

Report for exercise 3 from group A

Tasks addressed: 5
Authors: ALEX PASQUALI (03754113)
 NICOLA GUGOLE (03753996)

Last compiled: 2021-12-09
Source code: <https://github.com/AlexPasqua/MLCMS-exercises>

The work on tasks was divided in the following way:

ALEX PASQUALI (03754113)	Task 1	50%
	Task 2	50%
	Task 3	50%
	Task 4	50%
	Task 5	50%
NICOLA GUGOLE (03753996)	Task 1	50%
	Task 2	50%
	Task 3	50%
	Task 4	50%
	Task 5	50%

Report on task 1, Vector fields, orbits, and visualization

This task consists of displaying the phase portraits displayed in figure 2.5 of [2]. In particular they are a *node* and a *focus* phase portraits, both *stable* and *unstable*, plus an *unstable saddle* phase portrait.

The linear dynamical system in object has a state space $X = \mathbb{R}^2$, $I = \mathbb{R}$ and parameter $\alpha \in \mathbb{R}$. Its flow is defined as follows:

$$\left. \frac{\partial \phi_\alpha(t, x)}{\partial t} \right|_{t=0} = A_\alpha x \quad (1)$$

where $x, t \in I$ and A is a 2×2 real matrix parametrized by α . Specifically, we obtained these portraits with the following configurations (where λ_1 and λ_2 indicate the 2 eigenvalues of A):

- **Unstable focus:** $\alpha = 0.1$, $A = \begin{bmatrix} \alpha & \alpha \\ -\frac{1}{4} & 0 \end{bmatrix}$, $\lambda_1 = 0.05 + 0.15i$, $\lambda_2 = 0.05 - 0.15i$
- **Unstable saddle:** $\alpha = -0.2$, $A = \begin{bmatrix} -\alpha & \alpha \\ -\frac{1}{4} & 0 \end{bmatrix}$, $\lambda_1 = 0.345$, $\lambda_2 = -0.145$
- **Unstable node:** $\alpha = 2$, $A = \begin{bmatrix} \alpha & \alpha \\ -\frac{1}{4} & 0 \end{bmatrix}$, $\lambda_1 = 1.707$, $\lambda_2 = 0.293$
- **Stable node:** $\alpha = -0.25$, $A = \begin{bmatrix} \alpha & 0 \\ 0 & -1 \end{bmatrix}$, $\lambda_1 = -0.25$, $\lambda_2 = -1$
- **Stable focus:** $\alpha = 0.1$, $A = \begin{bmatrix} -\alpha & \alpha \\ -\frac{1}{4} & 0 \end{bmatrix}$, $\lambda_1 = -0.05 + 0.15i$, $\lambda_2 = -0.05 - 0.15i$

These systems are of course **not** topologically equivalent because their phase portraits (Figure 1) show qualitatively distinct behaviors around their fixed point. In particular, some of them are stable (Figure 1(d) and 1(e)) and some unstable (Figure 1(a), 1(b) and 1(c)). Moreover, Figure 1(a)-(e) represent focuses, Figure 1(b) represents a saddle and Figure 1(c)-(d) represent nodes.

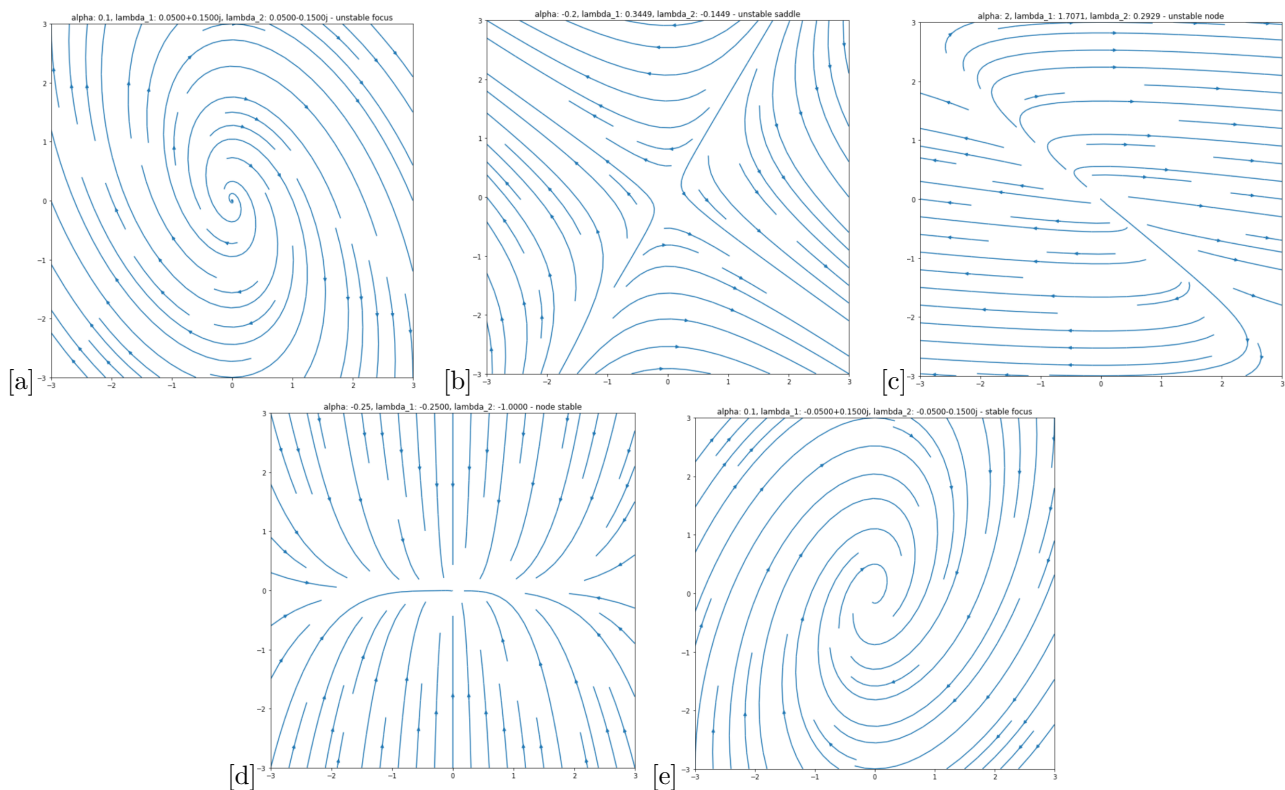


Figure 1: Phase portraits of task 1, namely: (a) Unstable focus; (b) Unstable saddle; (c) Unstable node; (d) Stable node; (e) Stable focus.

Report on task 2, Common bifurcations in nonlinear systems

This task asks for the analysis of the bifurcation for two dynamical systems:

$$\dot{x} = \alpha - x^2$$

and

$$\dot{x} = \alpha - 2x^2 - 3$$

The bifurcation diagrams for the two systems are described in **Figure 2**.

The two systems express a similar behaviour, with both having no steady state before a parametric value coincides with the *bifurcation point* (e.g. $\alpha = 0$ for the *left* model). From then on the system has two different steady states. The bifurcation type for these two systems is called *saddle-node bifurcation*, with one of the two steady states being repulsive and one attractive, colored in *orange* and *blue* in the **Figure 2**.

To answer the rest of the task questions it is relevant to cite ([2], p.59): "For instance, it is natural to expect that two equivalent systems have the same number of equilibria and cycles of the same stability types. The "relative position" of these invariant sets and the shape of their regions of attraction should also be similar for equivalent systems."

Therefore it is safe to assert that the two systems are not topologically equivalent at $\alpha = 1$ since the system on the *left* has two steady states and system on the *right* has no steady states. On the contrary, the two systems are topologically equivalent at $\alpha = -1$, since they have the same number of steady states and they actually have the same *normal form*. One might hypothesize the two systems have the same normal form since their bifurcation diagrams are qualitatively equivalent and taking a look at the graphics there are some hints giving more proofs, using the two differences in the system (the x^2 coefficient and the x^0 coefficient):

- the basic coefficient difference is having the +3 at the end of the equation, an element that simply delays the whole bifurcation. In fact the bifurcation point difference between the two systems is exactly 3.
- the coefficient multiplying x^2 is simply changing the amplitude of the y coordinate, stretching it by two times, which coincides with the multiplying coefficient, which is 2.

It is therefore possible to obtain a mapping between the two dynamical systems, using a delay followed by a multiplication to map back the *right* system into the *left* one.

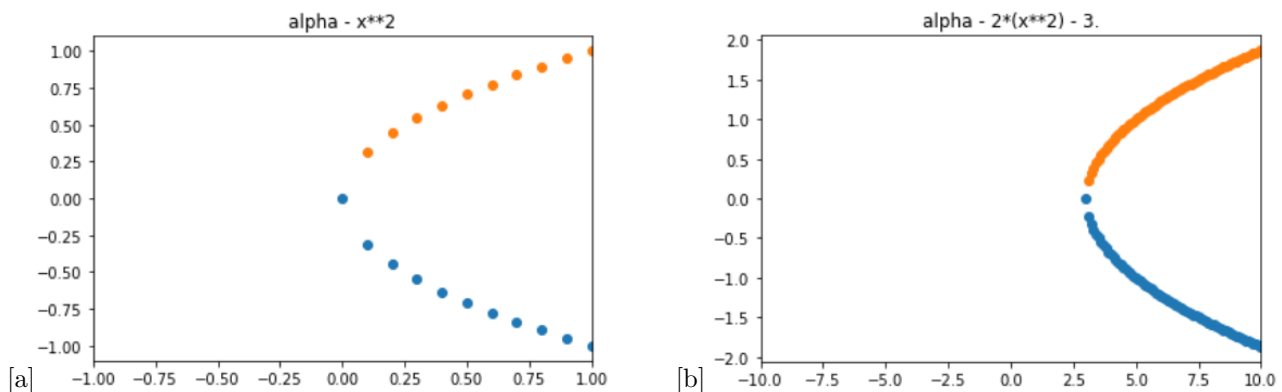


Figure 2: bifurcation diagrams for the two requested systems - *alpha* on the x axis, *x* coordinate on the y axis

Report on task 3, Bifurcations in higher dimensions

In this task an important bifurcation for systems with one parameter (but for two-dimensional state spaces) is analyzed: *the Andronov-Hopf bifurcation*, representable with the vector field in normal form:

$$\dot{x}_1 = \alpha x_1 - x_2 - x_1(x_1^2 + x_2^2)$$

$$\dot{x}_2 = x_1 + \alpha x_2 - x_2(x_1^2 + x_2^2)$$

Task 3.1: The change in behaviour of such a system can be shown to intuitively understand the bifurcation that is happening. **Figure 3** gives three different portraits by varying the α parameter, specifically setting it to -1, 0 and 1:

- for $\alpha = -1$ a single attractive stable point can be located in the origin.
- for $\alpha = 1$ an unstable point is located at the origin and a stable *limit cycle* is present near the origin.
- for $\alpha = 0$, as reported in [2] (p 85), a *weakly attractive focus* is present in the origin. This parametric value represents the bifurcation point, moving from a stable fixed point to a limit cycle behaviour.

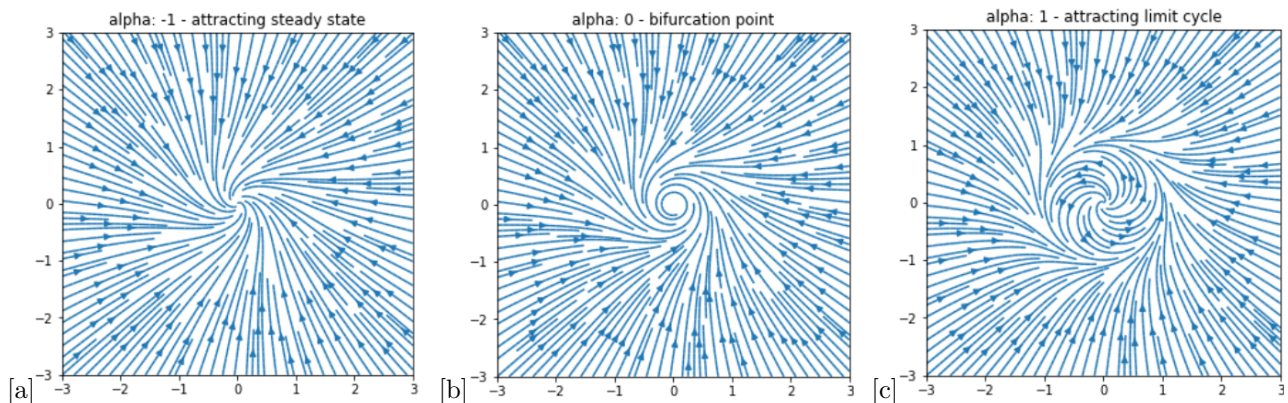


Figure 3: bifurcation shown three phase diagrams at representative values of α .

Task 3.2: The second part of the task asks for the numerical computation and visualization of two orbits (starting at $[2, 0]$ and $[0.5, 0]$) given a specific α value (in the requested case $\alpha = 1$). The two requested orbits have been computed taking advantage of *Euler's method*. Given the particular α value one should expect for any point not starting in the origin to converge to the limit cycle, either by repulsion from the origin or by attraction from outside the limit cycle.

Figure 4 show the method implemented to perform the task, where different parameters gave the possibility of tuning the number of iterations as well as the time step for the algorithm. The core of the method is highlighted by the *orange box*, where the two odes are assigned to be used by the underlying Euler's method step.

```
def plot_traj(system: list, start_pos, alpha=1, ts=1e-3, iter_num=10000, fig_size=10):
    """
    Function to plot a trajectory for a certain amount of iterations, given a starting point and a ode system.
    @param system: 2 ode system to get movement vector
    @param start_pos: initial coordinates of point
    @param alpha: system's parameter
    @param ts: step length
    @param iter_num: number of iterations to execute
    @param fig_size: gives width and height of plotted figure
    """
    w = 3
    X, Y = [], [] # lists to store positioning update
    x = start_pos
    for i in range(int(iter_num)):
        x1, x2 = x[0], x[1]
        X.append(x1)
        Y.append(x2)
        # find vectors for next movement
        u = eval(system[0])
        v = eval(system[1])
        # execute step - Euler's method
        x = x + ts * np.array([u, v])
    plt.figure(figsize=(fig_size, fig_size))
    # plot trajectory
    plt.plot(X, Y)
    plt.scatter(X[0], Y[0]) # put the starting point in evidence
    plt.title(f"Trajectory [{start_pos[0]}, {start_pos[1]}]")
    plt.xlim(-w, w)
    plt.ylim(-w, w)
    plt.show()
```

Figure 4: snippet for trajectory plotting

Finally **Figure 5** plots the result, which follow the expectations. The blue dot denotes the starting point for each trajectory.



Figure 5: system orbits starting at point (2,0) and (0.5,0) for $\alpha = 1$

Task 3.3: The last subtask asks for the analysis of a double parametered bifurcation, the *cusp bifurcation*, with the following normal form:

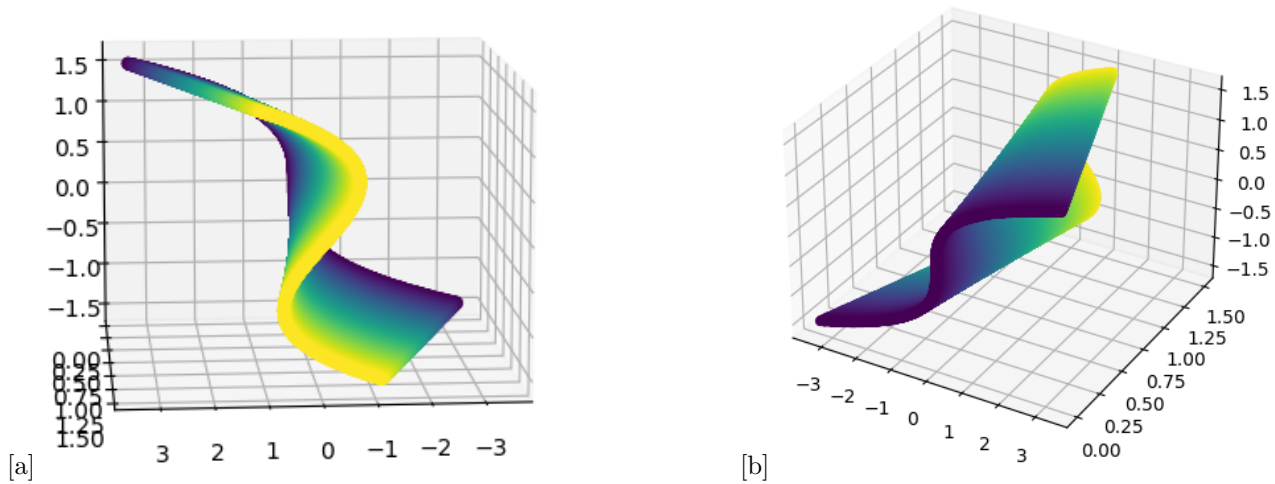
$$\dot{x} = \alpha_1 + \alpha_2 x - x^3$$

Following the hint provided in the exercise sheet, the core snippet shown in **Figure 6** generates the data triples by first sampling α_2 and x values, to then compute the relative α_1 values by reversing the normal form (we know that a fixed point is a point such that $\dot{x} = 0$).

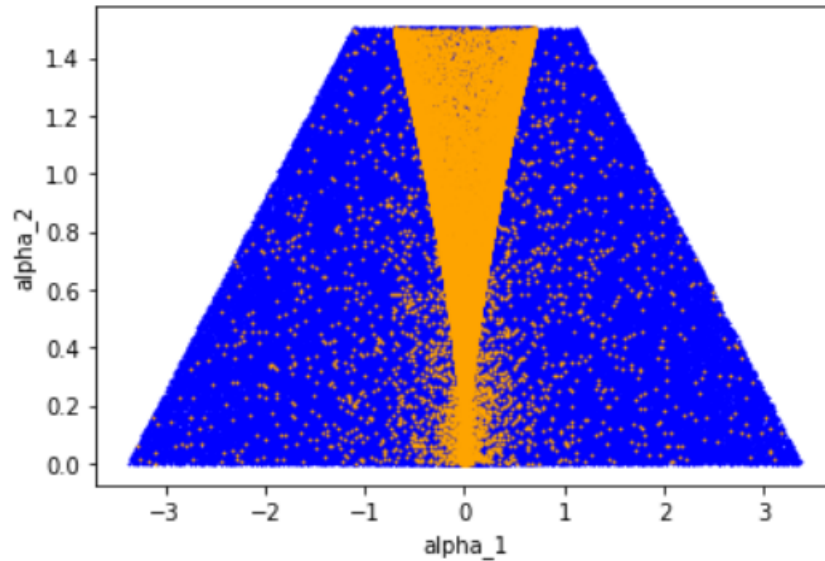
```
# sample alpha_2 and x n times
alphas_two_sampled = np.random.uniform(0, alpha_two_limit, n)
x_values_sampled = np.random.uniform(-x_limit, x_limit, n)
# create placeholders for final alphas_1, alphas_2 and x values and fill them
alphas_two = []
alphas_one = []
x_values = []
for a2, x in zip(alphas_two_sampled, x_values_sampled):
    # evaluate alphas_1 reversing the equation (dx = a1 + a2*x - x**3 = 0)
    alphas_one.append(-a2 * x + x ** 3)
    alphas_two.append(a2)
    x_values.append(x)
```

Figure 6: *cusp* implementation core snippet

The 3D plot of such a system (with an adequate colormap to try and highlight better the surface shape) is shown in **Figure 7** from two different points of view. An *hysteresis phenomenon* is visible and is responsible for the cusp generation.

Figure 7: *cusp* bifurcation system in 3D from 2 different angulations

The bifurcation gets his name in fact by the presence of a *cusp* shape when projecting the x dimension in the α_1 and α_2 plane. The particular surface shown in **Figure 7** does in fact imply that for a given (α_1, α_2) configuration there can be one or more than one steady states, implying a bifurcation. In **Figure 8** is shown a projection where coordinates with multiple x values are shown in *orange* and ones with only one value are shown in *blue*. To achieve such a feat the alphas values coming out of the snippet shown in **Figure 6** have been rounded to overcome floating point granularity not giving the expected result. This procedure was nevertheless not flawless, bringing also some noise due to the original value sampling fused with the rounding previously explained.

Figure 8: *cusp* projection on the alpha space

Report on task 4, Chaotic dynamics

In this task we analyze some aspects of the *logistic map* system, expressed by the following discrete map:

$$x_{n+1} = rx_n(1 - x_n) \quad (2)$$

with $x_n \in \mathbb{R}$, $n \in \mathbb{N}$ and $r \in (0, 4]$.

To analyze this system, we created some helping functions that have been put in the script `utilities.py`, in particular:


```

def plot_logistic_map_bifurcations(min_r: float = 0., max_r: float = 4.):
    """
    Plots the bifurcations of the logistic map system
    :param min_r: minimum r to test the system with
    :param max_r: maximum r to test the system with
    """
    n = 5000 # number of runs
    r = np.linspace(min_r, max_r, n)
    iterations = 1000 # number of steps per run
    last = 100 # number of steps, from the last one, to plot for each run
    x = 1e-5 * np.ones(n) # starting x for each run

    fig, ax = plt.subplots(1, 1, figsize=(8, 8))
    for i in range(iterations):
        x = logistic(r, x) # compute a step for each run
        # if we're in the last 'last' iterations, plot the points for each run
        if i >= (iterations - last):
            ax.plot(r, x, 'k', alpha=.25)
    ax.set_title("Logistic map bifurcation diagram")
    ax.set_xlabel("r")
    ax.set_ylabel("x")
    plt.tight_layout()
    plt.show()

```

Figure 9: Code of the function `plot_logistic_map_bifurcations`

- `logistic`: computes the new state x_{n+1} of the system (Eq. 2) given the current state x_n and the r parameter.
- `plot_logistic_map_bifurcations`: takes 2 values of the parameter r - a minimum (r_{min}) and a maximum one (r_{max}) - and plots the behavior of the system of Eq. 2 $\forall r \in [r_{min}, r_{max}]$. To do this, the system is run n times for m iterations, of which, only the last p values are kept. This is because, starting from an initial point x_0 , the system will need a certain number of steps to reach a steady state, and since the objective of this plot is to show the steady states associated to different values of r , all the states assumed by the system in this initial phase are not of interest, and are therefore discarded. Note: we do not keep only the last state assumed by the system during the m iterations because the system may find itself in a limit cycle or in a chaotic behaviour, therefore it is needed to have more than 1 value to see if there are a certain number of recurrent states associated to a certain value of r . The code of this function is shown in Figure 9.
- `logistic_map_cobweb_plot`: shows the cobweb plot, which is a tool used to investigate the qualitative behavior of one-dimensional iterated functions, such as the logistic map [1].

Task 4.1: By varying r from 0 to 2, the bifurcation diagram obtained with the function `plot_logistic_map_bifurcations` illustrated previously is shown in Figure 10(a).

It is possible to see that there is one steady state for each value of r , in particular:

- $r \in (0, 1] \implies x = 0$
- $r \in (1, 2] \implies x = \frac{r-1}{r}$

In this case, there is no bifurcation and the steady state of the system is always one ($\forall r \in (0, 2]$). This fixed point is attractive, because, since each point in the bifurcation diagram is obtained plotting the last p states assumed by the system for a certain value of r , if the steady state in object was repulsive, we would have

multiple points per value of r , instead this is not the case.

Furthermore, in the cobweb plot in Figure 11 it is possible to see how the system tends to a certain steady state: 0 for $r = 0.8$, 0.44 for $r = 1.8$.

Task 4.2: Varying r from 2 to 4, the number of steady states continue to double more and more often, with the exceptions of few stability islands. This behavior is shown in Figure 10(b). Moreover, Figure 12 shows the various limit cycles with an increasing number of steady states as r increases.

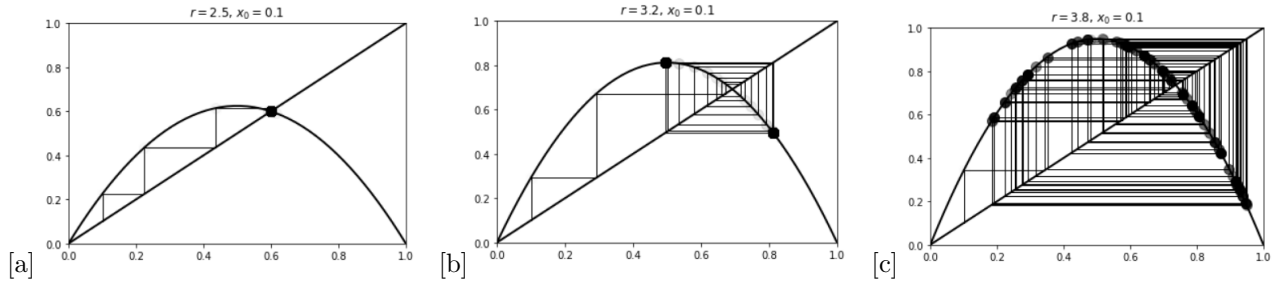


Figure 12: Cobweb plot with (a) $r = 2.5$, (b) $r = 3.2$ and (c) $r = .8$.

Task 4.3: Figure 10(c) shows the bifurcations of the logistic map for $r \in (0, 4]$.

- $\forall r \in (0, 1] : x = 0$ is a steady state
- $\forall r \in (1, 3) : x = \frac{r-1}{r}$ is a steady state
- $\forall r \geq 3$ there are limit cycles that include a different number of points, initially an oscillation between two states, then at roughly 3.4 the oscillation becomes between four states. This number tend to increase (doubling) as r increases, until at approximately $r = 3.57$ the behaviour becomes chaotic with some exceptions being sporadic stability islands where once again oscillation between a little number of states is present.

Part 2 - Lorenz attractor: The focus of this part is to analyze the *Lorenz attractor* [3]. The system is defined as follows:

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

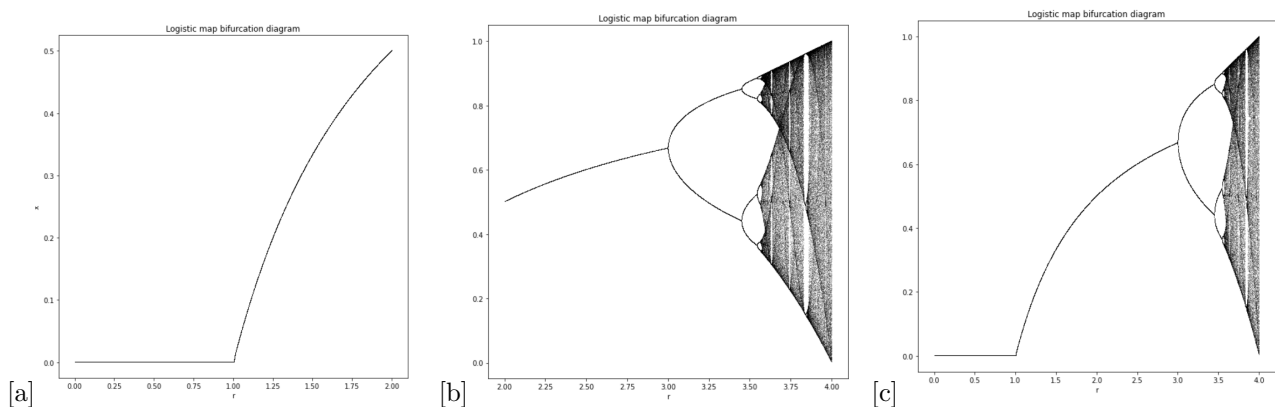
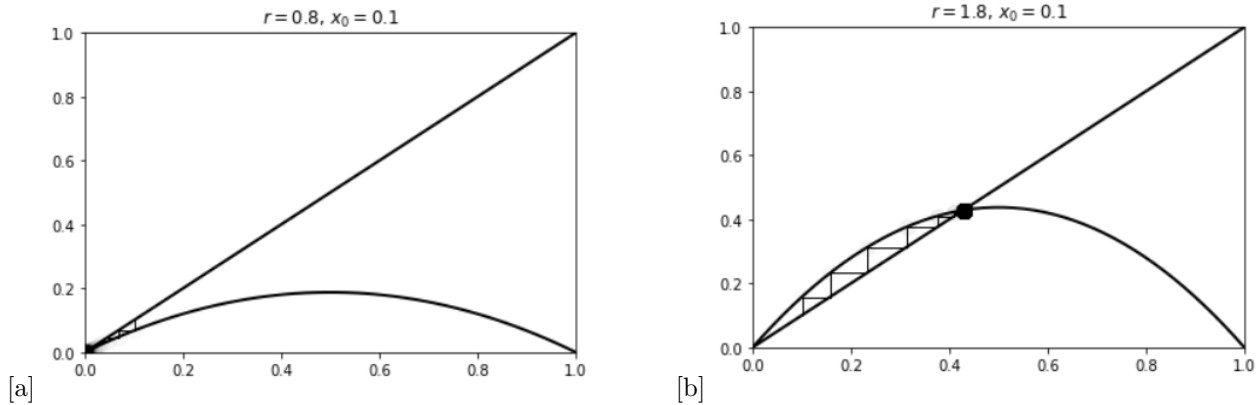
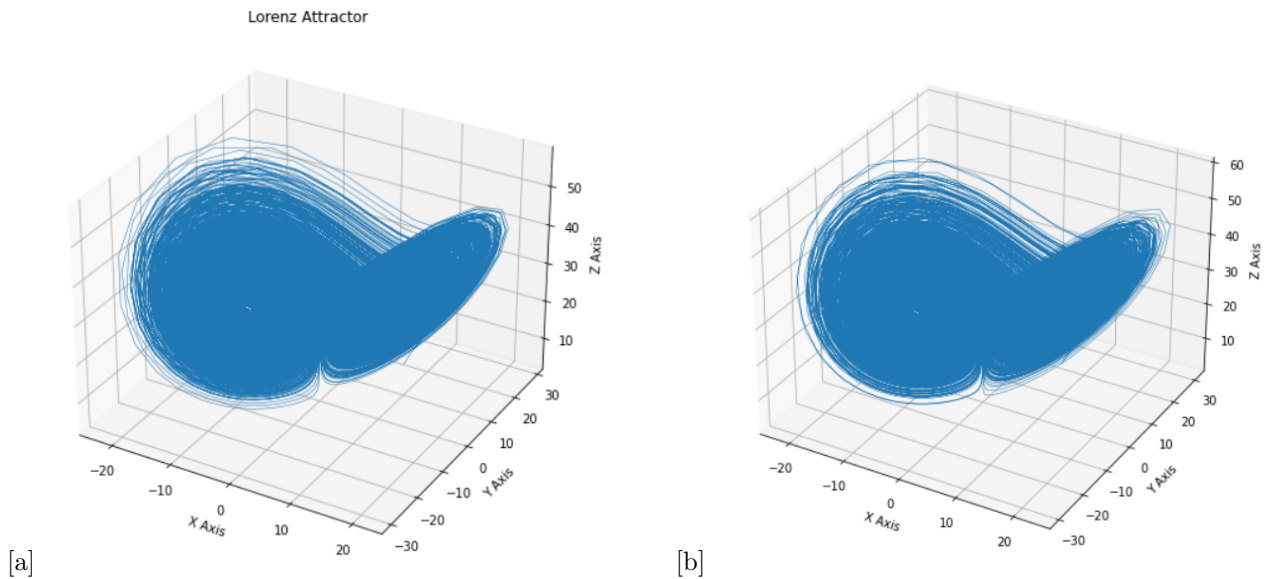


Figure 10: Bifurcations of the logistic map with: $r \in (0, 2]$ (a); $r \in [2, 4]$ (b); $r \in (0, 4]$ (c).

Figure 11: Cobweb plot with (a) $r = 0.8$ and (b) $r = 1.8$.

where x , y and z represent the coordinates of a point/state of the system, while σ , ρ and β are parameters. The system has been simulated for 1000 seconds and its trajectory has been plotted. Specifically, starting from $x_0 = (10, 10, 10)$ we obtain the plot depicted in Figure 13(a), while 13(b) shows the plot starting from $\hat{x}_0 = (10 + 10^{-8}, 10, 10)$. In both cases the parameters were set to $\sigma = 10$, $\beta = 8/3$, $\rho = 28$. Since the image is a little dense, simulations have been run with the same configurations but for 200 seconds instead of 1000. This is shown in Figure 14.

Since the system is chaotic, the minimal perturbation in the starting point (going from x_0 to \hat{x}_0) is visible on a macroscopic scale when plotting the trajectory of the system starting from that point. The difference between the two trajectories, expressed as $\|x(t) - \hat{x}(t)\|^2 \forall t$, is shown in Figure 15, where it is possible to notice how this difference assumes relatively large values. Specifically, this distance becomes larger than 1 after 13.48 seconds.

Figure 13: Trajectories of the *Lorenz attractor* simulated for 1000 seconds, starting from (a) $x_0 = (10, 10, 10)$ and (b) $\hat{x}_0 = (10 + 10^{-8}, 10, 10)$ with $\sigma = 10$, $\beta = 8/3$, $\rho = 28$.

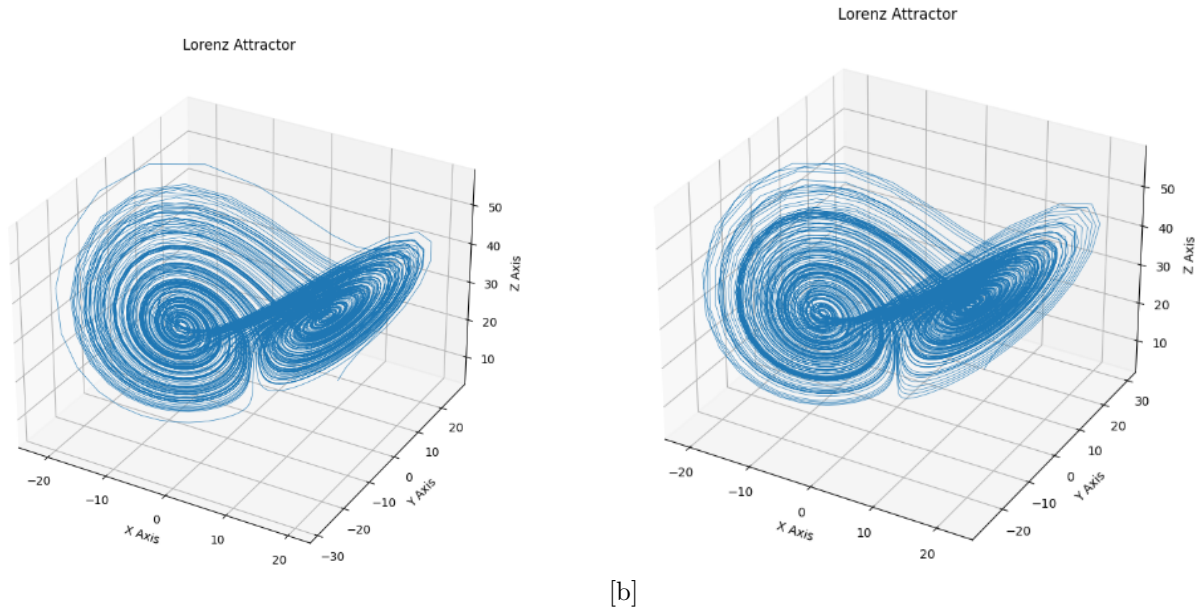


Figure 14: Trajectories of the *Lorenz attractor* simulated for 200 seconds, starting from (a) $x_0 = (10, 10, 10)$ and (b) $\hat{x}_0 = (10 + 10^{-8}, 10, 10)$ with $\sigma = 10$, $\beta = 8/3$, $\rho = 28$.

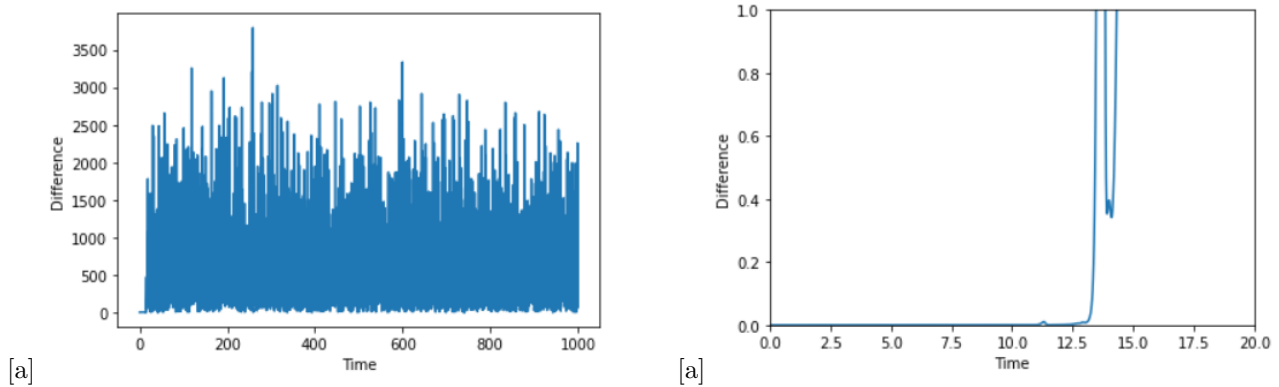


Figure 15: The difference between the trajectory of the *Lorenz attractor* starting from $x_0 = (10, 10, 10)$ and $\hat{x}_0 = (10 + 10^{-8}, 10, 10)$. For each value t on the x-axis the plot shows $\|x(t) - \hat{x}(t)\|^2$. (b) shows simply a close-up of (a): the plot is the same but the x and y axes have been shortened.

Now, setting $\rho = 0.5$ and repeating these tests, it is possible to observe in Figure 16 that the trajectories are identical to each other, but very different from before, indicating that a bifurcation occurs for a certain value of ρ between 0.5 and 28.

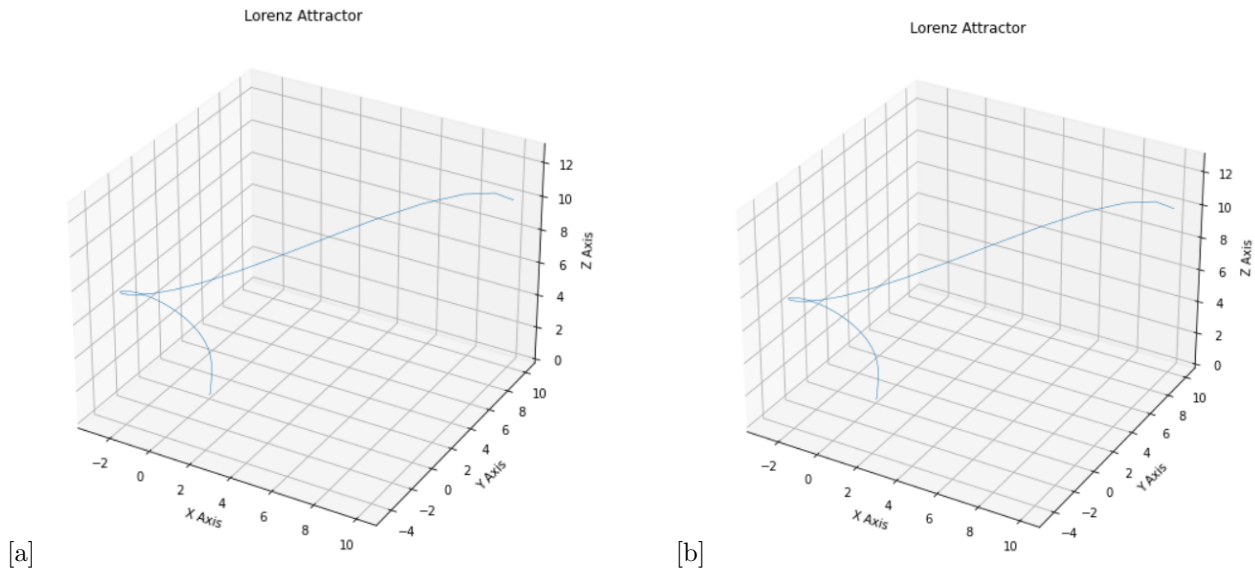


Figure 16: Trajectories of the *Lorenz attractor* simulated for 1000 seconds, starting from (a) $x_0 = (10, 10, 10)$ and (b) $\hat{x}_0 = (10 + 10^{-8}, 10, 10)$ with $\sigma = 10$, $\beta = 8/3$, $\rho = 0.5$.

Report on task 5, Bifurcations in crowd dynamics

Task 5.1 - 5.2: The Jupyter notebook `sir_bed_mode_unfinished.ipynb`, as the name suggests, is not complete. The main aspect that had to be addressed is the fact that the equations that define the SIR model are not implemented (Figure 17). It was straight forward to implement those equations having all the parameters needed (passed as arguments to the function): μ_0 , μ_1 , β , A , d , ν and b .

Other than that all the functions present in the notebook (namely `mu`, `R0`, `h` and `model`) have been moved to a new script called `sir_utilities.py`. Furthermore some code present in the notebook - for example the one to plot the system's trajectories and the behavior of the 3 variables (S, I, R) over time - has been put in specific functions placed in the new script to make everything more modular, tidy and reusable.

As a final point, the documentation has been improved or added when (partially) missing.

```
def model(t, y, mu0, mu1, beta, A, d, nu, b):
    """
    SIR model including hospitalization and natural death.

    Parameters:
    -----
    mu0
        Minimum recovery rate
    mu1
        Maximum recovery rate
    beta
        average number of adequate contacts per unit time with infectious individuals
    A
        recruitment rate of susceptibles (e.g. birth rate)
    d
        natural death rate
    nu
        disease induced death rate
    b
        hospital beds per 10,000 persons
    """
    S, I, R = y[:]
    m = mu(b, I, mu0, mu1)

    dSdt = -S # add the correct model here
    dIdt = 0
    dRdt = 0

    return [dSdt, dIdt, dRdt]
```

[a]

```
def model(t, y, mu0, mu1, beta, A, d, nu, b):
    """
    SIR model including hospitalization and natural death.

    Parameters:
    -----
    mu0
        Minimum recovery rate
    mu1
        Maximum recovery rate
    beta
        average number of adequate contacts per unit time with infectious individuals
    A
        recruitment rate of susceptibles (e.g. birth rate)
    d
        natural death rate
    nu
        disease induced death rate
    b
        hospital beds per 10,000 persons
    """
    S, I, R = y[:]
    m = mu(b, I, mu0, mu1)

    dSdt = A - d * S - (beta * S * I) / (S + I + R)
    dIdt = -(d + nu) * I - m * I + (beta * S * I) / (S + I + R)
    dRdt = m * I - d * R

    return [dSdt, dIdt, dRdt]
```

[b]

Figure 17: The code implementing the SIR model's equations. (a) the unfinished code given in the example notebook; (b) the completed code.

Task 5.3: The parameterized proposed scenario follows the ruled behaviour described in [2], Section 4, where the study of parameterized models being non *globally asymptotically stable* in E_0 is carried out. E_0 is in

our case the initial condition $[200, 0, 0]$ and our parameterized system is non *globally asymptotically stable* in that point since $R_0 < 1$ (0.99) and $\beta > d + \nu + \mu_0$ ($11.5 > 11.1$).

The scenario does therefore produce a bifurcation when varying the b parameter, as graphically portrayed in **Figure 19**. What is represented in the aforementioned figure is a trajectory plotting by varying b in the range $(0.01, 0.03)$ (with 0.001 as step), with each subfigure showing the trajectory of three different initial points:

- first point is initiated with **[195.3, 0.052, 4.4]** as SIR values. Its trajectory is colored in **red** and its initial (**x symbol**) and final (**rombold symbol**) trajectory landmarks are colored in **black**.
- second point is initiated with **[195.7, 0.03, 3.92]** as SIR values. Its trajectory is colored in **green** and its initial (**x symbol**) and final (**rombold symbol**) trajectory landmarks are colored in **light violet**.
- last point is initiated with **[193, 0.08, 6.21]** as SIR values. Its trajectory is colored in **blue** and its initial (**x symbol**) and final (**rombold symbol**) trajectory landmarks are colored in **orange**.

To discuss the behaviour change when increasing b :

- for $b < 0.022$ the system shows a local attractive point in $[193.666, 0.057, 5.71]$ for the three points in consideration, resembling ever more a stable focus as b increases in value (**Figure 19.c-d**).
- at $b = 0.022$ the system encounters its bifurcation point, with $[193.666, 0.057, 5.71]$ being a weak stable point since it is attracting the nearest point (the red plot) while a new behaviour shows up, with the other two points being attracted to a limit cycle (**Figure 19.f**). In particular the limit cycle is attractive both from the inside (the green plot) and from the outside (the blue plot).
- for $b > 0.022$ the system slowly converges once again to having an attractive steady state, which is actually E_0 $[200, 0, 0]$ (**Figure 19.g-h**).

Just to address it, **Figure 19** is composed of 2D images since 3D images such as the one portrayed in **Figure 18** do in our opinion make it harder to visualize single points while not adding much more qualitative information with respect to 2D.

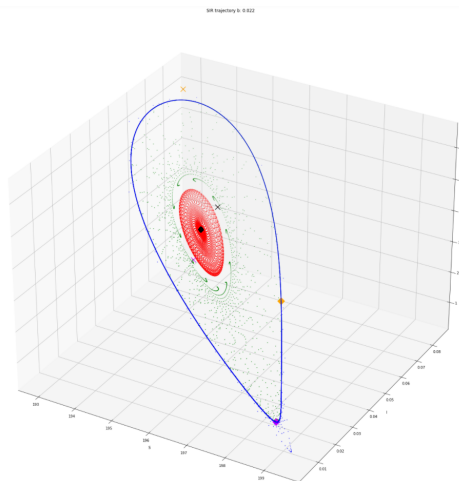


Figure 18: bifurcation point in 3D

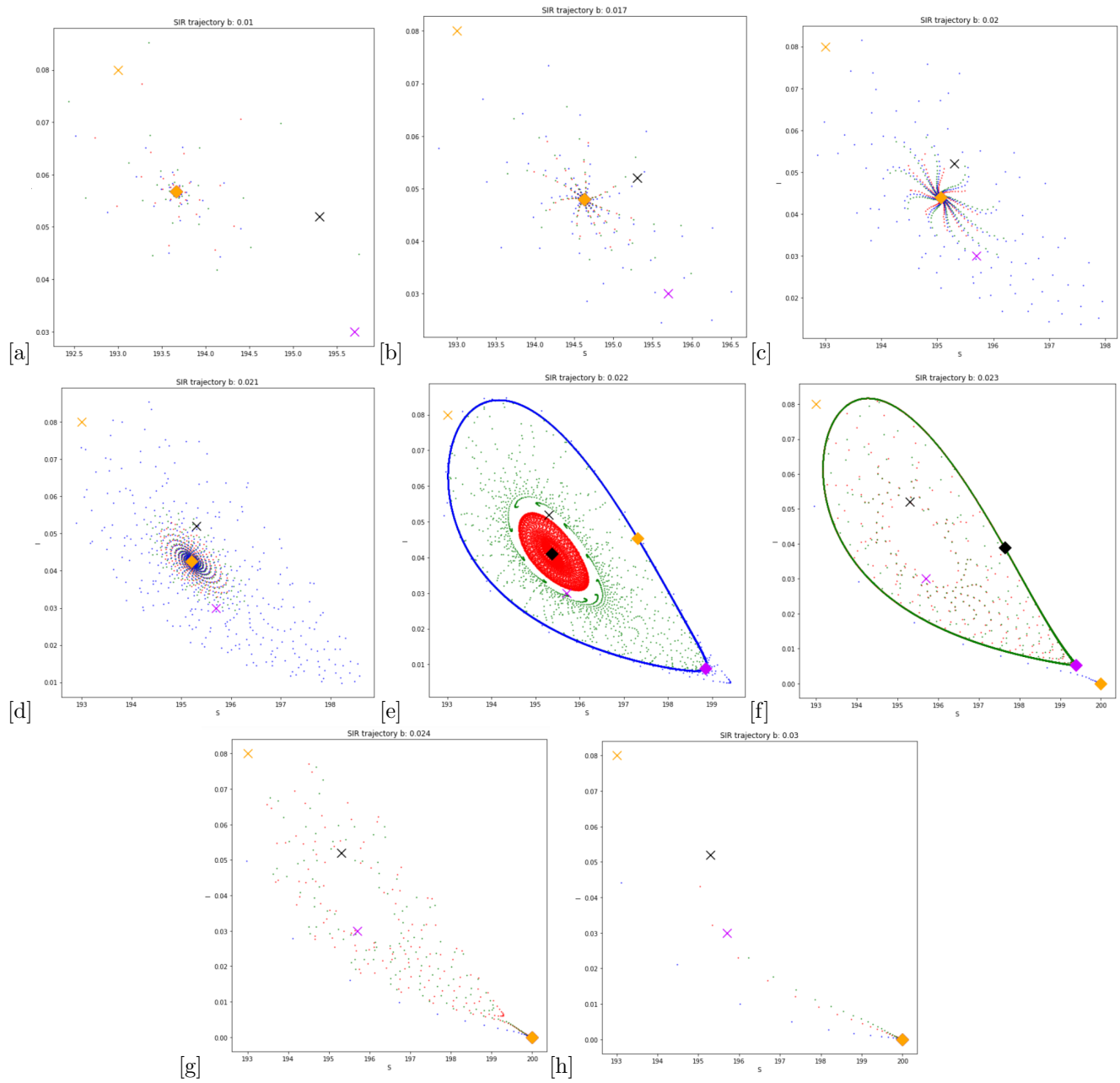


Figure 19: showing SIR behaviour change by varying b - time evolution from left to right and up to down

Task 5.4: The bifurcation shown in the previous paragraph appears at $b = 0.022$. Following the **Lemma 4.3** in [2] it is possible to demonstrate it is a *Hopf Bifurcation*, going locally from a steady state to a limit cycle as we previously have seen in **Section Task 3**. The calculations are shown in file `EX3/task5.ipynb` following the lemma conditions. Eventually, the normal form is the one mentioned in the same aforementioned section:

$$\begin{aligned}\dot{x}_1 &= \alpha x_1 - x_2 - x_1(x_1^2 + x_2^2) \\ \dot{x}_2 &= x_1 + \alpha x_2 - x_2(x_1^2 + x_2^2)\end{aligned}$$

Task 5.5: The *basic reproduction number* is defined in [4] as:

$$\mathbb{R}_0 = \frac{\beta}{d + \nu + \mu_1} . \quad (4)$$

Therefore, the variables used to compute it are β , d , ν and μ_1 , and they represent respectively the average number of contacts per unit of time with infectious people (β), the natural death rate per capita (d), the disease-induced death rate per capita (ν) and the maximum recovery rate (dependant of the number of available hospital beds) (μ_1). This reproduction rate indicates how easily the infection spreads among the population, so, since β is at the numerator, if it increases or decreases, \mathbb{R}_0 will behave the same way (increasing or decreasing). In [4], it is stated that, for classic epidemiological models with different demographics, the dynamics are almost completely characterized by \mathbb{R}_0 . The disease will be eliminated if $\mathbb{R}_0 < 1$, otherwise the disease will persist.

Task 5.6: The *disease free equilibrium* is defined in [4] as $E_0 \left(\frac{A}{d}, 0, 0 \right)$, and, for $\mathbb{R}_0 < 1$ it is an **attracting node**. This means that, for values of the S, I and R variables that are near to $\frac{A}{d}$, 0 and 0, the system will quickly reach E_0 and it will not move from that state as the simulation goes on. Practically speaking, the infection will not be able to spread through the population, since the steady state's component corresponding to the number of infected people (I) is going to stay 0.

If we get further from E_0 , though, things change: for $b > 0.022$ the system always gets to E_0 , while for $b < 0.022$ we found another attractive point in (193.666, 0.057, 5.71). This happens only when starting from a point that is not too near and not too far from E_0 . This behavior can be observed in Figure 20.

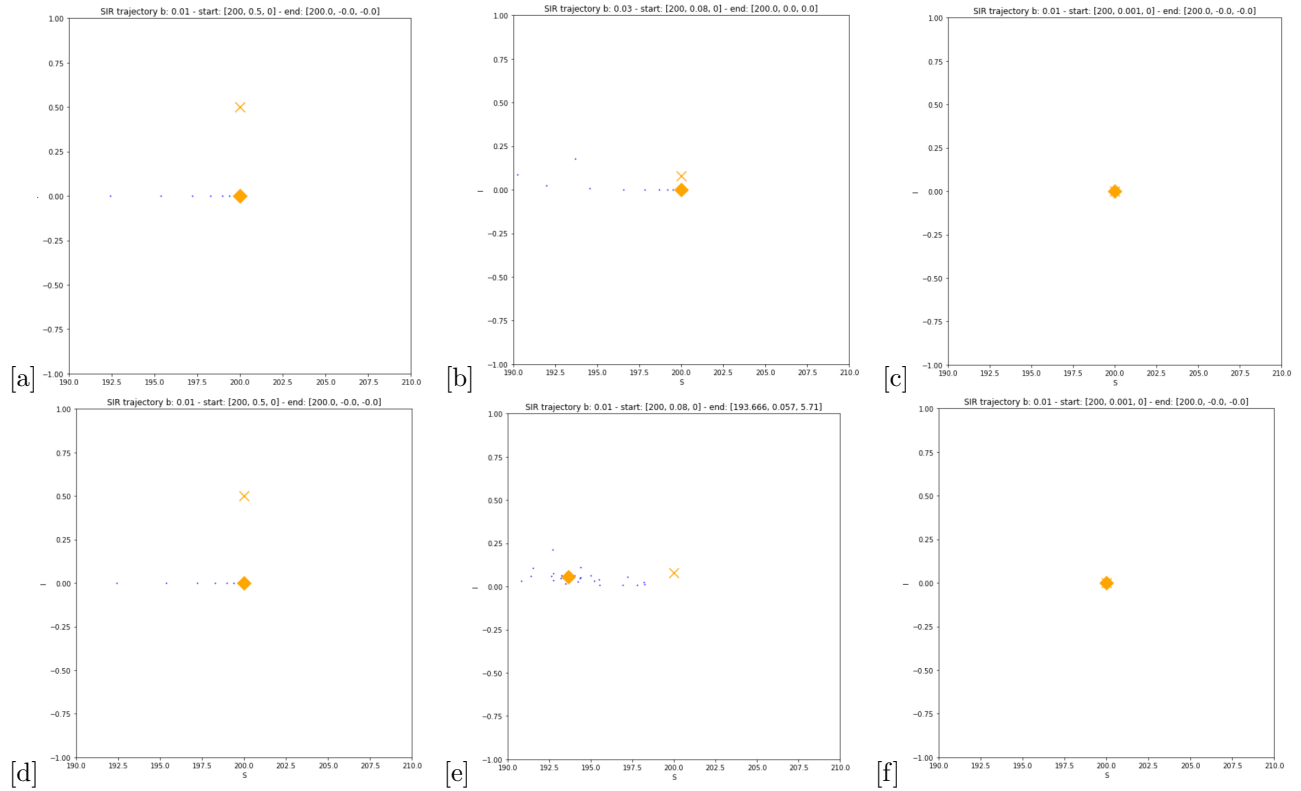


Figure 20: With $b < 0.022$ the system always reaches $E_0 = (\frac{A}{d} = 200, 0, 0)$ ((a), (b), (c)).

With $b > 0.022$: (d) starting far from E_0 , the system reaches E_0 anyway; (f) starting near E_0 , the system reaches E_0 anyway; (e) starting in between near and far from E_0 , the system reaches the point $[193.666, 0.057, 5.71]$.

It is worth noticing that this point gets reached from many different starting points, even if in the figure only one example is shown for the sake of simplicity.

References

- [1] Cobweb plot wikipedia. https://en.wikipedia.org/wiki/Cobweb_plot.
- [2] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory*. Springer New York, 2004.
- [3] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130 – 141, 1963.
- [4] Chunhua Shan and Huaiping Zhu. Bifurcations and complex dynamics of an sir model with the impact of the number of hospital beds. *Journal of Differential Equations*, 257(5):1662–1688, 2014.