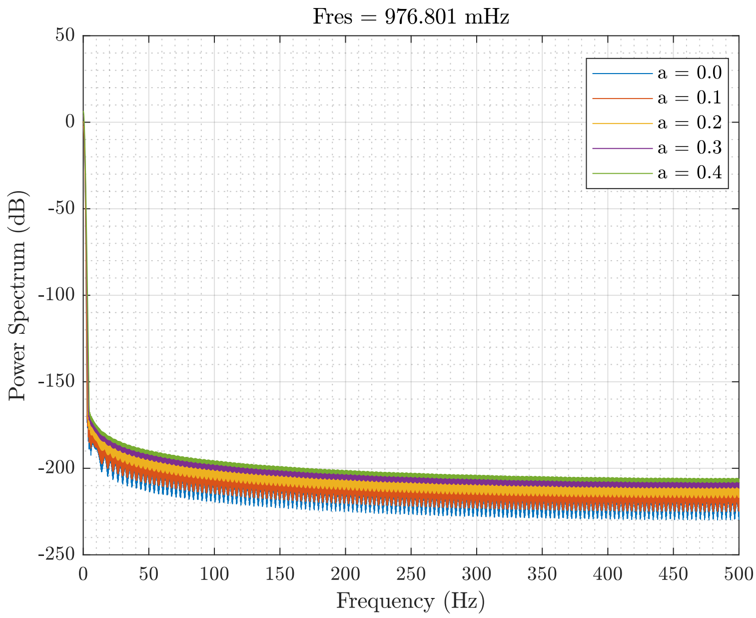

Figure 54 Power Spectrum of z

The global view shows that the greater the a parameter is, the higher the power spectrum value at which it stabilises.

3 Codes I

3.1 Exercise 2

```
%-------------------------------------------------------------------------%
% Exercise 2
%-------------------------------------------------------------------------%

% Date: 14/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Declaration of solution vectors
Delta_t = 0.001;
t_max = 10;
t_sol = 0:Delta_t:t_max;

% Initial conditions
% a)
%     a = 1; b = 2; c = 3; d = 1;
%     x_0_1 = -1.25; y_0_1 = 1;
%     x_0 = -1.25; y_0 = 1;

% b)
    a = 1; b = -2; c = 3; d = 1;
%     x_0_2 = 0.2; y_0_2 = 1.5;
    x_0 = 0.2; y_0 = 1.5;

% c)

% alpha = 1.9;
% beta  = 14;
% x_0 = alpha*x_0_1 + beta*x_0_2;
% y_0 = alpha*y_0_1 + beta*y_0_2;

% RK4
t_sol = 0:Delta_t:t_max;
x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
x_sol(1) = x_0;
y_sol(1) = y_0;

% Definition of functions
F1 = @ (t,x,y) a*x + b*y;
F2 = @ (t,x,y) c*x + d*y;

for i =1:length(t_sol)-1
    % Computation of coefficients sub 1
```

```
    i1 = F1(t_sol(i),x_sol(i),y_sol(i));
    j1 = F2(t_sol(i),x_sol(i),y_sol(i));
    % Computation of coefficients sub 2
    i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
    j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
    % Computation of coefficients sub 3
    i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
    j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
    % Computation of coefficients sub 4
    i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
    j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
    % Compute next step
    x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
    y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);

end

% t-x, t-y
h = figure(1);
plot(t_sol,x_sol,t_sol,y_sol);
grid on;
grid minor;
box on;
tlt = title("Two-dimensional linear systems");
ylab = ylabel("variable");
xlab = xlabel("time");
leg  = legend("x","y", 'location', 'southwest');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',16);

% x-y

h2 = figure(2);
plot(x_sol,y_sol);
grid on;
grid minor;
box on;
tlt = title("Two-dimensional linear systems");
ylab = ylabel("y");
xlab = xlabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);


%% Save data

% Save as pdf
% print2pdf(h,"Fig_2_");
% print2pdf(h2,"Fig_2_");
```

3.2 Exercise 2 comparison

```
%-------------------------------------------------------------------------%
% Exercise 2: Comparison
%-------------------------------------------------------------------------%

% Date: 14/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Comparison

alpha = 1.9;
beta  = 14;

load('TDS_Ex2_S1_IC1.mat');
x_sol_S1_IC1 = alpha*x_sol;
y_sol_S1_IC1 = alpha*y_sol;

load('TDS_Ex2_S1_IC2.mat');
x_sol_S1_IC2 = beta*x_sol;
y_sol_S1_IC2 = beta*y_sol;

load('TDS_Ex2_S2_IC1.mat');
x_sol_S2_IC1 = alpha*x_sol;
y_sol_S2_IC1 = alpha*y_sol;

load('TDS_Ex2_S2_IC2.mat');
x_sol_S2_IC2 = beta*x_sol;
y_sol_S2_IC2 = beta*y_sol;

x_S1_compare = x_sol_S1_IC1 + x_sol_S1_IC2;
y_S1_compare = y_sol_S1_IC1 + y_sol_S1_IC2;
x_S2_compare = x_sol_S2_IC1 + x_sol_S2_IC2;
y_S2_compare = y_sol_S2_IC1 + y_sol_S2_IC2;

load('TDS_Ex2_S1_IC3.mat');
x_sol_S1_IC3 = x_sol;
y_sol_S1_IC3 = y_sol;

load('TDS_Ex2_S2_IC3.mat');
x_sol_S2_IC3 = x_sol;
y_sol_S2_IC3 = y_sol;

% Time domain

h = figure(1);
plot(t_sol,x_sol_S1_IC3,t_sol,y_sol_S1_IC3,t_sol,x_S1_compare,t_sol,y_S1_compare);
grid on;
grid minor;
box on;
```

```
tlt = title("System 1");
ylab = ylabel("variable");
xlab = xlabel("time");
leg  = legend("$x_3$","$y_3$","$\alpha x_1 + \beta x_2$","$\alpha y_1 + \beta y_2$", 'location
    ', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',16);

h2 = figure(2);
plot(t_sol,x_sol_S2_IC3,t_sol,y_sol_S2_IC3,t_sol,x_S2_compare,t_sol,y_S2_compare);
grid on;
grid minor;
box on;
tlt = title("System 2");
ylab = ylabel("variable");
xlab = xlabel("time");
leg  = legend("$x_3$","$y_3$","$\alpha x_1 + \beta x_2$","$\alpha y_1 + \beta y_2$", 'location
    ', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',16);

% Phase portrait

h3 = figure(3);
plot(x_sol_S1_IC3,y_sol_S1_IC3,x_S1_compare,y_S1_compare);
grid on;
grid minor;
box on;
tlt = title("System 1");
ylab = ylabel("y");
xlab = xlabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);

h4 = figure(4);
plot(x_sol_S2_IC3,y_sol_S2_IC3,x_S2_compare,y_S2_compare);
grid on;
grid minor;
box on;
tlt = title("System 2");
ylab = ylabel("y");
xlab = xlabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);

%% Save data

% Save as pdf
print2pdf(h,"Fig_2_1_txty_compare");
print2pdf(h2,"Fig_2_2_txty_compare");
print2pdf(h3,"Fig_2_1_xy_compare");
print2pdf(h4,"Fig_2_2_xy_compare");
```

3.3 Exercise 3

```
%------------------------------------------------------------------------%
% Exercise 3: Fixed points and classification
%------------------------------------------------------------------------%

% Date: 14/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = -0.45:0.045:0.45;
y_0 = -2:0.5:2;

% Definition of functions
F1 = @ (t,x,y) -x+4*x.^3;
F2 = @ (t,x,y) -2*x;

for i = 1:1:length(x_0)

    % Declaration of solution vectors
    Delta_t = 0.001;
    t_max = 2;
    t_sol = 0:Delta_t:t_max;

    t_sol = 0:Delta_t:t_max;
    x_sol = zeros(1,length(t_sol));
    y_sol = zeros(1,length(t_sol));
    x_sol(1) = x_0(i);

    for j = 1:1:length(y_0)
    y_sol(1) = y_0(j);



    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i));
        % Computation of coefficients sub 2
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
        % Compute next step
```

```
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);

    end

    h = figure(1);
    plot(x_sol,y_sol);
    grid on;
    grid minor
    box on;
    hold on

    end
end

ylab = ylabel("y");
xlab = xlabel("x");
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);

%% Save data

% Save as pdf
% print2pdf(h,"Fig_3");
```

3.4 Exercise 4

```
%-------------------------------------------------------------------------%
% Exercise 4: Plot the direction field
%-------------------------------------------------------------------------%

% Date: 14/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Map

x = -2.5:0.01:1.5;
y = -2.5:0.01:1.5;
dxdt = zeros(1,length(x));
dydt = zeros(1,length(y));
i = 1;
while i <= length(x)
    dxdt(i) = x(i)+exp(0);
    dydt(i) = 0;
    i = i+1;
end

% Position of the quiver vectors
position_x = size(length(x),length(y));
position_y = size(length(x),length(y));

for i = 1:20:length(x)
   for j = 1:20:length(y)
       position_x(i,j) = x(i);
       position_y(i,j) = y(j);
       val_dx_dt(i,j) = x(i)+exp(-y(j));
       val_dy_dt(i,j) = -y(j);
   end
end

% Figure plot
h = figure(1);
plot(x,dxdt,'b',y,dydt,'r');
hold on
qv = quiver(position_x,position_y,val_dx_dt,val_dy_dt,'c');
xlim([-2.5 1.5])
ylim([-1.5 1.5])
grid on;
grid minor;
box on;

tlt = title("Direction Field");
ylab = ylabel("y");
xlab = xlabel("x");
leg  = legend("dx/dt = 0","dy/dt = 0", 'location', 'southeast');
```

```
set(qv,'AutoScale','on', 'AutoScaleFactor', 100)
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',16);

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 0:25:100;
y_0 = 0:25:100;

for i = 1:1:length(x_0)

    % Declaration of solution vectors
    Delta_t = 0.001;
    t_max = 3;
    t_sol = 0:Delta_t:t_max;

    t_sol = 0:Delta_t:t_max;
    x_sol = zeros(1,length(t_sol));
    y_sol = zeros(1,length(t_sol));
    x_sol(1) = x_0(i);
    y_sol(1) = y_0(i);

    % Definition of functions
    F1 = @ (t,x,y) x + exp(-y);
    F2 = @ (t,x,y) -y;

    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i));
        % Computation of coefficients sub 2
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t);
        % Compute next step
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);

    end

    h2 = figure(2);
    plot(t_sol,x_sol,'r',t_sol,y_sol,'b');
    grid on;
    grid minor
    box on;
    hold on
end

tlt = title("Initial conditions distributed in the phase space");
ylab = ylabel("variable");
xlab = xlabel("time");
leg  = legend("x","y", 'location', 'northwest');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',16);
```

```
%% Save data

% Save as pdf
% print2pdf(h,"Fig_4");
% print2pdf(h2,"Fig_4_3");
```

4 Codes II (Chaos)

4.1 Exercise 1

```
%-------------------------------------------------------------------------%
% Exercise 1: Lorentz system
%-------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 0.1:0.005:0.105;
y_0 = 0.1:0.005:0.105;
z_0 = 0.1:0.005:0.105;
sigma = 10;
beta = 8/3;
rho_v = [21 24.15 30];
rho = rho_v(2);

% Definition of functions
F1 = @ (t,x,y,z) sigma*(y-x);
F2 = @ (t,x,y,z) rho*x-y-x*z;
F3 = @ (t,x,y,z) -beta*z +x*y;

% Declaration of solution vectors
Delta_t = 0.01;
t_max = 100;
t_sol = 0:Delta_t:t_max;

x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
z_sol = zeros(1,length(t_sol));

for j = 1:1:length(x_0)

    x_sol(1) = x_0(j);
    y_sol(1) = y_0(j);
    z_sol(1) = z_0(j);

    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        k1 = F3(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        % Computation of coefficients sub 2
```

```
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        k2 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        k3 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        k4 = F3(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        % Compute next step
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);
        z_sol(i+1) = z_sol(i) + (Delta_t/6)*(k1 + 2*k2 + 2*k3 + k4);

    end

    h = figure(1);
    plot(t_sol,x_sol);
    hold on
    h2 = figure(2);
    plot(t_sol,y_sol);
    hold on
    h3 = figure(3);
    plot(t_sol,z_sol);
    hold on
    h4 = figure(4);
    plot(x_sol,z_sol);
    hold on

end

h = figure(1);
grid on;
grid minor
box on;
tlt = title("Lorenz system");
ylab = ylabel("x");
xlab = xlabel("time");
leg  = legend("$x_0,y_0,z_0 = 0.1$","$x_0,y_0,z_0 = 0.105$", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h2 = figure(2);
grid on;
grid minor
box on;
tlt = title("Lorenz system");
ylab = ylabel("y");
xlab = xlabel("time");
leg  = legend("$x_0,y_0,z_0 = 0.1$","$x_0,y_0,z_0 = 0.105$", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
```

```
set(leg,'fontsize',10);
hold off

h3 = figure(3);
grid on;
grid minor
box on;
tlt = title("Lorenz system");
ylab = ylabel("z");
xlab = xlabel("time");
leg  = legend("$x_0,y_0,z_0 = 0.1$","$x_0,y_0,z_0 = 0.105$", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h4 = figure(4);
grid on;
grid minor
box on;
tlt = title("Lorenz system");
ylab = ylabel("z(t)");
xlab = xlabel("x(t)");
leg  = legend("$x_0,y_0,z_0 = 0.1$","$x_0,y_0,z_0 = 0.105$", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

%% Save data

% Save as pdf
print2pdf(h,"Fig_Chaos_1_11");
print2pdf(h2,"Fig_Chaos_1_22");
print2pdf(h3,"Fig_Chaos_1_33");
print2pdf(h4,"Fig_Chaos_1_44");
```

4.2 Exercise 2 ($\rho = 21$)

```
%------------------------------------------------------------------------%
% Exercise 1: Lorentz system
%------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 0.1:0.005:0.105;
y_0 = 0.1:0.005:0.105;
z_0 = 0.1:0.005:0.105;
sigma = 10;
beta = 8/3;
rho_v = [21,24.15,30];
rho = rho_v(1);
% Definition of functions
F1 = @ (t,x,y,z) sigma*(y-x);
F2 = @ (t,x,y,z) rho*x-y-x*z;
F3 = @ (t,x,y,z) -beta*z +x*y;

% Declaration of solution vectors
Delta_t = 0.01;
t_max = 50;
t_sol = 0:Delta_t:t_max;

x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
z_sol = zeros(1,length(t_sol));

for j = 1:1:length(x_0)

    x_sol(1) = x_0(j);
    y_sol(1) = y_0(j);
    z_sol(1) = z_0(j);

    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        k1 = F3(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        % Computation of coefficients sub 2
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
```

```
        k2 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        k3 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        k4 = F3(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        % Compute next step
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);
        z_sol(i+1) = z_sol(i) + (Delta_t/6)*(k1 + 2*k2 + 2*k3 + k4);

    end

    lyap_x(:,j) = x_sol;
    lyap_y(:,j) = y_sol;
    lyap_z(:,j) = z_sol;

end

h = figure(1);
delta = log(abs(lyap_x(:,2)-lyap_x(:,1)));
plot(t_sol,delta);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=300;
final=length(t_sol);
p = polyfit(t_sol(init:final),delta(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off



h2 = figure(2);
delta2 = log(abs(lyap_y(:,2)-lyap_y(:,1)));
plot(t_sol,delta2);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("y");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
```

```
hold on

init=300;
final=length(t_sol);
p = polyfit(t_sol(init:final),delta2(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off

h3 = figure(3);
delta3 = log(abs(lyap_z(:,2)-lyap_z(:,1)));
plot(t_sol,delta3);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("z");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=300;
final=length(t_sol);
p = polyfit(t_sol(init:final),delta3(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off


%% Save data

% Save as pdf
% print2pdf(h,"Fig_Chaos_2_x1");
% print2pdf(h2,"Fig_Chaos_2_y1");
% print2pdf(h3,"Fig_Chaos_2_z1");
```

4.3 Exercise 2 ($\rho = 24.15$)

```
%------------------------------------------------------------------------%
% Exercise 1: Lorentz system
%------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 0.1:0.005:0.105;
y_0 = 0.1:0.005:0.105;
z_0 = 0.1:0.005:0.105;
sigma = 10;
beta = 8/3;
rho_v = [21,24.15,30];
rho = rho_v(2);
% Definition of functions
F1 = @ (t,x,y,z) sigma*(y-x);
F2 = @ (t,x,y,z) rho*x-y-x*z;
F3 = @ (t,x,y,z) -beta*z +x*y;

% Declaration of solution vectors
Delta_t = 0.01;
t_max = 50;
t_sol = 0:Delta_t:t_max;

x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
z_sol = zeros(1,length(t_sol));

for j = 1:1:length(x_0)

    x_sol(1) = x_0(j);
    y_sol(1) = y_0(j);
    z_sol(1) = z_0(j);

    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        k1 = F3(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        % Computation of coefficients sub 2
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
```

```
        k2 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        k3 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        k4 = F3(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        % Compute next step
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);
        z_sol(i+1) = z_sol(i) + (Delta_t/6)*(k1 + 2*k2 + 2*k3 + k4);

    end

    lyap_x(:,j) = x_sol;
    lyap_y(:,j) = y_sol;
    lyap_z(:,j) = z_sol;

end

h = figure(1);
delta = log(abs(lyap_x(:,2)-lyap_x(:,1)));
plot(t_sol,delta);
grid on;
grid minor
box on;
tlt = title("Lyapunov");
xlab = xlabel("t");
ylab = ylabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=150;
final=length(t_sol)/1.5;
p = polyfit(t_sol(init:final),delta(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off



h2 = figure(2);
delta2 = log(abs(lyap_y(:,2)-lyap_y(:,1)));
plot(t_sol,delta2);
grid on;
grid minor
box on;
tlt = title("Lyapunov");
xlab = xlabel("t");
ylab = ylabel("y");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
```

```
hold on

init=150;
final=length(t_sol)/1.5;
p = polyfit(t_sol(init:final),delta2(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off

h3 = figure(3);
delta3 = log(abs(lyap_z(:,2)-lyap_z(:,1)));
plot(t_sol,delta3);
grid on;
grid minor
box on;
tlt = title("Lyapunov");
xlab = xlabel("t");
ylab = ylabel("z");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=150;
final=length(t_sol)/1.5;
p = polyfit(t_sol(init:final),delta3(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off


%% Save data

% Save as pdf
print2pdf(h,"Fig_Chaos_2_x2");
print2pdf(h2,"Fig_Chaos_2_y2");
print2pdf(h3,"Fig_Chaos_2_z2");
```

4.4 Exercise 2 ($\rho = 30$)

```
%-------------------------------------------------------------------------%
% Exercise 1: Lorentz system
%-------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 0.1:0.005:0.105;
y_0 = 0.1:0.005:0.105;
z_0 = 0.1:0.005:0.105;
sigma = 10;
beta = 8/3;
rho_v = [21,24.15,30];
rho = rho_v(3);
% Definition of functions
F1 = @ (t,x,y,z) sigma*(y-x);
F2 = @ (t,x,y,z) rho*x-y-x*z;
F3 = @ (t,x,y,z) -beta*z +x*y;

% Declaration of solution vectors
Delta_t = 0.01;
t_max = 50;
t_sol = 0:Delta_t:t_max;

x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
z_sol = zeros(1,length(t_sol));

for j = 1:1:length(x_0)

    x_sol(1) = x_0(j);
    y_sol(1) = y_0(j);
    z_sol(1) = z_0(j);

    for i =1:length(t_sol)-1
        % Computation of coefficients sub 1
        i1 = F1(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        j1 = F2(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        k1 = F3(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
        % Computation of coefficients sub 2
        i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
        j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*
    Delta_t/2);
```

```
        k2 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
        % Computation of coefficients sub 3
        i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        k3 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
        % Computation of coefficients sub 4
        i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        k4 = F3(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
        % Compute next step
        x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
        y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);
        z_sol(i+1) = z_sol(i) + (Delta_t/6)*(k1 + 2*k2 + 2*k3 + k4);

    end

    lyap_x(:,j) = x_sol;
    lyap_y(:,j) = y_sol;
    lyap_z(:,j) = z_sol;

end

h = figure(1);
delta = log(abs(lyap_x(:,2)-lyap_x(:,1)));
plot(t_sol,delta);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("x");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=150;
final=length(t_sol)/3-300;
p = polyfit(t_sol(init:final),delta(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off



h2 = figure(2);
delta2 = log(abs(lyap_y(:,2)-lyap_y(:,1)));
plot(t_sol,delta2);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("y");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
```

```
hold on

init=150;
final=length(t_sol)/3-300;
p = polyfit(t_sol(init:final),delta2(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off

h3 = figure(3);
delta3 = log(abs(lyap_z(:,2)-lyap_z(:,1)));
plot(t_sol,delta3);
grid on;
grid minor
box on;
tlt = title("Lyapunov exponents");
xlab = xlabel("t");
ylab = ylabel("z");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
hold on

init=150;
final=length(t_sol)/3-300;
p = polyfit(t_sol(init:final),delta3(init:final),1);
f1 = polyval(p,t_sol(init:final));
plot(t_sol(init:final),f1,'LineWidth',2);
str = {strcat('$\lambda = $' , num2str(p(1)))};
str = [str , strcat('$\lambda = $' , num2str(p(1)))];
legend(str{2},'location','best');
hold off


%% Save data

% Save as pdf
print2pdf(h,"Fig_Chaos_2_x3");
print2pdf(h2,"Fig_Chaos_2_y3");
print2pdf(h3,"Fig_Chaos_2_z3");
```

4.5 Exercise 3

```
%-------------------------------------------------------------------------%
% Exercise 3: Henon Map
%-------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

a = 1.4;
b = 0.3;
iterations = 10000;
x = zeros(1,iterations+1);
y = zeros(1,iterations+1);
x(1) = 0.1;
y(1) = 0.1;

for (i = 1:1:iterations)
   x(i+1) = y(i) + 1 -a*x(i).^2;
   y(i+1) = b*x(i);
end

h = figure(1);
plot(x,y,'.');

grid on;
grid minor
box on;
tlt = title("Henon map");
ylab = ylabel("$y_n$");
xlab = xlabel("$x_n$");
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);

%% Save data

% Save as pdf
% print2pdf(h,"Fig_Chaos_3");
```

4.6 Exercise 4

```
%-------------------------------------------------------------------------%
% Exercise 4: Rossler system
%-------------------------------------------------------------------------%

% Date: 17/04/2021
% Author/s: Ivan Sermanoukian Molina
% Subject: Nonlinear systems, chaos and control in engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clc
clear all
close all

% LaTeX configuration
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% RUNGE-KUTTA RK4 option

% Initial conditions
x_0 = 1;
y_0 = 1;
z_0 = 1;

b = 2;
c = 4;
a_vect = 0:0.1:0.4;

% Declaration of solution vectors
Delta_t = 0.001;
t_max = 40;
t_sol = 0:Delta_t:t_max;

t_sol = 0:Delta_t:t_max;
x_sol = zeros(1,length(t_sol));
y_sol = zeros(1,length(t_sol));
z_sol = zeros(1,length(t_sol));
x_sol(1) = x_0;
y_sol(1) = y_0;
z_sol(1) = z_0;

for j = 1:1:length(a_vect)

    % Definition of functions
    F1 = @ (t,x,y,z) -y-z;
    F2 = @ (t,x,y,z) x+a_vect(j)*y;
    F3 = @ (t,x,y,z) b+z*(x-c);

        for i =1:length(t_sol)-1
            % Computation of coefficients sub 1
            i1 = F1(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
            j1 = F2(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
            k1 = F3(t_sol(i),x_sol(i),y_sol(i),z_sol(i));
            % Computation of coefficients sub 2
            i2 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1
    *Delta_t/2);
            j2 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1
    *Delta_t/2);
```

```
            k2 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i1*Delta_t/2,y_sol(i)+j1*Delta_t/2,z_sol(i)+k1*Delta_t/2);
            % Computation of coefficients sub 3
            i3 = F1(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
            j3 = F2(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
            k3 = F3(t_sol(i)+Delta_t/2,x_sol(i)+i2*Delta_t/2,y_sol(i)+j2*Delta_t/2,z_sol(i)+k2*Delta_t/2);
            % Computation of coefficients sub 4
            i4 = F1(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
            j4 = F2(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
            k4 = F3(t_sol(i)+Delta_t,x_sol(i)+i3*Delta_t,y_sol(i)+j3*Delta_t,z_sol(i)+k3*Delta_t);
            % Compute next step
            x_sol(i+1) = x_sol(i) + (Delta_t/6)*(i1 + 2*i2 + 2*i3 + i4);
            y_sol(i+1) = y_sol(i) + (Delta_t/6)*(j1 + 2*j2 + 2*j3 + j4);
            z_sol(i+1) = z_sol(i) + (Delta_t/6)*(k1 + 2*k2 + 2*k3 + k4);

        end

        h = figure(1);
        plot(t_sol,x_sol)
        hold on
        h2 = figure(2);
        plot(t_sol,y_sol);
        hold on
        h3 = figure(3);
        plot(t_sol,z_sol);
        hold on
        h4 = figure(4);
        plot3(x_sol,y_sol,z_sol);
        hold on
        h5 = figure(5);
        pspectrum(x_sol,t_sol)
        hold on
        h6 = figure(6);
        pspectrum(y_sol,t_sol)
        hold on
        h7 = figure(7);
        pspectrum(z_sol,t_sol)
        hold on

end

h = figure(1);
grid on;
grid minor
box on;
tlt = title("Rossler system");
ylab = ylabel("x");
xlab = xlabel("time");
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'northeast');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h2 = figure(2);
grid on;
grid minor
```

```
box on;
tlt = title("Rossler system");
ylab = ylabel("y");
xlab = xlabel("time");
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'southeast');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h3 = figure(3);
grid on;
grid minor
box on;
tlt = title("Rossler system");
ylab = ylabel("z");
xlab = xlabel("time");
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h4 = figure(4);
grid on;
grid minor
box on;
tlt = title("Rossler system");
xlab = xlabel("x");
ylab = ylabel("y");
zlab = zlabel("z");
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'best');
set(tlt,'fontsize',16);
set(xlab,'fontsize',16);
set(ylab,'fontsize',16);
set(leg,'fontsize',10);
hold off

h5 = figure(5);
grid on;
grid minor
box on;
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'best');
set(leg,'fontsize',10);
hold off

h6 = figure(6);
grid on;
grid minor
box on;
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'best');
set(leg,'fontsize',10);
hold off

h7 = figure(7);
grid on;
grid minor
box on;
leg  = legend("a = 0.0","a = 0.1","a = 0.2","a = 0.3","a = 0.4", 'location', 'best');
set(leg,'fontsize',10);
hold off

```

```
%% Save data

% Save as pdf
% print2pdf(h,"Fig_4_x");
% print2pdf(h2,"Fig_4_y");
% print2pdf(h3,"Fig_4_z");
% print2pdf(h4,"Fig_4_xyz");
% print2pdf(h5,"Fig_4_spec_x");
% print2pdf(h6,"Fig_4_spec_y");
% print2pdf(h7,"Fig_4_spec_z");
```

References

[1] Antonio J. Pons. "Linear Systems". Universitat Politècnica de Catalunya. 2021.

[2] Antonio J. Pons. "Chaos". Universitat Politècnica de Catalunya. 2021.

[3] Steven H Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[4] Antonio J. Pons. "Non-Linear Systems. Part 1." Universitat Politècnica de Catalunya. 2021.

[5] Antonio J. Pons. "Linear Systems. Part 2." Universitat Politècnica de Catalunya. 2021.