Report for exercise 4 from group G

Tasks addressed:	6
Authors:	Zhaozhong Wang (03778350)
	Anastasiya Damaratskaya (03724932)
	Bassel Sharaf (03794576)
	Thanh Huan Hoang (03783022)
	Celil Burak Bakkal (03712329)
Last compiled:	2024–12–19

The work on tasks was divided in the following way:

Zhaozhong Wang (03778350)	Task 1	20%
	Task 2	20%
	Task 3	20%
	Task 4	20%
	Task 5	20%
Anastasiya Damaratskaya (03724932)	Task 1	20%
	Task 2	20%
	Task 3	20%
	Task 4	20%
	Task 5	20%
Bassel Sharaf (03794576)	Task 1	20%
	Task 2	20%
	Task 3	20%
	Task 4	20%
	Task 5	20%
Thanh Huan Hoang (03783022)	Task 1	20%
	Task 2	20%
	Task 3	20%
	Task 4	20%
	Task 5	20%
Celil Burak Bakkal (03712329)	Task 1	20%
	Task 2	20%
	Task 3	20%
	Task 4	20%
	Task 5	20%

Report on task 0: General Implementation

Before documenting the tasks, we will first explain how the `solve_system` method of the `DynamicalSystem` class was implemented. The `solve_system` method solves a dynamical system for a given initial state (`init_state`) and a set of time steps (`t_eval`). The method distinguishes between continuous and discrete systems, applying different solutions for each.

- **For continuous systems** (`self.discrete` is `False`):
  - The method uses `solve_ivp` from SciPy[1] to solve the system of differential equations numerically.
  - The `solve_ivp` function is passed the vector field (`fun`), the time span (`t_span`), the initial state (`y0`), and the time points (`t_eval`) where the solution is required.
  - It returns the solution (`trajectory.y`) at the specified time steps.
- **For discrete systems** (`self.discrete` is `True`):
  - The method iterates over the time steps in `t_eval`, updating the state by calling the vector field (`fun`) at each time step.
  - The updated state is appended to the `trajectory` list, and the final trajectory is returned.

In both cases, depending on the task, the vector field `fun` is specified by assigning a value to `self.fun` in the respective class of the task, which is then used to solve the system.

Report on task 1: Vector Fields, Orbits, and Visualization

In this task, we investigated the behavior of hyperbolic equilibria in two-dimensional dynamical systems. We focused on constructing phase portraits similar to Fig. 2.5 in [8], analyzing the different types of equilibria and their topological relationships.

Our analysis centered on linear systems of the form $\dot{x} = Ax$, where A is a 2×2 matrix and x represents the state vector in $\mathbb{R}^2$. We chose to examine three fundamental cases of hyperbolic equilibria by carefully selecting different matrix configurations. For our analysis, we used diagonal matrices of the form $A = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$ where we chose different values of a and b to achieve the desired eigenvalues for each case.

The results of our analysis are shown in Figure 1, which displays three phase portraits representing different types of hyperbolic equilibria. Each subplot shows the flow field in the region $[-2, 2] \times [-2, 2]$, with streamlines indicating the direction of flow and the equilibrium point marked at the origin. In the first subplot, we examined a stable node configuration with eigenvalues $\lambda_1 = -1$ and $\lambda_2 = -2$. The phase portrait clearly shows all trajectories converging to the origin, which is characteristic of stable nodes. The negative eigenvalues ensure that any initial state will eventually be attracted to the equilibrium point.

The middle subplot shows an unstable node with eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 2$. The phase portrait for this case demonstrates trajectories diverging from the origin in all directions. This behavior is typical of unstable nodes, where positive eigenvalues cause solutions to move away from the equilibrium point over time. The rightmost subplot displays a saddle point configuration with eigenvalues $\lambda_1 = -1$ and $\lambda_2 = 1$. The phase portrait reveals the distinctive saddle structure, where trajectories are attracted along one eigendirection and repelled along another. This creates the characteristic "X" pattern in the flow field.

Regarding topological equivalence between these systems, we found several important distinctions. The stable and unstable nodes cannot be topologically equivalent because their flows move in opposite directions - no homeomorphism could preserve both the topology and time orientation. Moreover, neither type of node can be topologically equivalent to the saddle point because saddles possess a fundamentally different phase space structure. The presence of both attractive and repulsive directions in a saddle creates qualitatively different behavior that cannot be transformed into that of a node while maintaining topological properties. Our visualizations effectively demonstrate these theoretical concepts, showing how the eigenvalues of the system matrix directly influence the qualitative behavior of solutions near equilibrium points. The phase portraits clearly illustrate the distinct characteristics of each type of hyperbolic equilibrium and help explain why certain systems cannot be topologically equivalent.

[1] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html`

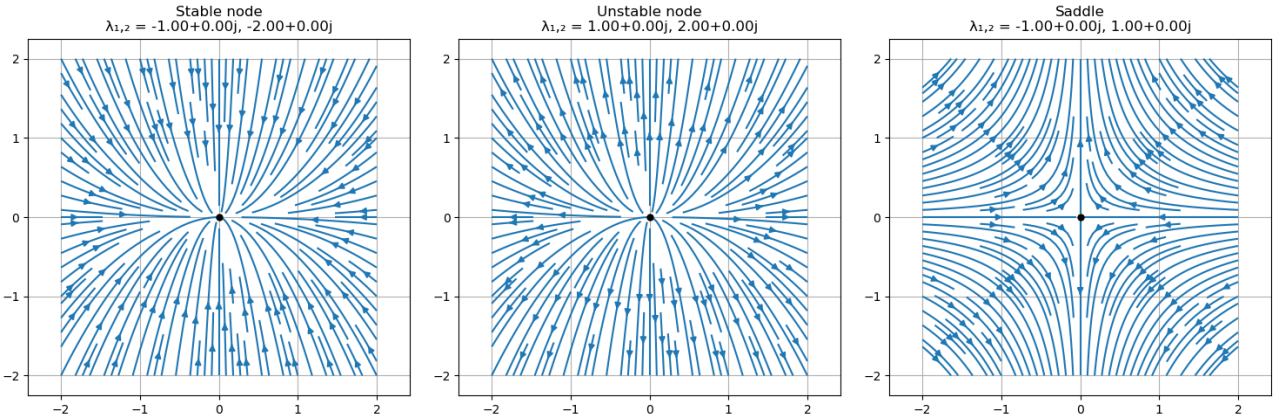

Figure 1: Phase portraits of three types of hyperbolic equilibria. Left: Stable node ($\lambda_{1,2} = -1, -2$). Middle: Unstable node ($\lambda_{1,2} = 1, 2$). Right: Saddle point ($\lambda_{1,2} = -1, 1$).

Report on task 2: Common Bifurcations in Nonlinear Systems

In this task, we investigated the bifurcation behavior of two related nonlinear dynamical systems. As graduate students in the Master's program, we were particularly interested in understanding how qualitative changes in system behavior emerge as parameters vary.

We first analyzed the system described by: $\dot{x} = \alpha - x^2$ where α is our bifurcation parameter. Through our analysis and numerical simulations, shown in Figure 2, we found that this system exhibits a saddle-node bifurcation at $\alpha = 0$. When $\alpha < 0$, no equilibrium points exist. At $\alpha = 0$, a single equilibrium appears, and for $\alpha > 0$, the system has two equilibrium points at $x = \pm\sqrt{\alpha}$. The upper branch (blue) represents stable equilibria, while the lower branch (red) represents unstable equilibria.

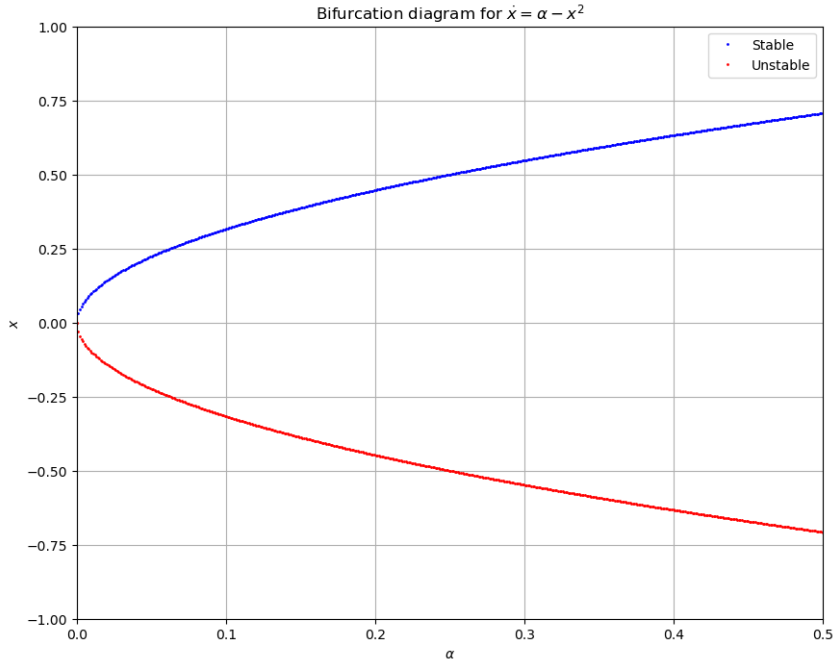

Figure 2: Bifurcation diagram for the system $\dot{x} = \alpha - x^2$ exhibiting a saddle-node bifurcation at $\alpha = 0$. Blue curves represent stable equilibria, red curves represent unstable equilibria.

The second system we investigated is given by: $\dot{x} = \alpha - 2x^2 - 3$ As shown in Figure 3, this system also exhibits a saddle-node bifurcation, but at $\alpha = 3$. We can see that for $\alpha < 3$, no equilibrium points exist. At the critical

value $\alpha = 3$, a single equilibrium point emerges, and for $\alpha > 3$, the system splits into two equilibrium points, following a similar pattern to the first system but shifted both horizontally and vertically.

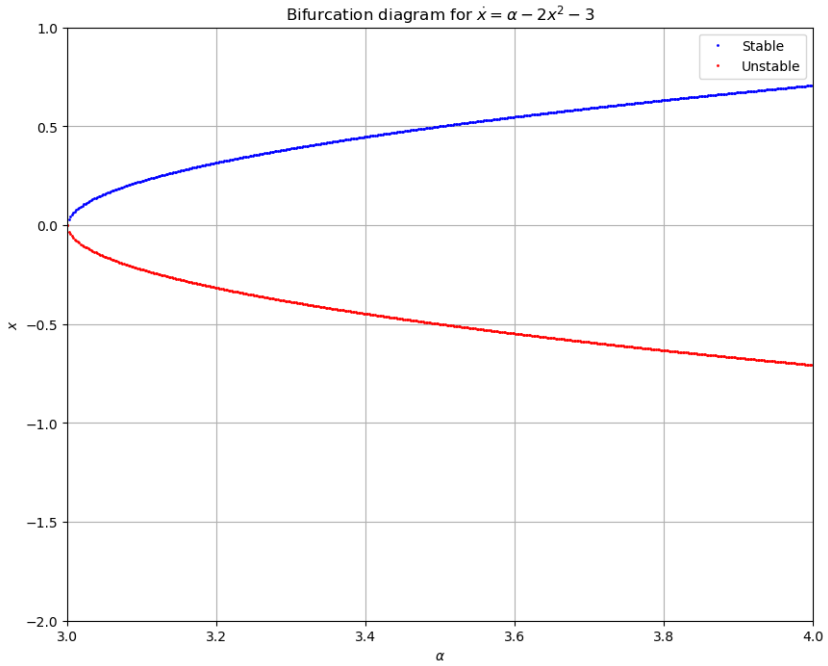

Figure 3: Bifurcation diagram for the system $\dot{x} = \alpha - 2x^2 - 3$ showing a saddle-node bifurcation at $\alpha = 3$. Blue curves represent stable equilibria, red curves represent unstable equilibria.

Regarding topological equivalence at $\alpha = 1$, we found that these systems are indeed topologically equivalent. Our reasoning is based on the observation that both systems possess qualitatively similar phase spaces: each has exactly two equilibrium points (one stable and one unstable), and the flow between these points follows the same pattern. We can construct a homeomorphism between their phase spaces that preserves the direction of flow.

Similarly, at $\alpha = -1$, the systems are also topologically equivalent, but for a different reason: neither system has any equilibrium points, and all trajectories move consistently in the same direction. The qualitative behavior of both systems in this regime is identical.

Both systems can be transformed into the standard saddle-node normal form $\dot{y} = \beta + y^2$ through appropriate coordinate and parameter transformations. This is evident in the similar parabolic structure seen in the bifurcation diagrams, where both systems demonstrate the characteristic "square root" behavior of equilibrium branches emerging at their respective bifurcation points.

Report on task 3: Bifurcations in Higher Dimensions

In this task, we focus on two types of bifurcations: the Andronov-Hopf bifurcation and the cusp bifurcation. The Andronov-Hopf bifurcation occurs in systems with a two-dimensional state space and one parameter, while the cusp bifurcation occurs in a one-dimensional state space with two parameters. We analyze these bifurcations through visualizations, numerical simulations, and a 3D plot for the cusp bifurcation. The code for this task can be found in the Jupyter Notebook `task3.ipynb` and in the `utils.py` file.

Andronov-Hopf Bifurcation

The Andronov-Hopf bifurcation has the vector field in normal form as follows:

$$\begin{aligned} \dot{x}_1 &= \alpha x_1 - x_2 - x_1(x_1^2 + x_2^2), \\ \dot{x}_2 &= x_1 + \alpha x_2 - x_2(x_1^2 + x_2^2). \end{aligned} \tag{1}$$

The system will exhibit different types of behavior depending on the value of α. Figure 4 (created using the function `plot_phase_portrait()` in the `utils.py` file, where users can specify the system of ordinary differential equations (ODEs) as a list of strings and the value of α) visualizes the system's bifurcation through the phase diagrams at three representative values of α: negative ($\alpha = -1.0$), zero ($\alpha = 0.0$), and positive ($\alpha = 1.0$):

- For $\alpha = -1.0$: The system has a stable fixed point at the origin, with trajectories spiraling in towards the origin, indicating a stable attracting focus.
- For $\alpha = 0.0$: The origin becomes a neutral fixed point (a center), with trajectories forming closed orbits around it, exhibiting neutral oscillations. The system is in weakly attractive focus.
- For $\alpha = 1.0$: The origin is now an unstable fixed point, and the system exhibits a stable limit cycle, with trajectories moving toward and oscillating around this cycle.

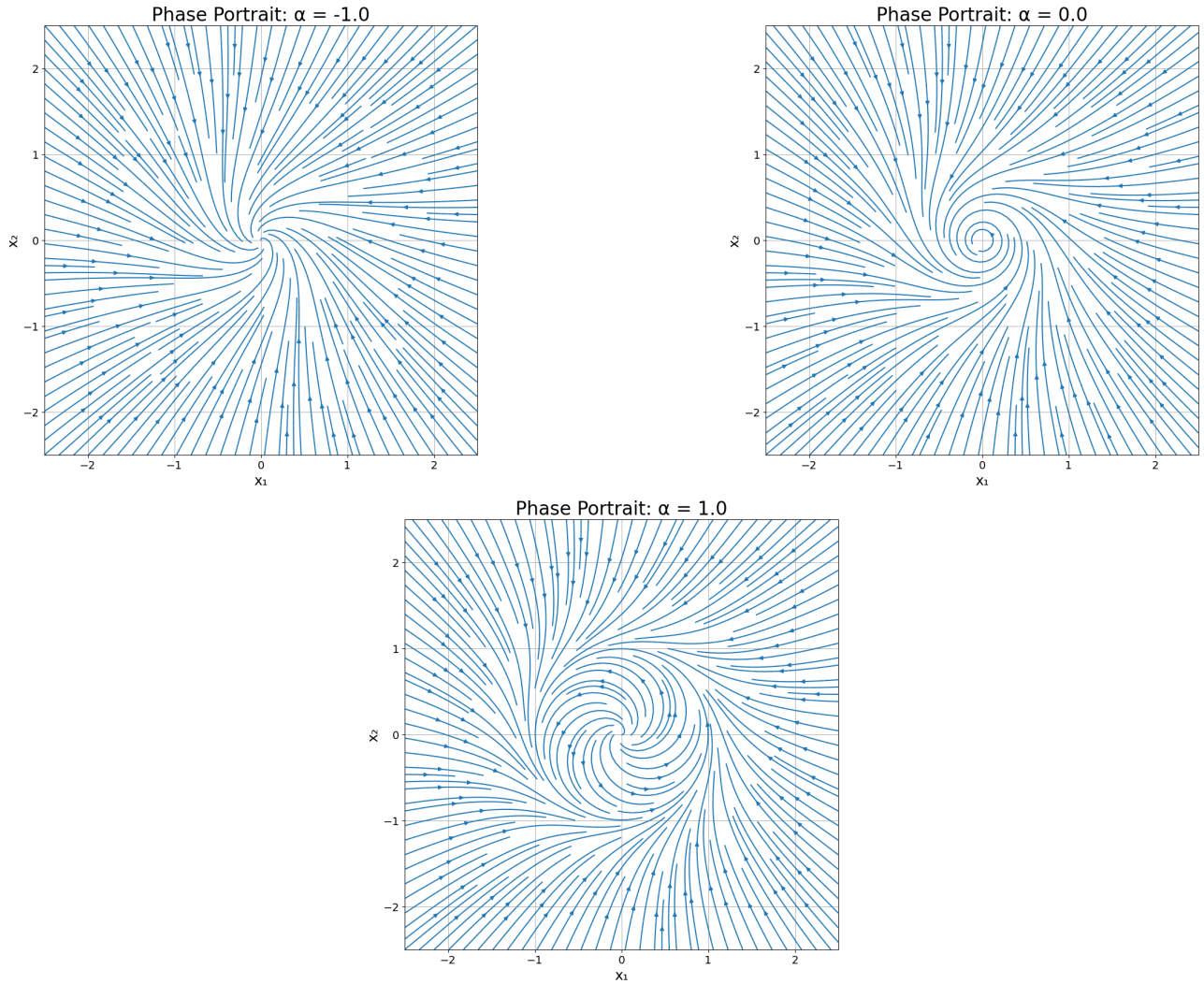

Figure 4: Phase diagrams of the Andronov-Hopf bifurcation at different α values

As we move from negative to zero to positive values of α, we can see a clear shift from inward spiraling towards a fixed point, to neutral circular motion around the origin, and finally to stable, periodic oscillations around a limit cycle. This progression highlights the dynamics of the Andronov-Hopf bifurcation, where the system's behavior changes qualitatively as the bifurcation parameter α crosses critical values.

To visualize the orbits of the system forward in time when $\alpha = 1.0$, Euler's method is used with a very small time step for computation. The Euler update step approximates the next position in the system by adding

the rate of change (calculated from the ODEs) multiplied by the time step to the current position. It uses the formula:

$$x(t + \Delta t) = x(t) + \Delta t \cdot [u, v]$$

where $[u, v]$ are the derivatives at the current position. This method iteratively updates the state in small steps to simulate the system's evolution over time. The same function, `plot_phase_portrait()`, is used for this purpose. Originally, for the task above, since we do not need to visualize the orbit, the parameter `plot_orbit` is set to `False` by default, thus preventing the orbit from being plotted on the same graph. One can also specify other parameters, such as the starting point `start_pos`, the time step `time_step`, and the number of iterations `num_iterations`. By default, the time step and the number of iterations are set to 10^{-3} and 10,000, respectively.

Figure 5 shows how the orbits of the system look with two starting points (2, 0) and (0.5, 0). In both cases, the orbits will move toward the limit cycle, and once near it, they will follow the cycle, oscillating in a periodic motion. The key feature is that the origin is unstable, and any starting point that is not the origin (in this case, (0, 0)) will result in the trajectory converging to the limit cycle.

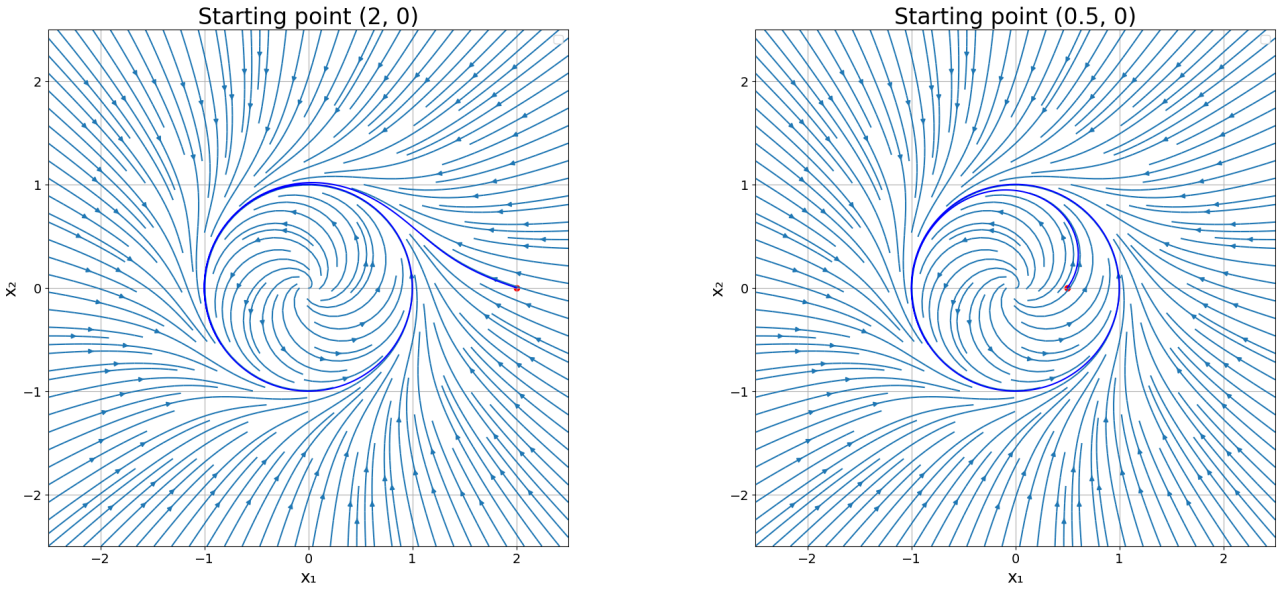

Figure 5: Orbits of the systems with starting points (2, 0) and (0.5, 0). The red dot indicates the starting point, whereas the dark blue curve represents the orbit.

Cusp Bifurcation

The cusp bifurcation, which occurs already in a one-dimensional state space but with two parameters, has the vector field in normal form as follows:

$$\dot{x} = \alpha_1 + \alpha_2 x - x^3 \tag{2}$$

The function used to visualize the bifurcation surface of the cusp bifurcation in 3D is `plot_cusp_bifurcation()` from `utils.py`. As the exercise suggests, points (x, α_2) are sampled uniformly, and the surface is plotted as a function α_1 of (x, α_2). The `plot_cusp_bifurcation()` function allows users to specify the range for x (`x_range`), the range for α_2 (`alpha2_range`), and the number of uniformly sampled points (`num_samples`). By default, `num_samples` is set to 2000.

Figure 6 shows the bifurcation surface of the cusp bifurcation in 3D. The ranges for x, α_2, and the number of sampled points for these ranges are $(-2, 2)$, $(0, 1.5)$, and 2000, respectively. Three different angles of the graph are displayed, as viewing from a single angle may make it difficult to observe certain parts of it.

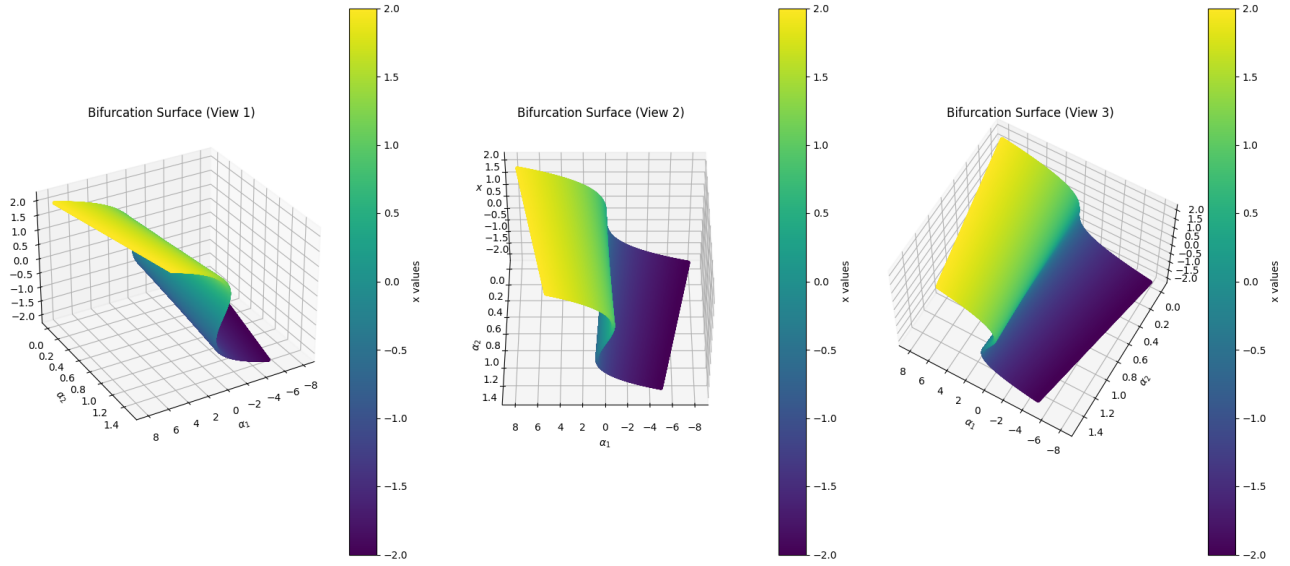

Figure 6: Bifurcation surface of the cusp bifurcation in 3D: Calculate α_1 based on x and α_1 values

The system is called a cusp bifurcation because, when the variable x is projected onto the parameter space (α_1, α_2), the bifurcation surface folds sharply back on itself, forming a pointed, cusp-like geometry. This cusp not only defines the shape but also separates regions in the parameter space where the system transitions between having one steady state and multiple steady states, highlighting the qualitative changes in the system's dynamics.

Report on task 4: Chaotic Dynamics

Note: All of the plots demonstrated in this task can be found and recreated in the file task4.ipynb with Jupyter Notebook.

Logistic Map

In the first part of task 4, we perform bifurcation analyses on the discrete dynamical system "logistic map", which is defined by the following equation:

$$x_{n+1} = r x_n (1 - x_n),$$

, with $r \in (0, 4]$, $x \in [0, 1]$ and $n \in N$. It is a discrete map.

Logistic Map with r from 0 to 2

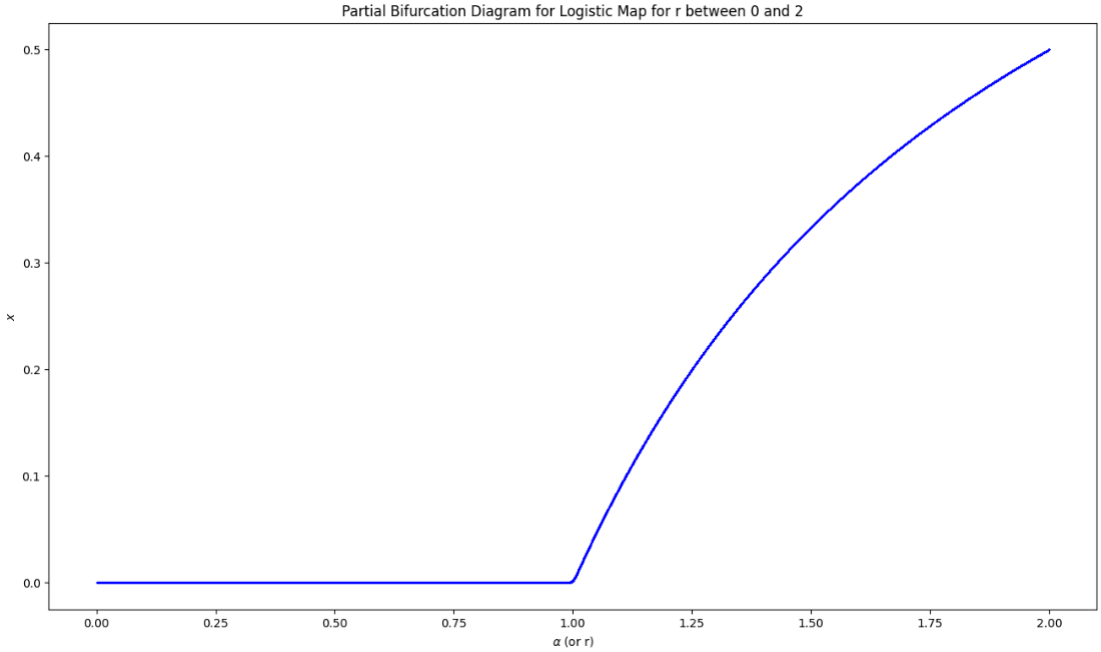

Figure 7: Logistic map partial bifurcation diagram for r between 0 and 2

In Figure 7, the r parameter of logistic map is varied from 0 to 2 on horizontal axis, where x is between 0 and 1 on vertical axis (in this r range, x does not exceed 0.5). For every r a total of 600 time steps are done. but the first 400 ones are skipped and only the last 200 steps are plotted in order skip the initial transient behavior and let it converge first [12, 13]. Here initial state is $x_0 = 0.5$. The idea of skipping transient steps and choice of initial state is inspired from [12, 13]. The plotting function used for Figure 7 is named **plot_bifurcation_diagram** and it can be found in the file **utils.py**. The same function is also used for Figure 8 and Figure 9.

As defined in Definition 2 of exercise sheet, we can find the fixed points by finding the values of x which satisfies the equation: f(x) = x, which is $x_{n+1} = r \cdot x_n \cdot (1 - x_n) = x_n$ for our case [13, 5]. From this we can infer that logistic map has 2 fixed points at $x^* \in \{0, 1 - \frac{1}{r}\}$, as they satisfy this equation [12, 13, 5].

When we plug these fixed point values in the logistic map equation it can be seen how they satisfy the equation:
For $x_n = 0$, we get:

$$x_{n+1} = r \cdot 0 \cdot (1 - 0) = 0 = x_n$$

For $x_n = 1 - \frac{1}{r}$, we get:

$$x_{n+1} = r \cdot \left(1 - \frac{1}{r}\right) \cdot \left(1 - \left(1 - \frac{1}{r}\right)\right) = (r - 1) \cdot \frac{1}{r} = 1 - \frac{1}{r} = x_n$$

It can also be seen by solving the following equation:

$$x - f(x) = x(1 - r(1 - x)) = x(1 - r + rx) = rx(x - \left(1 - \frac{1}{r}\right)) = 0[12]$$

We can check the stability of these fixed points by checking the absolute value of the derivative, i.e. $|f'(x)| = |r(1 - 2x)| = |r - 2rx|$, where $f(x) = rx(1 - x)$. [13, 9, 6]. For $|f'(x)| < 1$ it is stable, for $|f'(x)| > 1$ it is unstable [13, 9, 6].

For $r \in [0, 1)$ logistic map converges to the fixed point $x^* = 0$, which can be seen in Figure 7. [7, 13]. Furthermore, when 0 is plugged into absolute value of first derivative of logistic map, it results as follows: $|f'(0)| = |r - 2r \cdot 0| = |r|$. This means the fixed point $x^* = 0$ is stable and a steady state for $|r| < 1$, which

includes the interval $r \in [0, 1)$. [6]

For r between 1 and 2, it converges to the fixed point $x^* = 1 - \frac{1}{r}$ [6, 13]. For example, for r = 2, it converges to the state $x = 1 - \frac{1}{2} = 0.5$, which can be seen in Figure 7. When we plug this fixed point into the absolute value of derivative, we get $|f'(1 - \frac{1}{r})| = |r - 2r \cdot \left(1 - \frac{1}{r}\right)| = |r - 2r + 2| = |2 - r|$ [6]. Therefore, these fixed points are stable and steady state for $1 < r < 3$, as this interval satisfies the condition $|2 - r| < 1$ [6].

As can be seen in Figure 7, a bifurcation happens at r = 1, which is a transcritical bifurcation [7]. This is also shown with a red vertical line in Figure 9. In this type of bifurcation two fixed points collide and exchange stability, i.e. both before and after the bifurcation point there is one stable and one unstable fixed point [9, 4, 7]. In our case of logistic map, after r = 1 the fixed points $x_1^* = 0$ and $x_2^* = 1 - \frac{1}{r}$ exchange stability, i.e. $x_2^* = 1 - \frac{1}{r}$ becomes stable and $x_1^* = 0$ becomes unstable. [7]. We can also sense this bifurcation visually from Figure 7, as the intervals before and after r = 1 have totally different shapes.

Logistic Map with r from 2 to 4

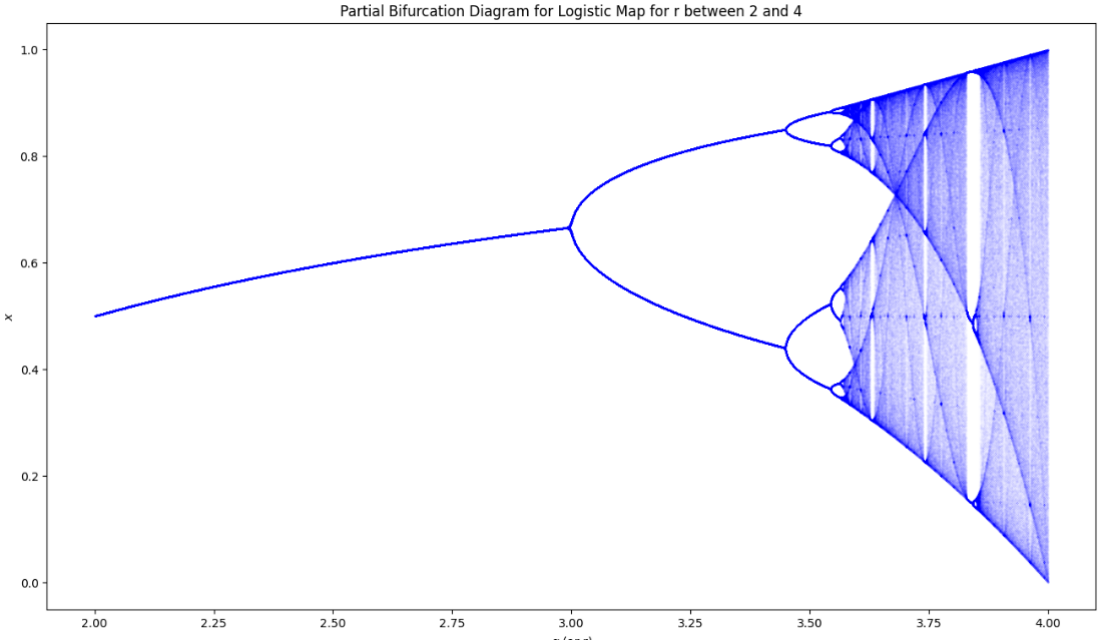

Figure 8: Logistic map partial bifurcation diagram for r between 2 and 4

The partial bifurcation diagram of logistic map for values of r between 2 to 4 is demonstrated in Figure 8.

For r between 2 and 3 again it approaches to the steady state $1 - \frac{1}{r}$ just like when it was between 1 and 2 [6, 13] and has a similar slightly curved linear shape. It also matches our previous explanation where we showed that fixed points with value $1 - \frac{1}{r}$ are stable for $1 < r < 3$ [6].

Another bifurcation occurs at r = 3, which is a period-doubling bifurcation [7, 1]. Here, the shape changes from a single slightly curved line to a shape of two slightly curved lines. After r = 3, the fixed point $x^* = 1 - \frac{1}{r}$ becomes unstable [7]. It can be seen from the absolute value of the first derivative previously mentioned, i.e. $|f'(1 - \frac{1}{r})| = |r - 2r \cdot \left(1 - \frac{1}{r}\right)| = |r - 2r + 2| = |2 - r|$ [7, 6] . When we plug r = 3 into this result, we get $|2 - r| = |2 - 3| = |-1| \not< 1$, which means it is no more stable [7].

After r = 3, the region of sequence of period-doubling bifurcations begin [12, 13]. As r grows larger, oscillations between 2 values, then 4 values, then 8, 16, 32, ... appear, i.e. a period-doubling cascade, where a sequence of period-doubling bifurcations occur successively. [12, 13]. In Figure 8 we see that the shapes of these bifurcations resemble each other and get closer and smaller the more bifurcated they get and then stop at some point and

the behaviour gets very chaotic.

For r between 3 and $(1+\sqrt{6}) \approx 3.44949 \approx 3.45$, we oscillate between two values, i.e. a period-2-cycle [12, 7, 13]. These two values are expressed as follows: $x_{\pm} = \frac{1}{2r}\left(r+1 \pm \sqrt{(r-3)(r+1)}\right)$ [7, 13]. These two values are fixed points of the two successive iterations of logistic map but not directly of the logistic map itself, i.e. they satisfy the equation f(f(x)) = x, where $f(x) = r \cdot x \cdot (1-x)$ [7]. It is also to note that we find the the values further period-n-cycles oscillate between the same way, which is finding the fixed points of n successive iterations of $f(x) = r \cdot x \cdot (1-x)$ [7].

With r between 3.44949 and 3.54409, another period-doubling bifurcation happens, and system oscillates between 4 values, i.e. a period-4-cycle [12, 7, 13]. As mentioned, we find these 4 values by finding the fixed points of 4 successive iterations of $f(x)$, i.e. $f(f(f(f(x)))) = x$ [7].

For r between 3.54409 and 3.56441 another period-doubling bifurcation occurs and period is doubled again [12, 13]. Now system oscillates between 8 values, i.e. a period-8-cycle [12, 13]. This region of successive period-doubling bifurcations with period numbers 2^n go on like this forever and accumulate at $r \approx 3.56995$ [12, 13]. At some point they get so small that it is no more possible to see them from a plot like Figure 8.

The value $r \approx 3.56995$ is approximately the point where the period-doubling-cascade ends and chaotic region begins [7, 12, 13], which looks like a dense set of points. This chaotic region is sensitive to initial conditions [11, 12, 13]. It can show significantly different results depending on the initial condition and is unpredictable [11, 13].

Even in chaotic region (after $r \approx 3.56995$), there are still some intervals where non-chaotic periodic behavior can be observed [12, 13]. For example, at $r = (1+\sqrt{8}) \approx 3.8284 \approx 3.83$ a period-3-cycle can be observed [12, 13].

Full Bifurcation Diagram with r between 0 and 4

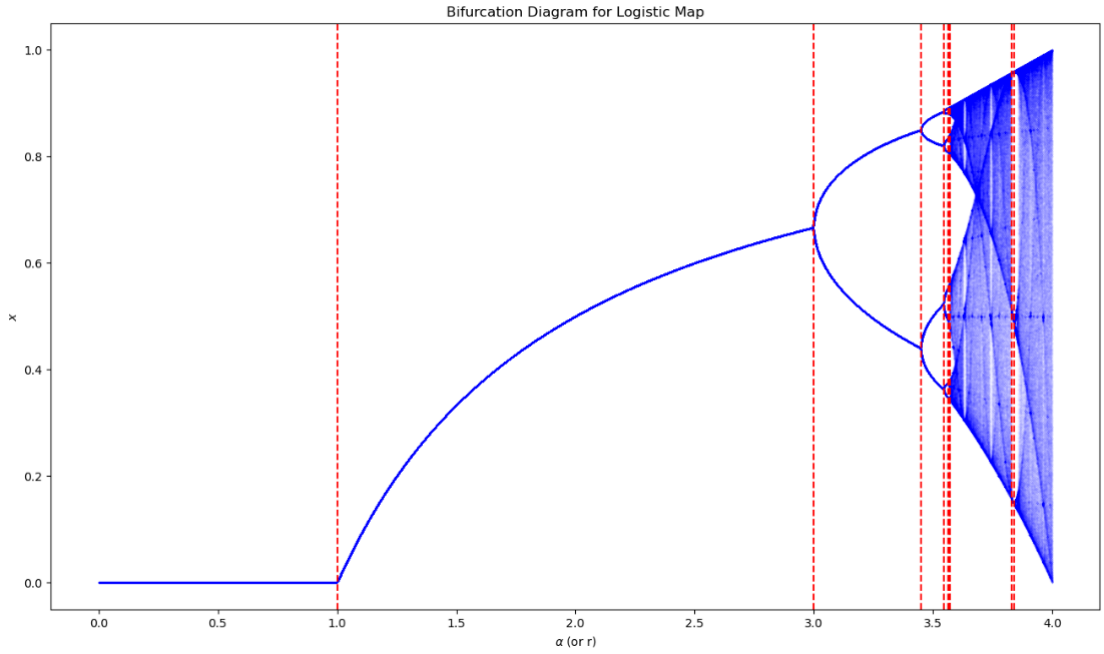

Figure 9: Full Logistic map partial bifurcation diagram for r between 0 and 4

In this plot Figure 9, full bifurcation diagram of logistic map with r between 0 and 4 visualized. In this diagram, the positions of steady states and limit cycles are indicated with vertical red dashed lines. More clearly, these vertical red lines are on the bifurcation points and the steady states and limit cycles, which are explained in

the previous sections, lie between these red lines. So these lines show us the start and end positions of these steady states and limit cycles. The vertical lines on bifurcation points logic is inspired from [12]

Lorenz System

In part 2 of task 4, we analyze the Lorenz system. It is a system in three-dimensional space that forms a strange attractor, a fractal set on which the dynamics are chaotic. It has 3 ordinary differential equations, which are also called Lorenz equations and defined as follows:

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

, where x, y, z are state variables, and these equations describe the rate of change of these variables with respect to time [2].

Lorenz Visualization with $x_0 = (10, 10, 10)$

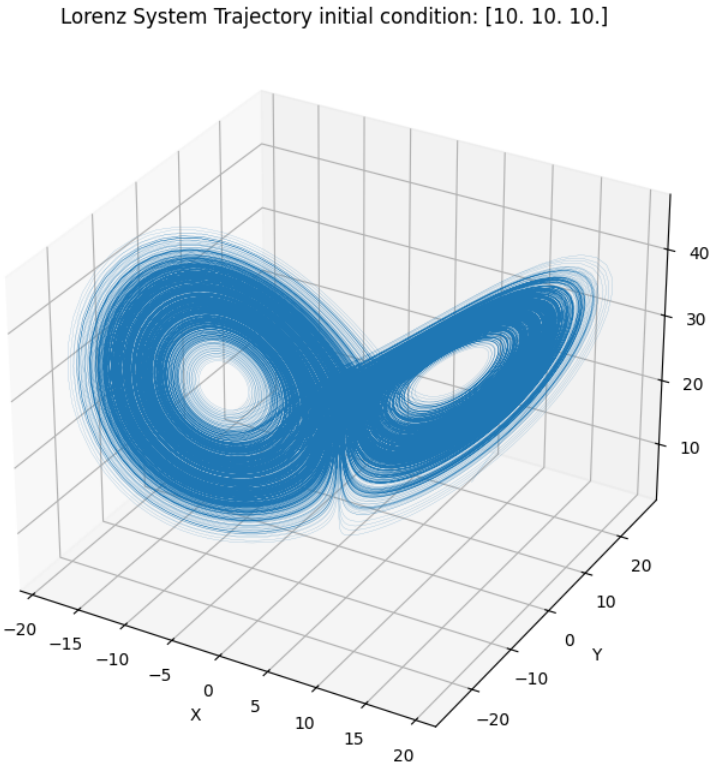

Figure 10: Lorenz system with $x_0 = (10, 10, 10)$

In Figure 10, a single trajectory of the Lorenz system is visualized until the end time of $T_{\text{end}} = 1000$ is reached. It is also important to note that T_{end} is not the iteration count, but the simulated time. We made 100 times of T_{end} iterations for the plot, i.e. 100000 iterations, to make it smooth. The simulation starts at $x_0 = (10, 10, 10)$ and uses the parameters $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$. The plotting function used for Figure 10 is named **plot_trajectory_lorenz_system** and it can be found in the file **utils.py**. The same function is also used for Figure 11 and Figure 13.

In Figure 10 we see the Lorenz attractor, which is a strange attractor and has fractal structure. It looks like a butterfly with two loops resembling its wings placed next to each other. There are circular spaces in the middle of both of these two wings. The trajectory looks like it rotates around these circular spaces. The two wings look alike and look symmetrical.

Lorenz Visualization with $x_0 = (10 + 10^{-8}, 10, 10)$

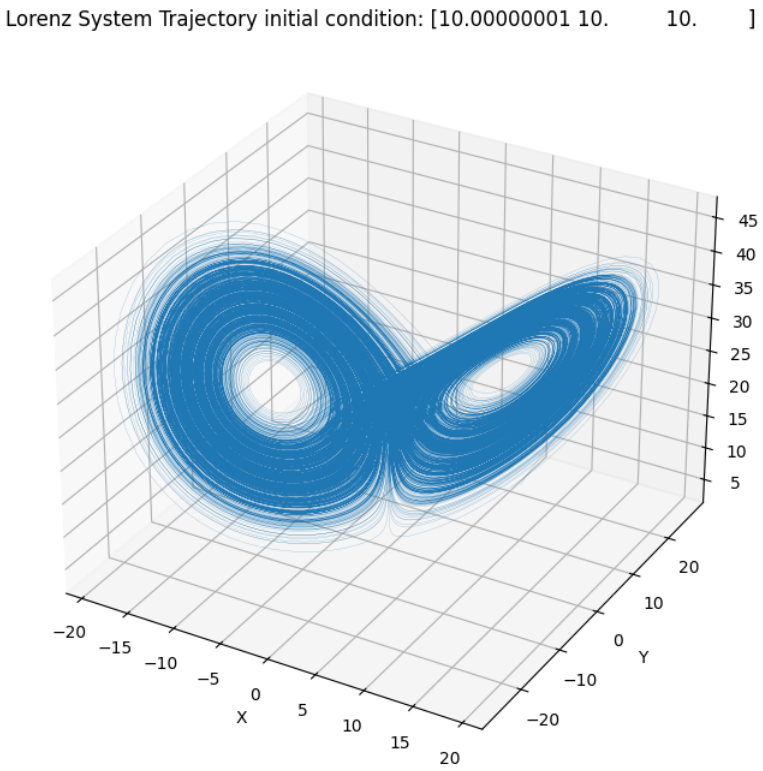

Figure 11: Lorenz system with $x_0 = (10 + 10^{-8}, 10, 10)$

In Figure 11, just like the Figure 10 in previous section, a single trajectory of the Lorenz system is visualized with the same parameters. Only difference between the previous plot in Figure 10 and current plot in Figure 11 is that this time the initial condition $x_0 = (10 + 10^{-8}, 10, 10)$ is used, which has a small perturbation, i.e. 10^{-8}, added in the first variable. Although the differences between these trajectories are not directly very clear in these two plots, the difference plot Figure 12 illustrated in the next section shows us how much they actually diverge.

Difference between the Two Trajectories Over Time

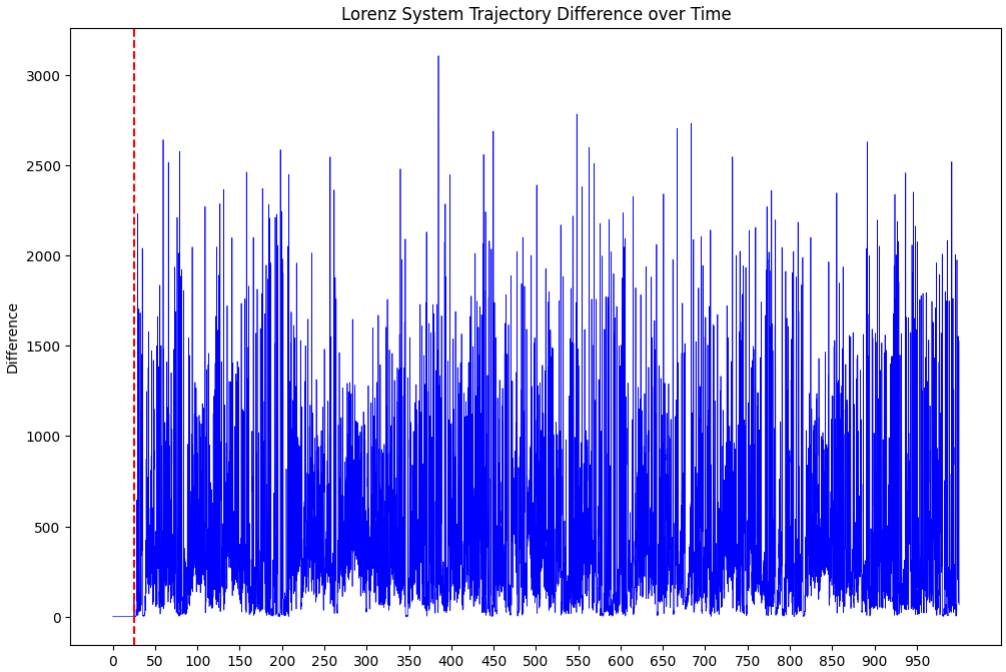

Figure 12: Difference between the Trajectories of Lorenz systems with Original and Perturbed Conditions

In Figure 12, the difference between the two trajectories of two Lorenz systems over time is demonstrated. The plotting function used for Figure 12 is named **plot_difference_lorenz_system** and it can be found in the file **utils.py**. The same function is also used for Figure 14. Here, first Lorenz system has the initial condition $x_0 = (10, 10, 10)$ and the second one has $x_0 = (10 + +10^{-8}, 10, 10)$. Horizontal axis represents time and vertical axis represents the difference which is defined as follows:

$$\|x(t) - \hat{x}(t)\|^2$$

, where $x(t), \hat{x}(t) \in \mathbb{R}^3$ are the two trajectories for the two initial conditions.

As it can be seen from the Figure 12, approximately at time t = 25 the difference between the points on the trajectory gets larger than 1. The position of it is indicated with a vertical dashed red line. Figure 12 shows us the chaotic nature of the system and it can be inferred that even a very small perturbation such as 10^{-8} can lead to great differences between the two trajectories later. The Figure 12 demonstrates that the two trajectories are nearly identical in the beginning such that the differences between them is nearly zero until time t = 25 and after that time point they get exponentially larger. From this it can be inferred that it is highly sensitive to initial conditions.

Trajectories and Differences with changed parameter $\rho = 0.5$

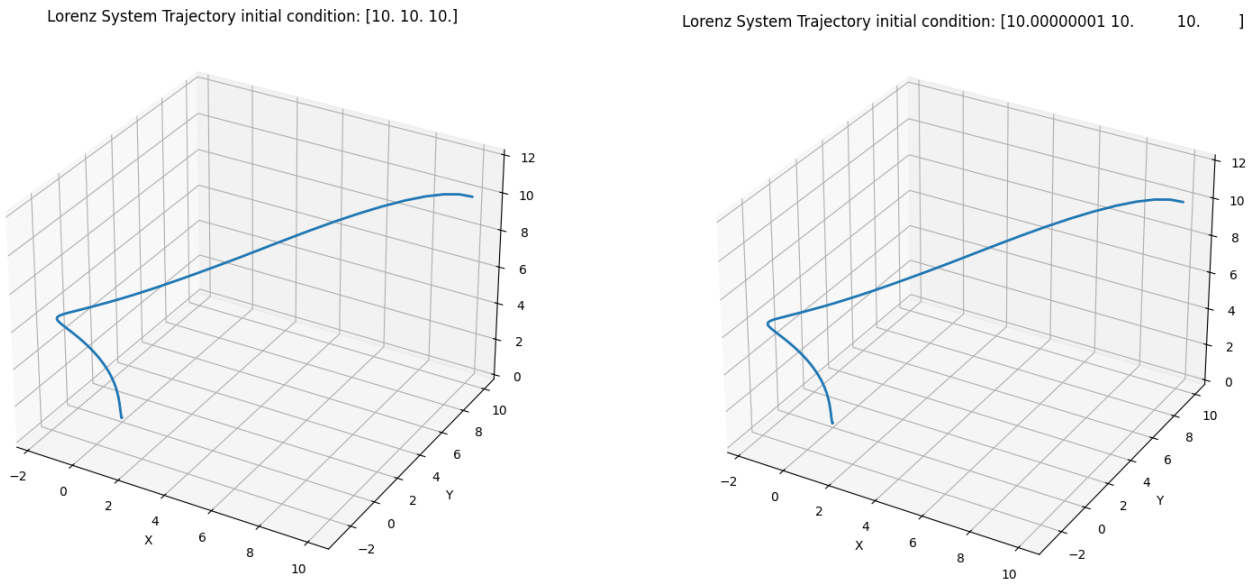

Figure 13: Trajectories of Lorenz systems with different initial conditions and $\rho = 0.5$

In Figure 13 the trajectories of two Lorenz systems with different initial conditions, just like in the previous sections, is illustrated. The difference now is the changed parameter $\rho = 0.5$, which was equal to 28 in the previous sections. Just like in the previous sections, left one has the initial condition $x_0 = (10, 10, 10)$ and the right one uses $x_0 = (10 + 10^{-8}, 10, 10)$.

In these plots it can be seen that after the parameter ρ is changed from 28 to 0.5, the trajectories no longer show a chaotic butterfly shape. Instead, they have simpler shapes that looks a like 3D curved line and look very similar to each other.

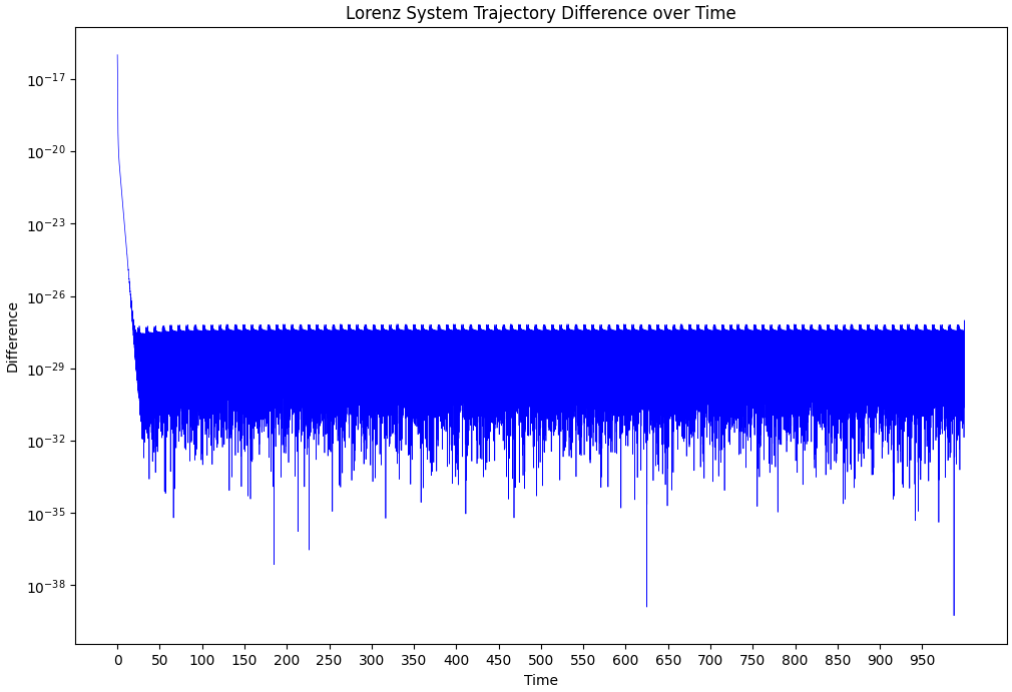

Figure 14: Difference between the Trajectories of Lorenz systems with Original and Perturbed Conditions

In Figure 14 the differences between the two trajectories are demonstrated. Logarithmic scale is applied to the vertical axis of differences, because otherwise they are impossible to see in the plot due to the very small size of differences. The same formula in Figure 12 is used for computing the differences. The differences are approximately between 10^{-38} and 10^{-17}, which are extremely small contrary to the differences in Figure 12, in which they become exponentially larger. It shows us that with reduced $\rho = 0.5$, the system is not highly sensitive to initial conditions like the previous system with $\rho = 28$.

With these results, it can be understood that there is a bifurcation (or multiple ones) between the value 0.5 and 28. Because as it is expressed in Definition 3 of exercise sheet, these two systems with different ρ parameters show topologically non-equivalent behaviour.

Report on task 5: Bifurcations in Crowd Dynamics

SIR Model

In this task, we analyze and describe the SIR model provided by Shan et al. [10]. The model is based on the following differential equations, where **S** is the number of susceptible, **I** the number of infectious and **R** the number of recovered individuals at time t:

$$\frac{dS}{dt} = A - \delta S - \frac{\beta SI}{S+I+R},$$
$$\frac{dI}{dt} = -(\delta+\nu)I - \mu(b,I)I + \frac{\beta SI}{S+I+R},$$
$$\frac{dR}{dt} = \mu(b,I)I - \delta R.$$

where the recovery rate $\mu(b, I)$ of infectious individuals is defined as:

$$\mu(b,I) = \mu_0 + (\mu_1 - \mu_0)\frac{b}{b+I}$$

with μ_0 and μ_1 the minimum and maximum recovery rates based on the number of available beds. The standard incidence rate is:

$$\frac{\beta SI}{S+I+R}.$$

The constant $\beta > 0$ is the average number of adequate contacts per unit time with infectious persons **I**, $\delta > 0$ is per capital natural death rate, A is the recruitment rate, and b is the number of beds per 10,000 persons. In the classic SIR model, we use the linear recovery rate $\mu I(t)$, and typically there is no bistability and periodicity. The reproduction number R_0 characterizes the dynamics in a way that the disease will die out over time only if $R_0 < 1$; otherwise, if $R_0 > 1$, the disease will spread in the population because each infectious individual causes more than one subordinate infection [10]:

$$R_0 = \frac{\beta}{d+\nu+\mu_1}.$$

In this equation, β represents the average number of adequate contacts, and raising β means more contacts and a higher potential for disease transmission. With increasing β, R_0 also increases, leading to a rapidly spreading disease and more infectious people over time. If, on the other hand, β decreases, meaning fewer contacts occur, R_0 will decrease too, indicating that the disease spreads more slowly and the number of infectious persons declines over time. Other variables used in the equation are d, standing for natural death rate, ν, representing the disease-induced death rate of infectious individuals, and μ_1, denoting the maximum recovery rate. Each of these variables, by increasing, reduces R_0, as it means either that the individual died and is not in the model anymore, died fast and could not spread the illness that often, or recovered fast so that the spreading cannot occur anymore. In summary, R_0 provides a significant insight into how the current situation should be treated, whether the people should be isolated (decreases β) or better treatment should be developed (increases μ_1). Referring to E_0, the disease-free equilibrium, defined as:

$$E_0 = (\frac{A}{d}, 0, 0)$$

being an attractive node, it means that E_0 is a stable equilibrium node and any change in **S**, **I** or **R** close to E_0 will eventually be reversed as demonstrated in the first plot of Figure 15. The model will return to E_0 in the course of time, implying that the disease will naturally be eliminated since the susceptible population stabilizes at **S**$= A/d$ (balance between birth and death rate); **R** remains at zero as no one has contracted the disease for a long period of time; and **I** converges to zero as the disease cannot preserve itself, **I** $\to 0$. Likewise, the recovery rate is demonstrated in the second plot of Figure 15, which remains constant, approaching a steady value over time, similar to **S**. It supports the fact that the system reaches a disease-free equilibrium over time, meaning the recovery dynamics remain constant.

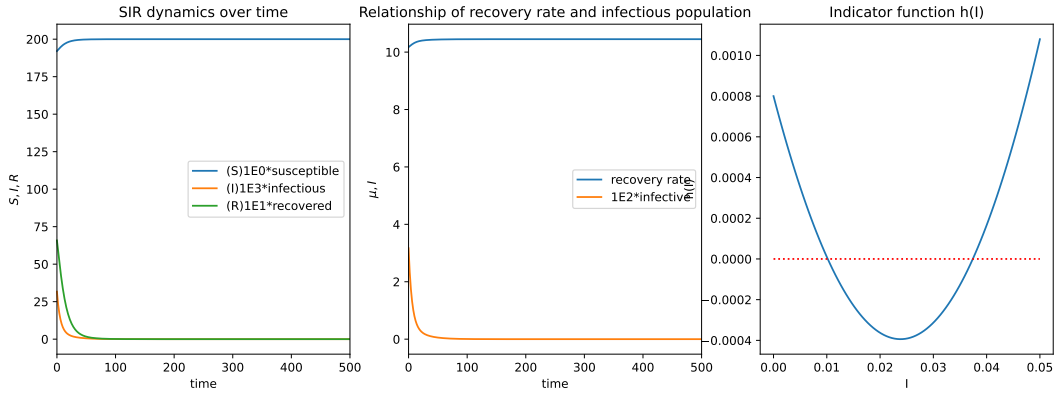

Figure 15: Plot 1: a disease-free equilibrium, indicating that the system approaches a stable equilibrium where infections die out; Plot 2: the relationship between the recovery rate μ and **I**; Plot 3: indicator function of Hopf bifurcation. The parameters are set: $A = 20$, $d = 0.1$, $\beta = 11.5$, $\nu = 1$, $\mu_0 = 10$, $\mu_0 = 10.45$, $b = 0.2$.

SIR Model Implementation

To implement the SIR model, we used the provided `sir_model.py` and `task5.ipynb` files. In `sir_model.py`, we implemented such methods as calculation of $\mu(b, I)$, computation of R_0, the saddle-node's codimension in case $R_0 = 1$, determining the indicator function for bifurcation h and returning the results of the differential equations of the SIR model. In `task5.ipynb`, we set the parameters described above, analyze the bifurcation after solving the model, and visualize the results in 2D and 3D plots.

Bifurcation

The third plot of Figure 15 demonstrates the parabolic indicator function h in relation to the infectious population **I**. The h is defined as:

$$h(I) = c_3 I^3 + c_2 I^2 + c_1 I + c_0$$

with

$$c_3 = d(\beta - \nu), \quad c_2 = (\mu_1 - \mu_0)b\nu + 2bd(\beta - \nu) + dA$$
$$c_0 = b^2 dA, \quad c_1 = b\left((\mu_0 - \mu_1 + 2d)A + (\beta - \nu)bd\right).$$

The red dotted line indicates a threshold condition $h(I) = 0$. The function $h(I)$ has a minimum at approximately (0.022, -0.0004), lying under the threshold and suggesting a critical point where the system dynamics change. After crossing the threshold, a limit cycle emerges, leading to sustained oscillations, which can designate a Hopf bifurcation.

In order to more precisely identify and demonstrate the bifurcation, we increase parameter b from 0.01 to 0.03 in small increments of 0.001 and visualize its influence on **S** and **I** in Figure 16 using three different initial conditions provided in the exercise sheet. We can directly identify that the plots where $b < 0.021$ have a stable equilibrium as trajectories converge to a single point; however, at $b = 0.021$, the equilibrium becomes marginally stable, and trajectories begin spiraling without fully stabilizing. Right after reaching the value of 0.021, at $b = 0.022$, a limit cycle occurs, indicating a Hopf bifurcation. The following b-changes ($b > 0.022$)

influence trajectories that are now spiraling into the periodic orbit instead of the fixed point. The detailed analysis of the trajectories dynamics is demonstrated in Table 1

Table 1: Behavior of the system for increasing b-values for all three initial conditions.

b-**Range**	**IC1:** $(195.3, 0.052, 4.4)$	**IC2:** $(195.7, 0.03, 3.92)$	**IC3:** $(193, 0.08, 6.21)$
$0.010 \leq b < 0.020$	Damped oscillations & stable	Damped oscillations & stable	Faster decay & stable
$0.020 \leq b \leq 0.021$	Weak oscillations	Weak oscillations	Weak oscillations (higher A)
$b \approx 0.022$	Hopf bifurcation (limit cycle)	Hopf bifurcation (limit cycle)	Hopf bifurcation (limit cycle)
$0.022 < b < 0.024$	Stable limit cycle	Stable limit cycle	Slow decay to equilibrium
$0.024 \leq b < 0.030$	Large amplitude limit cycle	Large amplitude limit cycle	Slow decay to equilibrium
$b \geq 0.030$	Stabilizes back to equilibrium	Stabilizes back to equilibrium	Slow decay to equilibrium

The increase in parameter b affects the recovery rate $\mu(b, I)$ but does not directly influence R_0, determining the average number of infections caused by a single infectious individual, as well as the system dynamics in general. Reaching $R_0 > 1$ would mean the unique endemic equilibrium exists. However, R_0 is still insufficient to fully establish the dynamic behavior of the nonlinear system since it only describes the initial growth or decline of infections when the system is close to the disease-free equilibrium, assuming a linear approximation. In the case of Hopf bifurcation, the system undergoes a change in stability, leading to oscillations, which cannot be inferred from R_0 and requires a nonlinear analysis.

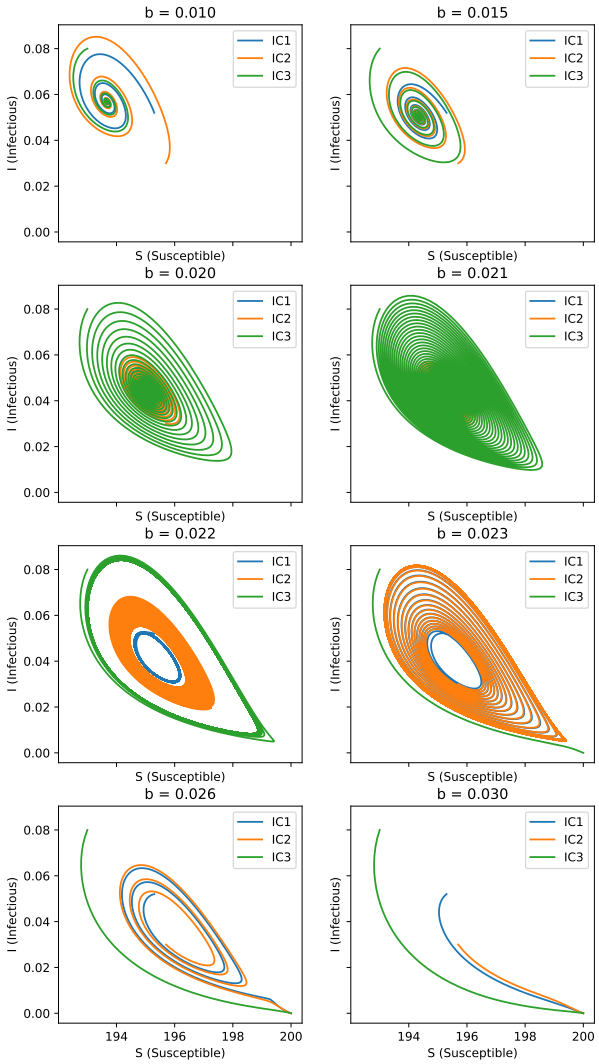

Figure 16: Visualization of how changing b parameter influences **S** and **I**, where IC is an initial condition: IC1 = [195.3, 0.052, 4.4], IC2 = [195.7, 0.03, 3.92], IC3 = [193, 0.08, 6.21]

Bonus: Saddle-Node Bifurcation in SIR Model

A saddle-node bifurcation is a local bifurcation in which two fixed points (or equilibria) of a dynamical system collide and annihilate each other [3]. In [10], it is claimed in **Theorem 3.1** that if $\mathbb{R}_0 < 1$ and $\beta > d + \nu + \mu_0$, the system has two endemic equilibria if and only if $\Delta_0 > 0$ and $\mathscr{B} < 0$. Given that at the endemic equilibrium I_1 and I_2 are the roots of a quadratic equation $\mathscr{A}I^2 + \mathscr{B}I + \mathscr{C} = 0$, we can simply set $\Delta_0 = 0$ and see if there is a valid bifurcation at the zero points. In this task we choose β as the changeable parameter and keep all other parameters the same as stated in the exercise sheet. To visualize the zero points, we plot Δ_0 against β, and the result is shown in Figure 17. We see that there are two zero points, and the numerical result is [10.56313 11.44318]. This result is calculated with resolution set to $1e-5$. Later when calculating Δ_0 in `class endemic_equilibrium`, we allow some tolerance so that if out of numerical reasons Δ_0 is a negative number whose absolute value is smaller than the tolerance, we treat it as zero. We set the tolerance to $1e-10$ here.

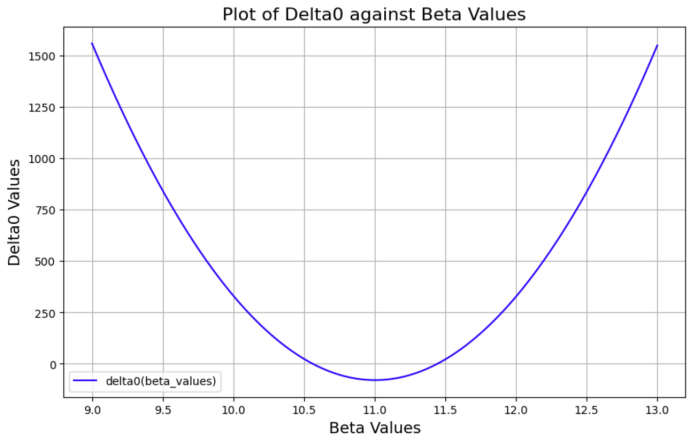

Figure 17: Δ_0 against β

We can use the mathematical conditions proposed in [10] to verify if around these two zero points there are two endemic equilibria, but it would involve a lot of calculations that are rather tedious to exhibit here. So we continue with some easier graphical methods. We plot I_1 and I_2 calculated from the quadratic equation against β to see if those I values can be valid. The result is shown in Figure 18. In this figure we can see that the left zero point cannot represent an endemic equilibrium because the corresponding I_1 and I_2 are both negative. So without much calculation we can come to the conclusion that only the right zero point can be valid.

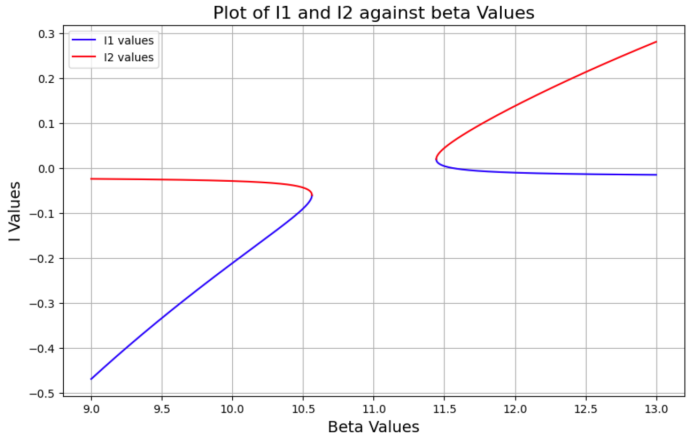

Figure 18: I_1 and I_2 at endemic equilibrium plotted against β

To confirm the result, we choose a β value that can make both I values positive and use the closed-form formulas to calculate S and R at endemic equilibrium to see if those values are sensible. We choose $\beta = 11.5$, and the

resulting (S, I, R) configurations are [199.55163 0.00391 0.4054] and [195.06278 0.04392 4.45408], corresponding to I_1 and I_2 in Figure 18, respectively. These results are seemingly acceptable. Also, at $\beta = 11.5$, $\frac{dS}{dt}$ is numerically zero in both configurations (with tolerance $1e-10$), so we can conclude that around Δ_0's right zero point $\beta = 11.44318$, a saddle-node bifurcation will take place.

Before visualizing the bifurcation, it's good to see which endemic equilibrium around the zero point $\beta = 11.44318$ is stable. We test it at $\beta = 11.5$ using the model without tolerance in `sir_model.py`, setting the initial (S, I, R) configuration to exactly the endemic equilibrium-configurations, the result is shown in Figure 19. It's obvious to see that the second configuration represents a stable endemic equilibrium while the first doesn't, because the small numerical error already drives the first endemic equilibrium to the disease-free equilibrium. Thus we need to slightly adjust the model in `sir_model.py` to counter this issue. We create a new function called `model_with_tolerance`, which will treat the calculated derivative as zero if its absolute value lies within the tolerance (here tolerance is set to $1e-10$). With this small change, the two equilibria do not deviate from their start points now, and we can continue the following visualization.

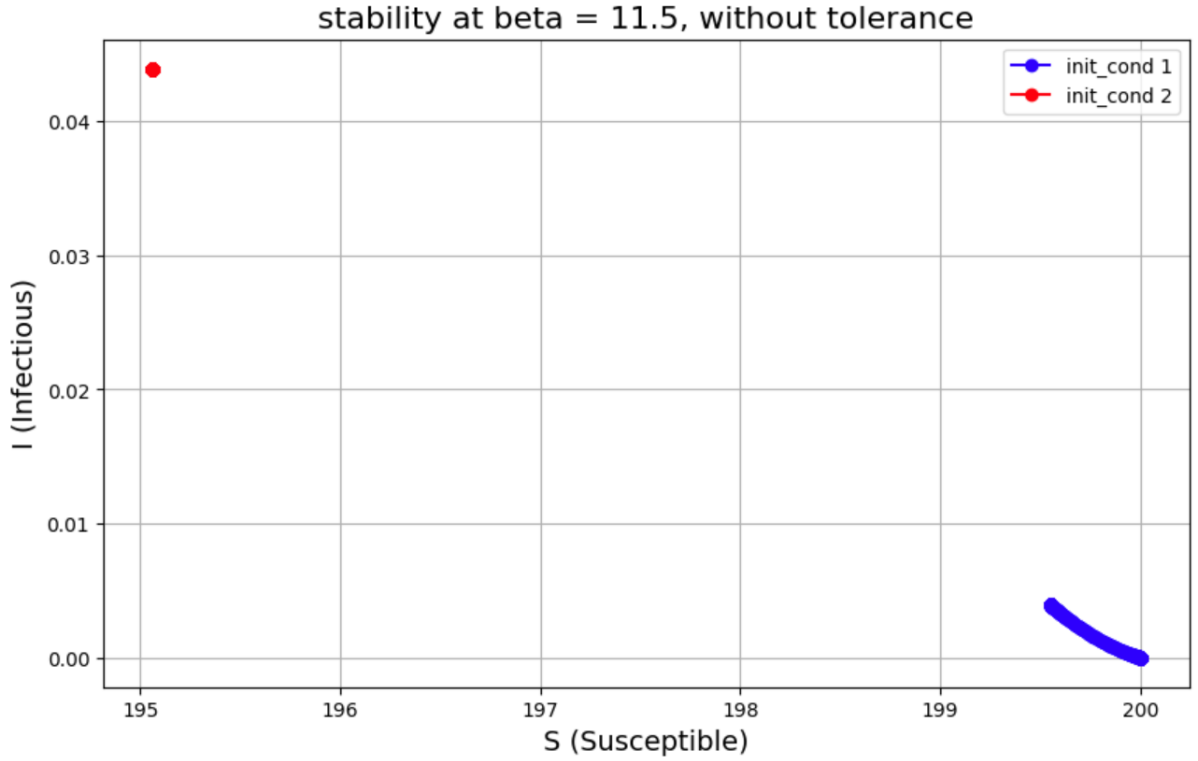

Figure 19: Test equilibrium stability at $\beta = 11.5$

To show that the two endemic equilibria approach each other and finally merge at the right zero point $\beta = 11.44318$, we create a set of β that gradually approaches $\beta = 11.44318$ from 11.5. This is shown in Figure 20. In this figure we can see that the two endemic equilibria indeed become closer to each other as β comes nearer to the right zero point of Δ_0. Out of numerical reason, the exact merging moment can not be shown here, because the zero point $\beta = 11.44318$ is not accurate enough, and using it to calculate Δ_0 will make Δ_0 a negative number, even with some tolerance introduced. To avoid tedious parametrization, we just leave a qualitative result here, which should be enough to show this whole process.

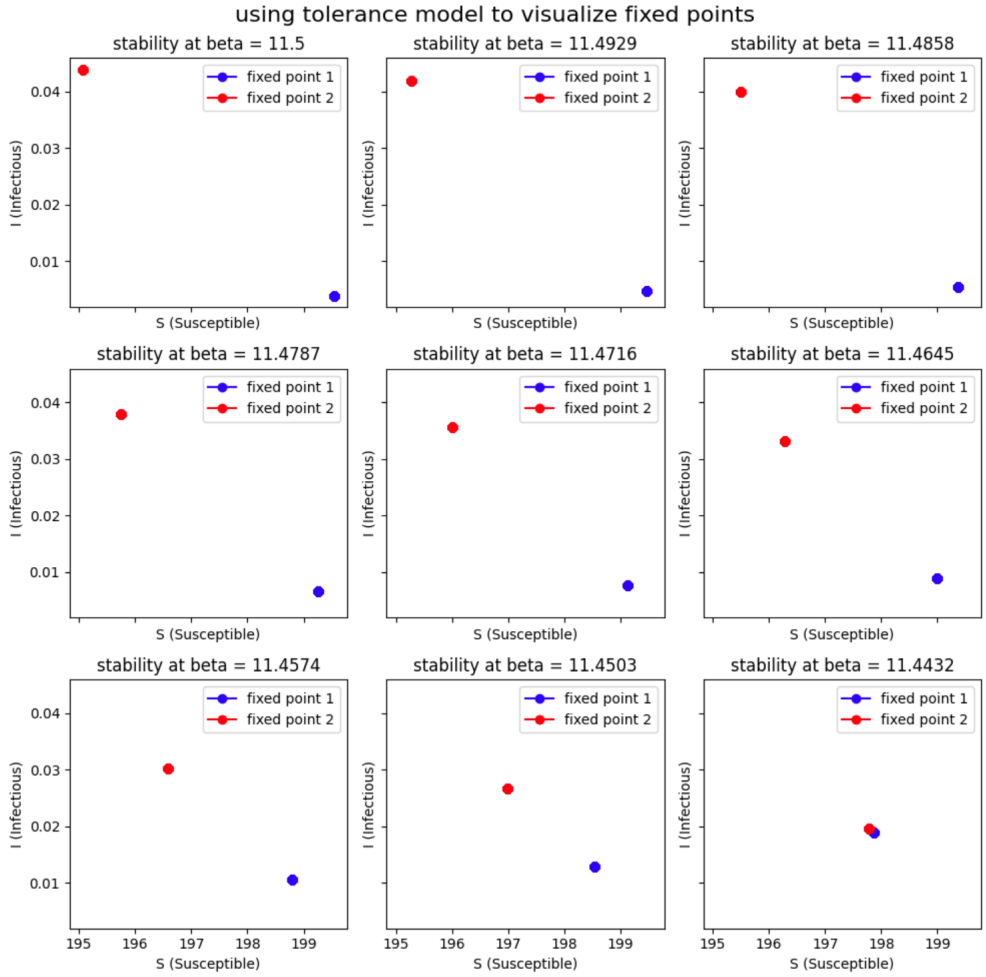

Figure 20: Evolution of endemic equilibria with β approaching the right zero point of Δ_0

After the the two endemic equilibria merge at $\beta = 11.44318$, if we keep reducing β, Δ_0 will fall below zero, which will lead to the disappearance of the endemic equilibrium. This also corresponds to the characteristic of saddle-node bifurcation where the two equilibria "annihilate each other", meaning that they both disappear after colliding each other.

References

[1] Wikipedia contributors. Bifurcation diagram. `https://en.wikipedia.org/wiki/Bifurcation_diagram`, 2024. Accessed: 2024-12-13.

[2] Wikipedia contributors. Lorenz system. `https://en.wikipedia.org/wiki/Lorenz_system`, 2024. Accessed: 2024-12-13.

[3] Wikipedia contributors. Saddle-node bifurcation — Wikipedia, the free encyclopedia, 2024. Accessed: 2024-12-13.

[4] Wikipedia contributors. Transcritical bifurcation. `https://en.wikipedia.org/wiki/Transcritical_bifurcation`, 2024. Accessed: 2024-12-13.

[5] Yale University Department of Mathematics. Logistic map fixed points. `https://gauss.math.yale.edu/fractals/Chaos/FixedPoints/LogisticFixedPoints/LogisticFixedPoints.html`, 2024. Accessed: 2024-12-13.

[6] Yale University Department of Mathematics. Stable fixed point proof for the logistic map. https://gauss.math.yale.edu/fractals/Chaos/FixedPoints/LogisticFixedPoints/StableFPProof.html, 2024. Accessed: 2024-12-13.

[7] Yanghong Huang. Part 5: Bifurcation of maps. https://personalpages.manchester.ac.uk/staff/yanghong.huang/teaching/MATH4041/part5.pdf, 2024. Accessed: 2024-12-13.

[8] Yuri A Kuznetsov. *Elements of Applied Bifurcation Theory*. Springer New York, 2004.

[9] Pierre Roussel. Stability maps. https://people.uleth.ca/~roussel/nld/stability_maps.pdf, 2024. Accessed: 2024-12-13.

[10] 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.

[11] Eric W. Weisstein. Chaos. https://mathworld.wolfram.com/Chaos.html, 2024. From MathWorld–A Wolfram Web Resource, Accessed: 2024-12-13.

[12] Weisstein, Eric W. Logistic map. https://mathworld.wolfram.com/LogisticMap.html, 2024. Accessed: 2024-12-13.

[13] Wikipedia contributors. Logistic map — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Logistic_map, 2024. Accessed: 2024-12-13.