Dynamical modeling of pathogenic bacteria in surface drinking water

MASTER THESIS IN APPLIED AND COMPUTATIONAL MATHEMATICS

Farhan Omar

UNIVERSITY OF BERGEN
Faculty of Mathematics and Natural Sciences

Department of Mathematics

September 16, 2020

Abstract

Dynamical modeling of ecological systems has been an interesting tool to understand the interactions between species in nature. In this thesis, we construct a dynamical model for interactions between pathogenic bacteria and two potential predator microorganisms, namely Daphnia and Protozoa in drinking water resources. Here we have multistage predation as both predators prey on bacteria, but one of the predators (Daphnia) additionally preys on the other predator (protozoa). There is also a direct inflow of bacteria into the system by fecal contamination. Linear functional response (Lotka-Volterra type) is considered for simplicity.

Furthermore, we studied the boundedness of the solution of the system, existence, local stability of biologically feasible equilibrium points, and global stability of the coexistence point. We also performed sensitivity analysis of parameters of the system, observability, and possible existence of chaos in the system. Bacteria data from Bergen municipality are analyzed along with numerical simulations to verify our analytical results.

The key findings of this study are that one of the predators may go to extinction or all species will coexist at stable concentration, periodically or chaotically. We also find out that the ratio between the food conversion efficiency of the two predators is the most sensitive parameter in the system. A comparison between our numerical simulation and bacteria data analysis showed that the model managed to demonstrate the general trend in the data, especially for the parameter set that gave chaotic behavior.

Keywords— Dynamical model, functional response, boundedness, nondimensionalization, stability, sensitivity, chaos, observability

Acknowledgments

This work would not have been possible without help from good people. First, I must thank my mother for her commitment and insistence that I should attend and continue school, without her, I would have been out of school, maybe after the sixth grade. Second, I want to thank my adviser Guttorm Alendal for nice and valuable discussions and more importantly his patience and support to the topic of my choice.

Special thanks to my friend Abdul Majid Murad for his general advice, support, and for reading the draft and providing valuable feedback and of course for our tea\lunch meetings. Additionally, I want to thank my friend Sunniva for being beside me in difficult situations and listening to my frustrations. Finally, I want to thank Takuya Iwanaga one of the sensitivity analysis library contributors for helping with sensitivity analysis and improve the performance of my code for that and others from the open-source community.

This work obtained a grant from Hordaland county (now Vestland county) for writing about a topic concerning the environment, so I am grateful to that.

Contents

Chapter 1

Introduction

Mathematical modeling of ecological systems, especially predator-prey systems, has been an important tool to understand the relationship between different species that live together in nature. This is important for example in controlling harmful species that invade a region or extinction of rare species due to unbalance in the food chain caused by natural or human-made factors. This field has been in continuous development since the first model by Lotka-Volterra (1928) [60]. Now there is a special direction in mathematics to study biological phenomena called mathematical biology, along with a special field of informatics called Bioinformatics.

There are two ways to study the interactions between biological species, the first one is the experimental method which is used by biologists and the second one is theoretical used by mathematicians. To better analyze and understand a biological system, we need to consider both methods together. The question which arises here is, can we make a dynamical model for the whole food chain in nature? Theoretically, it should be possible. But the system will be huge and very complicated and maybe impossible to analyze. This can be attributed to the different types of interactions among species such as predation, competition, mutualism, symbiosis, and parasitism, etc. The involvement of many species makes the system even more complex. Therefore, it will be more beneficial to study small systems or divide large ones into several subsystems to simplify the analysis.

Recently I have been interested in freshwater ecology because freshwater bodies are used to supply communities with drinking water in many countries, especially in Norway as the drinking water is mainly from surface water resources such as lakes. In these bodies of water, there exist many pathogenic bacteria that live naturally or due to fecal contamination, alongside with other microorganisms. That is why it is of interest to know whether the water in the different surface water resources is completely safe to drink.

In Norway, each municipality has a mechanism to remove bacteria, viruses, and other harmful species from the water before sending it to distribution networks. They also take samples from these water resources regularly to study the development of these pathogenic bacteria. In 2019, Askøy municipality in Norway, hundreds of people became sick because the water in one of the storing mediums became contaminated with pathogenic bacteria from the feces of some animals. In the aftermath of that, I became interested in making a dynamical model for the bacteria living in surface water using available data from Bergen and Askøy municipality. The goal was to investigate the dynamics of interaction between these species and the opportunity to clean the water from pathogenic bacteria biologically and find possible solutions to make the water resources safe to drink from even without any traditional treatment.

1.1 Problem outline

In this master thesis, we are trying to make a dynamical model of interactions between pathogenic bacteria, Daphnia, and protozoa in surface drinking water bodies such as lakes, rivers, and other resources. The goal is to study the interaction between them, the time evolution of the system, longtime predictability, and investigate the possible total extinction of bacteria or other microorganisms being studied.

1.2 Biological theory about the species considered in this work

In this section, we will briefly present some biological background regarding bacteria and microorganism species that are included in the study. The pathogenic bacteria included are E. coli, Campylobacter, Clostridium perfringens, and Intestinal enterococci. The microorganisms are Daphnia and protozoa.

1.2.1 Coliform bacteria

They are non-spore-forming bacteria and considered as indicator organisms. They are divided into total coliform, fecal coliform, and E. coli. E. coli is a subgroup of fecal coliform, which in its turn is a subgroup of total coliform. Factors that affect the survival of E. coli in water are temperature, pH value, Salinity, Predation, Stream-bed resuspension and sunlight intensity. Even though the predation reduce the amount of viable E. coli, the temperature remain the most important factor as it affect the inactivation rate of E. coli. The Q_{10} equation can be used to model the dependency of the inactivation rate of E.coli on the temperature in various sources of water. Mathematically, it is given by,

$$\frac{k}{k_*} = Q_{10}^{\frac{(T_c - T_{c,*})}{10}}$$

where

Q_{10}: is a measure of the rate of change in the inactivation rate as a consequence of increases in temperature by 10 °C 10 increments. $T_{c,*}$:is the reference temperature.

k_*: is the inactivation rate at the reference temperature.

k: is the inactivation rate at Celsius temperature T_c. Note that k and k_* have the same units, which are day^{-1}.

The Q_{10} equation is more accurate when water from the same source is tested at different temperatures. It is known that E. coli can survive at $4\,^{\circ}\mathrm{C}$ as well as at higher temperatures. For further reading see [5, 29]

1.2.2 Campylobacter jejuni (C.jejuni)

These are non-spore-forming bacteria that can cause disease in humans. They inhabit most of the freshwater systems. The factors that affect the survival of C.jejuni in water are:

- Temperature

  Optimal survival of Campylobacter has regularly been identified at low temperatures with a maximal survival of greater than 4 months at densities exceeding $1.10^4\, CFU\, ml^{-1}$ reported for sterile stream-water held at $4\,^{\circ}\mathrm{C}$.

  The temperature threshold at which survival becomes significantly affected has been identified to be in the region of $16-22$ °C.

- Light.

  Light affects changes in the uptake of nutrients and progressively inhibit the active transport and biosynthetic processes of exposed E. coli populations.

- Oxygen.

  The survival of C.jejuni increases at lower oxygen concentrations.

- Nutrients.

  When bacteria are translocated from their intestinal niche into the aquatic environment,they confront a host of potential stressors such as suboptimal temperatures, nutrient deprivation, biological interactions, U.V. irradiation, and disinfectants. Consequently, they may undergo a temporal physiological and morphological transition into a viable but non-culturable (VNC) stage, retaining basal metabolic activity yet failing to grow or multiply in bacteriological culture.

- Biotic interaction.

  Such as antagonism, competition, and predation. The predatory and competitive activities by autochthonous aquatic microflora especially protozoa reduce the survival times of C. jejuni in water. In addition to the antagonistic role, Protozoa act also as endosymbionts of coliforms and bacterial pathogens including C.jejuni, providing protection from survival antagonists including chlorine.

For further reading see [58]

1.2.3 Clostridium perfringens (C.perfringens)

C.perfringens are anaerobic, gram-positive, spore-forming rod-shaped bacteria. They are found in soil, sewage, intestinal tracts of animals and humans, drinking water, and food that is contaminated with the bacteria. They can live at a temperature of 50 °C and are able to make spores and resist heating, chlorination, and other stress factors. Thus, if the food is contaminated by spores, they will germinate after the food is cooked.

These bacteria can lead to food poisoning which in terms leads to abdominal cramps, watery diarrhea, and nausea. They are therefore considered as an indicator in water quality measurements and toxic substances as well.

1.2.4 Intestinal enterococci

These are spore-forming bacteria that can be found in most freshwater systems. Factors that affect the survival of enterococci in secondary habitats as environmental water, aquatic vegetation, or sediment are,

- Sunlight
  Sunlight has been a suspected stressor of bacteria, i.e., the reduction in levels of enterococci is enhanced at higher temperatures (i.e., during summer months and in warmer climates).
- Salinity.
  These bacteria can grow in the presence of salt (6.5% $NaCl$) more than fecal Coliforms and E.coli.
- Disinfection.
- Starvation.
  Enterococci can enter the viable-but-non-culturable (VBNC) state. This phenomenon describes a state in which bacteria that can normally be cultured under a defined set of conditions lose that ability while retaining viability, as assessed by measurements of membrane potential, infectivity, mRNA expression, the ability to reproduce, or cell envelope integrity.

- Predation.
  Such as grazing by protozoa.

It was shown that many bacterial pathogens resist digestion within protozoa and can replicate inside them. This is the most important concern in terms of health risk management for drinking water since the number of bacteria contained in a single organism can easily exceed the infectious dose for humans. Another concern is that those bacterial pathogens can survive within the resisting form of their host, i.e., the cysts, considering that cysts can resist various extreme conditions and have a prolonged survival time in natural environments.

For further reading, see [4, 8]

1.2.5 Protozoa

Protozoa are one-celled animals that are found in nearly every habitat on earth and can be very diverse in freshwaters. This group includes (i) autotrophs, i.e. they can produce their own food, using materials from inorganic sources and (ii) heterotrophs, i.e. they cannot synthesize their own food and rely on other organisms for nutrition. Many types survive by ingesting particles or absorbing dissolved organic carbon. They are very important predators of bacteria in the aquatic environment. Some bacterial pathogens have evolved to become resistant to protozoa digestion. Many survive, and even multiply, within the intracellular environment of certain protozoa. [49]. Protozoa can reduce the survival of C. jejuni in unsterilized lake waters and influences E. coli survival as well. [48] Some types of protozoa can cause diseases in humans and symptoms can range from mild to life-threatening. [46]

1.2.6 Daphnia

Daphnia are planktonic crustaceans that belong to Phyllopoda and include over 100 known species of freshwater plankton organisms. They inhabit most types of standing fresh except for extreme habitats, such as hot springs waters. Adults range from 1 mm to 5 mm in size. They feed on small, suspended particles in the water, which range in size from 1 μm up to 50 μm in diameter. The food is made up of planktonic algae or bacteria. In most habitats, Daphnia have low density or completely disappear during part of the year, usually the cold or the dry season. Growth is slowed down by density-dependent competition, usually because of food shortage, predators, or parasites. The peak in Daphnia density usually follows a peak in algae density and may be followed by the clear-water phase in which Daphnia effectively remove most of the phytoplankton from the water. [14]

Daphnia carinata, which is a member of the Daphnia family, can graze Campylobacter jejuni cells at a rate of 7 $\% individual^{-1}\, h^{-1}$ under simulated natural conditions in the presence of an algal food source. Daphnia will also effectively graze Escherichia coli and predates on protozoa as well. [31]

1.3 Types of functional response (FR) in ecological models

In this section, we will review the most used functional responses in ecological models.

Definition 1.3.1 (Functional response). The functional response represents the number of prey consumed per predator per unit time and can be prey dependent, predator dependent, or ratio-dependent.

1.3.1 Prey dependent functional response

In this functional response, the number of prey consumed per predator depends only on the prey population without taking into account the predator population. The following are some examples of this type of functional response (see Figure 1.1)

1. Lotka-Volterra or Holling type I

   This type of functional response assumes a linear increase in the intake rate with food density. Mathematically, it is given by

   $$g(N) = aN$$

   , where N is the population density of prey species at time t and a is the predation rate.

2. Holling type II [12]

   This functional response is given by the following equation,

   $$g(N) = \frac{aN}{1 + ahN}$$

where

a: is the searching efficiency or searching rate or attack rate.
h: is the handling time, the time spent by the predator handling each prey encountered, and during which it stops searching.

Now let us take the limits at infinity and zero,

$$\begin{aligned}\lim_{N\to\infty} g(N) =& \quad \lim_{N\to\infty} \frac{aN}{1+ahN} = \lim_{N\to\infty} \frac{a}{ah} = \frac{1}{h}\\ \lim_{N\to 0} g(N) =& \quad \lim_{N\to 0} \frac{aN}{1+ahN} = 0\end{aligned}$$

In this type of functional response, the predator spends much time in finding and killing the prey species leading to maximum mortality rate, while at a high prey density ($N \to \infty$) the mortality rate flattens out (and equals $\frac{1}{h}$) as the predator spend much of its time in handling the prey rather than in searching.

Monod (1949) [34] used an equivalent function to describe the growth of bacteria cultures,

$$\begin{aligned}R(C) &= R_k \frac{C}{C_1 + C} && \text{dividing the numerator and denominator by } C_1\\ R(C) &= \frac{\frac{R_k}{C_1} C}{1 + \frac{1}{C1} C} && \text{this become equivalent to Holling type II}\end{aligned}$$

where

R: is the specific growth rate of a bacteria population.
C: is the concentration of the limiting nutrient.
R_k: the maximum specific growth rate of the culture.
C_1: is the concentration of the nutrient at which the specific growth rate is half the maximum.

Michaelis-Menten (1913) [23] used another equivalent functional response to study enzyme kinetics. It is given by the following equation,

$$g(N) = \frac{mN}{A+N}$$

m: perdition coefficient or the maximum specific growth rate of the predator.
A: the half saturation constant when $A = N \to g(N) = \frac{1}{2}m$. It is equivalent to Monod and so to Holling type II

3. Holling type III [12]

   The type III functional response is similar to type II, but the variable N is squared. Mathematically, it is given by the following equation,

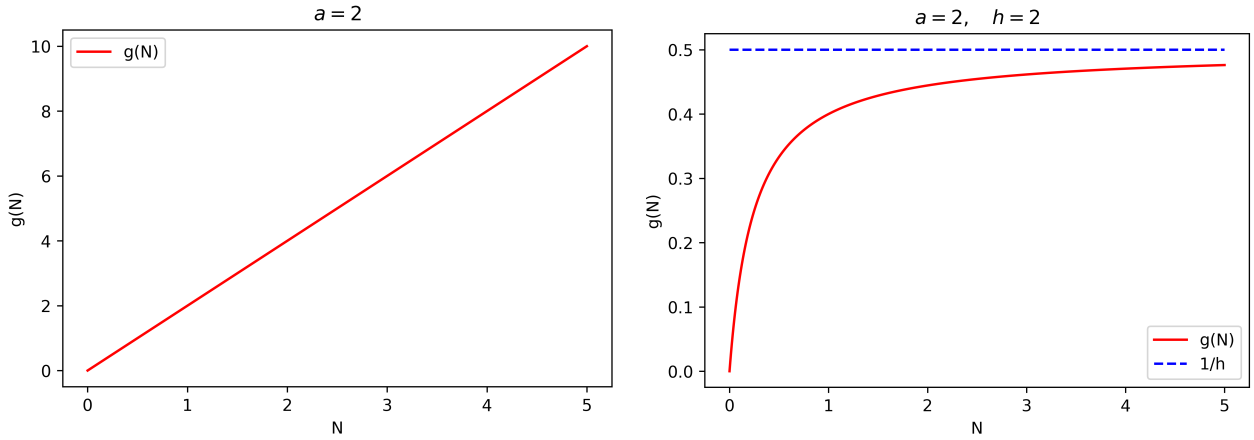
   $$g(N) = \frac{aN^2}{1 + ahN^2}$$

   where a and h are the same as in Holling type II.

   Now let us take the limit at infinity and at zero,

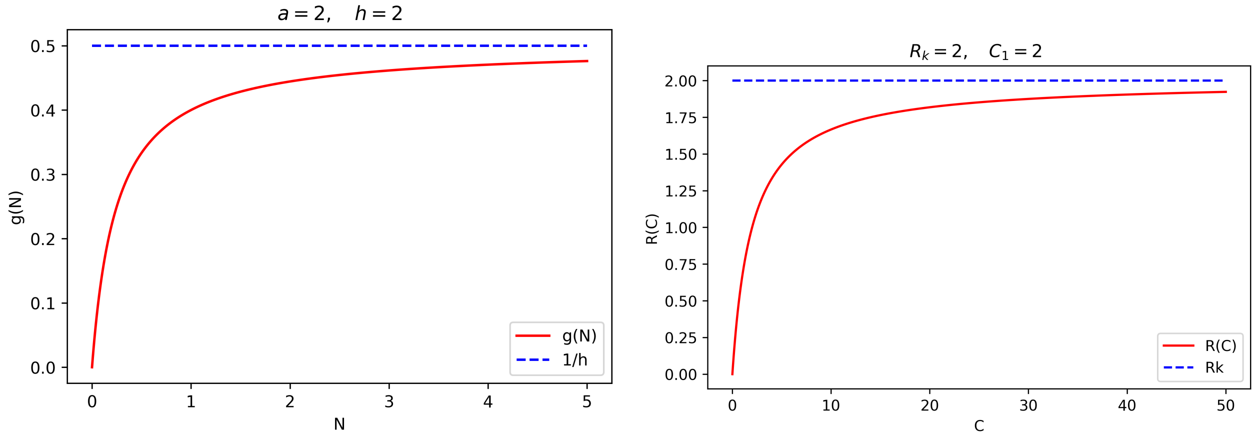
   $$\begin{aligned} \lim_{N\to\infty} g(N) = &\quad \lim_{N\to\infty} \frac{aN^2}{1 + ahN^2} = \lim_{N\to\infty} \frac{a}{ah} = \frac{1}{h} \\ \lim_{N\to 0} g(N) = &\quad \lim_{N\to 0} \frac{aN^2}{1 + ahN^2} = 0 \end{aligned}$$

Note that both type II and type III of Holling functional response have the same asymptote $\frac{1}{h}$ when prey density tends toward infinity.



(a) Lotka-Volterra.

(b) Holling Type I



(c) Holling Type II.

(d) Monod

Figure 1.1: Different types of Prey Dependent Functional Response (PDFR) for a chosen set of parameters.

1.3.2 Predator dependent functional response

In this functional response, the number of prey consumed per predator depends on both prey population and predator population, as both appear in the mathematical function that represents it. The following are some examples of this type of functional response,

1. Hassell-Varley

$$g(N, P) = \frac{\alpha N}{P^m}$$

2. Beddington-DeAngelis

$$g(N,P) = \frac{aN}{1+ahN+cP}$$

3. Arditi-Akçakaya.

$$g(N,P) = \frac{\alpha N P^{-m}}{P+\alpha h N P^{-m}}$$

m: the mutual interference between predators. $m \in [0,1]$

1.3.3 Ratio dependent functional response

In this functional response, the number of prey consumed per predator depends on the ratio of the prey population to the predator population. The following are some examples of this type of functional response (see Figure 1.2),

1. Contois (1959)

   Contois [10] built the following model for bacterial growth from the model suggested by Monod [34],

$$\begin{aligned} R &= \frac{u_m S}{BP+S} \qquad \text{dividing the numerator and denominator by P} \\ \Rightarrow R(\frac{S}{P}) &= \frac{u_m \frac{S}{P}}{B+\frac{S}{P}} \\ u_m &= \frac{R_m}{1+b} \\ B &= \frac{b u_m}{c R_m} \\ A &= b S_0 \end{aligned}$$

   where

   R: is the specific growth rate of the bacteria population

   S: is the concentration of limiting nutrients (prey).

   S_0: the initial concentration of the limiting nutrients

   P: the population density of the bacteria (predator)

   u_m, B: are growth constants that are constant under a defined condition.

   R_m: is the maximum specific growth rate of the culture (bacteria)

   A: is the concentration of the nutrient at which the specific growth rate is half the maximum $\frac{1}{2}R_m$

b: proportionality constant.

c: the yield coefficient

2. Arditi-Ginzburg I: [3]

$$g(\frac{N}{P}) = a\frac{N}{P}$$

3. Arditi-Ginzburg II [3]

$$\begin{aligned} g(\frac{N}{P}) = & \frac{\alpha\frac{N}{P}}{1+\alpha h\frac{N}{P}} = \frac{\alpha N}{P+\alpha hN} & \text{dividing the numerator and denominator by } \alpha h \\ = & \frac{\frac{1}{h}N}{\frac{1}{\alpha h}P+N} & \text{this is equivalent to the Contois functional response} \end{aligned}$$

Where

N: the population density of prey species at time t.

P: the population density of prey species at time t.

h: Has the same unit as in Holling's type II function (a time). Geometrically it is the reciprocal of the upper asymptote.

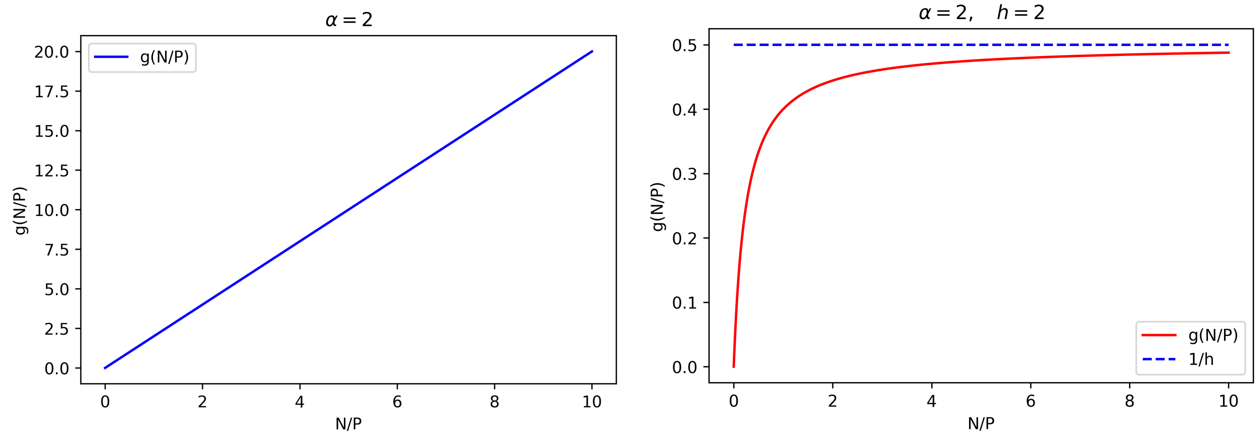
α: is the rate at which the prey population is made available to the predator population. Whatever the abundance of predators, the prey death rate due to predation cannot exceed α . Geometrically it is the slope of g at the origin.

Now let us take the limit at infinity and at zero,

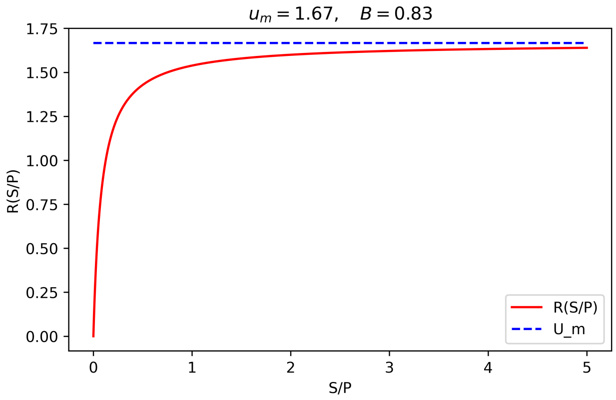
$$\lim_{\frac{N}{P}\to\infty} g(\frac{N}{P}) = \lim_{N\to\infty} g(\frac{N}{P}) = \frac{1}{h}$$

$$\lim_{\frac{N}{P}\to 0} g(\frac{N}{P}) = \lim_{N\to 0} g(\frac{N}{P}) = \lim_{P\to\infty} g(\frac{N}{P}) = 0$$

Thus, the functional response tends to the same limit as Holling's type II and III.



(a) Arditi-Ginzburg I. (b) Arditi-Ginzburg II.



(c) Contois.

Figure 1.2: Different types of Ratio Dependent Functional Response (RDFR) for a chosen set of parameters.

1.4 Logistic equation

The logistic equation is developed by Belgian mathematician Verhulst 1838 and is given by the following equation,

$$\frac{dN}{dt} = rN - aN^2 = rN\left(1 - \frac{N}{K}\right)$$

Here N represents the number of individuals at time t, r the intrinsic growth rate, a the intraspecific competition between the individuals, and $K = \frac{r}{a}$ the carrying capacity of the species N, which represents the maximum number of individuals that the environment can support.

Solving the equation with initial condition $N(0) = N_0$ we obtain,

$$N(t) = \frac{K}{1 + \left(\frac{K}{N_0} - 1\right)\exp(-rt)}.$$

Let us take the limit at zero and infinity,

$$\begin{aligned}\lim_{t\to\infty} N = & \lim_{t\to\infty} \frac{K}{1 + \left(\frac{K}{N_0} - 1\right)\exp(-rt)} = \frac{K}{1} = K,\\ \lim_{t\to 0} N(t) = & \frac{K}{1 + \left(\frac{K}{N_0} - 1\right)} = N_0.\end{aligned}$$

Thus, the number of individuals tends to the carrying capacity when times tend to infinity and to the initial population value at the beginning ($t = 0$) (see Figure 1.3).

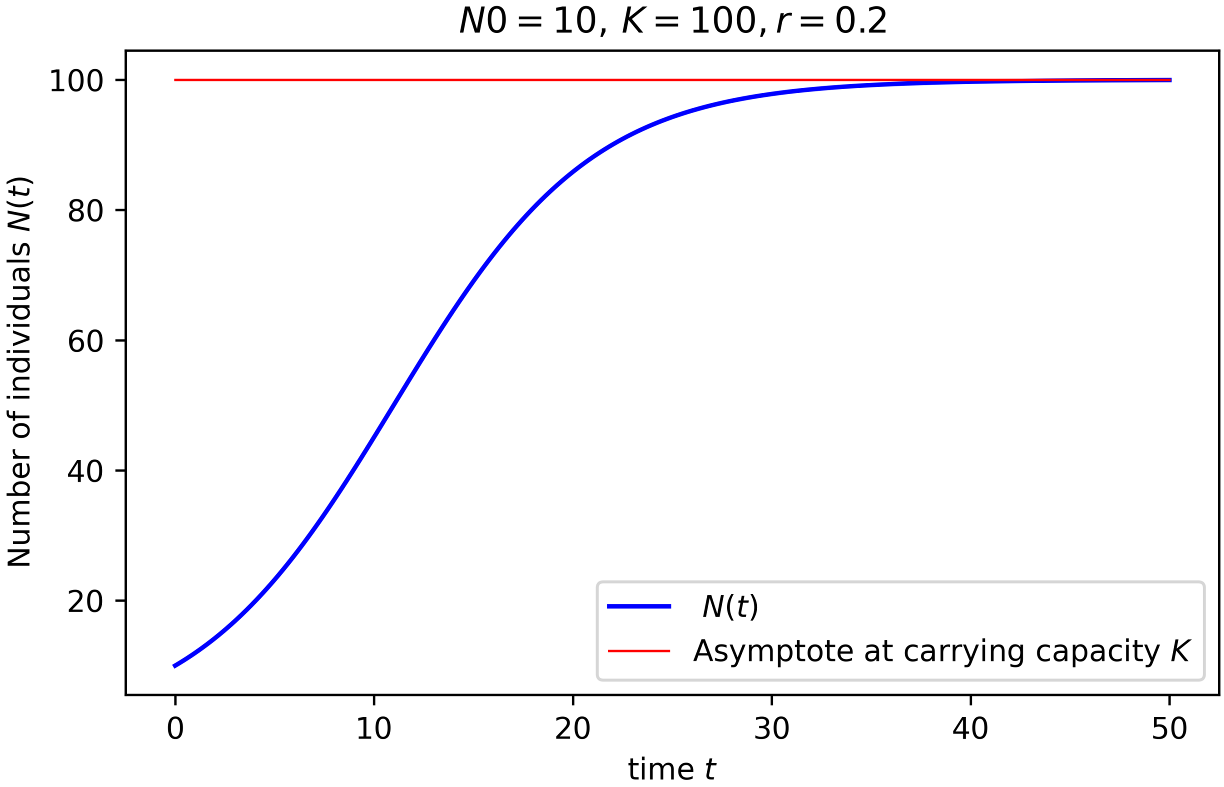


Figure 1.3: Logistic equation for a chosen set of parameters.

1.5 Examples of ecological models

In this section, we will review some ecological models and how they are built from the assumption given and finally build the model for the problem we are studying.

1.5.1 Two competing species

Assume that two species compete for limited nutrient resources in a closed environment and do not prey on each other. Let x and y be the population density of the first and second species, respectively. Assumptions of the model are as follows:

1. In the absence of the other species, the population of the two species grows according to the logistic equation.
2. The two species compete for nutrients and this will reduce the density of the two populations through the mixed term $a_{ij}x_ix_j$

Based on the above assumption, the dynamics of the system are described by:

$$\begin{aligned}
\frac{dx_1(t)}{dt} =& r_1x_1 - a_{11}x_1x_1 - a_{12}x_1x_2 = r_1x_1(1-\frac{a_{11}}{r_1}x_1) - a_{12}x_1x_2\\
=& r_1x_1(1-\frac{x_1}{k_1}) - a_{12}x_1x_2,\ x_1(0) > 0\\
\frac{dx_2(t)}{dt} =& r_2x_2 - a_{22}x_2x_2 - a_{21}x_1x_2 = r_2x_2(1-\frac{a_{22}}{r_2}x_2) - a_{21}x_1x_2\\
=& r_2x_2(1-\frac{x_2}{k_2}) - a_{21}x_1x_2,\ x_2(0) > 0
\end{aligned}$$

where

$r_{i,i=1,2}$: is the intrinsic growth rate of prey species.

a_{ij}: Measure the degree at which the species j interferes with species i.

$k_{i,i=1,2}$: is the environmental carrying capacity for ith species.

a_{ii}: are the intraspecific interference coefficient of the two species.
This model is constructed by Volterra in [60] and analyzed using linear stability theory in litterateur, for example [7] and [36]

1.5.2 Predator-Prey Equation by Lotka-Volterra [60]

Assume that there are two species in a closed environment and one of them (the predator) feeds on the other species (the prey) while the prey species get nutrients from other resources. Let x and y be the population density of the first and second species, respectively.

Assumption

- In the absence of the predator, the prey population grows at a rate proportional to the current population density. This is given by the term ax
- In the absence of the prey, the predator population dies out at a rate proportional to the current population. This is given by the term $-cy$
- The interference between the predator and prey will increase the growth rate of the predator by a term γxy and decrease the growth rate of the prey by a term $-\beta xy$.

Based on the above assumption, the dynamics of the system are described by:

$$\begin{aligned} \frac{dx(t)}{dt} &= \quad ax - \beta xy, \\ \frac{dy(t)}{dt} &= \quad - cy - \gamma xy, \end{aligned}$$

where

a: is the growth rate of the prey species.

c: is the death rate of the predator species

β, γ: is the effect of the interaction between the two species.

This model is constructed by Volterra in [60] and analyzed using linear stability theory in many litterateurs, for example [7] and [36]

1.5.3 Two prey one-predator model

Elettreby [15] studied a two prey-one predator model in which the prey teams help each other. The two teams of preys have densities $x(t), y(t)$, respectively, interacting with one team of predators with density $z(t)$.

The assumptions of this model are as follows:

1. In the absence of any predation, each team of preys grows logistically; this is the terms $ax(1-x)$ and $ay(1-y)$.

2. The effect of the predation is to reduce the prey growth rate by a term proportional to the prey and predator populations; this is the $-xz$ and $-yz$ terms.
3. The teams of preys help each other against the predator, e.g. in foraging and in early warning against predation that is a xyz term exist.
4. In the absence of any prey for sustenance, the predator's death rate results in inverse decay, that is, the term $-cz^2$.
5. The prey's contribution to the predator growth rate are dxz, eyz; that is proportional to the available prey as well as the size of the predator population.

Based on the previous assumptions, he proposed the following model,

$$\begin{aligned}\frac{dx(t)}{dt} =&ax(1-x)-xz+xyz,\\ \frac{dy(t)}{dt} =&by(1-y)-yz+xyz,\\ \frac{dz(t)}{dt} =&-cz^2+dxz+eyz,\end{aligned}$$

where the coefficients $,b,c,d$ and e are positive constants and $x(0), y(0), z(0) > 0$.

It is worth noting that the interspecific competition between prey species is not considered in this model, which is usually done in most biological models. The author assumed that the two prey species are gazelles and zebras, which do compete on food in nature. Furthermore, the author studied the local and global stability of the equilibrium points. In case the prey species help each other, there are two locally stable interior eqm. points, while only one in the case they don't help each other. Finally, numerical simulation was done to verify the analytical result.

1.5.4 Dynamical behavior of three species predator-prey system with mutual support between non refuge prey

D.Pal et al. [39] studied the effect of refuge and harvesting in a three-species model of two prey and one predator, where the two non refuge prey groups are helping each other and shielding from the predator group.

Assumptions

1. In the absence of any predator, each team of preys grows logistically.
2. Due to the effect of predation, the prey growth rate is reduced by a quantity proportional to the prey and predator population.
3. The non refuge teams of prey help each other against the predator.
4. Predator species is harvested with the harvesting effort E.

5. Prey's contribution to predator growth rate is proportional to available prey and size of predator population.
6. Some preys of each team are refuge, i.e., αx and βy .
7. When the team of 2nd prey helps first prey against the predator, then not only the total populations of the second team involved in this interaction but also the non refuge population of the first team involve in this interaction. The same situation occurs for the first prey support to the second prey.

Using the above assumptions, the following predator-prey model is proposed:

$$\begin{aligned}
\frac{dx(t)}{dt} &= ax(1-x)-(1-\alpha)xz+(1-\alpha)xyz,\\
\frac{dy(t)}{dt} &= by(1-y)-(1-\beta)yz+(1-\beta)xyz,\\
\frac{dz(t)}{dt} &= -cz^2+d(1-\alpha)xz+e(1-\beta)yz-Ez,
\end{aligned}$$

where the coefficients a, b, c, d and e are positive constants. With initial densities $x(0) > 0, y(0) > 0$ and $z(0) > 0$.

In the proposed model, linear functional response is used, but competition between prey species is not considered which could be important, especially if they feed on the same limited resource. However, the intraspecific competition between the predators species is considered the term$-cz^2$. Furthermore, they studied the local stability of the equilibrium points and global stability of the interior equilibrium. In this model there are two globally stable nonnegative interior equilibrium points under some assumptions, which means that the species can coexist. Finally, numerical simulation is carried out to verify the results in case of interior equilibrium point and the effect of harvesting and refuge.

1.5.5 Dynamics of species in a two preys-one predator model system with help

Ali et al. [2] studied a model consisting of two teams of prey interacting with one team of predators. The assumptions used in the model's construction are as follows.

Assumptions

1. Each species grows logistically in the absence of predation and other competing species.
2. Holling-type II or Michaelis-Menten functional response for predation is used.
3. The effect of intraspecific competition among predators is considered her as well. This is expressed in the term $-HX_3^2$.
4. The two prey teams help each other in prevention of the predator's attack so as to increase the growth rate of prey species. This is shown by the terms $\sigma_1 X_1 X_2 X_3$ and $\sigma_2 X_1 X_2 X_3$.

Based on the following assumptions, they proposed the following model:

$$\begin{aligned}
\frac{dX_1(T)}{dT} &= r_1(1-\frac{X_1}{K_1}) - \frac{M_1}{A_1+X_1}\cdot X_1X_3 + \sigma_1 X_1X_2X_3, \quad X_1(0)>0,\\
\frac{dX_2(T)}{dT} &= r_2(1-\frac{X_2}{K_2}) - \frac{M_2}{A_2+X_2}\cdot X_2X_3 + \sigma_2 X_1X_2X_3, \quad X_2(0)>0,\\
\frac{dX_3(T)}{dT} &= -HX_3^2 - DX_3 + \frac{E_1M_1}{A_1+X_1}\cdot X_1X_3 + \frac{E_2M_2}{A_2+X_2}\cdot X_2X_3, \quad X_3(0)>0,
\end{aligned}$$

where
X_1, X_2: are the population densities of prey species.
X_3: is the population density of predator species.
$r1, r2$: biotic potential
K_1, K_2: environmental carrying capacities of two prey species.
σ_1, σ_2: are coefficients of the help between two prey teams.
M_1, M_2 are predation coefficients (the maximal growth rate of the species).
E_1, E_2: are conversion factors of captured prey species into new predators.
A_1, A_2: are half-saturation constants.

All constants are assumed to be positive. This model differs from the previous models in using the Holling-type II functional response.

Further, they studied the local stability of nonnegative equilibrium points using linear theory and the Routh-Hurwitz criterion. The global stability of the interior equilibrium was also studied, and it was proved to be globally stable under some condition on the parameters given by the Lyapunov function method.

1.5.6 Persistence of Two Prey-One Predator System with Ratio-dependent Predator Influence

Kesh et al. [28] studied a mathematical model of two competing prey and one predator system. They used similar assumptions as the previous models but used a ratio dependent functional response for predation.

$$\begin{aligned}
\frac{dx_1(t)}{dt} &= r_1x_1(1-\frac{x_1}{k_1}) - \frac{a_1}{1+x_1+\alpha_1 y}x_1y - a_{12}x_1x_2, \quad x_1(0)>0,\\
\frac{dx_2(t)}{dt} &= r_2x_2(1-\frac{x_2}{k_2}) - \frac{a_2}{1+x_2+\alpha_2 y}x_2y - a_{21}x_1x_2, \quad x_2(0)>0,\\
\frac{dy(t)}{dt} &= -\epsilon y + \frac{\beta a_1}{1+x_1+\alpha_1 y}x_1y + \frac{\beta a_2}{1+x_2+\alpha_2 y}x_2y, \quad y(0)>0,
\end{aligned}$$

where
$x_i(t)(i=1,2)$: the population density of competing prey species which grow logistically.
$y(t)$: the population density of predator species at time t.
r_i: the specific growth rate of the prey species x_i.
k_i: the carrying capacity of the prey species x_i.
a_{ij}: the inter-specific competition coefficient
a_i: the uptake rate.
α_i: the interference coefficient of the predator.
β: the food conversion efficiency of the predator.
ϵ: the mortality rate of the predator
Here all the parameters are assumed to be positive.

The proposed model has six equilibrium points, but only one at which all species coexist. Furthermore, they studied the local stability of the equilibrium points and global stability as well using the Lyapunov function and Bendixson-Dulac criterion. They proved that the dynamics do not change by changing the functional response from ratio-dependent to linear in the Lotka-Volterra systems.

1.5.7 Persistence and stability of a two prey one predator system

T. K. Kara et al. [27] studied a model consisting of two competing prey x and y, and one predator z with the presence of time delay due to the gestation.

The assumptions of the model are as follows:

1. In the absence of the predator the prey population density grows, according to the logistic law of growth.
2. The two prey species compete for food resources.
3. One prey is much higher in abundance and more vulnerable compared to the other.
4. The handling time for one prey x is negligible and for that Holling type I functional response is used, whereas the predator needs sufficient handling time for other prey y, thus Holling type II functional response is used for that.
5. There is a reaction time for predator.

Based on the previous assumptions, the following model is proposed:

$$\begin{aligned}
\frac{dx(t)}{dt} &= rx(1-\frac{x}{K})-a_1xy-\omega_1xz,\\
\frac{dy(t)}{dt} &= sy(1-\frac{y}{L})-a_2xy-\frac{\omega_2}{m+y}yz,\\
\frac{dz(t)}{dt} &= -cz+b_1\omega_1x(t-\tau)z+b_2,\frac{\omega_2}{m+y}y(t-\tau)z,
\end{aligned}$$

r, s: are the intrinsic growth rate of the two prey species
K and L: are the carrying capacities of the two prey species.
c: is the mortality rate coefficient of the predator.
a_1, a_2: are inter-specific interference coefficient of two prey species.
ω_1, ω_2: is the first and second type prey species searching the efficiency of the predator.
b_1, b_2: are the conversion factors denoting the number of newly born predators for each captured of first and second prey respectively.
m: is the half-saturation co-efficient.
$(\tau > 0)$: A discrete-time delay to allow for reaction time for the predator.

Further, they studied the conditions for the existence of non-negative equilibrium points and their local stability using linear stability theory. They also investigated the global stability of the equilibrium points as well as the persistence of the system. Finally, numerical simulation was carried out to verify the analytical results. The main finding from the numerical simulation was that time delay influences the stability of the interior equilibrium point in the view that there was a critical value for which it was stable.

1.5.8 Dynamics of a two prey and one predator system with time interruption and random fluctuations

Reddy et al. [42] studied a model consisting of one predator feeding on two competing prey with the presence of time delay and stochastic noise. The assumptions of the model as follows:

1. In the absence of the predator the prey population density grows, according to the logistic law of growth.
2. The two prey species compete for food resources.
3. The death rate of the predator in the absence of the prey species is $-dz$.
4. The intraspecific reaction between predators is taken into account and the functional response is taken as Holling type II.
5. There is a time delay τ in the system which appears in the predator equation only.

Based on the previous assumptions, they constructed the following model:

$$\begin{aligned}
\frac{dx(t)}{dt} &= r_1 x(1-\frac{x}{k_1}) - \alpha_{13}xz,\\
\frac{dy(t)}{dt} &= r_2 y(1-\frac{y}{k_2}) - \alpha_{23}yz,\\
\frac{dz(t)}{dt} &= -dz + -\alpha_{33}z^2 + \alpha_{31}x(t-\tau)z(t-\tau) + \alpha_{32}y(t-\tau)z(t-\tau),
\end{aligned}$$

$x(t), y(t), z(t)$: represent the population density of prey1, prey2 and predator species, respectively.
r_1, r_2: represent the intrinsic growth rates of prey1 and prey2, respectively.
k_1, k_2: represent the carrying capacities of prey1 and prey2, respectively.
α_{13}, α_{23}: represent the decrease rates of prey1 and prey2, respectively, due to predation.
α_{31}, α_{32}: represent the gain rates of the predator due to the predation of prey1 and prey2, respectively.

d: the mortality rate of the predator.
α_{33}: denote the decreased rate of the predator due to intraspecific competition.
τ: represents the time delay parameter.
All the parameters are assumed to constants.

They studied the local and global stability of the interior equilibrium points using linear theory along with the Routh-Hurwitz criterion and Lyapunov function. They also studied the effects of time delay and random environmental fluctuations on the stability of the model around the interior equilibrium point and finally numerical simulation to verify the analytical results for a set of chosen parameter values.

1.5.9 Analysis of Dynamics in Two-prey, one-predator model: Effect of the Remained carcass

Lee [30] has studied a model consisting of two prey and one predator in which one of the prey species h_1 is called existent prey and the other h_2 is called invader prey. After the predator p has consumed the existent prey, its carcass is considered as food for the invader prey. The following functional response of the existent prey to the benefit of remained carcasses is used :

$$\Psi(h_1, h_2, p) = \frac{\tau_1 h_2 p}{\tau_2 h_1 + \tau_3 h_2}$$

where τ_1, τ_2, τ_3 are positive constants.

Assumption of the model are as follows:

1. The existent preys have two negative effects by the intraspecific competition of themselves and the obstruction of invader preys, to have the remained carcass.
2. The two negative effects are proportional to the density of the existent prey and invader prey respectively, and to simplify the model, the real amount of remained carcasses which the existent prey has is proportional to the total amount of remained carcasses divided by the negative effects.
3. The existent prey and the predator have already approached to stable coexistence density before the invader prey invades.
4. The gain of the existent prey given by the predator is less than loss when the existent prey and the invader prey approach their carrying capacities, respectively.

Based on the previous assumption, he proposed the following model:

$$\begin{aligned}
\frac{dh_1(t)}{dt} &= \epsilon_1 h_1 (1 - \frac{h_1}{k_1}) - a_1 h_1 p + \frac{\tau_1}{\tau_2 h_1 + \tau_3 h_3} h_1 h_2 p, \\
\frac{dh_2(t)}{dt} &= \epsilon_2 h_2 (1 - \frac{h_2}{k_2}) - a_2 h_2 p \\
\frac{dp(t)}{dt} &= -\delta p + b_1 h_1 p + b_2 h_2 p,
\end{aligned}$$

where

ϵ_i: the growth rate of preys in the absence of a predator.

k_i: The carrying capacity of preys

a_i: The consumption rate of preys by predator

b_i: The rate of increase in the number of predators from ingesting the prey.

δ: The death rate of the predator.

Further, the linear stability analysis was performed and there was a unique interior equilibrium point under some condition on the parameter. Numerical simulation was done to study the stability of the interior equilibrium point as well as the effect of the remained carcass on the stability and persistence of the system. They showed that for the set of parameters satisfying the persistence condition, the system had a chaotic attractor.

1.5.10 Two Predators Competing for Two Prey Species: An Analysis of MacArthur's Model

S.B.Hsu and S.P.Hubbell [20] studied the MacArthur model which describes two predators feeding on two prey species.

The assumption of the model are as follows:

1. In the absence of predation, the growth rates of prey species are logistic and the linear functional response for predation is used
2. The two predator species x_1, x_2 prey on either prey species ,R_1, R_2 .
3. The two predator species compete only by lowering the population of shared prey with no interference between rivals.
4. Death rates are assumed to follow a “type III” survivorship in which the number dying is proportional to the number currently alive.
5. No significant time lags in the system.

Based on the previous assumption, they proposed the following model:

$$\begin{aligned}
\frac{dR_1(t)}{dt} &= r_1R_1(1-\frac{R_1}{K_1}) - k_{11}R_1x_1 - k_{21}R_1x_2, \quad R_1 > 0,\\
\frac{dR_2(t)}{dt} &= r_2R_2(1-\frac{R_2}{K_2}) - k_{12}R_2x_1 - k_{22}R_2x_2, \quad R_2 > 0,\\
\frac{dx_1(t)}{dt} &= -D_1x_1 + b_{11}x_1R_1 + b_{12}x_1R_2, \quad x_1 > 0,\\
\frac{dx_2(t)}{dt} &= -D_2x_2 + b_{21}x_2R_1 + b_{22}x_2R_2, \quad x_2 > 0
\end{aligned}$$

where $R_i(t)$ the number of ith prey at time t, $x_i(t)$ the number of ith predator at time t, r_i the intrinsic rate of increase for ith prey, K_i the carrying capacity for ith prey, b_{ij}: the birth rate per predator (predator species i) per unit prey (prey species j) consumed, k_{ij}: the feeding rate per predator (predator species i) per unit prey (prey species j) consumed, and D_i the death rate of ith predator.

Further, they studied the conditions for global stability of non-negative equilibrium points, especially the case in which the two-predator, two-prey system catastrophically collapses to a one-predator, two-prey system, or even to a one-predator, one-prey system.

Chapter 2

The mathematical model

The model consist of four competing prey species which are

1. Coliform Bacteria (E. coli)
2. Campylobacter
3. Clostridium perfringens
4. Intestinal enterococci,

nutrients and two competing predators

1. protozoa
2. Daphnia

The bacteria feed on nutrients. Daphnia and protozoa both feed on bacteria, but Daphnia in addition can feed on Protozoa, thus we have multistage predation in the system. Furthermore, we denote the prey species as $x_{i,i=1,2,3,4}$, the predators as $y_{j,j=1,2}$ and nutrients as N
The following diagram demonstrate the system and the interactions between the species:

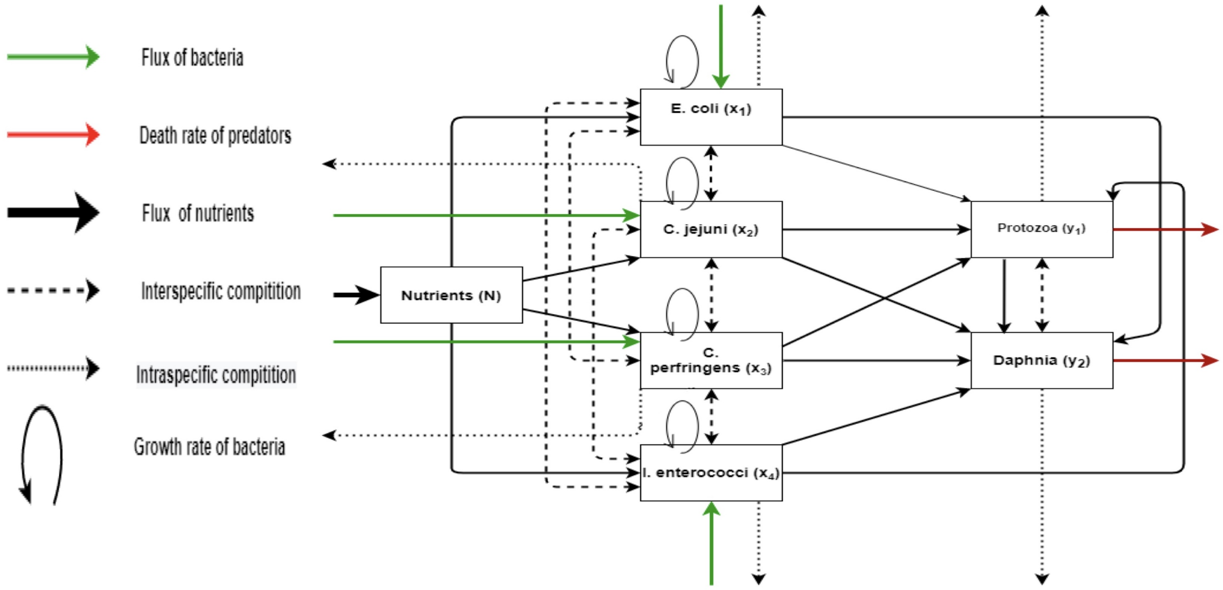


Figure 2.1: Flow diagram of the system 2.1. The continuous arrows represent the input to the system or output from the system (flux of bacteria or nutrients, predation, death rate of predators). The dashed and dotted arrows refer to the interspecific and intraspecific competition between species, respectively. Finally, the looped arrows represent the growth rate of bacteria.

2.1 Assumptions

1. There is an inflow of nutrients into the system which may be constant or time-dependent. The amount of nutrients reduces only due to consumption by bacteria.
2. In the absence of predators and interspecific competition, the bacteria will grow according to the logistic equation.
3. the four prey species compete for nutrients and this is incorporated into the equation by the mixed terms.
4. One predator, Daphnia prey on the other one (Protozoa), and they compete for other prey species.
5. For simplicity, linear preys dependent functional response will be considered here.
6. The predators have no preference when choosing their prey.

7. The concentration of pathogenic bacteria (E.coli, Intestinal enterococci, Campylobacter, Clostridium perfringens) in surface water at any time depends on the fecal contaminant input, thus we add a source term s_i.

Based on the previous assumption, we propose the following model:

2.2 The model

$$
\begin{aligned}
\frac{dN(t)}{dt} &= R(t) - c_1x_1 - c_2x_2 - c_3x_3 - c_4x_4\\
\frac{dx_1(t)}{dt} &= r_1x_1 - \sigma_{11}x_1x_1 - \sigma_{12}x_1x_2 - \sigma_{13}x_1x_3 - \sigma_{14}x_1x_4 - f_{11}x_1y_1 - f_{12}x_1y_2 + s_1(t),\\
\frac{dx_2(t)}{dt} &= r_2x_2 - \sigma_{22}x_2x_2 - \sigma_{21}x_2x_1 - \sigma_{23}x_2x_3 - \sigma_{24}x_2x_4 - f_{21}x_2y_1 - f_{22}x_2y_2 + s_2(t),\\
\frac{dx_3(t)}{dt} &= r_3x_3 - \sigma_{33}x_3x_3 - \sigma_{31}x_3x_1 - \sigma_{32}x_3x_2 - \sigma_{34}x_3x_4 - f_{31}x_3y_1 - f_{32}x_3y_2 + s_3(t),\\
\frac{dx_4(t)}{dt} &= r_4x_4 - \sigma_{44}x_4x_4 - \sigma_{41}x_4x_1 - \sigma_{42}x_4x_2 - \sigma_{43}x_4x_3 - f_{41}x_4y_1 - f_{42}x_4y_2 + s_4(t),\\
\frac{dy_1(t)}{dt} &= -\epsilon_1y_1 - \omega_{11}y_1^2 - (g_{12} + \omega_{12})y_1y_2 + \beta_{11}f_{11}y_1x_1 + \beta_{21}f_{21}y_1x_2 + \beta_{31}f_{31}y_1x_3 + \beta_{41}f_{41}y_1x_4,\\
\frac{dy_2(t)}{dt} &= -\epsilon_2y_2 - \omega_{22}y_2^2 + (\gamma_{12}g_{12} - \omega_{21})y_1y_2 + \beta_{12}f_{12}y_2x_1 + \beta_{22}f_{22}y_2x_2 + \beta_{32}f_{32}y_2x_3 + \beta_{42}f_{42}y_2x_4,
\end{aligned}
$$

with initial conditions $N(0) > 0, x_{1..4}(0) > 0, y_1(0) > 0, y_2(0) > 0$.

After some manipulation we obtain:

$$
\begin{aligned}
\frac{dN(t)}{dt} &= R(t) - c_1x_1 - c_2x_2 - c_3x_3 - c_4x_4, \quad N(0) > 0\\
\frac{dx_1(t)}{dt} &= x_1\left[r_1\left(1 - \frac{\eta_{11}x_1 + \eta_{12}x_2 + \eta_{13}x_3 + \eta_{14}x_4}{N}\right) - f_{11}y_1 - f_{12}y_2\right] + s_1(t), \quad x_1(0) > 0,\\
\frac{dx_2(t)}{dt} &= x_2\left[r_2\left(1 - \frac{\eta_{21}x_1 + \eta_{22}x_2 + \eta_{23}x_3 + \eta_{24}x_4}{N}\right) - f_{21}y_1 - f_{22}y_2\right] + s_2(t), \quad x_2(0) > 0,\\
\frac{dx_3(t)}{dt} &= x_3\left[r_3\left(1 - \frac{\eta_{31}x_1 + \eta_{32}x_2 + \eta_{33}x_3 + \eta_{34}x_4}{N}\right) - f_{31}y_1 - f_{32}y_2\right] + s_3(t), \quad x_3(0) > 0\\
\frac{dx_4(t)}{dt} &= x_4\left[r_4\left(1 - \frac{\eta_{41}x_1 + \eta_{42}x_2 + \eta_{43}x_3\eta_{44}x_4}{N}\right) - f_{41}y_1 - f_{42}y_2\right] + s_4(t), \quad x_4(0) > 0,\\
\frac{dy_1(t)}{dt} &= y_1\left[-\epsilon_1 - \omega_{11}y_1 - \psi_{12}y_2 + \beta_{11}f_{11}x_1 + \beta_{21}f_{21}x_2 + \beta_{31}f_{31}x_3 + \beta_{41}f_{41}x_4\right], \quad y_1(0) > 0,\\
\frac{dy_2(t)}{dt} &= y_2\left[-\epsilon_2 + \phi_{12}y_1 - \omega_{22}y_2 + \beta_{12}f_{12}x_1 + \beta_{22}f_{22}x_2 + \beta_{32}f_{32}x_3 + \beta_{42}f_{42}x_4\right], \quad y_2(0) > 0.
\end{aligned}
\tag{2.1}
$$

Where

$$
\begin{aligned}
i,k =&1,2,3,4 \quad , \quad j,l = 1,2, \quad K_i = \frac{r_i}{\sigma_{ii}}, \quad \phi_{12} = g_{12} + \omega_{12}, \\
\psi_{12} =&\gamma_{12}g_{12} - \omega_{21}, \quad K = \eta_{ii}K_i, \quad K = mN, \quad \eta_{ik} = \frac{\sigma_{ik}}{\sigma_{ii}}\eta_{ii},
\end{aligned}
$$

N : represent the concentration of the nutrients in the surface water.

x_i : represent the population density of E.coli, Intestinal enterococci, Campylobacter, and Clostridium perfringens respectively.

y_j : represent population density of protozoa and Daphnia at time t respectively

$R(t)$: the net flow of nutrients into the water resource.

c_i : The decreased rate of nutrients du to consumption by bacteria species x_i r_i: is the intrinsic growth rate for prey species.

s_i : the amount of bacteria species i can be added to the system through fecal contamination.

K_i : the environmental carrying capacity for ith prey species.

σ_{ik}: are inter-specific competition coefficient between the prey species i and k.

$\omega_{jl,j\neq l}$: are interspecific competition coefficient of the two predator species i.e. the decreased rate of the predator species j due interspecific competition with the predator species l

ω_{jj} : denote the decreased rate of the predator due to intraspecific competition.

β_{ij}, γ_{12} : are the conversion factors denoting the birth rate per predator (predator species j) per unit prey (prey species i) consumed.

ϵ_j : is the death rate of the predator j in the absence of prey species.

f_{ij} : represent the decreased rate of prey species i due to predation by predator species j.

g_{12}:represent the decreased rate of predator species 1 due to predation by predator species 2.

2.2.1 Units of the parameters and variables in the system

The units of variables and parameters in the system are given in the following table:

N $\frac{[mass\ N]}{[vol]}$	x_i $\frac{[num\ x_i]}{[vol]}$	y_j $\frac{[num\ y_i]}{[vol]}$	$R(t)$ $\frac{[mass\ N]}{[vol][time]}$
c_i $\frac{[mass\ N]}{[num\ x_i][time]}$	r_i $\frac{1}{[time]}$	σ_{ik} $\frac{[vol]}{[num\ x_k][time]}$	η_{ik} $\frac{[mass\ N]}{[num\ x_k]}$
K_i $\frac{[num\ x_i]}{[vol]}$	K $\frac{[mass\ N]}{[vol]}$	f_{ij} $\frac{[volume]}{[num\ y_j][time]}$	g_{12} $\frac{[vol]}{[num\ y_2][time]}$
s_i $\frac{[num\ x_i]}{[vol][time]}$	ϵ_j $\frac{1}{[time]}$	$\omega_{jl_{,j\neq l}}$ $\frac{[vol]}{[num\ y_l][time]}$	ω_{jj} $\frac{[vol]}{[num\ y_j][time]}$
γ_{12} $\frac{[num\ y_2]}{[num\ y_1]}$	β_{ij} $\frac{[num\ y_j]}{[num\ x_i]}$	$\frac{dN(t)}{dt}$ $\frac{[mass\ N]}{[vol][time]}$	$\frac{dx_i(t)}{dt}$ $\frac{[num\ x_i]}{[vol][time]}$
$\frac{dy_j(t)}{dt}$ $\frac{[num\ y_j]}{[vol][time]}$	mass $\frac{mg}{100\ ml}$	vol $100\ ml$	time day

Table 2.1: Units of the parameters and variables in the system 2.1.

Chapter 3

Theoretical background

In this chapter, we will give some theoretical background to understand this work, which includes stability analysis, linear algebra, sensitivity analysis, observability and chaos in dynamical systems.

3.1 Stability analysis of a dynamical system

Definition 3.1.1 (dynamical system). A dynamical system is a system that describes the evolution of a physical system over time and can be given by an ordinary differential equation of the form: $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t)$. If the time does not appear explicitly on the right-hand side, then the system is said to be autonomous.

Definition 3.1.2 (Equilibrium point). An equilibrium point of the system $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{F}(\boldsymbol{x}, t)$ is a point for which $\boldsymbol{F}(\boldsymbol{x}, t) = 0$, i.e. the system is in a steady state at that point. Thus, if the system starts at it will stay there as times goes to infinity.

Definition 3.1.3 (Lyapunov stability). An equilibrium point x_e is called Lyapunov stable if for every $\epsilon > 0$, there exists a $\delta > 0$ such that , for all x_0 with $\|\boldsymbol{x_0} - \boldsymbol{x_e}\| < \delta$ we have $\|\boldsymbol{x}(t) - \boldsymbol{x_e}\| < \epsilon$ for all $t \geq 0$. This mean that all solutions that start close enough to the equilibrium point will remain near it for $t \geq 0$. An equilibrium point is called unstable if it is not Lyapunov stable.

Definition 3.1.4 (Asymptotic stability). An equilibrium point x_e is called asymptotically stable if there exists a $\delta > 0$ such that , for all x_0 with $\|\boldsymbol{x_0} - \boldsymbol{x_e}\| < \delta$ we have $\lim_{t\to\infty} \|\boldsymbol{x}(t) - \boldsymbol{x_e}\| = 0$. This mean that all solutions that start close enough to the equilibrium point will converge to it as time tends to infinity.

Definition 3.1.5 (Invariant set). A set $A \subseteq \mathbb{R}^n$ is called invariant with respect to a dynamical system $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t)$ if for every trajectory $\boldsymbol{x}$ we have $\forall \boldsymbol{x_0} \in A \Rightarrow\ \boldsymbol{x}(t) \in A$ for all $t \in \mathbb{R}$. If we restrict t to nonnegative values, then it will be called positive invariant set. Consequently, if a trajectory enters or starts in A, it will stay in A for all times. Equilibrium points and limit cycles are examples of invariant sets.

Definition 3.1.6 (Basin of attraction). The basin of attraction of an equilibrium point $B(x_e)$ is the set of points in the domain of the solution that converge to it as times tend to infinity.

Definition 3.1.7 (Global stability). An equilibrium point is called globally stable if all solutions of the system converge to it as the time tend to infinity.

3.1.1 Stability of one dimensional ODE

Let us assume that we have an ODE in the following general form $\frac{dx}{dt} = f(x)$. We linearize it as follows: Set $\epsilon = x - x^*$ where ϵ is a small perturbation then

$$x = x^* + \epsilon \Rightarrow \frac{d\left(x^* + \epsilon\right)}{dt} = f\left(x^* + \epsilon\right)$$

$$\left.\begin{aligned} \frac{dx^*}{dt} + \frac{d\epsilon}{dt} &= f\left(x^*\right) + \epsilon f'(x^*) + O(\epsilon^2) \\ \frac{dx^*}{dt} &= 0 \text{ since } x^* \text{ is constant} \\ f(x^*) = 0 \quad &\text{since } x^* \text{ is an equilibrium point} \end{aligned}\right\} \Rightarrow \frac{d\epsilon}{dt} = \epsilon f'(x^*) \Rightarrow \epsilon = \epsilon_0 e^{f'(x^*)t}$$

Thus x^* will be stable if $f'(x^*) < 0$ since $\epsilon \to 0$ as $t \to \infty$ which implies that $x(t) \to x^*$as $t \to \infty$ and unstable if $f'(x^*) > 0$ since $\epsilon \to \infty$ as $t \to \infty$ which implies that $x(t)$ will not converge to x^* as $t \to \infty$.

Usually, we study the sign of eigenvalues when we investigate stability. For one dimensional ODE, the eigenvalues will be the derivative of the right-hand side at equilibrium points, and we get the same conclusion about stability, i.e. if the eigenvalues for a given equilibrium point are negative then the equilibrium point will be stable, and if they are positive it will be unstable.

The dynamical systems can be divided into two types, linear and nonlinear:

3.1.2 Stability of linear systems

Here we will just discuss the stability of linear homogeneous autonomous systems. Consider the following linear autonomous system with constant coefficients:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}$$

where A is a $n \times n$ matrix of coefficients and $\boldsymbol{x}(t) \in R^n$ is the vector containing the solutions of the system.

3.1.2.1 Stability of a 2D linear system

A two-dimensional linear autonomous system of ODE is given by:

$$\dot{x}(t) = ax + b$$
$$\dot{y}(t) = cx + d$$

The system can be written in matrix form as:

$$\begin{bmatrix} \dot{x}(t) \\ \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} a & b \\ \\ c & d \end{bmatrix} \begin{bmatrix} x \\ \\ y \end{bmatrix} \Leftrightarrow \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}$$

where $A = \begin{bmatrix} a & b \\ \\ c & d \end{bmatrix}$ and $\dot{\boldsymbol{x}} = \begin{bmatrix} x \\ \\ y \end{bmatrix}$

If the matrix A is invertible, then the system will have only one equilibrium point which is $\boldsymbol{x} = 0$. Furthermore, if the matrix A has two eigenvectors λ_1 and λ_2, then the classification of the zero equilibrium point will be as follows:

1. $\lambda_1 > \lambda_2 > 0$ implies that the equilibrium point is unstable and is called an unstable node.
2. $\lambda_1 = \lambda_2$ >0 then the equilibrium point will be unstable. Furthermore, it will be called a proper node (star node) if there are two distinct eigenvectors or an improper node (degenerate node) if there is only one independent eigenvector.
3. $\lambda_1 < \lambda_2 < 0$ implies that the equilibrium point is asymptotically stable and called a stable node.

4. $\lambda_1 = \lambda_2 < 0$ implies that the equilibrium point is asymptotically stable and called a node. Furthermore, it will be called a proper node (star node) if there are two distinct eigenvectors or an improper node (degenerate node) if there is only one independent eigenvector.

5. If one of the eigenvalues is positive, and the other is negative, then the equilibrium point will be unstable and is called a saddle point.

6. λ_1, λ_2 are complex with a positive real part, then the equilibrium point will unstable and it is called a spiral point.

7. λ_1, λ_2 are complex with a negative real part, then the equilibrium point will be asymptotically stable and it is called a spiral point.

8. λ_1, λ_2 are complex with zero real part then the equilibrium point will be stable and it is called a center.

3.1.2.2 Stability of an n-dimensional linear autonomous system

Let us assume that we have the following n-dimensional linear autonomous system in vector form:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}$$

where A is $n \times n$ matrix of coefficients and $\boldsymbol{x}(t) \in R^n$ is the vector containing the solutions of the system. Then the classification of equilibrium points will be as follows:

1. If the matrix A has negative real eigenvalues or complex eigenvalues with negative real parts, then the equilibrium point will be asymptotically stable.

2. If the matrix A has at least one positive real eigenvalue or one complex eigenvalue with a positive real part, then the equilibrium point will be unstable.

3.1.3 Stability of a nonlinear system of ODE

3.1.3.1 Stability of a 2-dimensional nonlinear system

Let us assume that we have a nonlinear system of ODE of the form:

$$\begin{aligned} \dot{x}(t) &= f(x, y) \\ \dot{y}(t) &= g(x, y) \end{aligned} \tag{3.1}$$

and suppose that $EP(x^*, y^*)$ is an equilibrium point, i.e. $f(x^*, y^*) = 0$ and $g(x^*, y^*) = 0$

The first step is to linearize the system around the equilibrium point as follows:

Let $\epsilon = x - x^*, \eta = y - y^*$ be a disturbance from the equilibrium point. Following a similar procedure, as in the one-dimensional case, we get the differential equations for the disturbance as follows:

$$\begin{aligned}\dot{\epsilon} &= \epsilon\frac{\partial f}{\partial x} + \eta\frac{\partial f}{\partial y}\\ \dot{\eta} &= \epsilon\frac{\partial g}{\partial x} + \eta\frac{\partial g}{\partial y}\end{aligned}$$

which can be written as

$$\begin{bmatrix}\dot{\epsilon}\\ \\ \dot{\eta}\end{bmatrix} = \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\\ \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}\end{bmatrix}\begin{bmatrix}\epsilon\\ \\ \eta\end{bmatrix}$$

The matrix

$$\begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\\ \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}\end{bmatrix}_{EP}$$

is the Jacobian matrix which is an analog of $f(x^*)$ in case of the one dimensional ODE. Since we have neglected the quadratic terms in the derivation, we obtain a linearized system and classifying equilibrium points will be similar as for the linear system except for the case where we have complex eigenvalues with zero real part in which the stability is undetermined.

3.1.3.2 Limit cycles in 2-dimensional nonlinear system

Definition 3.1.8. A limit cycle is an isolated closed trajectory, i.e. the neighboring trajectories are not closed; they spiral either toward or away from the limit cycle. The limit cycle is called stable or attracting if all the neighboring trajectories approach it, unstable if the neighboring trajectories approach it from the interior of the limit cycle and spiral out from it from the exterior of it. The limit cycle may be also semi-stable if the nearby trajectories approach it from the interior and spiral out from it from the exterior or the opposite. [55]

There are several methods to prove the nonexistence of a limit cycle, and here we will state some of them:

1. If we can write our system as a gradient system which is of the form $\dot{\boldsymbol{x}} = -\boldsymbol{\nabla} V(\boldsymbol{x})$ where V is a continuously differentiable scalar function, then there can't be a limit cycle.

2. If we can find a Lyapunov function for the system, i.e. the equilibrium point is globally stable, then the system has no closed orbits. More about Lyapunov function in the global stability section.

3. Bendixson–Dulac theorem which can be stated as follows: Let D be a simply connected region (i.e. no holes) in R^2 on winch, the system 3.1 is defined and if there exists a continuously differentiable $V(\boldsymbol{x})$ such that $\frac{\partial(Vf)}{\partial x} + \frac{\partial(Vg)}{\partial y}$ has the same sign throughout D, then there is no limit cycle that can lie entirely in D.

4. Critical point criteria: A closed trajectory has a critical point in its interior. Thus if a region D is simply connected (i.e. no holes) in R^2 and has no critical (equilibrium) points, then it cannot contain any limit cycles.

3.1.3.3 Stability of an n-dimensional nonlinear system

Consider the following n-dimensional nonlinear system of ODE in vector form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x})$$

To study the local stability of the previous system we linearize it around an equilibrium point and obtain the following linearized system :

$$\dot{\boldsymbol{\epsilon}} = \boldsymbol{J}_{\boldsymbol{x}^*}\, \boldsymbol{\epsilon}$$

where $\boldsymbol{J}_{\boldsymbol{x}^*}$ is the Jacobian matrix evaluated at the equilibrium point. Then the classification of the equilibrium points will be as follows:

1. If the Jacobian matrix evaluated at the equilibrium point has negative real eigenvalues or complex eigenvalues with a negative real part, then the equilibrium point will be asymptotically stable.

2. If the Jacobian matrix evaluated at the equilibrium point has at least one positive real eigenvalue or one complex eigenvalue with a positive real part, then the equilibrium point will be unstable.

3. If at least one eigenvalue of the Jacobian matrix evaluated at the equilibrium point has zero real part and others have a negative real part, then it is not possible to determine the stability by this method.

3.1.4 Routh-Hurwitz criterion for stability

Sometimes it is difficult to calculate the roots of the characteristic polynomial of a linear system or linearized system. Routh-Hurwitz provides a criterion for the stability of a linear system without calculation of the roots explicitly. In this section, we provide the Routh-Hurwitz criterion for a two-dimensional and three-dimensional dynamical system.

Let us assume we have the following characteristic equation for an n-dimensional linear system:

$$P(\lambda) = a_n\lambda^n + a_{n-1}\lambda^{n-1} + a_{n-2}\lambda^{n-2} + \ldots + a_1\lambda + a_0 = 0$$

We first construct a Routh array as follows:

$$\begin{array}{r|cccc|}
\lambda^n & a_n & a_{n-2} & a_{n-4} & \cdots \\
\lambda^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\
\lambda^{n-2} & b_{n-1} & b_{n-3} & b_{n-5} & \cdots \\
\lambda^{n-3} & c_{n-1} & c_{n-3} & c_{n-5} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \cdots \\
\lambda^0 & h_{n-1} & & &
\end{array}$$

where

$$b_{n-1} = -\frac{1}{a_{n-1}}\begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix} \quad , \quad b_{n-3} = -\frac{1}{a_{n-1}}\begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}, \ldots$$

$$c_{n-1} = -\frac{1}{b_{n-1}}\begin{vmatrix} a_{n-1} & a_{n-3} \\ b_{n-1} & b_{n-3} \end{vmatrix} \quad , \quad c_{n-3} = -\frac{1}{b_{n-1}}\begin{vmatrix} a_{n-1} & a_{n-5} \\ b_{n-1} & b_{n-5} \end{vmatrix}, \ldots$$

The Routh-Hurwitz criterion states that the number of roots with a positive real part is equal to the number of the sign changes in the first column of the Routh array. Thus the system will be stable if all the coefficients in the first column of the Routh array are of the same sign.

3.1.4.1 Two dimensional linear system

The characteristic equation for a two-dimensional linear system is given by:

$$a_2\lambda^2 + a_1\lambda + a_0 = 0$$

The Routh array becomes:

$$\begin{array}{c|cc|} \lambda^2 & a_2 & a_0 \\ \lambda^1 & a_1 & 0 \\ \lambda^0 & b_1 & 0 \end{array}$$

where

$$b_1 = -\frac{1}{a_1}\begin{vmatrix} a_2 & a_0 \\ a_1 & 0 \end{vmatrix} = a_0$$

Thus, the condition for stability of a $2D$ system is that all the coefficients should have the same sign.

3.1.4.2 Three dimensional linear system

The characteristic equation for a two-dimensional linear system is given by:

$$a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$

The Routh array becomes:

$$\begin{array}{c|cc|} \lambda^3 & a_3 & a_1 \\ \lambda^2 & a_2 & a_0 \\ \lambda^1 & b_1 & 0 \\ \lambda^0 & c_1 & 0 \end{array}$$

where

$$b_1 = -\frac{1}{a_2}\begin{vmatrix} a_3 & a_1 \\ a_2 & a_0 \end{vmatrix} = \frac{a_1a_2 - a_0a_0}{a_3} \quad , \quad c_1 = -\frac{1}{b_1}\begin{vmatrix} a_2 & a_0 \\ b_1 & 0 \end{vmatrix} = a_0$$

Thus, the condition for stability of a 3D system is that all the coefficients should be positive and $a_1a_2 > a_0a_3$. For special cases and more details, see [13] and [43]

3.2 Global stability and Lyapunov direct method

In this section, we will go through some method and theorems for global stability of dynamical systems

Definition 3.2.1 (Positive definite functions). Let W be a region of the phase space containing the equilibrium point x^* and let $V(\boldsymbol{x}) : W \subseteq \mathbb{R}^n \to \mathbb{R}$ be a continuous and differentiable function. The function V is called positive definite [positive semi-definite] if the following conditions are satisfied:

1. $V(\boldsymbol{x}^*) = 0$
2. $V(\boldsymbol{x}) > 0 \quad [V(\boldsymbol{x}) \geq 0] \quad \forall\, x \in W \setminus \{\boldsymbol{x}^*\}$

Theorem 3.2.1 (Lyapunov direct method for stability). *Let us consider the following autonomous nonlinear system of ODE:*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x(t)})$$

and let $\boldsymbol{x}^*$ *be an equilibrium point of the system. Suppose that there exist a real-valued positive definite function* $V(\boldsymbol{x}) : W \subseteq \mathbb{R}^n \to \mathbb{R}$ *then:*

1. *if* $\dot{V}(\boldsymbol{x}) \leq 0 \quad \forall\, x \in W \setminus \{\boldsymbol{x}^*\}$ *then* x^* *is Lyapunov stable.*
2. *if* $\dot{V}(\boldsymbol{x}) < 0 \quad \forall\, x \in W \setminus \{\boldsymbol{x}^*\}$ *then* x^* *is asymptotically stable.*
3. *if* $\dot{V}(\boldsymbol{x}) \geq 0 \quad \forall\, x \in W \setminus \{\boldsymbol{x}^*\}$ *then* x^* *is unstable.*

Theorem 3.2.2 (Barbashin-Krasovskii-theorem). *Let us consider the following autonomous nonlinear system of ODE:*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x(t)})$$

and let $\boldsymbol{x}^*$ *be an equilibrium point of the system. Suppose that there exists a continuously differentiable real-valued positive definite function* $V(\boldsymbol{x}) : W \subseteq \mathbb{R}^n \to \mathbb{R}$*. Let* $S = \Big\{x \in W : \dot{V(\boldsymbol{x})} = \boldsymbol{0}\Big\}$ *and suppose that no other solution except the trivial solution* $\boldsymbol{x}(t) = 0$ *can stay in S, then* $\boldsymbol{x}^*$ *is asymptotically stable. If in addition* $V(\boldsymbol{x})$ *is radially unbounded i.e* $\|\boldsymbol{x}\| \to \infty \Rightarrow V(\boldsymbol{x}) = 0$ *then* $\boldsymbol{x}^*$ *is globally stable.*

3.2.1 Global stability for generalized Lotka–Volterra Systems

Let us consider the following generalized Lotka–Volterra competitive-predator-prey system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F(x, p)} = \boldsymbol{x}^T(r - A\boldsymbol{x}) \tag{3.2}$$

where $\boldsymbol{p}$ represent all the parameter in the system, $\boldsymbol{F} = [F_1, F2, F3, \ldots, F_n]^T$ is the right-hand side of the system, $\boldsymbol{x} = [x_1, x_2, x_3, \ldots x_n]^T$ is n dimensional state vector, $\boldsymbol{r} = [r_1, r_2, r_3, \ldots r_n]^T$ is the n-dimensional real vector that contain growth rates and death rates of the species involved in the interaction(The component that represent growth rates will be positive and the ones that represent death rates will be negative), and $A = (a_{ij})$ is the $n \times n$ interaction matrix that contain competition and predation coefficients which may be represented by using the concept of carrying capacity.

Theorem 3.2.3. *Suppose that the Lotka-Volterra system 3.2 has a unique interior equilibrium point* $\boldsymbol{x}^* = A^{-1}\boldsymbol{r}$*, then this equilibrium point will be globally stable on positive* $\mathbb{R}^n$ *if there exist a positive definite matrix* D *such that the symmetric matrix* $M = DA + A^TD$ *is positive definite.*

Proof. Let $V : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0} \to$ be defined as:

$$V(\boldsymbol{x}) = \sum_{i=1}^{n} p_i \left(x_i - x_i^* - x_i^* log\left(\frac{x_i}{x_i^*}\right)\right)$$

where $p_i \in \mathbb{R}$ are positive constants to be found.

The derivative of the function is given by:

$$\dot{V}(\boldsymbol{x}) = \boldsymbol{\nabla} V(\boldsymbol{x})\boldsymbol{F} = \left[\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, \frac{\partial V}{\partial x_3}, \dots \frac{\partial V}{\partial x_n}, \right] .[F_1, F2, F3, \dots, F_n]^T$$

which can be written as

$$\begin{aligned}\dot{V}(\boldsymbol{x}) &= -\left(\boldsymbol{x} - \boldsymbol{x}^*\right)^T \left(A^TD\right)\left(\boldsymbol{x} - \boldsymbol{x}^*\right) = -\boldsymbol{X^T}\left(\frac{1}{2}\left(A^TD + DA\right) + \frac{1}{2}\left(A^TD - DA\right)\right)\boldsymbol{X} \\ &= -\frac{1}{2}\boldsymbol{X}^T\left(DA + A^TD\right)\boldsymbol{X} - \underbrace{\frac{1}{2}\boldsymbol{X}^T\left(DA - A^TD\right)\boldsymbol{X}}_{=0} = -\frac{1}{2}\boldsymbol{X^T}M\boldsymbol{X}\end{aligned}$$

where

$$X = [x_1 - x_1^*, x_2 - x_2^*, x_3 - x_3^*, \vdots, x_n - x_n^*]^T \quad , \quad D = diag(p_1, p_2, p_3, \dots, p_n)$$

Here we used the idea that we can write A^TD as a sum of a symmetric and skew-symmetric matrices i.e.

$$A^TD = \frac{A^TD + \left[A^TD\right]^T}{2} + \frac{A^TD - \left[A^TD\right]^T}{2} = \frac{1}{2}\left(A^TD + DA\right) + \frac{1}{2}\left(A^TD - DA\right)$$

The function $V(\boldsymbol{x})$ satisfies the following:

1. $V(\boldsymbol{x}^*) = 0$
2. $V(\boldsymbol{x}) > 0 \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \setminus \{\boldsymbol{x}^*\}$

thus by Lyapunov function stability theorem the interior equilibrium point $\boldsymbol{x}^*$ will be globally stable if M is positive definite. □

3.2.2 Positive definite matrices

Definition 3.2.2 (Positive definite matrix). A real $n \times n$ symmetric matrix A is called positive (negative) definite if $\boldsymbol{x^T} A\boldsymbol{x} > 0 \quad \big(\boldsymbol{x^T} A\boldsymbol{x} < 0\big)$ for all nonzero vector $\boldsymbol{x} \in \mathbb{R}^n$. If the inequality is not strict then it will be positive (negative) semi-definite .

To prove that the matrix is positive definite we can use the following theorems.

Theorem 3.2.4 (Eigenvalues). *A symmetric matrix A is positive definite if all its eigenvalues are positive. The eigenvalues λ's can be found by solving the algebraic equation $det(A - \lambda I) = 0$*

Theorem 3.2.5 (Sylvester's criterion). *A symmetric matrix A is positive definite if all its leading principal minors are positive. The leading principals minors are all the upper left determinants of the matrix.*

Theorem 3.2.6 (Cholesky decomposition). *Every real symmetric positive definite matrix A has a unique Cholesky decomposition. A Cholesky decomposition of the matrix A is defined as $A = LL^T$ where L is a lower triangular matrix with real and positive entries.*

For further reading we refer to [32] and [54]

3.3 Chaos theory

In this section, we briefly present chaos theory and the essential property of a chaotic dynamical system.

3.3.1 Some definitions

Definition 3.3.1 (Chaos). Here we adopt a slightly modified version of the definition mentioned in the book by Strogatz [55] which is given as follows,

Chaos is aperiodic long-term behavior in a deterministic nonlinear bounded system that exhibits sensitive dependence on the initial conditions.

1. aperiodic long-term behavior means that there are trajectories that do not converge to equilibrium points, periodic orbits, or quasi-periodic orbits as time tends towards infinity. Consequently, the system does not have any stable equilibrium points.
2. nonlinear, i.e. the system cannot be written in the form $\dot{\boldsymbol{x}} = A\boldsymbol{x}$ for any matrix A.

3. bounded, i.e the trajectories do not tend to infinity, but bounded in a defined domain. Hence, the trajectory converges to an attractor which is called a strange attractor.
4. deterministic system means that it is not stochastic, i.e. the system has no random or noisy inputs or parameters. The irregular behavior arises from the non-linearity of the system rather than from noisy driving forces.
5. sensitive dependence on initial conditions means that two infinitesimally close trajectories separate exponentially fast, i.e. the system has a positive Lyapunov exponent.

Definition 3.3.2 (Attractor). In general terms, an attractor can be defined as a set to which all nearby trajectories converge and therefore the stable eqm. points and stable limit cycle considered as attractors. More precisely, it is defined as follows, An attractor is a closed set A that have the following properties,

1. A is invariant set: any trajectory $\boldsymbol{x}\left(\boldsymbol{t}\right)$ that starts in or inter A stays in A for all time.
2. A attracts an open set of initial conditions: there is an open set U containing A such that if $\boldsymbol{x_0} \in U$, then the distance from $\boldsymbol{x}\left(\boldsymbol{t}\right)$ to A tend to zero as $t \to \infty$. This means that A attracts all trajectories that start sufficiently close to it. The largest such U is called the basin of attraction.
3. A is minimal: there is no proper subset of A that satisfy conditions 1 and 2

Definition 3.3.3 (strange attractor). The strange attractor is defined to be an attractor that exhibits sensitive dependence to initial conditions.

3.3.2 Lyapunov exponent

The Lyapunov exponent is a quantity that characterizes the rate of separation of two infinitesimally close trajectories on an attractor. Mathematically, it is defined as follows,

Given two trajectories a and b in the phase space with initial separation $\boldsymbol{\delta_0}$. Suppose that $\boldsymbol{x}\left(\boldsymbol{t}\right)$ is a point on the attractor at time t, and consider a nearby point $\boldsymbol{x}\left(\boldsymbol{t}\right) + \boldsymbol{\delta_0}$, where $\boldsymbol{\delta_0}$ is a tiny separation vector of initial length $\|\boldsymbol{\delta_0}\| = 1 \times 10^{-8}, 1 \times 10^{-15}$. The growth of the initial separation is exponential and is given by

$$\|\boldsymbol{\delta}\left(t\right)\| = \|\boldsymbol{\delta_0}\| e^{\lambda(t)}$$

The quantity λ is called the Lyapunov exponent. The positive value of λ indicates that the system is sensitive to initial conditions, i.e. chaotic given that other properties are satisfied.

If $\lambda = 0$ the system exhibit periodicity and finally λ is negative, the system will converge to a stable equilibrium point. There are as many Lyapunov exponents as the number of state variables and conventionally $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$ where n the number of state variables. For a continuous $3D$ dynamical system, we have the flowing configurations [51],

λ_1	λ_2	λ_3	Type of attractor	dimension	Dynamic
negative	negative	negative	Equilibrium point	0	static
0	negative	negative	Limit cycle	1	periodic
0	0	negative	Attracting 2-torus	2	Quasiperiodic
0	0	0	Invariant torus	1 or 2	(Quasi)periodic
$+$	0	-	Strange	2 or 3	Chaotic

Table 3.1: Characteristics of the $3D$ bounded dynamical system

From the above table, we see that for chaos the largest Lyapunov exponent is positive, the second is zero, and the third is negative and the largest one is only positive in a chaotic system. Therefore to determine chaos it is usually enough to calculate the largest Lyapunov exponent which is defined as,

$$\lambda_1 = \lim_{t\to\infty} \lim_{\boldsymbol{\delta_0}\to 0} \frac{1}{t} \ln\left(\frac{\|\boldsymbol{\delta}\,(t)\|}{\|\boldsymbol{\delta_0}\|}\right)$$

and its positivity can assure that the system is chaotic.

Remark. The sum of Lyapunov exponents for a $3D$ equals the trace of the Jacobian matrix averaged over a long time, thus one can calculate the third Lyapunov exponent as follows

$$\sum_{i=0}^{3} \lambda_i =< tr\,(j) >\to \lambda_1 + 0 + \lambda_3 \to \lambda_3 = \sum_{i=0}^{3} \lambda_i - \lambda_1$$

In a chaotic system, it is no longer possible to predict the long future of the system and one can define a time horizon $t_{horizon}$ for which the evolution of the system is not predictable(see Figure 3.1). It is defined as follows [55],

Assume again that there are two initial conditions which are almost indistinguishable (with initial separation of order $\boldsymbol{\delta_0} = 1 \times 10^{-13}$). After some time t, the separation became $\|\boldsymbol{\delta}\,(t)\| = \|\boldsymbol{\delta_0}\| e^{\lambda(t)}$. Let a be a measure of our tolerance, i.e if a prediction is within a of the true initial state, we consider it acceptable. Then our prediction becomes intolerable when $\|\boldsymbol{\delta}\,(t)\| \geq a$; this occurs after a time

$$t_{horizon} \sim O\left(\frac{1}{\lambda}\ln\left(\frac{a}{\|\boldsymbol{\delta_0}\|}\right)\right)$$

As we can see the $t_{horizon}$ depends logarithmically on the initial separation $\|\boldsymbol{\delta_0}\|$, and that's why making a long-term prediction in chaotic systems is impossible.

For further reading the reader is referred to [53, 55]

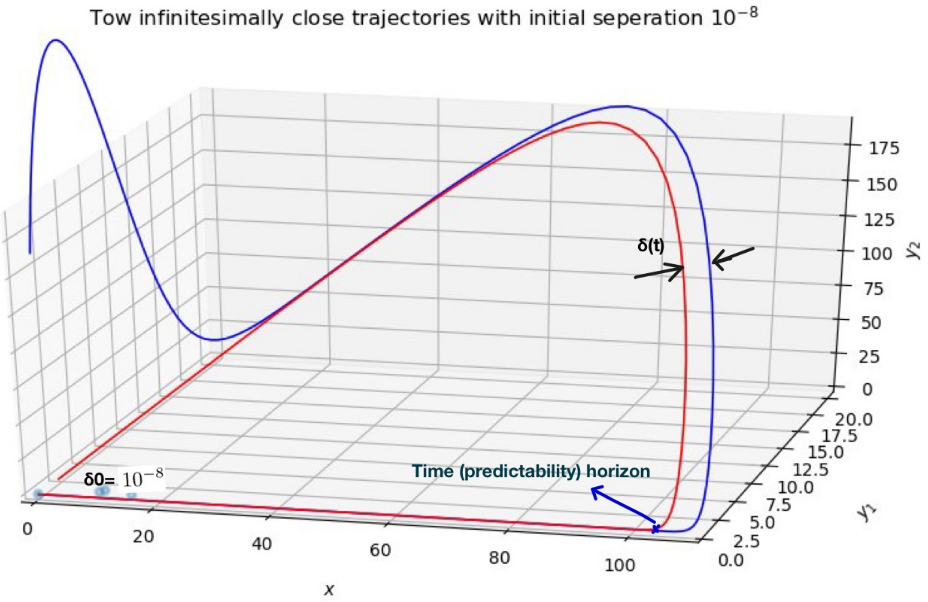


Figure 3.1: Sensitivity to initial condition and time horizon

3.4 Global sensitivity analysis

In this section, we will define sensitivity analysis and explain shortly some methods used in constructing the global sensitivity analysis of mathematical models.

Definition 3.4.1 (Sensitivity analysis). Mathematical models usually depend on a set of input factors which could be parameters or independent variables. These input factors are usually uncertain and vary in some intervals. The sensitivity analysis is the study of how much the change of each input factor contributes to the change of the model output. It is therefore used to determine which parameters have the most effect on the model output and which have a negligible effect. This is very useful in model improvements such as model reduction and parameter estimation. If, for example, a parameter has only a small effect on the model output, then we can safely set it to a value in its interval of variation and conduct parameter estimation only for the sensitive parameters.

Sensitivity analysis is divided into a local approach and a global approach. The local approach is a derivative based at some reference point, for example, an eqm. point of an ODE system. The global approach however is statistics-based, in the sense that all the input factors are varied simultaneously and the sensitivity is computed over the entire range of each input factor.

There are several methods to perform global sensitivity analysis (GSA) and here we present the most used ones,.

3.4.1 Sobol method

The Sobol method is a variance-based method for global sensitivity analysis. Let us assume that we have a mathematical model $Y = f(\boldsymbol{X})$ where $\boldsymbol{X}$ is a vector of k uncertain model input factors $\{X_1, X_2, ..., X_k\}$. To use this method the output of the model must be scalar, but we can study the sensitivity of a model with multiple outputs by doing separate sensitivity analysis for each output. We assume that the scalar function Y is square-integrable and defined over Ω, the k-dimensional unit hypercube,

$$\Omega = (X | 0 \leq x_i \leq 1; i = 1, 2, ..., k)$$

We further suppose that the input factors are uniformly distributed in the unit hypercube, i.e. $\boldsymbol{x_i} \in [0, 1]$. The steps for the variance based method is as follows:

- The function f can be decomposed as follows:

  $$f = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + ... + f_{12...k}$$

  where $f_i = f_i(X_i)$, $f_{ij} = f_{ij}(X_i, X_j)$ and so on for a total of 2^k terms, including f_0. Each term is square integrable over Ω. The condition for this decomposition is:

  $$\int_0^1 f_{i_1, i_2, ..., i_s}\left(X_{i_1}, X_{i_2}, ..., X_{i_s}\right) dX_{i_w}$$

  where $1 \leq i_1 < i_2 < ... < i_s < k$ and $i_w \in \{i_1, i_2, ..., i_s\}$. The functions $f_{i_1, i_2, ..., i_s}$ are obtained from:

$$\begin{aligned} f_0 =& E(Y), \\ f_i =& E_{\boldsymbol{x}\sim i}\left(Y|X_i\right) - E(Y), \\ f_{ij} =& E_{\boldsymbol{x}\sim ij}\left(Y|X_i, X_j\right) - f_i - f_j - E(Y), \end{aligned}$$

and similarly, for higher orders. Not that E is the expected value and the notation $\boldsymbol{x} \sim i$ means the set of all variables except $\boldsymbol{x}_i$.

- Relation between functions $f_{i_1,i_2,\dots,i_s}$ and partial variance is as follows:

$$\begin{aligned} V_i =& V\left(f_i\left(X_i\right)\right) = V_{X_i}\left[E_{\boldsymbol{x}\sim i}(Y|X_i)\right] \\ V_{ij} =& V\left(f_{ij}\left(X_i, X_j\right)\right) = V_{X_iX_j}\left[E_{\boldsymbol{x}\sim ij}(Y|X_i, X_j)\right] - V_{X_i}\left[E_{\boldsymbol{x}\sim i}(Y|X_i)\right] - V_{X_j}\left[E_{\boldsymbol{x}\sim j}(Y|X_j)\right] \end{aligned}$$

and so on for higher terms. All terms are linked by:

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \dots + V_{12\dots k}$$

Dividing both sides of equation by $V(Y)$, we obtain:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \dots + S_{12\dots k} = 1$$

- Sensitivity indices are:
  1. First order sensitivity index is given by:

     $$S_i = \frac{V_i}{V(Y)}$$

     and it measures the first order or additive effect of X_i on the model output. Note that $0 \leq \sum_{i=1}^{k} S_i \leq 1$ and it will equal unity only if all interaction terms are zero.
  2. Second-order sensitivity index is given by:

     $$S_{ij} = \frac{V_{ij}}{V(Y)}$$

     and it measures the second-order effect or the effect of interaction of X_i, X_j on the model output. Higher-order sensitivity indices can be computed equivalently.

3. Total sensitivity index is given by:

$$S_{T_i} = \frac{E_{\boldsymbol{X}\sim i}\left[V_{X_i}(Y|X_i)\right]}{V(Y)} = 1 - \frac{V_{\boldsymbol{X}\sim i}\left[E_{X_i}(Y|X \sim i)\right]}{V(Y)}$$

   where

   $E_{\boldsymbol{X}\sim i}\left[V_{X_i}(Y|X_i)\right]$: is the expected variance that would be left if all factors except X_i could be fixed.

   $V_{\boldsymbol{X}\sim i}\left[E_{X_i}(Y|X \sim i)\right]$: is the expected reduction in variance that would be obtained if all factors but X_i could be fixed.

   S_{T_i} measures the total effect, i.e. first and higher order effects (interaction) of input factor X_i. One way to visualize this is by considering that $V_{\boldsymbol{X}\sim i}\left[E_{X_i}(Y|X \sim i)\right]$ is the first order effect of $\boldsymbol{X} \sim i$, so that $V(Y) - V_{\boldsymbol{X}\sim i}\left[E_{X_i}(Y|X \sim i)\right]$ must give the contribution of all terms in the variance decomposition which do not include X_i. Note that $\sum_{i=1}^{k} S_{T_i} \geq 1$ and it will be equal unity only if all interaction terms are zero.

3.4.1.1 Computational cost

The method uses the Saltelli sampling approach and the number of samples is N then the number of model runs is $N(k+1)$ if the second-order indices are not computed and $N(2k+1)$ if the second-order indices are computed.

For more reading, see [17, 44, 45, 50]

3.4.2 Morris method

Morris's method [35] is OAT(One-factor-At-a-Time) method which is used to determine which input factor has effects which are (a) negligible, (b)linear and additive, or (c) nonlinear or involved in the interaction with other factors. Each input factor may assume a discrete number of values, called levels, which are chosen within the factor range of variation. For each input factor, two sensitivity indices are computed:

μ: the mean of the distribution F_i of the elementary effects of each input which measures the overall influence of the factor on the output.

σ : the standard deviation of the distribution F_i the elementary effects of each input which estimates the ensemble of the second -and higher-order effects in which the factor is involved, including curvatures and interaction effects.

3.4.2.1 Outline of the method

Let us assume that we have a mathematical model $y = f(\boldsymbol{X})$ where $\boldsymbol{X}$ is a vector of k uncertain model input factors $\{X_1, X_2, ..., X_k\}$. Assume that the k-dimensional vector $\boldsymbol{X}$ of the model input has components X_i each of which can assume integer values in the set $\left\{0, \frac{1}{p-1}, \frac{2}{p-1}, \frac{3}{p-1}, .., \frac{p-1}{p-1} = 1\right\}$. The region of experimentation Ω will be a k-dimensional p-level grid. Further, the factors are assumed to be uniformly distributed in $[0, 1]$ and then transformed from the unit hypercube to their actual distributions. The elementary effect for the ith input is defined as follows:

Let Δ be a predetermined multiple of $\frac{1}{p-1}$. For a given value $\boldsymbol{x}$ of $\boldsymbol{X}$, the elementary effect of the ith input factor is defined as

$$d_i(\boldsymbol{x}) = \frac{[y(x_1, ..., x_{i-1}, x_i + \Delta, x_{i+1}, ..., x_k) - y(\boldsymbol{x})]}{\Delta}$$

where $\boldsymbol{x} = (x_1, x_2, ..., x_k)$ is any selected value in Ω such that the transformed point $(\boldsymbol{x} + \boldsymbol{e}_i\Delta)$, where $\boldsymbol{e}_i$ is a vector of zeros but with a unit as its ith component, is still in Ω for each index $i = 1, 2, ..., k$. The finite distribution of elementary effects associated with the ith input factor is obtained by randomly sampling different $\boldsymbol{x}$ from Ω, and is denoted by F_i.

Original Morris design works as follows:

1. Start by randomly selecting a base value $\boldsymbol{x}^*$ for the vector $\boldsymbol{X}$. Each component x_i is sampled from the set $\left\{0, \frac{1}{p-1}, \frac{2}{p-1}, \frac{3}{p-1}, .., 1\right\}$. The vector $\boldsymbol{x}^*$ is used to generate other sampling points but it is not one of them. The model is never evaluated at $\boldsymbol{x}^*$.

2. The first sampling point $\boldsymbol{x}^{(1)}$ is obtained be increasing one or more components of $\boldsymbol{x}^*$ by quantity Δ while keeping other values at the base value.

3. The second sampling point $\boldsymbol{x}^{(2)}$ is generated from $\boldsymbol{x}^*$ with the property that it differs from $\boldsymbol{x}^{(1)}$ in its ith component that has either been increased or decreased by Δ i.e. $\boldsymbol{x}^{(2)} = x_1^{(1)}, ..., x_{i-1}^{(1)}, x_i^{(1)} \pm \Delta, x_{i+1}^{(1)}, ..., x_k^{(1)} = (\boldsymbol{x}^{(1)} \pm \boldsymbol{e}_i\Delta)$. The index i is randomly selected in the set $\{1, 2, ..., k\}$

4. The third sampling point $\boldsymbol{x}^{(3)}$ is generated from $\boldsymbol{x}^*$ with the property that it differ from $\boldsymbol{x}^{(2)}$ in only one component $j \neq i$ that has either been increased or decreased by Δ i.e. $\boldsymbol{x}_j^{(3)} = (\boldsymbol{x}_j^{(2)} \pm \boldsymbol{e}_j\Delta)$.

5. The design proceeds by producing a $(k + 1)$ sampling points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(k+1)}$ with the property that two consecutive points differ in only one component. Furthermore,

any component of the base vector $\boldsymbol{x}^*$ is selected at least once to be increased by Δ to calculate the elementary effect for each factor.

6. The succession of sampling points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(k+1)}$ define a trajectory in the input space. It also defines a design matrix $B^{\star}$, with dimensions $(k+1)\times k$, whose rows are the vectors $\boldsymbol{x}^{(l)}, \boldsymbol{x}^{(l)} \ldots \boldsymbol{x}^{(l+1)}$, and is called orientation matrix. Once a trajectory has been constructed and the model evaluated at its points, an elementary effect for each factor $i, i = 1, 2, .., k$ can be computed. If $\boldsymbol{x}^{(l)}$ and $\boldsymbol{x}^{(l+1)}$,where $l \in \{1, 2, ..., k\}$, are two sampling points differing in their ith component, the elementary effect associated with the factor i is
$$d_i(\boldsymbol{x}^l) = \frac{\left[y(\boldsymbol{x}^{l+1}) \pm y(\boldsymbol{x}^l)\right]}{\Delta}$$

7. The procedure is repeated r times with different base vector value each time which result in $r\,(k+1)$ runs, where k is the number of parameters or input variables. r is usually chosen in the interval $[10, 50]$.

Campanologo et al. [9] suggested an enhancement of the sampling strategy to better scan the input domain without increasing the number of model evaluations. The idea is to select the r trajectories in such a way to maximize their dispersion in the input space. We start by generating a high number of Morris trajectories $M = 500 - 1000$, and then we choose for examples, $r = 10$ with the highest spread. He also proposed a third sensitivity measure μ^* which is the estimate of the mean of the distribution of the absolute values of the elementary effects G_i i.e. $|d_i(\boldsymbol{x})| \sim G_i$. The use of μ^* solves the problem of the effects of opposite signs which occur when the model is non-monotonic. The drawback is the loss of information on the sign of the effect. Nevertheless, this information can be recovered by the simultaneous examination of μ and μ^*, as an estimate of μ comes at no extra computational cost.

If μ and μ^* are both high, the sign of the effect is always the same, i.e. the output function is monotonic with respect to that factor. If, in contrast, μ is low while μ^* is high, the factor carries the effects of different signs, depending on the values assumed by the other factors. μ^* also provides an answer to the screening problem by identifying the subset of non-influential factors in the sense that they can be fixed to any value within their ranges of uncertainty without significantly affecting the model outcomes. However, to examine the effects due to interactions, we still use the μ measure proposed by Morris and consider the standard deviation of the distribution F_i.

The formula for computing μ and σ is given by the following expressions:

$$\mu_i = \frac{1}{r}\sum_{j=1}^{r} d_{i,j}$$

$$\sigma_i = \sqrt{\frac{1}{r-1}\sum_{j=1}^{r} \left(d_{i,j} - \mu_i\right)^2}$$

$$\mu_i^\star = \frac{1}{r}\sum_{j=1}^{r} |d_{i,j}|$$

3.4.2.2 Interpretations of the sensitivity results

To properly rank factors, one must consider the values of both μ and σ at the same time, since a factor with elementary effects of different signs (i.e. which cancel each other out) would have a low value of μ but a considerable value of σ . The higher value of μ, the greater the influence of the corresponding parameters on the model output. A higher value of σ indicates an input with a non-linear effect on the model output, or an input involved in interactions with other factors. To rank factors in order of importance, it is advisable to use μ^*, as it provides an estimate of the overall factor importance and can be compared to the total sensitive index S_T from the Sobol method.

3.4.2.3 Computational cost

This method uses the Morris sampling approach and the number of model runs is $N(k+1)$ where N is the number of samples and k is the number of input factors. For further reading we recommend [9], [35], and [45, p. 94-108]

3.4.3 Random Balance Designs - Fourier Amplitude Sensitivity Test(RBD-FAST)

This procedure combines Satterthwaite's random balance designs (RBD) with the Fourier Amplitude Sensitivity Test(FAST) to calculate the main effect S_1 for the input factors.

The computational cost (i.e. number of model runs)for this method is equal to the sample size N while computing S_1 using Sobol method have computational cost $N(k+2)$, thus it is much cheaper and should be considered if one just interested in the main effect.

For further reading see [57], [40] and [59]

3.4.4 Delta Moment-Independent Measure (DMIM)

This method is introduced by Borgonovo [6] and Plischke [41] and the proposed GSA indicator is called δ that consider the entire distribution both of the input and the output in a moment dependent fashion. This method holds when the input factors are correlated while the Sobol method holds under the assumption that the input factors are independent. In this method, there is a possibility to study the importance of a group of parameters and interactions.

Sobol method and DMIM method agree in identifying the less relevant parameters with respect to the output uncertainty, but the ranking of the most relevant parameters could be different due to the different scope of the indicators.

3.4.4.1 Outline of the method

Let

1. $$\underline{\mathrm{X}} = (X_1, X_2, X_n,) \in \mathbb{R}^n$$

   be the set of uncertain parameters;

2. $$g\left(\underline{\mathrm{X}}\right) \ \ , g\left(\underline{\mathrm{X}}\right) : E \subseteq \mathbb{R}^n \to \mathbb{R}$$

   be the functional relationship between output Y and input $\underline{\mathrm{X}}$.

3. $\underline{\mathrm{x}} = (x_1, x_2, ..., x_n)$ is the realization of $\underline{\mathrm{X}}$.

4. $F_{\underline{\mathrm{X}}}\left(\underline{\mathrm{x}}\right) \ \ , f_{\underline{\mathrm{X}}}\left(\underline{\mathrm{x}}\right)$ are the cumulative distribution and corresponding joint density of $\underline{\mathrm{X}}$ respectively.

5. $f_{X_i}\left(x_i\right)$ is the marginal density of x_i and is related to the joint density by

   $$f_{X_i}\left(x_i\right) = \int ... \int f_{\underline{\mathrm{X}}}\left(\underline{\mathrm{x}}\right) \prod_{s \neq 0} dx_s$$

6. $F_Y\left(y\right) \ \ , f_Y\left(y\right)$ are the cumulative distribution and corresponding joint density of the model output Y respectively.

7. $f_Y|X_i\left(y\right)$ is the conditional density of Y given that one of the parameters, X_i, assumes a fixed value.

In this method, we can define the following sensitivity measures:

Definition 3.4.2. The quantity δ_i is defined as follows

$$\delta_i = \frac{1}{2} E_{X_i} \left[s \left(X_i \right) \right]$$

where

$$s \left(X_i \right) = \int |f_Y \left(y \right) - f_Y | X_i \left(y \right) | dy$$

is the shift between unconditional density distribution of Y obtained when all parameters free to vary in their range and the one when X_i is fixed at $x_i^\star$. It depends on X_i, thus it is a function of a random variable.

$$E_{X_i} \left[s \left(X_i \right) \right] = \int f_{X_i} \left(x_i \right) \left[\int |f_Y \left(y \right) - f_Y | X_i \left(y \right) | dy \right] dx_i$$

is the expected shift of $s \left(X_i \right)$. The quantity δ_i is called the expected distance between the density of Y and the conditional density of Y given X_i and assumes values between 0 and 1 i.e. $0 \leq \delta_i \leq 1$. It will equal 0 if Y is independent of X_i .

Definition 3.4.3. Let $\underline{\mathrm{R}} = (X_{i_1}, X_{i_1}, ..., X_{i_r})$ be any group of parameters. Then

$$\begin{aligned} \delta_{i_1, i_2, \ldots, i_r} &= \frac{1}{2} E_{\underline{\mathrm{R}}} \left[s \left(\underline{\mathrm{R}} \right) \right] \\ &= \int f_{X_{i_1}, X_{i_2}, \ldots, X_{i_r}} \left(x_{i_1}, x_{i_2}, ..., x_{i_r} \right) \left[\int |f_Y \left(y \right) - f_Y | X_{i_1}, X_{i_2}, ..., X_{i_r} \left(y \right) | dy \right] dx_{i_1}, x_{i_2}, ..., x_{i_r} \end{aligned}$$

where

$$f_{X_{i_1}, X_{i_2}, \ldots, X_{i_r}} \left(x_{i_1}, x_{i_2}, ..., x_{i_r} \right) = \int ... \int f_{\underline{\mathrm{X}}} \left(\underline{\mathrm{x}} \right) \prod_{k \neq i_1, i_2, \ldots, i_r} dx_k$$

From the above definition it has been proven that $\delta_{i_1, i_2, \ldots, i_r}$ i.e. the importance of all parameters equal unity. Note that this doesn't mean that $\sum_{i=1}^{r} \delta_i = 1$.

Definition 3.4.4. Let us consider the group of two parameters $\underline{\mathrm{R}} = (X_i, X_j)$, then the delta of X_i and X_j is given by

$$\delta_{ij} = \frac{1}{2} E_{X_i, X_j} \left[s \left(X_i, X_j \right) \right]$$

where

$$s \left(X_i, X_j \right) = \int |f_Y \left(y \right) - f_Y | X_i, X_j \left(y \right) | dy$$

is the shift obtained fixing X_i at x_i and X_j at x_j or the expected distance between the density of Y and the conditional density of Y given X_i and X_j . If Y is independent of X_j i.e. no contribution to the model uncertainty comes from X_j then $\delta_{ij} = \delta_i$. However, if there exists contribution from X_j to the model uncertainty δ_{ij} will be greater than δ_i.

Now let's define a residual delta $\delta_{j|i}$ which represent the expected distance between the conditional density of Y given X_i and the conditional density of Y given X_i and X_j

$$\delta_{j|i} = \frac{1}{2} E_{X_i X_j}\left[\int \left|f_{Y|X_i}(y) - f_Y|X_i, X_j(y)\right| dy\right]$$

Note that $\delta_{j|i} \geq 0$ with $\delta_{j|i} = 0$ only if Y is independent of X_j, thus $\delta_i \leq \delta_{ij} \leq \delta_i + \delta_{j|i}$. Again if Y is independent of X_j we get that $\delta_{ij} = \delta_i + \delta_{j|i}$.

The previous results can be summarized in the following table

No.	Property	Meaning
1	$0 \leq \delta_i \leq 1$	Bounds the possible values δ_i can assume.
2	$\delta_i = 0$	If Y is independent of X_i then $\delta_i = 0$
3	$\delta_{1,2,\ldots,n} = 1$	The importance of all parameters equals unity.
4	$\delta_{ij} = \delta_i$	If Y is dependent of X_i but independent of then X_j then $\delta_{ij} = \delta_i$
5	$\delta_i \leq \delta_{ij} \leq \delta_i + \delta_{j\|i}$	Bounds the possible values δ_{ij} can assume.
6	$\delta_{ij} = \delta_i + \delta_{j\|i}$	If Y is independent of X_j
7	$\sum_{i=1}^{n} \delta_i \ll 1$	Imply that the output uncertainty is mainly due to individual parameter contributions and not interactions.
8	$\sum_{i=1}^{n} \delta_i \approx \delta_{1,2,\ldots,n} = 1$	Imply that the interactions play a minor role in the output uncertainty.

Table 3.2: Properties of the uncertainty importance measure δ

Note that one can calculate the main effect index S_1 of parameters alongside δ_i based on PDF and not variance as it is involved in computing δ_i and SALib package in python has implemented this. see [19] for further details.

3.5 Observability in dynamical systems

Definition 3.5.1 (Observability). A system is said to be observable at time t_0 if, with the system in state $\boldsymbol{x}(t_0)$, it is possible to determine this state from the observation of the output over a finite time interval.

Mathematically it is defined as follows: Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be two continuously differentiable functions and for some $x^\star$ we have that $f(x^\star) = 0$ and $h(x^\star) = 0$. We consider the following observation system,

$$\begin{aligned}\dot{\boldsymbol{x}} =& \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}\right)\\ y(t) =& h\left(\boldsymbol{x}(t), \boldsymbol{p}\right)\\ \boldsymbol{x}^\star =& \boldsymbol{x}\left(t^\star, \boldsymbol{p}\right)\end{aligned} \tag{3.3}$$

where $\boldsymbol{p} \in \mathbb{R}^q$ is a real-valued vector of parameters, $\boldsymbol{u}(t) \in \mathbb{R}^r$ is the input vector, $\boldsymbol{x}(t)$ is the state variable vector, and $y(t) \in \mathbb{R}^m$ is called the measurable output or the observable vector.

The observation system 3.3 is called observable near the equilibrium point $x^\star$ over a given time interval $[0, T]$, if there exists a neighborhood N of $x^\star$ such that every other state $\boldsymbol{x}_1$ is distinguishable from $\boldsymbol{x}^\star$. Tow states $\boldsymbol{x}^\star \neq \boldsymbol{x}_1$ are said to be distinguishable if there exist some input $\boldsymbol{u}(t)$ such that $y(t, \boldsymbol{x}^\star, \boldsymbol{u}(t)) \neq y(t, \boldsymbol{x}_1, \boldsymbol{u}(t))$, where $y(t, \boldsymbol{x_i}, \boldsymbol{u}(t))$ denote the output function of the system for the input $\boldsymbol{u}(t)$ and initial state $\boldsymbol{x_i}(i = 0, 1)$.

Theorem 3.5.1 (observability of linear system). *Consider the linear system defined by*

$$\begin{aligned}\dot{\boldsymbol{x}} =& A\boldsymbol{x}\\ \boldsymbol{y} =& C\boldsymbol{x}\end{aligned} \tag{3.4}$$

where $\boldsymbol{x}$: state vector (n dimensional)
A: $n\times n$ matrix
C: $m\times n$ matrix

Now the above system is called completely observable if the $n \times nm$ matrix,

$$\left[C \,|\, CA \,|\, CA^2 \,|\, ... |\, CA^{n-1}\right]^T \tag{3.5}$$

has rank n.The previous matrix is usually called observability matrix and is denoted by O.

Theorem 3.5.2 (Observability of nonlinear system). *In case of a nonlinear system we study the local observability by linearizing the observation system 3.3 and defining*

$$A =: \frac{\partial \boldsymbol{f}}{\partial x}\left(\boldsymbol{x}^\star\right), \quad \boldsymbol{C} = \frac{\partial h}{\partial x}\left(\boldsymbol{x}^\star\right)$$

Now the system 3.3 is locally observable if the observability matrix

$$O = \left[C \,|\, CA \,|\, CA^2 \,|\, \ldots | \, CA^{n-1} \right]^T \tag{3.6}$$

is of rank n.

Chapter 4

Analysis of the model

For simplicity, let us consider the bacteria as one quantity, i.e. $x = \sum_{i=1}^{4} x_i$, and nutrition as constant, then the system reduces to:

$$\begin{aligned}
\frac{dx(t)}{dt} &= x\left[r\left(1-\frac{x}{k}\right) - f_1 y_1 - f_2 y_2\right] + s(t), \quad x(0) > 0, \\
\frac{dy_1(t)}{dt} &= y_1\left[-\epsilon_1 - \omega_{11} y_1 - \psi_{12} y_2 + \beta_1 f_1 x_1\right], \quad y_1(0) > 0, \\
\frac{dy_2(t)}{dt} &= y_2\left[-\epsilon_2 + \phi_{12} y_1 - \omega_{22} y_2 + \beta_2 f_2 x_1\right], \quad y_2(0) > 0,
\end{aligned} \tag{4.1}$$

where

$$j, l = 1, 2, \quad k = \frac{r}{\sigma}, \quad \psi_{12} = g_{12} + \omega_{12}, \quad \phi_{12} = \gamma_{12} g_{12} - \omega_{21},$$

and

x : represent the population density of bacteria E.coli, Intestinal enterococci, Campylobacter and Clostridium perfringens .

y_j: represent population density of protozoa and Daphnia at time t respectively

r: is the intrinsic growth rate for prey species.

s: the amount of bacteria species that can be added to the system through fecal contamination.

k : the environmental carrying capacity for ith prey species.

σ: are inter-specific competition coefficient of the prey species.

$\omega_{jl, j\neq l}$: are interspecific competition coefficient of the two predator species, i.e. decreased

rate of predator species j due to interspecific competition with predator species l
ω_{jj}: denote the decreased rate of the predator due to intraspecific competition.
β_j, γ_{12} : are the conversion factors denoting the birth rate per predator (predator species j) per unit prey consumed.
ϵ_j: is the death rate of the predator j in the absence of prey species.
f_j: represent the decreased rate of prey species due to predation by predator species j.
g_{12}: represent the decreased rate of predator species y_1 due to predation by predator species y_2.

The following table summarize the units of the parameters and variables in the system,

x	$y_{j,j=1,2}$	k	r	f_j
$\frac{[num\ x]}{[vol]}$	$\frac{[num\ y_j]}{[vol]}$	$\frac{[num\ x]}{[vol]}$	$\frac{1}{[time]}$	$\frac{[vol]}{[num\ y_j][time]}$
β_j	ϵ_j	ϕ_{12}	ψ_{12}	ω_{jj}
$\frac{[num\ y_j]}{[num\ x]}$	$\frac{1}{[time]}$	$\frac{[vol]}{[num\ y_1][time]}$	$\frac{[vol]}{[num\ y_2][time]}$	$\frac{[vol]}{[num\ y_j][time]}$
s	$\frac{dx(t)}{dt}$	$\frac{dy_j(t)}{dt}$	vol	time
$\frac{[num\ x]}{[vol][time]}$	$\frac{[num\ x]}{[vol][time]}$	$\frac{[num\ y_j]}{[vol][time]}$	100 ml	day

Table 4.1: Units of the parameters and variables in the system

4.1 Nondimensionalization

For simplicity we nondimensionalize the subsystem by using the following scaling:

$$\begin{aligned} t =& T\bar{t} = \frac{1}{r}\bar{t}, \quad x = X\bar{x} = \frac{r}{\beta 1 f_1}\bar{x}, \quad y_1 = Y_1\bar{y}_1 = \frac{r}{f_1}\bar{y}_1, \quad y_2 = Y_1\bar{y}_2 = \frac{\beta 2}{\beta 1}\frac{r}{f_2}\bar{y}_1, \\ s =& S\bar{s} = \frac{\beta 1 f_1}{r^2}, \end{aligned} \tag{4.2}$$

Then the system becomes:

$$\begin{aligned} \frac{d\bar{x}}{d\bar{t}} &= \bar{x}\left[1 - a\bar{x} - \bar{y}_1 - b\bar{y}_2\right] + \bar{s}(\bar{t}) = F_1, \quad \bar{x}(0) > 0 \\ \frac{d\bar{y}_1}{d\bar{t}} &= \bar{y}_1\left[-\delta_1 - \omega_1\bar{y}_1 - \psi b\bar{y}_2 + \bar{x}\right] = F_2, \quad \bar{y}_1(0) > 0 \\ \frac{d\bar{y}_2}{d\bar{t}} &= \bar{y}_2\left[-\delta_2 + \phi\bar{y}_1 - \omega_2 b y_2 + b\bar{x}\right] = F_3, \quad \bar{y}_2(0) > 0 \end{aligned} \tag{4.3}$$

where

$$
\begin{aligned}
a =&\frac{r}{\beta_1 f_1 k}, \quad b = \frac{\beta_2 f_2}{\beta_1 f_1}, \quad \omega_1 = \frac{\omega_{11}}{f_1}, \quad \omega_2 = \frac{\omega_{22}}{f_2}, \\
\delta_1 =&\frac{\epsilon_1}{r}, \quad \delta_2 = \frac{\epsilon_2}{r}, \quad \psi = \frac{\psi_{12}}{f_2} = \frac{g_{12}+\omega_{12}}{f_2}, \quad \phi = \frac{\phi_{12}}{f_1} = \frac{\gamma_{12} g_{12} - \omega_{21}}{f_1}
\end{aligned}
$$

In matrix form the system can be written as,

$$
\frac{d}{d\bar{t}} \begin{bmatrix} \bar{x}(\bar{t}) \\ \\ \bar{y}_1(\bar{t}) \\ \\ \bar{y}_2(\bar{t}) \\ \end{bmatrix} = \begin{bmatrix} \bar{x} & 0 & 0 \\ \\ 0 & \bar{y}_1 & 0 \\ \\ 0 & 0 & \bar{y}_2 \\ \end{bmatrix} \left(\begin{bmatrix} 1 \\ \\ -\delta_1 \\ \\ -\delta_2 \\ \end{bmatrix} - \begin{bmatrix} a & 1 & b \\ \\ -1 & \omega_1 & \psi b \\ \\ -b & -\phi & \omega_2 b \end{bmatrix} \begin{bmatrix} \bar{x} \\ \\ \bar{y}_1 \\ \\ \bar{y}_2 \\ \end{bmatrix} \right) + \begin{bmatrix} \bar{s} \\ \\ 0 \\ \\ 0 \\ \end{bmatrix}
$$

or more compactly as

$$
\frac{d\bar{\boldsymbol{x}}}{d\bar{t}} = \bar{\boldsymbol{X}}^T \left(\bar{\boldsymbol{r}} - A\bar{\boldsymbol{X}} \right) + \bar{\boldsymbol{s}} = \boldsymbol{F}(\bar{\boldsymbol{x}}, \boldsymbol{p})
$$

which is generalized Lotka-Volterra system if we consider $s = 0$.

where

$$
\begin{aligned}
\bar{\boldsymbol{x}} =& \left[\, \bar{x}, \bar{y}_1, \bar{y}_2 \,\right]^T, \quad \bar{\boldsymbol{r}} = \left[\, 1, -\delta_1, -\delta_2 \,\right]^T, \quad \boldsymbol{F} = \left[F1, F2, F3\right]^T, \\
\bar{\boldsymbol{s}} =& \left[\bar{s}, 0, 0\right]^T, \quad \boldsymbol{p} = \left[\, a, b, \delta_1, \delta_2, \psi, \phi, \omega_1, \omega_2\, \bar{s} \right], \\
\bar{X} =& \begin{bmatrix} \bar{x} & 0 & 0 \\ \\ 0 & \bar{y}_1 & 0 \\ \\ 0 & 0 & \bar{y}_2 \\ \end{bmatrix}, \quad A = \begin{bmatrix} a & 1 & b \\ \\ -1 & \omega_1 & \psi b \\ \\ -b & -\phi & \omega_2 b \end{bmatrix}, \quad \frac{d\bar{\boldsymbol{x}}}{d\bar{t}} = \left[\frac{d\bar{x}(\bar{t})}{d\bar{t}}, \frac{d\bar{y}_1(\bar{t})}{d\bar{t}}, \frac{d\bar{y}_2(\bar{t})}{d\bar{t}} \right]^T,
\end{aligned}
$$

We continue by dropping the bar notation and considering $\bar{\boldsymbol{s}} = \mathbf{0}$

4.2 Domain, positivity, and boundedness of the solution

In this section, we state some basic results of the system 4.3. The domain of the solution is the positive $\mathbb{R}^3_+ = \{(x, y_1, y_2) : (x, y_1, y_2) \geq 0\}$ and all the results are valid in this domain unless something else is mentioned explicitly.

Lemma 4.2.1 (Positivity of the solution). *Every solution of the system 4.3 with initial conditions* $x(0) > 0, y_1(0) > 0$ *and* $y_2(0) > 0$ *is positive i.e. exist in the interval* $[0, \infty]$ *for all* $\bar{t} \geq 0$

Proof. We follow the same procedure as described in [39]. Since the right-hand side of the system 4.3 and its derivatives are continuous, then there exists a unique solution for the system in some interval $0 \leq \xi < \infty$ according to existence and uniqueness theorem for nonlinear ODEs. see [7, p. 51-52]

From the system 4.3 with the given initial conditions we have:

$$\begin{aligned} x\left(t\right) &= x\left(0\right) exp\left(\int_0^t F_1(t) d\theta\right) \\ y_1(t) &= y_1(0) exp\left(\int_0^t F_2(t) d\theta\right) \\ y_2(t) &= y_2(0) exp\left(\int_0^t F_3(t) d\theta\right) \end{aligned}$$

where F_1, F_2 and F_3 are right-hand side of the equations. Clearly we can see that $x(t), y_1(t)$ and $y_2(t)$ are positive if the initial conditions and time are positive . Another way of seeing this is if we set $x = 0, y_1 = 0, y_2 = 0$ in the system 4.3 we obtain that $\frac{dx}{dt} = 0$, $\frac{dy_1}{dt} = 0$, $\frac{dy_2}{dt} = 0$ i.e. we stay in the positive octant $\mathbb{R}^3_+$. □

Lemma 4.2.2 (Boundedness of the solution). *All solution of the system that initiate in* $\mathbb{R}^3_+$ *are uniformly bounded within a region* $\boldsymbol{B} \subseteq \mathbb{R}^3_+$ *defined by:*

$$\boldsymbol{B} = \left\{(x, y_1, y_2) : (x, y_1, y_2) \in \mathbb{R}^3_+ : 0 \leq x \leq x_m, \quad 0 \leq y_1 \leq y_{1m}, \quad 0 \leq y_2 \leq y_{2m}\right\}$$

where

$$\begin{aligned} x_m =& max\left\{x_0, \frac{1}{a}\right\}, \quad y_{1m} = max\left\{y_{10}, \frac{B_1}{\omega_1}\right\}, \quad y_{2m} = max\left\{y_{20}, \frac{B_2}{\omega_2}\right\}, \\ B_1 =& x_m - \delta_1, \quad B_2 = b\, x_m + \phi\, y_{1m} - \delta_2 \end{aligned}$$

Proof. In the first equation, we have

$$\begin{aligned} \frac{dx}{dt} =& x\left(-ax - by_2 - y_1 + 1\right) \leq x\left(-ax + 1\right) \\ & \Downarrow \\ \frac{du}{dt} =& u(1 - au) \end{aligned}$$

solving it we obtain

$$\begin{aligned}
u\left(t\right) =&\frac{x_0 e^t}{a x_0 e^t - a x_0 + 1} = \frac{x_0}{a x_0 - \left(x_0 - 1\right) e^{-t}} \\
&\Downarrow \\
x\left(t\right) \leq&\frac{x_0}{a x_0 - \left(x_0 - 1\right)}
\end{aligned}$$

Taking the limit at 0 and ∞,

$$\begin{aligned}
\lim_{t\to 0} u\left(t\right) =&x_0 \\
\lim_{t\to \infty} u\left(t\right) =&\frac{1}{a}
\end{aligned}$$

We obtain that $x\left(t\right) \leq \max\left(x_0,\, \frac{1}{a}\right) = x_{max}$

For the second equation we have,

$$\begin{aligned}
\frac{dy_1}{dt} =&y_1\left(-b\psi y_2 - \delta_1 - \omega_1 y_1 + x\right) \leq y_1\left(-b\psi y_2 - \delta_1 - \omega_1 y_1 + x_{max}\right) \\
\leq&y_1\left(B_1 - \omega_1 y_1\right) \\
&\Downarrow \\
\frac{du}{dt} =&u\left(B_1 - \omega_1 u\right)
\end{aligned}$$

solving the last equation we obtain,

$$\begin{aligned}
u\left(t\right) =&\frac{B_1 y_{10} e^{B_1 t}}{B_1 + \omega_1 y_{10} e^{B_1 t} - \omega_1 y_{10}} = \frac{B1 y_{10}}{\omega_1 y_{10} + \left(B_1 - \omega_1 y_{10}\right) e^{-B_1 t}} \\
&\Downarrow \\
y_1\left(t\right) \leq&\frac{B_1 y_{10}}{\omega_1 y_{10} + \left(B_1 - \omega_1 y_{10}\right) e^{-B_1 t}}
\end{aligned}$$

Taking the limit at 0 and ∞,

$$\begin{aligned}\lim_{t\to 0} u(t) =& y_{1_0}\\ \lim_{t\to\infty} u(t) =& 0 \text{ if } \quad B_1 < 0\\ \lim_{t\to\infty} u(t) =& \frac{B_1}{\omega_1} \text{ if } \quad B_1 > 0\end{aligned}$$

we obtain that $\mathrm{y}_1(t) \leq \max\left\{y_{10},\, \frac{B_1}{\omega_1}\right\} = y_{1m}$

For the third equation, we have,

$$\begin{aligned}\frac{dy_2}{dt} =& y_2\left(-b\omega_2 y_2 + bx(t) - \delta_2 + \phi\, \mathrm{y}_1(t)\right) \leq y_2\left(bx_m + \phi y_{1m} - \delta_2 - \omega_2 y_2\right)\\ \leq& y_2\left(B_2 - \omega_2 y_2\right)\\ & \Downarrow\\ \frac{du}{dt} =& u\left(B_2 - \omega_2 u\right)\end{aligned}$$

solving the last equation we obtain,

$$\begin{aligned}u(t) =& \frac{B_2 y_{20} e^{B_2 t}}{B_2 + b\omega_2 y_{20} e^{B_2 t} - b\omega_2 y_{20}} = \frac{B_2 y_{20}}{b\omega_2 y_{20} + \left(B_2 - b\omega_2 y_{20}\right) e^{-B_2 t}}\\ & \Downarrow\\ \mathrm{y}_2(t) \leq& \frac{B_2 y_{20}}{b\omega_2 y_{20} + \left(B_2 - b\omega_2 y_{20}\right) e^{-B_2 t}}\end{aligned}$$

Taking the limit at 0 and ∞,

$$\begin{aligned}\lim_{t\to 0} u(t) =& y_{20}\\ \lim_{t\to\infty} u(t) =& 0 \text{ if } \quad B_2 < 0\\ \lim_{t\to\infty} u(t) =& \frac{B_2}{b\omega_2} \text{ if } \quad B_2 > 0\end{aligned}$$

we obtain that $\mathrm{y}_2(t) \leq \max\left\{y_{20},\, \frac{B_2}{b\omega_2}\right\} = y_{2m}$.

Now we have

$$\begin{aligned}\dot{u} >& f\left(x,\, u\right)\\ \dot{v} <& f\left(x,\, v\right)\\ \dot{w} =& f\left(x,\, w\right)\\ &\text{and}\\ u\left(t_0\right) =& v\left(t_0\right) = w\left(t_0\right)\end{aligned}$$

for each equation. Thus, by applying the Chaplygin inequality, we have,

$$u\left(t\right) > w\left(t\right) > v\left(t\right)$$

i.e. the solution of the system is bounded.

Another way of proving boundedness is by following ideas from [28] and [24][p 22-25].

The derivative of W along the trajectory of the system 4.3 is given by:

$$\begin{aligned}\frac{dW\left(t\right)}{dt} =& \frac{dx\left(t\right)}{dt} + \frac{1}{B_1}\frac{dy_1(t)}{dt} + \frac{1}{B_2}\frac{dy_2\left(t\right)}{dt} = x\left(1-ax\right) + \frac{1}{B_1}y_1\left(-B_1 - \omega_1 y_1\right) + \frac{1}{B_2}y_2\left(-B_2 - \omega_2 y_2\right)\\ =& x + y_1 + y_2 - \left(ax^2 + \frac{\omega_1}{B_1}{y_1}^2 + \frac{\omega_2}{B_2}{y_2}^2\right)\\ \leq& x + y_1 + y_2 - \mu\left(x^2 + {y_1}^2 + {y_2}^2\right)\\ \leq& x + y_1 + y_2 + \eta\left(x + y_1 + y_2\right)^2\\ \leq& W - \eta W^2\end{aligned}$$

Solving the obtained differential equation with initial condition $W\left(0\right) = W_0$ we obtain

$$W\left(t\right) \leq \frac{W_0}{W_0\eta - \left(W_0\eta - 1\right)e^{-t}}$$

Taking the limit at 0 and ∞,

$$\begin{aligned}\lim_{t\to 0} W\left(t\right) \leq& W_0\\ \lim_{t\to\infty} W\left(t\right) \leq& \frac{1}{\eta}\end{aligned}$$

where

$$\eta = \min\left\{\frac{1}{a},\, \frac{\omega_1}{B_1},\, \frac{\omega_2}{B_2}\right\},$$

we find that $W \leq \max\left\{W_0,\, \frac{1}{\eta}\right\}$ i.e. the solution of the system is uniformly bounded. □

4.3 Equilibrium points and condition for their biological feasibility

Let us assume the following assumption on the dimensional parameters:

1. The intraspecific competition is stronger than the interspecific competition if the species have the same size i.e. consume the resources equally. [1]
2. If one species is larger in size than the other species, then the interspecific competition is stronger on the smaller species because larger species normally consume more resources than smaller ones.
3. The intraspecific competition of the first predator species is equal to the one for the second predator.
4. The food conversion efficiency coefficient is less than the uptake rate coefficient, which represents the number of prey killed per predator.

Mathematically and in terms of dimensional parameters, this can be expressed as:

$$\begin{aligned}
&\omega_{12} >> \omega_{21} \quad , \quad \omega_{12} > \omega_{11} \quad , \quad \omega_{12} > \omega_{22} \quad , \\
&\omega_{22} = \omega_{11} \quad , \quad \omega_{21} < \omega_{11} \quad , \quad \omega_{21} < \omega_{22} \quad , \\
&\gamma_{12} < g_{12} \quad , \quad \beta_1 < f_1 \quad , \quad \beta_2 < f_2 \quad , \quad \beta_1 > \beta_2 \quad , \quad f_1 = f_2 \quad , \quad \epsilon_1 < \epsilon_2
\end{aligned}$$

In terms of non-dimensional parameters become:

$$\omega_1 = \omega_2, \quad \psi > \phi, \quad \delta_1 >> a, \quad \delta_2 >> a, \quad \delta_1 < \delta_2, \quad \psi > \omega_1, \quad \psi > \omega_2, \quad a < b \tag{4.4}$$

4.3.1 Equilibrium points

We obtain the equilibrium points of the system by solving the system of algebraic equations obtained by setting the right-hand side of the system 4.3 to zero. Here we use the SymPy (Python library for symbolic mathematics) to solve the system, and we obtain the following five equilibrium points:

1. $EP_1 = (0, 0, 0)$, i.e. none of the species exist.
2. $EP_2 = \left(\frac{1}{a}, 0, 0\right)$, i.e. only bacteria x is present in the system.

3. $EP_3\left(\frac{\delta_1+\omega_1}{a\omega_1+1},\ \ \frac{-a\delta_1+1}{a\omega_1+1},\ \ 0\right)$, i.e. only bacteria x and Protozoa y_1 are present and the second predator y_2 is extinct.
4. $EP_4 = \left(\frac{\delta_2+\omega_2}{a\omega_2+b},\ \ 0,\ \ \frac{-a\delta_2+b}{b(a\omega_2+b)}\right)$, this point represent the situation when only bacteria x and the second predator y_1 are present while the first predator is extinct.
5. $EP_5 = \left(\frac{E_0}{U_0}, \frac{E_1}{U_0}, \frac{E_2}{U_0 b}\right)$, this point represent the situation where all species exist together. Therefore, it is called the coexistence point or the interior equilibrium point.

where:

$$
\begin{aligned}
U_0 =& a\omega_1\omega_2 + a\phi\psi + b\omega_1 - b\psi + \omega_2 + \phi = \phi\left(1 + a\psi\right) + b\left(\omega_1 - \psi\right) + \omega_2\left(1 + a\omega_1\right) \\
E_0 =& \delta_1\left(\omega_2 + \phi\right) + \delta_2\left(\omega_1 - \psi\right) + \omega_1\omega_2 + \phi\psi, \\
E_1 =& - b\psi - \delta_1\left(a\omega_2 + b\right) + \delta_2\left(a\psi + 1\right) + \omega_2 \\
E_2 =& b\omega_1 - \delta_1\left(a\phi - b\right) - \delta_2\left(a\omega_1 + 1\right) + \phi,
\end{aligned}
$$

4.3.2 Existence conditions for positive equilibrium points

Let us divide the existence analysis into three cases:

4.3.3 Case 1 : $\delta_1 > \frac{1}{a}$ and $\delta_2 > \frac{b}{a}$

The condition $\delta_1 > \frac{1}{a}$ in terms of dimensional variables is $\delta_1 > k\beta_1 f_1$ i.e. the death rate for the first predator is greater than a critical value represented by the food conversion efficiency of the first predator. Similarly, the condition $\delta_2 > \frac{b}{a}$ in terms of dimensional variables is $\delta_2 > k\beta_2 f_2$ i.e. the death rate for the first predator is greater than a critical value represented by the food conversion efficiency of the second predator. In this case, there are only two ecologically feasible equilibrium points which are EP_1 and EP_2. Both points exist as the first doesn't depend on any parameters and the second depend on a which is always positive.

4.3.4 Case 2 : $\delta_1 = \frac{1}{a}$ and $\delta_2 = \frac{b}{a}$

In this case we have the following:

$$EP_3 = EP_4 = EP_5 = EP_2 = (1, 0, 0)$$

Proof. It is easy to check that $EP_3 = EP_4 = EP_2$

For EP_5 we have

$$\left.\begin{aligned} U_0 =& aE_0 \\ E_1 =& -b\psi - \delta_1\left(a\omega_2 + b\right) + \delta_2\left(a\psi + 1\right) + \omega_2 \\ =& -b\psi - \frac{1}{a}\left(a\omega_2 + b\right) + \frac{b}{a}\left(a\psi + 1\right) + \omega_2 \\ =& -b\psi - \omega_2 - \frac{b}{a} + b\psi + \frac{b}{a} + \omega_2 = 0 \\ E_2 =& 0 \quad \text{similarly} \end{aligned}\right\} \leftarrow \qquad EP_5 = EP_2$$

□

Thus there are only two ecologically feasible equilibrium points which are EP_1 and EP_2 and we have the same conclusion about existence as in Case 1.

4.3.5 Case 3 : $\delta_1 < \frac{1}{a}$ and $\delta_2 < \frac{b}{a}$

The condition $\delta_1 < \frac{1}{a}$ in terms of dimensional variables is $\epsilon_1 < k\beta_1 f_1$ i.e. the death rate for the first predator is less than a critical value represented by the food conversion efficiency. Similarly, condition $\delta_1 < \frac{1}{a}$ in terms of dimensional variables is $\epsilon_2 > k\beta_2 f_2$ i.e. the death rate for the first predator is less than a critical value represented by the food conversion efficiency. In this case all the equilibrium points may exist and this will be the focus during this study.

4.3.5.1 Existence of EP_1

This point always exists since it doesn't depend on any parameter.

4.3.5.2 Existence of EP_2

This point always exists since it depends on the parameter a only and we have $a > 0$.

4.3.5.3 Existence of EP_3

EP_3 exists if $\delta_1 < \frac{1}{a}$ which in terms of dimensional variables is $\epsilon_1 < k\beta_1 f_1$ i.e. the death rate for the first predator should be less than a critical value represented by the food conversion efficiency.

4.3.5.4 Existence of EP_4

EP_4 exist if $\delta_2 < \frac{b}{a}$ which in terms of dimensional variables is $\epsilon_2 < k\beta_2 f_2$ i.e. the death rate for the second predator should be less than a critical value represented by the food conversion efficiency.

4.3.5.5 Existence of EP_5

There are two cases

1. Case 1: $E_0, E_1, E_2, U_0 > 0$ then

$$\begin{aligned} U_0 > 0 \Leftrightarrow & a\omega_1\omega_2 + a\phi\psi + b\omega_1 - b\psi + \omega_2 + \phi > 0 \\ \Leftrightarrow & \phi > \frac{b(\psi - \omega_1) - \omega_2(1 + a\omega 1)}{1 + \psi} \qquad \text{satisfied if } \psi < \omega_1 \end{aligned}$$

$$\begin{aligned} E_0 > 0 \Leftrightarrow & \delta_1(\omega_2 + \phi) + \delta_2(\omega_1 - \psi) + \omega_1\omega_2 + \phi\psi > 0 \\ \Leftrightarrow & \delta_1 > \frac{\delta_2(\psi - \omega_1) - \omega_1\omega_2 - \psi\phi}{1 + \psi} \qquad \text{satisfied if } \psi < \omega_1 \end{aligned}$$

$$\begin{aligned} E_1 > 0 \Leftrightarrow & -b\psi - \delta_1(a\omega_2 + b) + \delta_2(a\psi + 1) + \omega_2 > 0 \\ \Leftrightarrow & \delta_1 > \frac{\delta_2(a\psi + 1) + \omega_2 - b\psi}{a\omega_2 + b} \end{aligned}$$

$$\begin{aligned} E_2 > 0 \Leftrightarrow & b\omega_1 - \delta_1(a\phi - b) - \delta_2(a\omega_1 + 1) + \phi > 0 \\ \Leftrightarrow & \delta_1 > \frac{\delta_2(1 - a\omega_1) + b\omega_1 + \phi}{a\phi - b} \end{aligned}$$

By assumptions 4.4 we have $\psi > \omega_1$ i.e. EP_5 may not exist in this case.

2. Case 2: $E_0, E_1, E_2, U_0 < 0$

$$\begin{aligned} U_0 < 0 \Leftrightarrow & a\omega_1\omega_2 + a\phi\psi + b\omega_1 - b\psi + \omega_2 + \phi < 0 \\ \Leftrightarrow & \phi < \frac{b(\psi - \omega_1) - \omega_2(1 + a\omega_1)}{1 + \psi} \end{aligned}$$

$$\begin{aligned} E_0 < 0 \Leftrightarrow & \delta_1(\omega_2 + \phi) + \delta_2(\omega_1 - \psi) + \omega_1\omega_2 + \phi\psi < 0 \\ \Leftrightarrow & \delta_1 < \frac{\delta_2(\psi - \omega_1) - \omega_1\omega_2 - \psi\phi}{1 + \psi} \qquad . \end{aligned}$$

$$\begin{aligned} E_1 < 0 &\Leftrightarrow -b\psi - \delta_1\left(a\omega_2+b\right) + \delta_2\left(a\psi+1\right) + \omega_2 < 0 \\ &\Leftrightarrow \delta_1 < \frac{\delta_2\left(a\psi+1\right)+\omega_2-b\psi}{a\omega_2+b} \end{aligned}$$

$$\begin{aligned} E_2 < 0 &\Leftrightarrow b\omega_1 - \delta_1\left(a\phi-b\right) - \delta_2\left(a\omega_1+1\right) + \phi < 0 \\ &\Leftrightarrow \delta_1 < \frac{\delta_2\left(1-a\omega_1\right)+b\omega_1+\phi}{a\phi-b} \end{aligned}$$

Under assumptions 4.4 we can prove that $U_0, E_0, E_1, E_2 < 0$ i.e. existence for EP_5 is guaranteed.

We can prove or observe that the following is in case $\delta_1 < \frac{1}{a} \quad and \quad \delta_2 < \frac{b}{a}$ and assumption 4.4:

$$\begin{aligned} &E_0, E_1, E_2 < U_0 \Rightarrow x_5, \quad y_{15}, \quad y_{25} < 1 \\ &x_3, \quad y_{13}, \quad x_4, \quad y_{14} < 1 \\ &x_3 < x_5 < x_4 < x_2 \\ &y_{15} < y_{13} \quad y_{25} < y_{24} \end{aligned}$$

Proof.

$$\left.\begin{aligned} x_3 =& \frac{\delta_1+\omega_1}{a\omega_1+1} = \frac{\delta_1+\omega_1}{a\left(\frac{1}{a}+\omega_1\right)} < 1 \quad \text{since } \delta_1 < \frac{1}{a} \\ x_4 =& \frac{\delta_2+\omega_2}{a\omega_2+b} = \frac{\delta_2+\omega_2}{a\left(\frac{b}{a}+\omega 2\right)} < 1 \quad \text{since } \delta_2 < \frac{b}{a} \end{aligned}\right\} \rightarrow x_3 < x_4 \quad if \quad \frac{\delta_2}{\delta_1} > b$$

$x_3 < x_4$: this conclusion is in agreement with ecology since protozoa feed only on bacteria, resulting in minimum bacteria concentration while Daphnia can feed on both bacteria and protozoa resulting in higher bacteria population.

$x_5 < x_4$: since both predators are present and both feed on bacteria resulting in lower concentration of bacteria.

$x_5 > x_3$: since in the interior steady state, the two predators compete for bacteria which leads to a decline in the population of both and increase in the population of bacteria. This doesn't happen at the equilibrium point where only bacteria and protozoa are present.

$$\begin{aligned} y_{13} =& \frac{-a\delta_1+1}{a\omega_1+1} < 1 \quad \text{since } \omega_1 > -\delta_1 \\ y_{24} =& \frac{-a\delta_2+b}{b\left(a\omega_2+b\right)} < 1 \quad \text{since } \omega_2 > -\delta_2 \end{aligned}$$

$y_{15} < y_{13}$: i.e. the concentration of Protozoa in the interior eqm. point is less than it in the eqm. point where it exists with bacteria because in the interior eqm. point it competes with Daphnia resulting in less population growth.
Similarly $y_{25} < y_{13}$ i.e. the concentration of Daphnia in the interior eqm. point is less than it in the eqm. point where it exists with bacteria because in the interior eqm. point it competes with Protozoa resulting in less population growth. □

In conclusion, the lowest bacteria concentration is in the equilibrium point where only Protozoa and bacteria are present and highest at the equilibrium point EP_2 in which the bacteria exist on its own. The concentration of microorganisms in the interior equilibrium point is lower than when only one of them exists with bacteria. This is important in controlling the system.

4.4 Local stability analysis

We start our analysis by linearizing the system and finding the Jacobian matrix at each equilibrium point and corresponding eigenvalues, then study its stability.

The Jacobian matrix of the system for an arbitrary point (x, y_1, y_2) is given by:

$$J = \begin{bmatrix} -2ax - by_2 - y_1 + 1 & -x & -bx \\ y_1 & -b\psi y_2 - \delta_1 - 2\omega_1 y_1 + x & -b\psi y_1 \\ by_2 & \phi y_2 & -2b\omega_2 y_2 + bx - \delta_2 + \phi y_1 \end{bmatrix}$$

4.4.0.1 First Equilibrium point EP_1

$$\begin{aligned} J_1 =& J(EP_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\delta_1 & 0 \\ 0 & 0 & -\delta_2 \end{bmatrix} \\ \lambda_1 =& \{1,\ -\delta_1,\ -\delta_2\} \end{aligned}$$

The first eigenvalue is positive and others are negative, thus EP_1 is unstable for all values of parameters in x-direction but stable in the y_1y_2 plane i.e. it is a saddle point. Biologically, it means the system can't go to extinction.

4.4.0.2 Second Equilibrium point EP_2

$$
\begin{aligned}
J_2 =& J(EP_2) = \begin{bmatrix} -1 & -\frac{1}{a} & -\frac{b}{a} \\ 0 & -\delta_1 + \frac{1}{a} & 0 \\ 0 & 0 & -\delta_2 + \frac{b}{a} \end{bmatrix} \\
\lambda_2 =& \left\{ -1,\ -\delta_1 + \frac{1}{a},\ -\delta_2 + \frac{b}{a} \right\}
\end{aligned}
$$

From the sign of eigenvalues we conclude that EP_2 asymptotically stable in xyz-space if $\delta_1 > \frac{1}{a}$, $\delta_2 > \frac{b}{a}$ and it will also be globally stable there. If $\delta_1 < \frac{1}{a}$, $\delta_2 < \frac{b}{a}$ then it is a saddle point because it is stable in the x direction but unstable in the $y_1 y_2$ plane.

4.4.0.3 Third Equilibrium point EP_3

$$
J_3 = J(EP_3) = \begin{bmatrix} -\frac{a(\delta_1+\omega_1)}{a\omega_1+1} & -\frac{\delta_1+\omega_1}{a\omega_1+1} & -\frac{b(\delta_1+\omega_1)}{a\omega_1+1} \\ \frac{-a\delta_1+1}{a\omega_1+1} & \frac{\omega_1(a\delta_1-1)}{a\omega_1+1} & \frac{b\psi(a\delta_1-1)}{a\omega_1+1} \\ 0 & 0 & \frac{b(\delta_1+\omega_1)-\delta_2(a\omega_1+1)-\phi(a\delta_1-1)}{a\omega_1+1} \end{bmatrix}
$$

$$
\lambda_3 = \left\{ \frac{\omega_1(a\delta_1-1-a)-a\delta_1-\sqrt{\mu_0}}{2(a\omega_1+1)},\quad \frac{\omega_1(a\delta_1-1-a)-a\delta_1+\sqrt{\mu_0}}{2(a\omega_1+1)},\quad \frac{E_2}{a\omega_1+1} \right\}
$$

$$
\begin{aligned}
\mu_0 =& a^2\delta_1^2\omega_1^2 + 2a^2\delta_1^2\omega_1 + a^2\delta_1^2 + 2a^2\delta_1\omega_1^2 + 2a^2\delta_1\omega_1 + a^2\omega_1^2 \\
&+ 4a\delta_1^2 - 2a\delta_1\omega_1^2 + 2a\delta_1\omega_1 - 2a\omega_1^2 - 4\delta_1 + \omega_1^2 - 4\omega_1 \\
& E_2 \text{is as before .}
\end{aligned}
$$

There are three distinct eigenvalues and two cases to consider,

1. Case $\mu_0 > 0$, the eigenvalues are all real, and we have the following,

   $\lambda_{31} < 0$ if $\sqrt{\mu_0} > \omega_1 (a\delta_1 - 1 - a) - a\delta_1$ which is satisfied as $a\delta_1 < 1$ from existence condition.

   $\lambda_{32} < 0$ if $\sqrt{\mu_0} > a\delta_1 - \omega_1 (a\delta_1 - 1 - a)$
   $\lambda_{32} > 0$ if $\sqrt{\mu_0} < a\delta_1 - \omega_1 (a\delta_1 - 1 - a)$

$\lambda_{33} < 0$ if $E_2 < 0$,
$\lambda_{33} > 0$ if $E_2 > 0$

EP_3 will be locally stable if $\sqrt{\mu_0} > a\delta_1 - \omega_1 (a\delta_1 - 1 - a)$ and $E_1 < 0$ and unstable if one of the conditions doesn't hold.

2. Case $\mu_0 < 0$,

   The eigenvalues λ_{31} and λ_{32} are complex with negative real part since

   $$\omega_1 (a\delta_1 - 1 - a) - a\delta_1 < 0$$

   , thus EP_3 will be stable if $E_2 < 0$ and unstable if $E_2 > 0$.

In the unstable case it will be a saddle point since it is stable in xy_1 plane and unstable in y_2 direction.

4.4.0.4 Fourth Equilibrium point EP_4

$$J_4 = J(EP_4) = \begin{bmatrix} -\frac{a(\delta_2+\omega_2)}{a\omega_2+b} & -\frac{\delta_2+\omega_2}{a\omega_2+b} & -\frac{b(\delta_2+\omega_2)}{a\omega_2+b} \\ 0 & \frac{-\delta_1(a\omega_2+b)+\delta_2+\omega_2+\psi(a\delta_2-b)}{a\omega_2+b} & 0 \\ \frac{-a\delta_2+b}{a\omega_2+b} & -\frac{\phi(a\delta_2-b)}{b(a\omega_2+b)} & \frac{\omega_2(a\delta_2-b)}{a\omega_2+b} \end{bmatrix}$$

$$\lambda_4 = \left\{ \frac{\omega_2(a\delta_2-b-a)-a\delta_2-\sqrt{\mu_1}}{2(a\omega_2+b)}, \frac{\omega_2(a\delta_2-b-a)-a\delta_2+\sqrt{\mu_1}}{2(a\omega_2+b)}, \frac{E_1}{a\omega_2+b} \right\}$$

$$\begin{aligned} \mu_1 = & a^2\delta_2^2\omega_2^2 + 2a^2\delta_2^2\omega_2 + a^2\delta_2^2 + 2a^2\delta_2\omega_2^2 + 2a^2\delta_2\omega_2 + a^2\omega_2^2 \\ & + 4ab\delta_2^2 - 2ab\delta_2\omega_2^2 + 2ab\delta_2\omega_2 - 2ab\omega_2^2 - 4b^2\delta_2 + b^2\omega_2^2 - 4b^2\omega_2. \end{aligned}$$

E_1 is as before

There are three distinct eigenvalues and two cases to consider,

1. Case $\mu_1 > 0$, the eigenvalues are all real, and we have the following,

   $\lambda_{41} < 0$ if $\sqrt{\mu_1} > \omega_2 (a\delta_2 - b - a) - a\delta_2$ which is satisfied as $a\delta_2 < b$ from existence condition.

   $\lambda_{42} < 0$ if $\sqrt{\mu_1} > a\delta_2 - \omega_2 (a\delta_2 - b - a)$
   $\lambda_{42} > 0$ if $\sqrt{\mu_1} < a\delta_2 - \omega_2 (a\delta_2 - b - a)$

$\lambda_{43} < 0$ if $E_2 < 0$,
$\lambda_{43} > 0$ if $E_2 < 0$

EP_3 will be locally stable if $\sqrt{\mu_1} > a\delta_2 - \omega_2\left(a\delta_2 - b - a\right)$ and $E_2 < 0$ and unstable if one of the conditions doesn't hold.

2. Case $\mu_0 < 0$,

   The eigenvalues λ_{41} and λ_{42} are complex with negative real part since

   $$\omega_2\left(a\delta_2 - b - a\right) - a\delta_2 < 0$$

   . Consequently, EP_4 will be stable if $E_1 < 0$ and unstable if $E_1 > 0$.

In the unstable case it will be a saddle point since it is stable in xy_2 plane and unstable in y_1 direction.

4.4.0.5 Fifth Equilibrium point EP_5

$$J_5 = J(EP_5) = \begin{bmatrix} \frac{U_1 a}{U_0} & \frac{U_1}{U_0} & \frac{U_1 b}{U_0} \\ -\frac{U_2}{U_0} & \frac{U_2\omega_1}{U_0} & \frac{U_2 b\psi}{U_0} \\ -\frac{U_3}{U_0} & -\frac{U_3\phi}{U_0 b} & \frac{U_3\omega_2}{U_0} \end{bmatrix}$$

Where:

$$\begin{aligned} U_1 =& -\delta_1\omega_2 - \delta_1\phi - \delta_2\omega_1 + \delta_2\psi - \omega_1\omega_2 - \phi\psi \\ U_2 =& a\delta_1\omega_2 - a\delta_2\psi + b\delta_1 + b\psi - \delta_2 - \omega_2 \\ U_3 =& a\delta_1\phi + a\delta_2\omega_1 - b\delta_1 - b\omega_1 + \delta_2 - \phi \end{aligned}$$

The set of eigenvalues for the fifth equilibrium point is given by

$$\lambda_5 = \{\lambda_{51}, \lambda_{52}, \lambda_{53}\}$$

where

$$
\begin{aligned}
\lambda_{51} =& -\frac{A_1}{3U_0} - \frac{\sqrt[3]{\mu_3}}{3} - \frac{A_1^2 - 3A_2}{3U_0^2\sqrt[3]{\mu_3}} \\
\lambda_{52} =& -\frac{A_1}{3U_0} - \frac{\sqrt[3]{\mu_3}\left(-\frac{1}{2} - \frac{\sqrt{3}i}{2}\right)}{3} - \frac{A_1^2 - 3A_2}{3U_0^2\sqrt[3]{\mu_3}\left(-\frac{1}{2} - \frac{\sqrt{3}i}{2}\right)} \\
\lambda_{53} =& -\frac{A_1}{3U_0} - \frac{\sqrt[3]{\mu_3}\left(-\frac{1}{2} + \frac{\sqrt{3}i}{2}\right)}{3} - \frac{A_1^2 - 3A_2}{3U_0^2\sqrt[3]{\mu_3}\left(-\frac{1}{2} + \frac{\sqrt{3}i}{2}\right)}
\end{aligned}
$$

and

$$
\begin{aligned}
\mu_3 =& \frac{2A_1^3 - 9A_1A_2 + 27A_3 + \sqrt{-4\left(A_1^2 - 3A_2\right)^3 + \left(2A_1^3 - 9A_1A_2 + 27A_3\right)^2}}{2U_0^3} \\
A_1 =& -U_1a - U_2\omega_1 - U_3\omega_2 \\
A_2 =& U_1U_2a\omega_1 + U_1U_2 + U_1U_3a\omega_2 + U_1U_3b + U_2U_3\omega_1\omega_2 + U_2U_3\phi\psi \\
A_3 =& U_1U_2U_3\left(-a\omega_1\omega_2 - a\phi\psi - b\omega_1 + b\psi - \omega_2 - \phi\right)
\end{aligned}
$$

As the eigenvalues are given by complex expressions that are functions of all parameters, we will study its stability by both numerical simulation and Lyapunov direct method.

4.5 Global stability analysis

In this section, we study the global stability of equilibrium points of the system and the existence of periodic solution and chaos using Lyapunov functions, Bendixson-Dulac criteria, and linear algebra.

Theorem 4.5.1 **(EP_2).** *The equilibrium point EP_2 is globally stable in the xy_1y_2-space if :$\delta_1 > \frac{1}{a}$ and $\delta_2 > \frac{b}{a}$.*

Proof. In case $\delta_1 > \frac{1}{a}$ *and* $\delta_2 > \frac{b}{a}$ the equilibrium EP_2 is the only locally stable equilibrium point in xy_1y_2-space so it must be globally stable there as well. □

Theorem 4.5.2 **(EP_3).** *The equilibrium point EP_3 is globally asymptotically stable in the positive xy_1-plane if it is locally asymptotically stable there.*

Proof. Let the function $H_2(x, y_1) = 1/(xy_1), F_1 = x\,(1 - ax - y_1)\ \ , F_2 = y_1\,(-\delta_1 - \omega_1 y_1 + x)$ Clearly $H_2(x, y_1) >$ in the $x_1\ y_1$-plane . Then we have

$$BD_2 = \frac{\partial}{\partial x}\,(H_2F_1) + \frac{\partial}{\partial y_1}\,(H_2F_2) = -\frac{a}{y_1} - \frac{\omega_1}{x} < 0$$

i.e BD_2 doesn't change sign and is not identically zero in the positive x_1y_1-plane. Thus, by Bendixson-Dulac theorem, there exist no limit cycle in the positive x_1y_1-plane. This implies that if EP_3 is locally asymptotically stable in the positive xy_1-plane, then it will be globally stable there. □

Theorem 4.5.3 (EP_4). *The equilibrium point EP_4 is globally asymptotically stable in the positive xy_1-plane if it is locally asymptotically stable there.*

Proof. Let the function $H_3(x, y_1) = 1/(xy_1),\ F_2 = x\,(1 - ax - by_2)\ \ , F_3 = y_2\,(-b\omega_2 y_2 + bx - \delta_2)$ Clearly $H_2(x, y_2) > 0$ in the $x_1\ y_1$-plane. Then we have

$$BD_3 = \frac{\partial}{\partial x}\,(H_3F_1) + \frac{\partial}{\partial y_2}\,(H_3F_3) = -\frac{a}{y_2} - \frac{b\omega_2}{x} < 0$$

i.e. BD_3 doesn't change sign and is not identically zero in the positive x_1y_1-plane. Thus, by Bendixson-Dulac theorem, there exists no limit cycle in the positive x_1y_1-plane. This implies that if EP_3 locally asymptotically stable in the positive xy_1-plane, then it will be globally stable there. □

Theorem 4.5.4 (EP_5). *Given an equilibrium point $ep = (x_e, y_{1_e}, y_{2_e})$. A candidate for Lyapunov function, which is frequently used in literature (see for example [39]),is the following:*

$$\begin{aligned} V(x, y_1, y_2) =& p_1\left(x - x_e \log\left(\frac{x}{x_e}\right) - x_e\right) + p_2\left(y_1 - y_{1e}\log\left(\frac{y_1}{y_{1e}}\right) - y_{1e}\right) \\ &+ p_3\left(y_2 - y_{2e}\log\left(\frac{y_2}{y_{2e}}\right) - y_{2e}\right) \end{aligned}$$

Where p_1, p_2, p_3 are positive constants

Clearly $V > 0$ for all $(x, y1, y2) \neq (x_e, y_{1_e}, y_{2_e})$ and $V\,(x_e, y_{1_e}, y_{2_e}) = 0$

The derivative of V is given by:

$$\begin{aligned}\frac{dV}{dt} =& p_1 x\left(1-\frac{x_e}{x}\right)(-ax-by_2-y_1+1)+p_2y_1\left(1-\frac{y_{1e}}{y_1}\right)(-b\psi y_2-\delta_1-\omega_1 y_1+x)\\ &+p_3y_2\left(1-\frac{y_{2e}}{y_2}\right)(-b\omega_2 y_2+bx-\delta_2+\phi y_1)\end{aligned}$$

After some algebraic manipulation and using techniques of transforming functions into quadratic form we obtain,

$$\frac{dV}{dt}=-\frac{1}{2}X^TMX$$

where

$$M = DA + A^TD$$

$$X=\begin{bmatrix} x-x_e \\ \\ y_1-y_1 \\ \\ y_2-y_{2e} \\ \end{bmatrix} \quad , \quad D=\begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix} \quad , \quad A=\begin{bmatrix} a & 1 & b \\ -1 & \omega_1 & b\psi \\ -b & -\phi & b\omega_2 \end{bmatrix}$$

If we can find p_1, p_2, p_3 that make the matrix M positive definite, then V will be a Lyapunov function and the interior equilibrium point EP_5 will be globally stable.

Proof. We have

$$M=\begin{bmatrix} 2ap_1 & p_1-p_2 & b(p_1-p_3) \\ p_1-p_2 & 2\omega_1 p_2 & bp_2\psi-p_3\phi \\ b(p_1-p_3) & bp_2\psi-p_3\phi & 2b\omega_2 p_3 \end{bmatrix}$$

. Choose $p_1=1, p_2=1, p_3=1$ then the matrix M becomes

$$M=\begin{bmatrix} 2a & 0 & 0 \\ 0 & 2\omega_1 & b\psi-\phi \\ 0 & b\psi-\phi & 2b\omega_2 \end{bmatrix}$$

. The matrix M is positive definite if all the leading principal minors are positive.

The first leading principal minor is $M_0 = \begin{vmatrix} 2a \end{vmatrix} \Rightarrow det(M_0) = 2a > 0$

The second leading principal minor is

$$M_1 = \begin{bmatrix} 2a & 0 \\ 0 & 2\omega_1 \end{bmatrix} \Rightarrow det(M1) = 4a\omega_1 > 0$$

The third leading principal minor is matrix M itself.

$$det(M) = 2a \left(4b\omega_1\omega_2 - (b\psi - \phi)^2\right) > 0$$

if $4b\omega_1\omega_2 > (b\psi - \phi)^2$ thus the matrix M is positive definite and EP_5 is globally stable if the last condition holds. $\square$

Chapter 5

Global Sensitivity Analysis (GSA)

In this section, we will study the global sensitivity of the parameters of the nondimensional model 4.3 together with the initial values to classify which parameters are influential and which are not influential on the model output. The model studied here has three outputs and each output is a time series. Therefore, we need to choose some variables of interest to study their sensitivity. Her we choose the max, mean, median, and equilibrium value for each output.

5.1 Methodology

The methodology we used to perform sensitivity analysis is as follows,

- All the results are conducted using the SALib package for sensitivity analysis in python. [19].
- We will use four methods of GSA which are Sobol, Morris, Random Balance Designs - Fourier Amplitude Sensitivity Test (RBD-FAST), and Delta Moment-Independent Measure (DMIM) and compare the results.
- The number of samples and the sampling scheme used for each method is as in the following table:

Sensitivity method	Sobol	Morris	RBD-Fast	DMIM
Number of samples N	30000	15000	100000	100000
Number of model evaluation	$D\,(N+2)$	$D\,(N+1)$	N	N
Sampling method used	Saltelli	Morris	Latin hypercube	Latin hypercube

Table 5.1: Number of samples and samplings scheme for different sensitivity analysis methods

where D is the number of parameters or input factors.

- For all methods, we use the following intervals for parameters which we assume they include the true values for the parameters. The intervals are chosen based on dimensional parameters which in turn chosen based on some literature, for example, [33] and [56].

$$\begin{aligned}
a =&[10^{-5}, 10^{-3}],\ b = [0.1, 10],\ \delta_1 = [0.1, 10],\ \delta_2 = [0.1, 10],\\
\psi =&[0.1, 10]\ \phi = [0.0001, 5],\ \omega_1 = [0.001, 10],\ \omega_2 = [0.001, 5],\\
x_0 =&[0, 10000],\ y_{1_0} = [0, 1000],\ y_{2_0} = [0, 100]
\end{aligned}$$

- Validating model output
  1. We run the model using Python ODE solver $solv_{ivp}$ with $RK45$ method which is an explicit method and with relative tolerance $rtol = 10^{-9}$ and absolute tolerance $atol = 10^{-9}$ which is proved to work best after extensive testing.
  2. We compute statistics for the model output for comparison's sake.
  3. We compute statistics for positive stable eqm. points (EP_3, EP_4, EP_5) and index of parameters for which there isn't any stable equilibrium point.
  4. We compute the max value of the output using boundedness of the solutions for all parameter sets and compute its statistics, we also find the max of the max here for comparison's sake.
  5. There are parameter sets for which the system does not have any stable equilibrium points (limit cycle, chaos). For these sets we choose the max of the last 500 outputs as equilibrium values.
  6. We filter out parameter sets that give unreasonable model output (output greater than the max value from boundedness, negative, not equal to stable eqm. points

values from the analytical solution given that the point is stable) and run the model again until we obtain the expected values. If the ODE solver can't compute the correct output after trying with all methods and options we set eqm. values equal to the ones from the analytical solution.

7. It worth noting that ODE solvers may give very small negative values of order 10^{-10} or less instead of zero, and it is best to set them to zero as they should be because this will lead an error if one will perform a log transformation of the model output before running the sensitivity analysis.

After validating the model output, we run the sensitivity analysis which needs another validation procedure which we will explain in the next step.

- Validating sensitivity analysis results

  We will summarize the criteria for reasonable sensitivity analysis results for each method in the following table:

Sensitivity method	Sobol	Morris	RBD-Fast	DMIM
Criteria	$\sum_i S_i \leq 1$ $\sum_i S_i = 1 \leftrightarrow$ No interactions $\sum_i S_{T_i} \geq \sum_i S_i$	None	$\sum_i S_i \leq 1$	$0 \leq \delta_i \leq 1$ $\sum_i S_i \leq 1$ $\sum_{i=1}^{n} \delta_i = 1 \leftrightarrow$ No interactions

Table 5.2: Criteria for reasonable sensitivity analysis results

The Delta method is sensitive to outliers. To reduce the effect of outliers we may need to apply log transformation or square root to the model output before applying sensitivity analysis, but our tests showed that these transformations change the sensitivity index of the parameters. Other options are modifying outliers to reduce their effects and for this, we use the IQR test to find outliers and then modify them. It worth noting that outliers appear when there is not any stable eqm. points in the system.

5.2 Sensitivity analysis results

The following tables summarize the results of sensitivity analysis. In these tables the green color indicates the highest values, then comes yellow, and finally, the red color indicates the lowest values. Additionally, all values are rounded to two decimals for the ease of analysis.

5.2.1 Maximum value of x, y_1 and y_2 respectively

Max	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM
a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
b	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.05
δ_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
δ_2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ψ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ϕ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_2	0.00	0.00	0.00	0.00	0.00	0.00	0.34	0.09	0.15
x_0	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.11	0.11
$\sum$	1.00	1.00	1.00	1.00	1.00	1.00	0.52	0.24	0.31

Table 5.3: Comparison of first-order sensitivity index S_1 from Sobol, RBD-FAST and DMIM methods for max value all variables

Max	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Morris	DMIM	Sobol	Morris	DMIM	Sobol	Morris	DMIM
a	0.00	0.00	0.02	0.00	0.39	0.03	0.00	38.35	0.02
b	0.00	0.00	0.02	0.00	76.65	0.03	0.53	3111.51	0.04
δ_1	0.00	0.00	0.02	0.00	15.04	0.03	0.00	80.51	0.02
δ_2	0.00	0.00	0.02	0.00	2.50	0.03	0.01	380.12	0.02
ψ	0.00	0.00	0.02	0.00	43.72	0.03	0.01	2206.60	0.02
ϕ	0.00	0.00	0.02	0.00	96.80	0.03	0.01	1876.67	0.02
ω_1	0.00	0.00	0.02	0.00	98.73	0.03	0.01	2024.91	0.02
ω_2	0.00	0.00	0.02	0.00	15.73	0.03	0.77	4141.93	0.03
x_0	1.00	10000.00	0.96	0.00	63.34	0.03	0.10	1651.83	0.02
y_{10}	0.00	0.00	0.02	1.00	998.55	0.97	0.00	273.36	0.02
y_{20}	0.00	0.00	0.02	0.00	30.29	0.03	0.14	329.42	0.41
$\sum$	1.00	10000.00	1.16	1.00	1441.74	1.27	1.58	16115.21	0.64

Table 5.4: Comparison between S_T Sobol, $\mu^\star$ Morris and δ from DMIM for max value of all variables

5.2.2 Mean value of x, y_1 and y_2 respectively

Mean	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM
a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
b	0.72	0.72	0.72	0.56	0.52	0.54	0.45	0.44	0.44
δ_1	0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.00	0.01
δ_2	0.08	0.07	0.07	0.02	0.02	0.02	0.03	0.03	0.03
ψ	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00
ϕ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00
x_0	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	0.02	0.02	0.03	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\sum$	0.84	0.83	0.83	0.63	0.59	0.62	0.49	0.47	0.48

Table 5.5: Comparison of first-order sensitivity index S_1 from Sobol, RBD-FAST, and DMIM methods for mean value all variables

Mean	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Morris	DMIM	Sobol	Morris	DMIM	Sobol	Morris	DMIM
a	0.00	0.04	0.07	0.00	0.00	0.13	0.00	0.04	0.11
b	0.88	5.34	0.41	0.90	0.48	0.16	0.99	4.10	0.74
δ_1	0.08	1.76	0.07	0.17	0.14	0.13	0.07	0.78	0.11
δ_2	0.15	1.54	0.15	0.28	0.12	0.13	0.47	3.16	0.09
ψ	0.02	0.54	0.07	0.10	0.12	0.13	0.03	1.45	0.11
ϕ	0.00	0.31	0.06	0.03	0.18	0.13	0.06	1.95	0.11
ω_1	0.04	1.63	0.07	0.05	0.17	0.13	0.02	1.46	0.11
ω_2	0.05	1.50	0.10	0.04	0.10	0.13	0.06	3.90	0.10
x_0	0.01	3.53	0.10	0.00	0.20	0.13	0.00	1.72	0.10
y_{10}	0.00	0.05	0.06	0.02	0.14	0.18	0.00	0.17	0.11
y_{20}	0.00	0.05	0.06	0.00	0.03	0.13	0.00	0.26	0.11
$\sum$	1.23	16.29	1.22	1.59	1.68	1.51	1.70	18.99	1.80

Table 5.6: Comparison between S_T Sobol, $\mu^\star$ Morris, and δ from DMIM for mean value of all variables

5.2.3 Median value of x, y_1 and y_2 respectively

Median	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast	DMIM
a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
b	0.72	0.72	0.72	0.56	0.53	0.55	0.42	0.43	0.43
δ_1	0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.00	0.01
δ_2	0.08	0.07	0.08	0.02	0.02	0.02	0.03	0.03	0.03
ψ	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00
ϕ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.02	0.02	0.02	0.00	0.00	0.00	0.01	0.00	0.00
x_0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
y_{20}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\sum$	0.83	0.82	0.83	0.61	0.58	0.60	0.47	0.46	0.47

Table 5.7: Comparison of first order sensitivity index S_1 from Sobol, RBD-FAST, and DMIM methods for median value all variables

Median	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Morris	DMIM	Sobol	Morris	DMIM	Sobol	Morris	DMIM
a	0.00	0.06	0.07	0.00	0.00	0.17	0.00	0.01	0.10
b	0.89	5.18	0.44	0.93	0.37	-44211.65	0.99	0.84	0.74
δ_1	0.08	1.66	0.08	0.18	0.04	0.16	0.08	0.13	0.10
δ_2	0.15	1.55	0.16	0.29	0.11	0.16	0.49	0.75	0.09
ψ	0.02	0.46	0.07	0.11	0.04	0.17	0.04	0.13	0.10
ϕ	0.00	0.29	0.07	0.03	0.05	0.17	0.06	0.27	0.10
ω_1	0.05	1.92	0.07	0.06	0.09	0.17	0.03	0.12	0.10
ω_2	0.05	1.50	0.10	0.05	0.06	0.17	0.07	0.41	0.10
x_0	0.00	2.37	0.07	0.00	0.13	0.17	0.00	0.34	0.10
y_{10}	0.00	0.04	0.07	0.00	0.00	0.17	0.00	0.00	0.10
y_{20}	0.00	0.05	0.07	0.00	0.01	0.17	0.00	0.02	0.10
$\sum$	1.24	15.08	1.27	1.65	0.90	-44209.97	1.76	3.02	1.73

Table 5.8: Comparison between S_T Sobol, $\mu^\star$ Morris and δ from DMIM for median all variables

5.2.4 Eqm. value of x, y_1 and y_2 respectively

~Eqm	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Rbd-Fast	DMIM	Sobol	Rbd-Fast 2	DMIM	Sobol	Rbd-Fast	DMIM
a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
b	0.72	0.70	0.70	0.41	0.22	0.23	0.10	0.06	0.06
δ_1	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00
δ_2	0.07	0.07	0.07	0.01	0.01	0.01	0.00	0.00	0.00
ψ	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
ϕ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω_1	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00
x_0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
y_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
y_{20}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\sum$	0.82	0.80	0.80	0.44	0.24	0.25	0.10	0.06	0.06

Table 5.9: Comparison of first order sensitivity index S_1 from Sobol, RBD-FAST, and DMIM methods for eqm. value of all variables

Eqm	Sensitivity of x variable			Sensitivity of y_1 variable			Sensitivity of y_2 variable		
Si method	Sobol	Morris	DMIM	Sobol	Morris	DMIM	Sobol	Morris	DMIM
a	0.00	0.25	0.09	0.00	0.06	0.46	0.00	6.18	0.10
b	0.89	5.78	0.42	1.02	0.60	7.95	0.27	7.40	0.76
δ_1	0.10	2.12	0.10	0.38	0.22	0.42	0.20	7.41	0.12
δ_2	0.18	1.93	0.15	0.69	0.30	0.79	0.81	13.91	0.23
ψ	0.02	0.83	0.09	0.28	0.19	0.38	0.36	3.75	0.14
ϕ	0.02	0.56	0.09	0.36	0.18	0.41	0.94	2.43	0.14
ω_1	0.07	2.01	0.09	1.05	0.26	0.26	0.23	4.60	0.14
ω_2	0.06	1.79	0.11	0.20	0.20	0.46	0.14	7.53	0.16
x_0	0.00	2.70	0.09	0.00	0.25	0.47	0.00	5.01	0.11
y_{10}	0.00	0.13	0.09	0.00	0.03	0.46	0.00	4.42	0.12
y_{20}	0.00	0.13	0.09	0.00	0.02	0.49	0.00	0.03	0.11
$\sum$	1.34	18.23	1.41	3.98	2.31	12.55	2.95	62.67	2.13

Table 5.10: Comparison between S_T Sobol, $\mu^\star$ Morris, and δ from DMIM for eqm. values of all variables

Parameter	a	b	δ_1	δ_2	ψ	ϕ	ω_1	ω_2	x_0	y_{10}	y_{20}	$\sum$
Median value of y_{10}	0.14	0.16	0.16	0.15	0.15	0.14	0.14	0.38	0.14	0.14	0.14	1.84
Eqm. value of y_{10}	0.18	0.17	0.17	0.17	0.17	0.18	0.18	0.18	0.18	0.18	0.18	1.94

Table 5.11: δ sensitivity index from DMIM for median and equilibrium value of y_1 after log-transformation.

5.3 Summary and concluding remarks

Let us summarize the information from the previous tables into three compact tables. The first table (Table 5.12) represents the most sensitive parameters overall, and the second (Table 5.13) and third (Table 5.14) tables represent the classification of parameters into sensitive and moderate or nonsensitive in decreasing order from left to right in terms of first-order effect and total/interactions effect, respectively.

S_1,Variable of interest	Max	Mean	Median	Eqm
x	x_0	b	b	b
y_1	y_{10}	b	b	b
y_2	ω_2, y_{20}	b	b	b
S_T,Variable of interest	Max	Mean	Median	Eqm
x	x_0	b	b	b
y_1	y_{10}	b, y_{10}	b, ω_2	b, ω_1
y_2	ω_2, y_{20}	b	b	ϕ, δ_2, b

Table 5.12: The most sensitive parameter for each variable of interest in terms of main effect total/interactions effects. Note that for variables there are more than one parameter due to the fact that all methods of sensitivity did not agree

	S_1	Max	Mean	Median	Eqm
x	Sobol	$[x_0]$, [all others)]	$[b, \delta_2, \omega_2, x_0, \delta_1]$, $[\omega_1, \psi, \phi, (a, y_{10}, y_{20})]$	$[b, \delta_2, \omega_2, \ \delta_1]$, $[\omega_1, \psi, \phi, (a, x_0, y_{10}, y_{20})]$	$[b, \delta_2, \omega_2, \ \delta_1]$, $[\omega_1, \phi, (a, \psi, x_0, y_{10}, y_{20})]$
	RBD-Fast	$[x_0]$, [all others)]	$[b, \delta_2, \omega_2, x_0, \delta_1]$, $[\omega_1, \psi, (a, \phi, y_{10}, y_{20})]$	$[b, \delta_2, \omega_2, \ \delta_1]$, $[\omega_1, \psi, (a, \phi, x_0, y_{10}, y_{20})]$	$[b, \delta_2, \omega_2, \delta_1]$, $[\omega_1, \psi(a, \phi, x_0, y_{10}, y_{20})]$
	DMIM	$[x_0]$, [all others)]	$[b, \delta_2, \omega_2, x_0, \delta_1]$, $[\omega_1, \psi, \phi, y_{20}, (a, y_{10})]$	$[b, \delta_2, \omega_2, \delta_1]$, $[\omega_1, \psi, (\phi, x_0), y_{20}, (a, y_{10})]$	$[b, \delta_2, \omega_2, \ \delta_1]$, $[\omega_1, \psi, \phi, (x_0, y_{20}), (a, y_{10})]$
y_1	Sobol	$[y_{10}]$, [(all others)]	$[b, y_{10}, \delta_2, \delta_1, \psi]$, $[\omega_1, \phi, \omega_2, x_0, \ y_{20}, a]$	$[b, \delta_2, \delta_1, \psi]$, $[\phi, \omega_1, \omega_2, (a, x_0, y_{10}, y_{20})]$	$[b, \delta_2, \delta_1, \psi]$, $[\phi, \omega_1, \omega_2, (a, x_0, y_{10}, y_{20})]$
	RBD-Fast	$[y_{10}]$, [(all others)]	$[b, y_{10}, \delta_2, \delta_1, \psi]$, $[(\omega_1, \omega_2), \phi, (a, x_0, y_{20})]$	$[b, \delta_2, \delta_1, \psi]$, $[\omega_1, \omega_2, \phi, (a, x_0, y_{10}, y_{20})]$	$[b, \delta_2, \delta_1]$, $[\psi, \omega_1, (a, \phi, \omega_2, x_0, y_{10}, y_{20})]$
	DMIM	$[y_{10}]$, [(all others)]	$[b, y_{10}, \delta_2, \delta_1, \psi]$, $[\omega_1, \omega_2, \phi, a, x_0, y_{20}]$	$[b, \delta_2, \delta_1, \psi]$, $[\omega_1, \omega_2, \phi, a, (x_0, y_{10}), y_{20}]$	$[b, \delta_2, \delta_1]$, $[\psi, \omega_1, \phi, \omega_2, y_{10}, (a, x_0), y_{20}]$
y_2	Sobol	$[\omega_2, y_{20}, b]$, $[x_0, \psi, \delta_2, \phi, (a, \delta_1, \omega_1, y_{10})]$	$[b, \delta_2, \delta_1]$, $[\psi, \omega_2, \omega_1, \phi, (a, x_0, y_{10}, y_{20})]$	$[b, \delta_2, \delta_1, \omega_2]$, $[\psi, (\phi, \omega_1), (a, y_{10}), (x_0, y_{20})]$	$[b]$, $[\delta_2, \omega_1, \omega_2, \delta_1, \phi, (a, \psi, x_0, y_{10}, y_{20})]$
	RBD-Fast	$[y_{20}, \omega_2, b]$, $[(a, \delta_1, \delta_2, \psi, \phi, \omega_1, x_0, y_{10})]$	$[b, \delta_2]$, $[\delta_1, (\psi, \phi), (\omega_1, \omega_2), (a, x_0, y_{10}, y_{20})]$	$[b, \delta_2]$, $[\delta_1, (\psi, \phi), (\omega_1, \omega_2), (a, x_0, y_{10}), y_{20}]$	$[b]$, $[\delta_2, (\delta_1, \phi, \omega_2), (a, \psi, \omega_1, x_0, y_{10}, y_{20})]$
	DMIM	$[\omega_2, y_{20}, b]$, $[x_0 \delta_2, \phi, \delta_1, (a, \psi, \omega_1, y_{10})]$	$[b, \delta_2, \delta_1, \psi, \phi, \omega_2, \omega_1]$, $[y_{10}, (a, x_0, y_{20})]$	$[b, \delta_2, \delta_1, \omega_2]$, $[\psi, \phi, \omega_1, y_{10}, (a, x_0, y_{20})]$	$[b]$, $[\delta_2, (\delta_1, \phi, \omega_2), \omega_1, (\psi, x_0, y_{10}), (a, y_{20})]$

Table 5.13: First(Main) effect sensitivity indices for all variables of interest. In each cell, the first array includes the most sensitive and sensitive parameters, while the array below it contains the least sensitive and nonsensitive parameters. Note also that the parameters in the parenthesis are equal in terms of sensitivity.

	Si index	Max	Mean	Median	Eqm
x	s_T	$[x_0]$, [(all others)]	$[b, \delta_2, \delta_1, \omega_2, \omega_1, \psi]$, $[x_0, \phi, (a, y_{10}, y_{20})]$	$[b, \delta_2, \delta_1, \omega_2, \omega_1, \psi]$, $[\phi, y_{10}, a, x_0, y_{20}]$	$[b, \delta_2, \delta_1, \omega_1, \omega_2, \psi, \phi]$, $[x_0, (a, y_{10}, y_{20})]$
	$\mu^\star$	$[x_0]$, [(all others)]	$[b, x_0, \delta_1, \omega_1, \delta_2, \omega_2]$, $[\psi, \phi, y_{20}, y_{10}, a]$	$[b, x_0, \omega_1, \delta_1, \delta_2, \omega_2]$, $[\psi, \phi, a, y_{20}, y_{10}]$	$[b, x_0, \delta_1, \omega_1, \delta_2, \omega_2]$, $[\psi, \phi, a, y_{20}, y_{10}]$
	δ	$[x_0]$, [(all others)]	$[b, \delta_2, x_0, \omega_2, \delta_1, \omega_1, \psi, a]$, $[(\phi, y_{20}, y_{10})]$	$[b, \delta_2, \omega_2, \delta_1]$, $[\omega_1, \psi, a, y_{20}, x_0, \phi, y_{10}]$	$[b, \delta_2, \omega_2, \delta_1]$, $[\omega_1, \psi, a, x_0, y_{20}, \phi, y_{10}]$
y_1	s_T	$[y_{10}]$, [(all others)]	$[b, \delta_2, \delta_1, \psi, \omega_1, \omega_2]$, $[\phi, y_{10}, x_0, y_{20}, a]$	$[b, \delta_2, \delta_1, \psi, \omega_1, \omega_2, \phi]$, $[x_0, y_{20}, (a, y_{10})]$	$[\omega_1, b, \delta_2, \delta_1, \phi, \psi, \omega_2]$, $[x_0, y_{20}, a, y_{10}]$
	$\mu^\star$	$[y_{10}]$, [(all others)]	$[b, x_0, \phi, \omega_1, \delta_1, y_{10}]$, $[\delta_2, \psi, \omega_2, y_{20}, a]$	$[b, x_0, \delta_2, \omega_1, \omega_2, \phi, \psi, \delta_1]$, $[y_{20}, y_{10}, a]$	$[b, \delta_2, \omega_1, x_0, \delta_1, \omega_2, \psi, \phi]$, $[a, y_{10}, y_{20}]$
	δ	$[y_{10}]$, [(all others)]	$[y_{10}, b]$, $[\psi, \delta_2, \omega_2, \phi, x_0, y_{20}, a, \omega_1, \delta_1]$	$[\omega_2, b, \delta_1, \delta_2, \psi]$, $[\phi, (a, \omega_1, x_0, y_{10}, y_{20})]$	$[(a, \phi, \omega_1, x_0, y_{10}, y_{20}, \omega_2)]$, $[b, \delta_2, \psi, \delta_1]$
y_2	s_T	$[\omega_2, b, y_{20}, x_0, \omega_1, \phi, \psi, \delta_2]$, $[y_{10}, (a, \delta_1)]$	$[b, \delta_2, \delta_1, \phi, \omega_2, \psi, \omega_1]$, $[x_0, y_{20}, (a, y_{10})]$	$[b, \delta_2, \delta_1, \omega_2, \phi, \psi, \omega_1]$, $[x_0, y_{10}, y_{20}, a]$	$[\phi, \delta_2, \psi, b, \omega_1, \delta_1, \omega_2]$, $[x_0, (a, y_{10}, y_{20})]$
	$\mu^\star$	$[\omega_2, b, \psi, \omega_1, \phi, x_0]$, $[\delta_2, y_{20}, y_{10}, \delta_1, a]$	$[b, \omega_2, \delta_2, \phi, x_0, \omega_1, \psi]$, $[\delta_1, y_{20}, y_{10}, a]$	$[b, \delta_2, \omega_2, x_0, \phi, \psi, \omega_1, \delta_1]$, $[y_{20}, a, y_{10}]$	$[\delta_2, \omega_2, \delta_1, b, a, x_0, \omega_1]$, $[y_{10}, \psi, \phi, y_{20}]$
	δ	$[y_{20}, b, \omega_2]$, $[\omega_1, \phi, \psi, \delta_2, \delta_1, x_0, y_{10}, a]$	$[b, \delta_1, \omega_1, \psi, y_{10}, a, y_{20}, \phi]$, $[x_0, \omega_2, \delta_2]$	$[b, \delta_1, \omega_1, \psi, y_{10}, a, \phi, y_{20}, x_0, \omega_2]$, $[\delta_2]$	$[b, \delta_2, \omega_2, \omega_1, \psi, \phi]$, $[\delta_1, y_{10}, x_0, y_{20}, a]$

Table 5.14: Total/interactions effect sensitivity indices for all variables of interest. In each cell, the first array includes the most sensitive and sensitive parameters, while the array below it contains the least sensitive and nonsensitive parameters. Note also that the parameters in the parenthesis are equal in terms of sensitivity.

5.3.0.1 General results

1. In tables 5.13 and 5.14 the first array in each cell include the most sensitive and sensitive parameters, while the array below it contains the least sensitive and nonsensitive parameters. The parameters in each vector are arranged in decreasing order in terms of sensitivity.

2. $\mu^\star$ from Morris method is comparable with Total effect S_T Sobol. Although the ranking

is different, Morris' method successfully classifies the most sensitive parameters as well as the nonsensitive ones.

3. δ index from DMIM gives mostly identical ranking as total effect S_T from Sobol method.
4. S_1 from Sobol, RBD-FAST Fast, and DMIM give mostly the same ranking for parameters and its values are approximately the same.
5. Max value x and y_1 are only sensitive to their corresponding initial values x_0 and y_{1_0}, while max value of y_2 is sensitive to ω_2, b, y_{2_0} as individual and interaction. This is because the sum of first-order indices does not equal unity, which indicates that there are interaction effects.
6. For mean, median, and eqm. values, the parameter b is the most sensitive one in terms of the first-order effect, and there is more effect of interactions than the first-order effect. This is clear by observing the sum of indices in tables 5.5, 5.7, and 5.9.
7. In terms of total or interactions effects the mean, median, eqm. values of x, y_1 and y_2 are sensitive to the parameters $\{b, \delta_2, \delta_1, \omega_2, \omega_1, \psi, \phi\}$ with b being the most sensitive parameter except for eqm. of y_1 as ω_1 is the most sensitive one and eqm. value of y_2 as ϕ is the most sensitive one.
8. Overall, we can state that the sensitive parameters are $\{b, \delta_2, \delta_1, \omega_2, \omega_1, \psi, \phi\}$ and the nonsensitive parameters are $\{a, x_0, y_{1_0}, y_{2_0}\}$ if max value is not in consideration.
9. The most important parameter is b while the least important parameter is a.

Chapter 6

Observability

In this chapter we will study the observability of the system 4.3 in different cases and determine for which variable and under which condition the system is observable.

6.1 Observing bacteria x only

Consider the system 4.3 and the following observation function $h(\boldsymbol{x}) = x - x_e$, then the observability matrix is given by,

$$\begin{aligned} O &= \begin{bmatrix} C & CJ & CJ^2 \end{bmatrix}^T \\ &= \begin{bmatrix} 1 & -2ax - by_2 - y_1 + 1 & O_{13} \\ 0 & -x & x\left(2ax - b\phi y_2 + b\psi y_2 + by_2 + \delta_1 + 2\omega_1 y_1 - x + y_1 - 1\right) \\ 0 & -bx & bx\left(2ax + 2b\omega_2 y_2 - bx + by_2 + \delta_2 - \phi y_1 + \psi y_1 + y_1 - 1\right) \end{bmatrix} \end{aligned}$$

Where,

$$J = \begin{bmatrix} -2ax - by_2 - y_1 + 1 & -x & -bx \\ y_1 & -b\psi y_2 - \delta_1 - 2\omega_1 y_1 + x & -b\psi y_1 \\ by_2 & \phi y_2 & -2b\omega_2 y_2 + bx - \delta_2 + \phi y_1 \end{bmatrix}$$

$$\begin{aligned} O_{13} =& 4a^2x^2 + 4abxy_2 + 4axy_1 - 4ax - b^2xy_2 + b^2y_2^2 + 2by_1y_2 - 2by_2 - xy_1 + y_1^2 - 2y_1 + 1 \\ C =& \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y_1} & \frac{\partial h}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

The determinant of the observability matrix O is given by,

$$det(O) = bx^2\left(-2b\omega_2 y_2 - b\phi y_2 + b\psi y_2 + bx + \delta_1 - \delta_2 + 2\omega_1 y_1 + \phi y_1 - \psi y_1 - x\right)$$

If it is different than zero, then the rank of the matrix will be 3 and the system will observable.

6.2 Observing bacteria y_1 only

In this case the observation function is given by $h(\boldsymbol{x}) = y_1 - y_{1_e}$, and the observability matrix is given by,

$$\begin{aligned} O &= \begin{bmatrix} C & CJ & CJ^2 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & y_1 & O_{13} \\ 1 & -b\psi y_2 - \delta_1 - 2\omega_1 y_1 + x & O_{23} \\ 0 & -b\psi y_1 & O_{33} \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} O_{13} =& y_1 \left(-2ax - b^2\psi y_2 - b\psi y_2 - by_2 - \delta_1 - 2\omega_1 y_1 + x - y_1 + 1\right) \\ O_{23} =& b^2\psi^2 y_2^2 + 2b\delta_1\psi y_2 + 4b\omega_1\psi y_1 y_2 - b\phi\psi y_1 y_2 - 2b\psi x y_2 \\ & + \delta_1^2 + 4\delta_1\omega_1 y_1 - 2\delta_1 x + 4\omega_1^2 y_1^2 - 4\omega_1 x y_1 + x^2 - x y_1 \\ O_{33} =& by_1 \left(2b\omega_2\psi y_2 + b\psi^2 y_2 - b\psi x + \delta_1\psi + \delta_2\psi + 2\omega_1\psi y_1 - \phi\psi y_1 - \psi x - x\right) \\ C =& = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

The determinant of the observability matrix O is given by,

$$det(O) = by_1^2 \left(2a\psi x + b^2\psi^2 y_2 - 2b\omega_2\psi y_2 + b\psi x + b\psi y_2 - \delta_2\psi + \phi\psi y_1 + \psi y_1 - \psi + x\right)$$

If it is different than zero, then the rank of the matrix will be 3 and the system will be observable by only observing the predator species y_1.

6.3 Observing bacteria y_2 only

In this case the observation function is given by $h(\boldsymbol{x}) = y_1 - y_{2_e}$, and the observability matrix is given by,

$$\begin{aligned} O &= \begin{bmatrix} C & CJ & CJ^2 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & by_2 & O_{13} \\ 0 & \phi y_2 & O_{23} \\ 1 & -2b\omega_2 y_2 + bx - \delta_2 + \phi y_1 & O_{33} \end{bmatrix} \end{aligned}$$

where

$$
\begin{aligned}
O_{13} =& y_2 \left(-2abx - 2b^2\omega_2 y_2 + b^2 x - b^2 y_2 - b\delta_2 + b\phi y_1 - by_1 + b + \phi y_1\right) \\
O_{23} =& y_2 \left(-2b\omega_2\phi y_2 - b\phi\psi y_2 + b\phi x - bx - \delta_1\phi - \delta_2\phi - 2\omega_1\phi y_1 + \phi^2 y_1 + \phi x\right) \\
O_{33} =& 4b^2\omega_2^2 y_2^2 - 4b^2\omega_2 x y_2 + b^2 x^2 - b^2 x y_2 + 4b\delta_2\omega_2 y_2 \\
& - 2b\delta_2 x - 4b\omega_2\phi y_1 y_2 - b\phi\psi y_1 y_2 + 2b\phi x y_1 + \delta_2^2 - 2\delta_2\phi y_1 + \phi^2 y_1^2 \\
C =& = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

The determinant of the observability matrix O is given by,

$$
det(O) = y_2^2 \left(2ab\phi x - b^2\phi\psi y_2 + b^2\phi y_2 - b^2 x - b\delta_1\phi - 2b\omega_1\phi y_1 + b\phi x + b\phi y_1 - b\phi - \phi^2 y_1\right)
$$

If it is different than zero, then the rank of the matrix will be 3 and the system will be observable by only observing the predator species y_2.

Note that since the system is observable by observing each species alone, then it should be observable by observing any combination of them. Note also that the system is not observable at the zero equilibrium point regardless of the variable being observed because the determent of the observability matrix will be zero then.

Chapter 7

Bacteria and microorganisms data

In this chapter, we represent some available data regarding bacteria and microorganisms in surface drinking water in west Norway. The data regarding bacteria are obtained from Bergen municipality, and reports from Rådgivende Biologer website [21, 22].

7.1 Bacteria data from the untreated water from all water resources in Bergen

Bergen Municipality takes samples from all the water resources that are used to supply the community with drinking water. The samples are taken several times each month, but the frequency differs between months. In each sample, the number of pathogenic bacteria, Clostridium Perfringens (CP), Coliform Bacteria (CB), E. coli (EC), and Intestinal Enterococci (IE) are measured in number per $100ml$. The data analyzed in this work is from 2010 to 2019. Note that the Campylobacter jejuni bacteria type is not measured, thus we do not have any data concerning it.

The following tables give us the statistics of each individual pathogenic bacteria in all drinking water resources in Bergen municipality in Norway. In these tables

1. MeanAll is the mean of bacteria considering all water resources.
2. 25% ,50%, 75% are the 25, 50, 75 percentile respectively, and defined as the values below which 25%, 50%, 75% of data could be found.
3. std is the standard deviation.

	Espeland	Svartediket	Jordalen	Kismul	Risnæs	Sædal	Bogetveit	Gamsebotn	Raudtjørn	Sætervatn	MeanAll
count	285.00	304.00	305.00	304.00	305.00	301.00	139.00	139.00	137.00	138.00	986.00
mean	0.32	0.52	1.68	0.17	0.14	0.52	0.55	2.58	0.38	0.35	1.57
std	0.69	0.80	1.80	0.42	0.59	0.95	1.00	6.04	0.84	1.03	3.29
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50%	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
75%	0.00	1.00	3.00	0.00	0.00	1.00	1.00	3.00	0.00	0.00	2.00
max	5.00	4.00	10.00	2.00	7.00	7.00	8.00	52.00	6.00	9.00	53.00

Table 7.1: Statistics for Clostridium Perfringens bacteria in all waters in Bergen

	Espeland	Svartediket	Jordalen	Kismul	Risnæs	Sædal	Bogetveit	Gamsebotn	Raudtjørn	Sætervatn	MeanAll
count	584.00	510.00	590.00	502.00	607.00	503.00	204.00	199.00	202.00	202.00	986.00
mean	65.29	107.66	25.58	15.26	188.95	248.01	107.00	146.96	62.83	79.40	441.21
std	192.15	379.71	53.16	74.85	368.38	525.15	305.11	441.25	118.87	199.68	817.30
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	1.00	3.00	5.00	1.00	10.00	3.00	5.00	7.50	6.25	11.00	25.00
50%	9.00	9.00	11.00	3.00	43.00	23.00	13.50	22.00	22.50	28.00	108.00
75%	43.25	26.00	25.00	10.00	206.50	200.00	41.25	59.00	58.00	68.75	446.25
max	2420.00	2420.00	687.00	1553.00	2420.00	2420.00	2000.00	2420.00	816.00	2420.00	7481.00

Table 7.2: Statistics for Coliform Bacteria in all waters in Bergen

	Espeland	Svartediket	Jordalen	Kismul	Risnæs	Sædal	Bogetveit	Gamsebotn	Raudtjørn	Sætervatn	MeanAll
count	585.00	510.00	591.00	504.00	607.00	503.00	234.00	229.00	202.00	202.00	986.00
mean	13.06	1.33	1.83	0.23	39.19	1.33	0.86	0.61	0.23	0.55	34.95
std	50.77	3.92	3.32	0.85	187.92	3.67	3.93	1.77	0.74	2.03	158.16
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50%	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	3.00
75%	5.00	1.00	2.00	0.00	10.00	1.00	1.00	0.00	0.00	0.00	12.00
max	687.00	63.00	34.00	12.00	2420.00	44.00	57.00	17.00	8.00	19.00	2431.00

Table 7.3: Statistics for E.coli Bacteria in all waters in Bergen

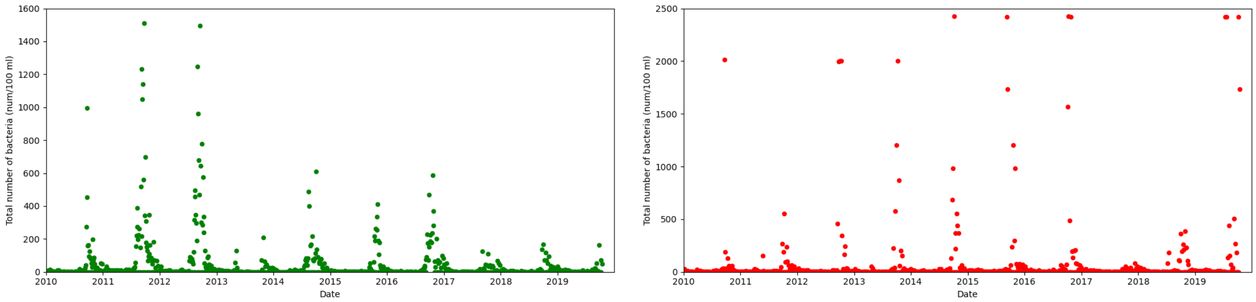
	Espeland	Svartediket	Jordalen	Kismul	Risnæs	Sædal	Bogetveit	Gamsebotn	Raudtjørn	Sætervatn	MeanAll
count	476.00	496.00	495.00	491.00	495.00	494.00	228.00	196.00	198.00	227.00	986.00
mean	0.23	0.37	0.74	0.05	2.43	0.18	0.55	0.78	0.07	0.15	2.34
std	1.05	1.03	1.67	0.38	7.31	0.58	6.50	3.65	0.48	0.54	6.80
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
75%	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	2.00
max	13.00	12.00	17.00	7.00	69.00	5.00	98.00	33.00	5.00	4.00	98.00

Table 7.4: Statistics for Intestinal Enterococci Bacteria in all waters in Bergen

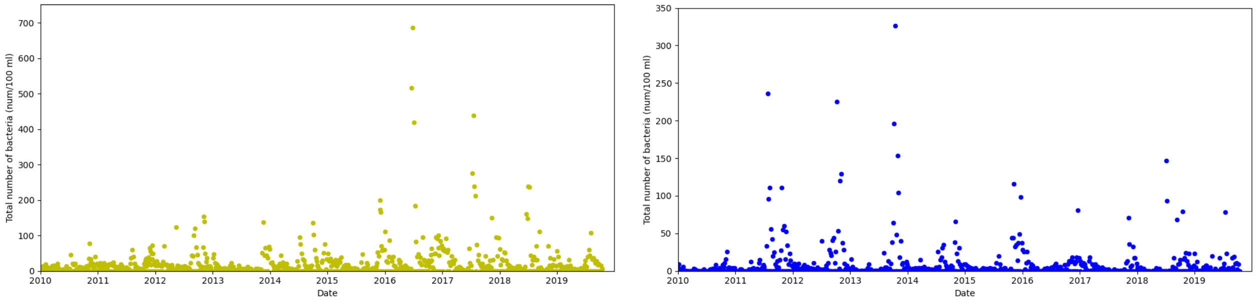
Table 7.5 and Figures (7.1 , 7.2 , 7.3) demonstrates the total number of pathogenic bacteria in all drinking water resources in Bergen municipality in Norway. Here again MeanTotal represent the mean of total number of bacteria considering all water resources.

	Espeland	Svartediket	Jordalen	Kismul	Risnæs	Sædal	Bogetveit	Gamsebotn	Raudtjørn	Sætervatn	MeanTotal
count	986	986	986	986	986	986	986	986	986	986	986
mean	47	57	17	8	140	127	22	30	13	16	48
std	180	279	44	54	422	396	145	207	60	96	88
min	0	0	0	0	0	0	0	0	0	0	0
25%	0	0	0	0	0	0	0	0	0	0	3
50%	1	0	4	0	6	0	0	0	0	0	12
75%	15	10	16	4	92	26	0	0	0	0	48
max	2687	2426	687	1553	4842	2464	2002	2428	818	2420	752

Table 7.5: Total number of pathogenic bacteria in different water resources in Bergen. The maximum value of bacteria occurs in Risnæs with value equal to 4842 and minimum value is in Jordalen 687

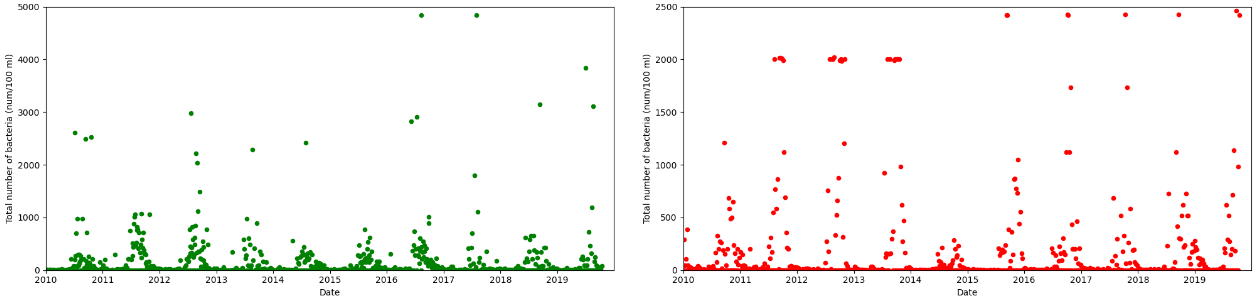


(a) Espeland (b) Svartediket



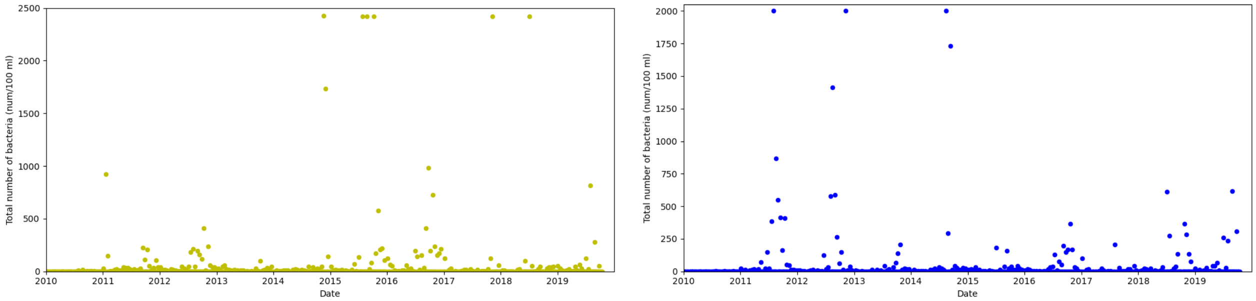
(c) Jordalen (d) Kismul

Figure 7.1: Total number of bacteria in Espeland, Svartediket, Jordalen, and Kismul water resources in Bergen



(a) Risnæs

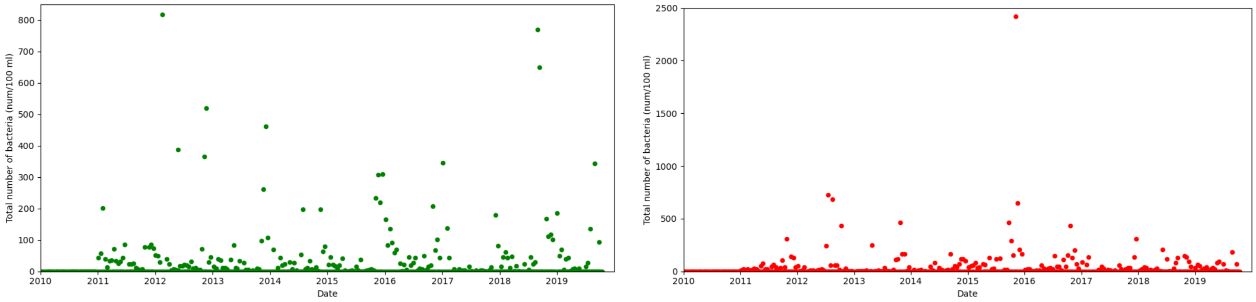
(b) Sædal



(c) Bogetveit

(d) Gamsebotn

Figure 7.2: Total number of bacteria in Risnæs, Sædal, Gamsebotn, and Bogetveit water resources in Bergen.



(a) Raudtjørn

(b) Sætervatn

Figure 7.3: Total number of bacteria in Raudtjørn, and Sætervatn water resources in Bergen.

From figures 7.1 ,7.2 , 7.3 we see some kind of periods of up-springs of bacteria and periods with zero or low numbers of bacteria which indicate some kind of cyclic behavior, but it doesn't seem to be periodic.

7.2 Daphnia

The following table which is taken from the report [21], gives us an indication about the total number of Daphnia (Cladocera) in five water resources in Bergen municipality.

Water resource, Date	20 Jun.	19 Jul.	15 Aug.	4 Sep.	2 Oct.	24 Oct.
Svartediket	4193	1970	957	610	246	210
Storediket	2131	709	314.3	473	454	207
Tarlebøvatnet	12234	3218	1077	903.4	376	1287.4
Nodre Gløvervatnet	1551.3	795	2791	1905	627.7	1908.7
Søndre Gløvervatnet	3832.4	1065	2621	6862.5	1172.5	3929

Table 7.6: Total number of all types of Daphnia in 5 water resources in Bergen in 2000 in num/m^3

Total number of Daphnia in Jordalsvannet on 19.Oct.2016 is 53 num/m^3 and Indrevatnet on 6.Oct.2016 is 912 num/m^3.

From the previous table, we can see that the total number of Daphnia population in Svartediket and Storediket decreases in the wintertime, while in the remaining water resources they vary monthly.

7.3 Protozoa

We haven't been able to find any data about the Protozoa for water resources in Bergen or West of Norway in general.

Chapter 8

Numerical simulations

In this chapter, we present a numerical simulation of the different solutions of the system 4.3 using different sets of parameters. For this purpose, we use Python 3.7.7 via Anaconda and Jupyter Notebook. We will mainly use the ODE solver RK45 with absolute tolerance 1×10^{-8} and relative tolerance 1×10^{-8} to solve the system numerically. For calculating the Lyapunov exponents we use Matalb2020a and the code by Vasiliy Govorukhin [18] which is based on Wolf method. We also wrote a python code (see appendix B.3) to calculate the largest Lyapunov exponent (LLE) based on Sprott method [52]. Additionally, we use Julia version 1.5.1 Dynamical system library [11] that has a function to calculate the spectrum of Lyapunov exponents and another one for the largest Lyapunov exponent.

We will study the effect of changing parameters, taking into consideration the sensitivity analysis. We will also study the stability of the interior equilibrium point. In general, the solution will converge to an eqm. point if there exists a stable one, but if there is not any stable equilibrium point, the solution is either periodic, chaotic or exactly stays in one of the unstable eqm. points if we start from it.

Let us assume the following values for the dimensional parameters,

k	r	f_1	f_2	β_1	β_2	ϵ_1	ϵ_2	ψ_{12}	ϕ_{12}	ω_{11}	ω_{22}
5000	0.28,27	1.6	1.8	0.47	0.47/4	0.47/2	0.25/2	2.8	0.34	0.5	0.5

Table 8.1: Biological values of the dimensional parameters.

Note that there are four other parameters which are included in the calculation of ψ_{12} and

ϕ_{12} and they are $g_{12} = 0.9, \gamma_{12} = 0.6, \omega_{12} = 1.9, \omega_{21} = 0.2$. As we did't perform parameter estimation, these values of parameters are given based on [25,33,47,48,56] and bacteria data we obtained from Bergen municipality. For example, we took k as the max number of sum of bacteria taking in account all water resources in Bergen.

We calculate the non-dimensional parameters from the previous dimensional ones and call it base parameter set

$$PM_{base} = [7.18 \times 10^{-5}, 0.28, 0.01, 0.46, 1.56, 0.21, 0.31, 0.28]$$

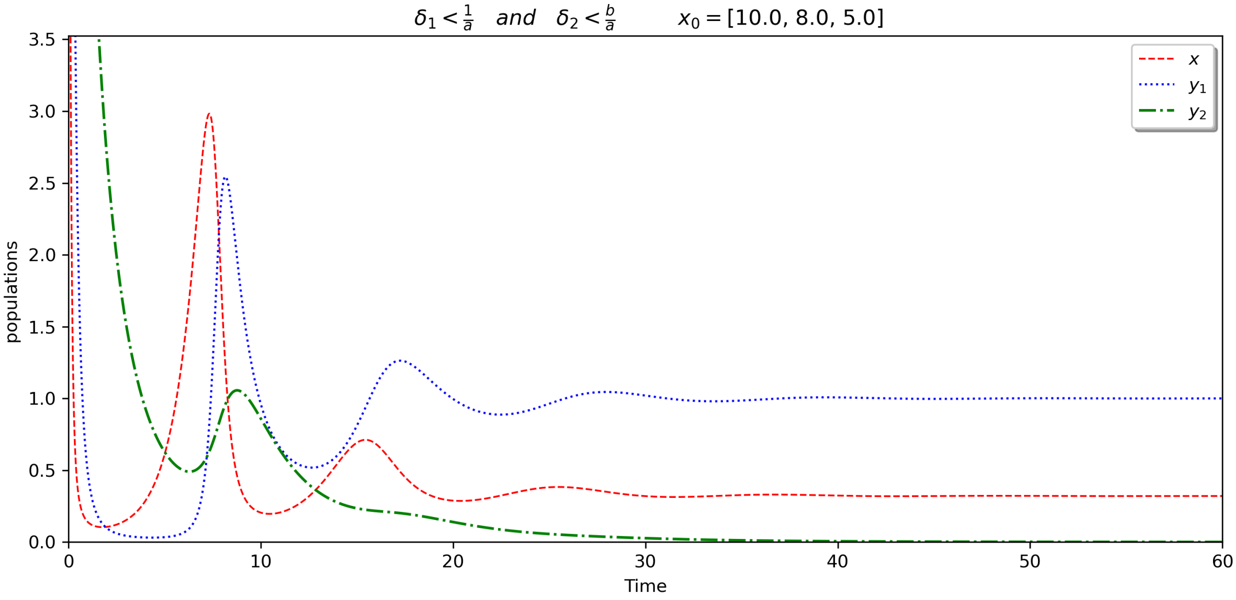
, then we study the effect of each parameter by changing only one parameter at a time in their sensitivity ranges. Note that we denote the parameter set as PM, and we will order it in a vector in the following order $PM = [a, b, \delta_1, \delta_2, \psi, \phi, \omega_1, \omega_2]$ with a subscript for each individual case.

For the base parameter set, local stability analysis is given in Table 8.2.

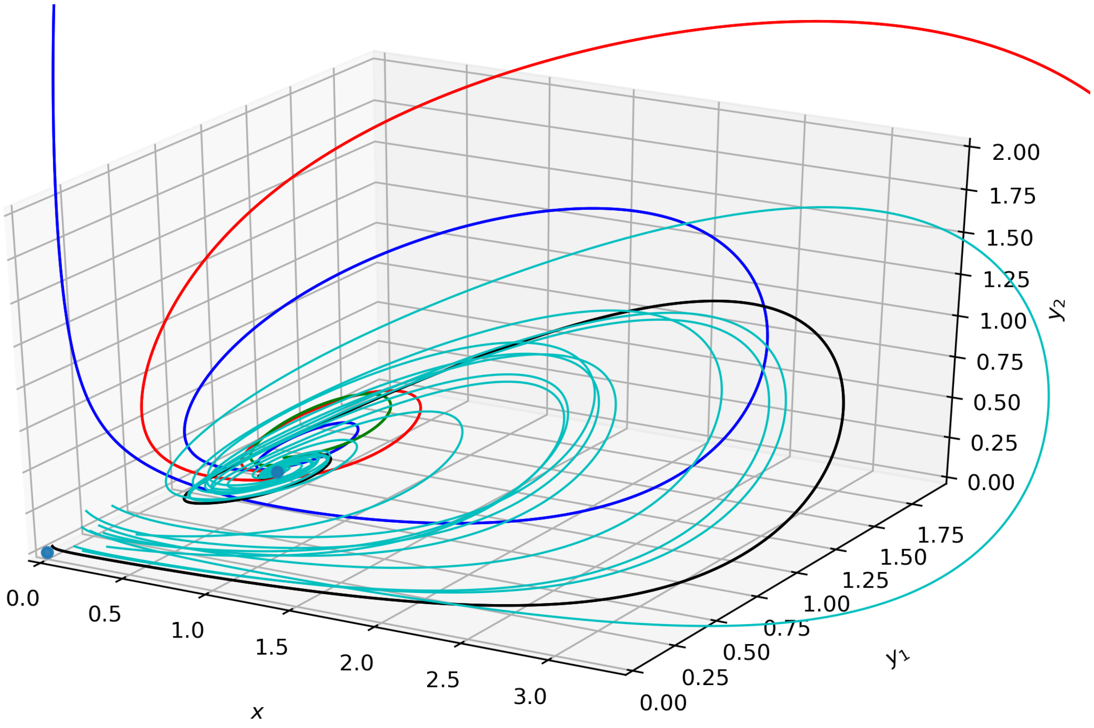
Equilibrium point	Eigenvalues	(Local) Stability
$(0, 0, 0)$	$[(1 + 0i), (-0.01 + 0i), (-0.46 + 0i)]$	Unstable
$(13927.58, 0, 0)$	$[(-1 + 0i), (13927.567 + 0i), (3899.261 + 0i)]$	Unstable
$(0.32, 1.0, 0.0)$	$[(-0.155 + 0.544i), (-0.155 - 0.544i), (-0.16 + 0i)]$	Stable
$(2.64, 0.0, 3.57)$	$[(-0.14 + 0.849i), (-0.14 - 0.849i), (1.073 + 0i)]$	Unstable
$(-1.11, 2.15, -4.09)$	$[(1.508 + 0i), (0.128 + 0i), (-1.981 + 0i)]$	Unstable

Table 8.2: Equilibrium points and their stability for the base parameter set PM_{base}.

From the above table, it is clear that the system has only one stable eqm. point $EP_3 = (0.32, 1.0, 0.0)$ and the solution converges to it as shown in Figure 8.1. Furthermore, the system does not have (positive) interior eqm. point for this parameter set.



(a) Time series for PM_{base}. All populations converge to the eqm. point $EP_3 = (0.32, 1.0, 0.0)$.



(b) Phase portrait for PM_{base} and several initial conditions. All trajectories converge to the eqm. point $EP_3 = (0.32, 1.0, 0.0)$.

Figure 8.1: Time series and phase portrait of the system for base parameter set.

8.1 Effect of parameter b

This parameter is given by $b = \frac{\beta_2 f_2}{\beta_1 f_1}$. Hence, it represents the ratio between the reproduction rate of the second predator y_2 to the first predator y_1 and is considered as the most sensitive parameter in the system. The following table demonstrates the characteristics of the system for different values of this parameter,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$b = \in [0.1, 0.471]$	EP_4	Extinction of the first predator and no (positive) interior eqm point	Figure 8.2
$b = \in [0.472, 0.781]$	EP_3, EP_4	Extinction of the first or second predator depending on the initial condition. No stable interior eqm. point.	Figure 8.3
$b = \in [0.782, 10]$	EP_4	Extinction of the first predator and no (positive) interior eqm point	Figure 8.4

Table 8.3: Characteristics of the system for different values b.

In conclusion, increasing this parameter will lead to the extinction of the predator y_1 (see Figures mentioned in the previous table).

8.2 Effect of parameter δ_1

This parameter is given by $\delta_1 = \frac{\epsilon_1}{r}$. Thus, it represents the ratio between the death rate of the predator y_1 and the growth rate of the prey species x and is the second most sensitive parameter. The following table demonstrates the characteristics of the system for different values of this parameter,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$\delta_1 = \in [0.1, 0.582]$	EP_3	Extinction of the second predator	Figure 8.5
$\delta_1 \in [0.583, 1.0819]$	EP_5	Coexistence of all three species	Figure 8.6
$\delta_1 \in [1.0829, 10]$	EP_4	Extinction of the first predator	Figure 8.7

Table 8.4: Characteristics of the system for different values δ_1.

From the previous table we conclude that for large values of δ_1, the first predator goes extinct because then the death rate will be much greater than the bacteria intrinsic growth rate.

8.3 Effect of parameter δ_2

This parameter is given by $\delta_2 = \frac{\epsilon_2}{r}$. Thus, it represents the ratio between the death rate of the predator y_2 and the growth rate of the prey species x and is the third most sensitive parameter. The following table demonstrates the characteristics of the system for different values of this parameter,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$\delta_2 \in [0.1, 0.1499]$	EP_4	Extinction of the first predator	Figure 8.8
$\delta_2 \in [0.1599, 0.299]$	EP_5	Coexistence of all second species	Figure 8.9
$\delta_2 =\in [0.3, 10]$	EP_3	Extinction of the second predator	Figure 8.10

Table 8.5: Characteristics of the system for different values δ_2.

We can conclude from the previous table that for large values of δ_2, the second predator goes extinct because then the death rate will be much greater than the bacteria intrinsic growth rate.

8.4 Effect of parameter ψ

This parameter is defined as $\psi = \frac{\psi_{12}}{f_2} = \frac{g_{12}+\omega_{12}}{f_2}$. Thus, it represents the ratio between the sum of predation rate of the predator y_2 with respect to predator y_1 and decreased rate of predator species y_1 due to interspecific competition with the predator species y_2 and the predation rate of the predator y_2 with respect to prey species x i.e. the ratio between two negative effects on the predator species y_1. This parameter is also among the sensitive ones. The characteristics of the system for different values of this parameter are presented in the following table,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$\psi \in [0.1, 2.63299]$	EP_3	Extinction of the second predator and no coexistence eqm. point	Figure 8.11
$\psi = \in [2.63399, 10]$	EP_3, EP_4	Extinction of the first or second predator depending on initial conditions and unstable coexistence eqm. point	Figure 8.12.

8.5 Effect of parameter ϕ

This parameter is defined as $\phi = \frac{\phi_{12}}{f_1} = \frac{\gamma_{12}g_{12} - \omega_{21}}{f_1}$. Thus, it represents the ratio between the difference between the reproduction rate of y_2 due to predation on y_1 and decreased rate of predator species y_2 due to interspecific competition with the predator species y_1 and the predation rate of the predator y_2 with respect to prey species x.

If $\phi \in [0.0001, 0.37]$, the system will have one stable eqm. point EP_3 and all solution that starts from points with $x > 0, y_1 > 0, y_2 \geq 0$ will converge to it (See Figure 8.13). For $\phi \in [0.371, 1.234]$ the system will have one stable eqm. Point EP_5 and all solutions that start from points with $x, y_1, y_1 > 0$ will converge to it (See Figure 8.14). Finally, if $\phi \geq 1.235$ then the system does not have any stable eqm. point and the solution tends toward a stable periodic orbit (see Figure 8.15). Thus, there is a bifurcation at $\phi = 0.37$ and $\phi = 1.234$ as the system changes character after these values.

8.6 Effect of parameter ω_1

This parameter is given by $\omega_1 = \frac{\omega_{11}}{f_1}$. Thus, it represents the ratio between intraspecific competition between the predator species y_1 and decreased rate of prey species x due to predation by the predator species y_1. The characteristics of the system for different values of this parameter are presented in Table 8.6,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$\omega_1 \in [0.1, 0.801]$	EP_3	Extinction of the second predator	Figure 8.18
$\omega_1 \in [0.901, 10]$	EP_5	Coexistence of all three species	Figure 8.19

Table 8.6: Characteristics of the system for different values of ω_1

It is evident from the previous table that for large values of ω_1, all three species will coexist for indefinite amount of time.

8.7 Effect of parameter ω_2

This parameter is given by $\omega_2 = \frac{\omega_{22}}{f_2}$. Thus, it represents the ratio between intraspecific competition between the predator species y_2 and decreased rate of prey species x due to predation by the predator species y_2. Our numerical tests showed that for all values of this parameter in its sensitivity range $[0.001, 5]$, the system converges to the third eqm. point i.e. the second predator is extinct. The following table gives some examples,

Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$\omega_2 = 0.1$	$EP_3 = (0.32, 1., 0.)$	Extinction of the second predator	Figure 8.20
$\omega_2 = 1$	$EP_3 = (0.32, 1., 0.)$	Extinction of the second species	Figure 8.21
$\omega_2 = 5$	$EP_3 = (0.32, 1., 0.)$	Extinction of the second predator.	Figure 8.22

Table 8.7: Characteristics of the system for different values of ω_2

However this is not surprising as the equilibrium values of x, y_1, y_2 are not very sensitive to ω_2 (see Tables 5.9 , 5.10 or Tables 5.13 , 5.14).

8.8 More than one stable eqm. point in the system

For some parameter sets, there are more than one stable equilibrium point. However, our numerical investigation showed that the only possible situation is two stable eqm. points which are EP_3 i.e. first predator y_1 goes extinct, and EP_4 i.e. second predator y_2 goes extinct. As an example, we give the following parameter set

$$PM_{34} = [0.000504, 0.68, 2.96, 5.53, 9.11, 0.51, 0.06, 1.83]$$

. In this case, the initial conditions determine which eqm. point the solution will converge to (see Figure 8.23.)

If we for example start with initial condition $x_0, y_{1_0}, y_{2_0} = (4, 3, 2)$ the solution tend to $EP_3 = (3.02, 1., 0.)$, but if we start with $x_0, y_{1_0}, y_{2_0} = (5, 0.02, 3)$ the solution tends to the stable eqm. point $EP_4 = (10.81, 0., 1.46)$.

8.9 Globally stable interior eqm. point EP_5

The interior equilibrium point is globally stable if $det(M) = 2a\left(4b\omega_1\omega_2 - (b\psi - \phi)^2\right) > 0$. There are many parameter sets that satisfy this condition and here we show one example

$$PM_{gs} = \left[6.95 \times 10^{-5}, 0.79, 5.58, 7.59, 3.1, 2.99, 1.75, 0.11\right]$$

and for this $det(M) = 4.17 \times 10^{-5} > 0$. To quantitatively check the local stability analysis of the system for this parameter set, we present the equilibrium points, their corresponding eigenvalues, and stability in Table 8.8.

Equilibrium point	Eigenvalues	(Local) Stability
$(0, 0, 0)$	$[(1 + 0i), (-5.58 + 0i), (-7.59 + 0i)]$	Unstable
$(14397.188, 0, 0)$	$[(-1 + 0i, (14391.61 + 0i), (11384.33 + 0i)]$	Unstable
$(7.33, 0.999, 0)$	$[(-0.88 + 2.56i, (-0.88 - 2.56i), (1.2 + 0i)]$	Unstable
$(9.723, 0.0, 1.263)$	$[(-0.05 + 2.77i, (-0.05 - 2.77i), (1.05 + 0i)]$	Unstable
$(8.127, 0.408, 0.748)$	$[(-0.17 + 3.0i), (-0.17 - 3.04i, (-0.43 + 0i)]$	Locally and globally asymptotically stable

Table 8.8: Equilibrium points and their stability for the parameter set PM_{gs}

We observe from the table that for the interior equilibrium point, the first two eigenvalues are complex with a negative real part and the third one is negative real, thus it is locally stable, and since it satisfied the global stability condition, it is globally stable as well. In terms of biology, this means that the population will coexist at a concentration equal to their equilibrium values. However, the concentrations will not exactly be equal to their equilibrium ones but will oscillate slightly around them.

8.10 Periodic behavior

For many parameter sets, the system has a periodic solution or tends toward a periodic solution after some transient oscillation. Here we present two examples of such behavior. The first one clearly shows a limit cycle while the other is periodic, but the limit cycle is difficult to find at least graphically in the phase portrait. The procedure to find the parameter sets which give periodic or chaotic solutions is as follows (See appendix B.3).

1. We sample parameters using the Saltelli sampling method from the SALib library for sensitivity analysis, which is done in the sensitivity analysis section.
2. We make a python function that classifies stable eqm. points and find parameter sets for which there is not any stable eqm. points and these parameters give a periodic or chaotic solution.

The first parameter set that gives a periodic solution is,

$$PM_{periodic_1} = [0.000622, 0.28, 5.20, 1.82, 1.96, 4.23, 0.42, 2.09]$$

and local stability analysis for this parameter set is given Table 8.9.

Equilibrium point	Eigenvalues	(Local) Stability
$(0, 0, 0)$	$[(1 + 0i), (-5.19 + 0i), (-1.82 + 0i)]$	Unstable
$(1607.06, 0, 0)$	$[(-1 + 0i), (1601.87 + 0i), (444.87 + 0i)]$	Unstable
$(5.61, 1.0, 0.0)$	$[(-0.21 + 2.36i), (-0.21 - 2.36i), (3.96 + 0i)]$	Unstable
$(14.0, 0.0, 3.57)$	$[(-1.04 + 1.67i), (-1.04 - 1.67i), (6.86 + 0i)]$	Unstable
$(6.64, 0.32, 2.41)$	$[(0.01 + 2.33i), (0.01 - 2.33i), (-1.57 + 0i)]$	Unstable

Table 8.9: Equilibrium points and their stability for the parameter set $PM_{periodic_1}$

From the table we observe that for each equilibrium point there is at least one positive real eigenvalue or complex eigenvalue with a positive real part, thus they are unstable. To investigate the periodicity of the solution, we analyze the time series of the solution with the initial condition $(6.67, 0.35, 2.44)$. We find the peaks and their corresponding time and the time difference between them for x, y_1 and y_2. Here we show these properties for the last five peaks only,

$$
\begin{aligned}
\begin{bmatrix} t_{X_{peaks}} \\ X_{peaks} \end{bmatrix} &= \begin{bmatrix} 337.609882 & 340.36990828 & 343.12660121 & 345.88662749 & 348.64665378 \\ 8.55990328 & 8.5607464 & 8.56154942 & 8.56229294 & 8.56297974 \end{bmatrix} \\
\begin{bmatrix} t_{Y_{1peaks}} \\ Y_{1peaks} \end{bmatrix} &= \begin{bmatrix} 337.999886 & 340.759912 & 343.516605 & 346.276631 & 349.036657 \\ 0.72170403 & 0.72190582 & 0.72210336 & 0.72228477 & 0.72245119 \end{bmatrix} \\
\begin{bmatrix} t_{Y_{2peaks}} \\ Y_{2peaks} \end{bmatrix} &= \begin{bmatrix} 338.239888 & 340.99658092 & 343.75660721 & 346.51663349 & 349.27665978 \\ 4.13092759 & 4.13180079 & 4.13262109 & 4.13337396 & 4.1340649 \end{bmatrix} \\
T_x &= [2.7566973, 2.7566973, 2.76003067, 2.7566973] \\
T_{y_1} &= [2.7566973, 2.7566973, 2.76003067, 2.7566973] \\
T_{y_2} &= [2.7566973, 2.760030672.7566973, 2.7566973]
\end{aligned}
$$

From above we see that the time differences between two successive peaks $T_x = T_{y_1} = T_{y_2} \approx 2.67$, are the same (see Figure 8.25 c) thus, the solution is periodic with period 2.67 non-dimensional time unit (to get the dimensional one we have to divide it by the specific growth rate of the bacteria r then the unit will be *day*).

Another way to prove the existence of a periodic solution is by calculating the Lyapunov exponent spectrum even though it is enough to just compute the largest one $LLE(\lambda_1)$. For a periodic solution (limit cycle), the largest Lyapunov exponent must be zero and the two others must be negative (see Table 3.1). For calculating the spectrum of Lyapunov exponents (LEs) and the largest Lyapunov exponent LLE, we followed the bellow procedure,

1. We wrote a python code (see appendix B.3) based on the Sprott method [52] and tested it for the Lorenz system with two reference values, and we got the expected results.

2. We used a Matlab code by [18] for calculating the whole spectrum of Lyapunov exponents based on [61]. This code gave reasonable, but not completely correct results for the Lorenz system and reference values, especially for the zero exponent for which the result was accurate until two decimals.

3. We used Julia Dynamical system library [11] which has a chaos tool and can calculate both the Largest Lyapunov exponent $LLE(\lambda_1)$ and the spectrum of Lyapunov exponents LEs based on [16] and [26], respectively.

4. As a final test we calculate the exponents for increasingly large time t to check convergence and also calculate the trace of the Jacobian matrix averaged over the orbit $\langle Tr_J \rangle_t$

for the same amount of time and compare it with the sum of Lyapunov exponents $\sum \lambda_i$ and if they are significantly equal we accept the values.

5. Note that the choice of the ODE solver plays an important role here and it is best to check the calculation for stiff and non stiff solver and compare the results.

The Lyapunov characteristics for the parameter set $PM_{periodic_1}$ are given in Table 8.10 :

method	Python Sprott method	Matlab Wolf method	Julia Dynamical system library
IC	$[6.67, 0.35, 2.44]$	$[6.67, 0.35, 2.44]$	$[6.67, 0.35, 2.44]$
t	100000	100000	100000
LEs, LLE	$1.827\,798 \times 10^{-6}$	$[0.000028, -0.027113, -1.513829]$	$[5.192\,108 \times 10^{-5}, -0.027137, -1.513803], 1.818\,734 \times 10^{-7}$
$\sum \lambda_i$		-1.486688	-1.540888
$\langle Tr_J \rangle_t$	-1.540914		

Table 8.10: Lyapunov exponents using different methods and algorithms for the parameter set $PM_{periodic_1}$

We observe from the previous table that the largest Lyapunov exponent is approximately zero and the two others are negative, thus there exists a limit cycle attractor in the system (see Figure 8.26).

The second parameter set that gives a periodic solution is

$$PM_{periodic_2} = [0.0002023,\ 0.16,\ 8.81,\ 3.22,\ 3.06,\ 4.0,\ 1.01,\ 0.09]$$

and local stability analysis is given in Table 8.11 :

Equilibrium point	Eigenvalues	(Local) Stability
$(0, 0, 0)$	$[(1 + 0i), (-9.79 + 0i), (-4.33 + 0i)]$	Unstable
$(4935.74, 0, 0)$	$[(-1 + 0i), (2494.87 + 0i), (697.14 + 0i)]$	Unstable
$(9.82, 1., 0.)$	$[(-0.37 + 3.22i), (-0.37 - 3.22i), (2.7 + 0i)]$	Unstable
$(20.06, 0., 6.04)$	$[(-0.09 + 2.11i), (-0.09 - 2.11i), (6.11 + 0i)]$	Unstable
$(11.12, 0.36, 3.86)$	$[(0.53 + 2.66i), (0.53 - 2.66i), (-1.44 + 0i)]]$	Unstable

Table 8.11: Equilibrium points and their stability for the parameter set $PM_{periodic_2}$

From the table we observe that for each equilibrium point there exists at least one positive real eigenvalue or complex eigenvalue with a positive real part, thus they are unstable. To check the periodicity of the solution, we analyze the time series for the solution starting from third initial condition (see Figure 8.27). We find the peaks, their corresponding times and the time difference between them for x, y_1 and y_2. Here we just show these properties for the last five peaks,

$$
\begin{aligned}
\begin{bmatrix} t_{X_{peaks}} \\ X_{peaks} \end{bmatrix} &= \begin{bmatrix} 970.01323338 & 976.20325401 & 982.3899413 & 988.57996193 & 994.76998257 \\ 60.65041116 & 60.65433949 & 60.66371714 & 60.67277167 & 60.68314734 \end{bmatrix} \\
\begin{bmatrix} t_{Y_{1peaks}} \\ Y_{1peaks} \end{bmatrix} &= \begin{bmatrix} 970.10656702 & 976.29658766 & 982.48327494 & 988.67329558 & 994.86331621 \\ 18.75377117 & 18.74584112 & 18.76084647 & 18.762193 & 18.76832601 \end{bmatrix} \\
\begin{bmatrix} t_{Y_{2peaks}} \\ Y_{2peaks} \end{bmatrix} &= \begin{bmatrix} 970.249901 & 976.439921 & 982.629942 & 988.819963 & 995.009983 \\ 62.2803533 & 62.26980152 & 62.31311859 & 62.27559734 & 62.3606805 \end{bmatrix} \\
T_x &= [6.19002063, 6.19002063, 6.18668729, 6.19002063, 6.19002063] \\
T_{y_2} &= [6.19002063, 6.19002063, 6.18668729, 6.19002063, 6.19002063] \\
T_{y_2} &= [6.18668729, 6.19002063, 6.19002063, 6.19002063, 6.19002063]
\end{aligned}
$$

From above we see that the time differences between two successive peaks $T_x = T_{y_1} = T_{y_2} \approx 6.19$, are the same (see Figure 8.27), thus the solution is periodic with period 6.19. To quantitatively investigate the periodicity of the solution, we calculate Lyapunov exponents that are shown in Table 8.12 :

method	Python Sprott method	Matlab Wolf method	Julia Dynamical system library
IC	$[11.155, 0.391, 3.893]$	$[11.155, 0.391, 3.893]$	$[11.155, 0.391, 3.893]$
t	100000	1000000	10000000
LEs, LE	$-5.386\,401 \times 10^{-5}$	$[-0.001379, -0.209325, -0.209373]$	$[-0.000196, -0.209965, -0.209965], -0.000196$
$\sum \lambda_i$		-0.420077	-0.420126
$\langle Tr_J \rangle_t$	-0.400393		

Table 8.12: Lyapunov exponents using different methods and algorithms for the parameter set $PM_{periodic_2}$

From the above table, we observe the largest Lyapunov exponent doesn't become significantly zero even though we used a large averaging time. This may be due to the fact that it is a transient state between periodic and chaotic systems.

The common feature that we can observe from the two examples and related figures, is that the bacteria x is the first to reach a peak, then comes the first predator protozoa which preys on the bacteria and reaches its peak concentration. Finally comes the second predator Daphnia which preys on both of them. After that, all species reach their minimum values with bacteria having the highest concentration and then the cycle repeats for an indefinite amount of time, i.e. the three species coexist but in a period fashion. This behavior is not very rare in nature as many populations exhibit cyclic behavior depending on food availability and other natural factors.

8.11 Chaotic behavior

There are many parameter sets for which the solution has a chaotic behavior, and here we study one of them. The parameter set is given by,

$$PM_{chaos} = [0.000399, 0.28, 9.79, 4.33, 0.16, 4.1, 0.74, 0.17]$$

and the local stability analysis is given in Table 8.13,

Equilibrium point	Eigenvalues	(Local) Stability
$(0, 0, 0)$	$[(1 + 0i), (-9.79 + 0i), (-4.33 + 0i)]$	Unstable
$(2504.65, 0, 0)$	$[(-1 + 0i), (2494.87 + 0i), (697.14 + 0i)]$	Unstable
$(10.52, 1., 0.)$	$[(-0.37 + 3.22i), (-0.37 - 3.22j), (2.7 + 0i)]$	Unstable
$(16.06, 0.0, 3.55)$	$[(-0.09 + 2.11i), (-0.09 - 2.11i), (6.11 + 0i)]$	Unstable
$(10.17, 0.39, 2.18)$	$[(0.53 + 2.66i), (0.53 - 2.66j), (-1.44 + 0i)]$	Unstable

Table 8.13: Equilibrium points and their stability for the parameter set PM_{chaos}

As we observe from the table, none of the equilibrium points are stable because for each of them there exists a positive real eigenvalue or a complex one with a positive real part. To check the periodicity or aperiodicity of the system, we plot the solution of the system for several initial conditions (see Figure 8.29) and analyze one of them. Here we find the peaks, their corresponding times, and compute the time difference between two successive peaks for all variables T_x, T_{y_1}, T_{y_2} for the initial condition $(0.1, 0.2, 0.3)$ (see Figure 8.29 b). These quantities for the last six peaks are given by the following,

$$\begin{bmatrix} t_{X_{peaks}} \\ X_{peaks} \end{bmatrix} = \begin{bmatrix} 948.666 & 961.137 & 966.683 & 979.983 & 984.457 & 996.0299 \\ 43.204 & 115.916 & 49.311 & 124.002 & 39.039 & 107.222 \end{bmatrix}$$

$$\begin{bmatrix} t_{Y_1 peaks} \\ Y_{1peaks} \end{bmatrix} = \begin{bmatrix} 948.796 & 961.189 & 966.799 & 980.037 & 984.597 & 996.087 \\ 13.431 & 44.199 & 16.001 & 47.664 & 11.684 & 40.475 \end{bmatrix}$$

$$\begin{bmatrix} t_{Y_2 peaks} \\ Y_{1peaks} \end{bmatrix} = \begin{bmatrix} 948.963 & 961.493 & 966.963 & 980.357 & 984.769 & 996.363 \\ 112.710 & 16.481 & 127.037 & 8.233 & 97.897 & 30.936 \end{bmatrix}$$

$$T_x = [12.470, 5.547, 13.3004.473, 11.573]$$

$$T_{y_1} = [12.393, 5.610, 13.236, 4.560, 11.490]$$

$$T_{y_2} = [12.530, 5.470, 13.393, 4.413, 11.593]$$

We clearly see that these time differences are not equal. Therefore, the system is not periodic. Now we know that the system is not periodic. Let us check sensitivity to initial conditions by calculating the Lyapunov exponents or the largest Lyapunov exponents. The Lyapunov exponents using different algorithms are given in the following table:

method	Python Sprott method	Matlab Wolf method	Julia Dynamical system library
IC	$[10.2, 0.416, 2.207]$	$[10.2, 0.416, 2.207]$	$[10.2, 0.416, 2.207]$
t	100000	100000	1000000
LEs, LE	0.045556	$[0.037707, 0.000077, -0.428533]$	$[0.046657, 0.066233, -0.503393], 0.046646$
$\sum \lambda_i$		-0.390749	-0.390503
$\langle Tr_J \rangle_t$	-0.270883		

Table 8.14: Lyapunov exponents using different methods and algorithms for the parameter set PM_{chaos}

It is evident from the previous table that the largest Lyapunov exponent is positive and the second one is very small positive and the third one is negative, indicating sensitive dependence on the initial condition, i.e. the system is chaotic for that parameter set (see Figures 8.29 , 8.30 , 8.31).

Furthermore, we observe again that the bacteria reach its peak concentration then got preyed on by the first predator y_1 which reach it peak concentration as well. After that, the second predator preys on the first predator and reaches its peak concentration. After some time and as the concentration of bacteria is very low, the second predator population decline significantly, and all populations reach their minimum concentration with the bacteria having the largest one around 0.0005 and then this repeats in a non-periodic fashion.

Remark. For the previous chaotic parameter, we can define the following basin of attraction

$$B = \{(x, y_1, y_2) : x > 0, y_1 > 0, y_2 > 0 \,\text{and}\, (x, y_1, y_2) \neq EP_5\}$$

. Therefore, the trajectories must start from B to converge to the strange attractor. Otherwise, they will converge to one of the equilibrium points.

Now we know that the parameter set PM_{chaos} gives rise to chaos, let us investigate if we will still have chaos if we change on of the parameter at a time in their sensitivity range. We increase each parameter with step 0.1 and check the characteristics of the system. The results are summarized in the following table:

Parameter	Chaotic	Periodic	Equilibrium point
b	$[0.22, 0.436]$	$[0.437, 0.443], [0.1, 0.21]$	$[0.444, 10]$
δ_1	$[7.53, 10]$	$[2.6399, 7.43]$	$[0.1, 2.6299]$
δ_2	$[2.8, 4.9]$	$[2.7, 2.79], [5, 6.3]$	$[0.2, 2.6], [6.4, 10]$
ψ	$[0.1, 2.2]$	$[2.3, 6]$	$[6.1, 10]$
ϕ	$[3.5001, 5]$	$[2.1001, 3.4001]$	$[0.0001, 2.0001]$
ω_1	$[0.001, 1.301]$	$[1.401, 9.901]$	None
ω_2	$[0.001, 0.201]$	$[0.301, 5]$	None

Table 8.15: Parameter ranges that give different solution taking the parameter set PM_{chaos} as a base and changing one parameter at a time

We observe from the previous table that the parameters b, δ_2, ψ drive the system from periodic or chaotic behavior to equilibrium solutions which is important result in terms of controlling the system.

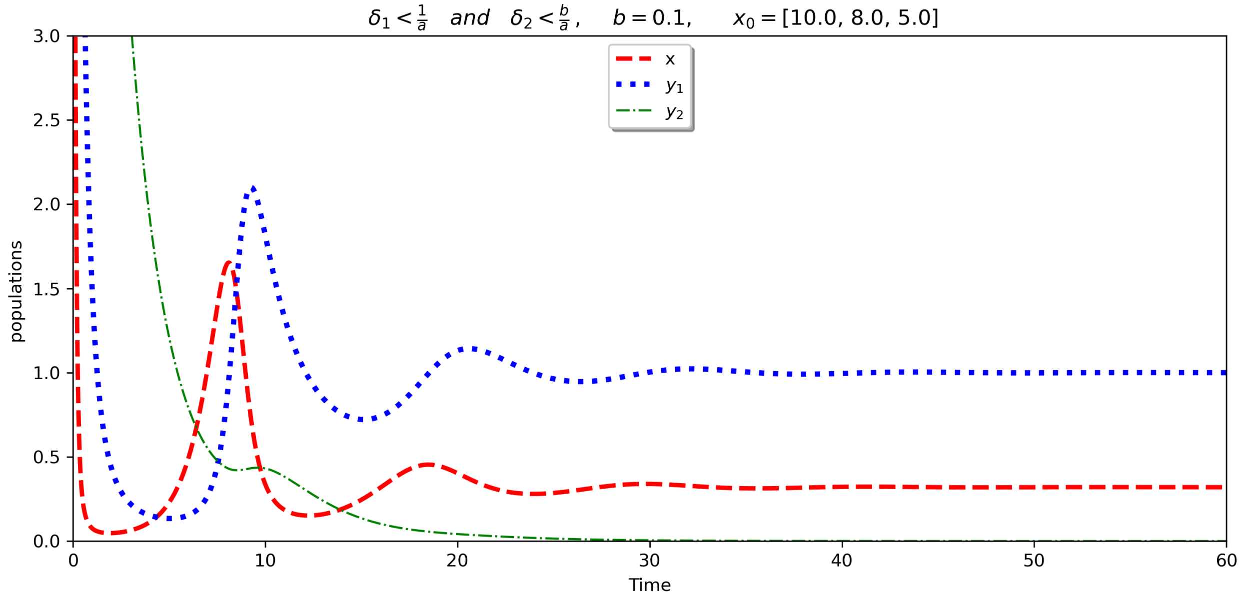
8.12 Effect of the parameter s on global stability of the coexistence equilibrium point.

Let us investigate the effect of bacteria inflow to the system on the coexistence of the three species. For this purpose, we use the parameter set PM_{gs} for which we know that the system has a globally stable interior equilibrium point. Our numerical tests showed that for $s \in [0.1, 3]$ the system will have a globally stable interior equilibrium point, but for $s \geq 4$ the first predator became extinct. The following table gives characteristics of the system for different values of this parameter:

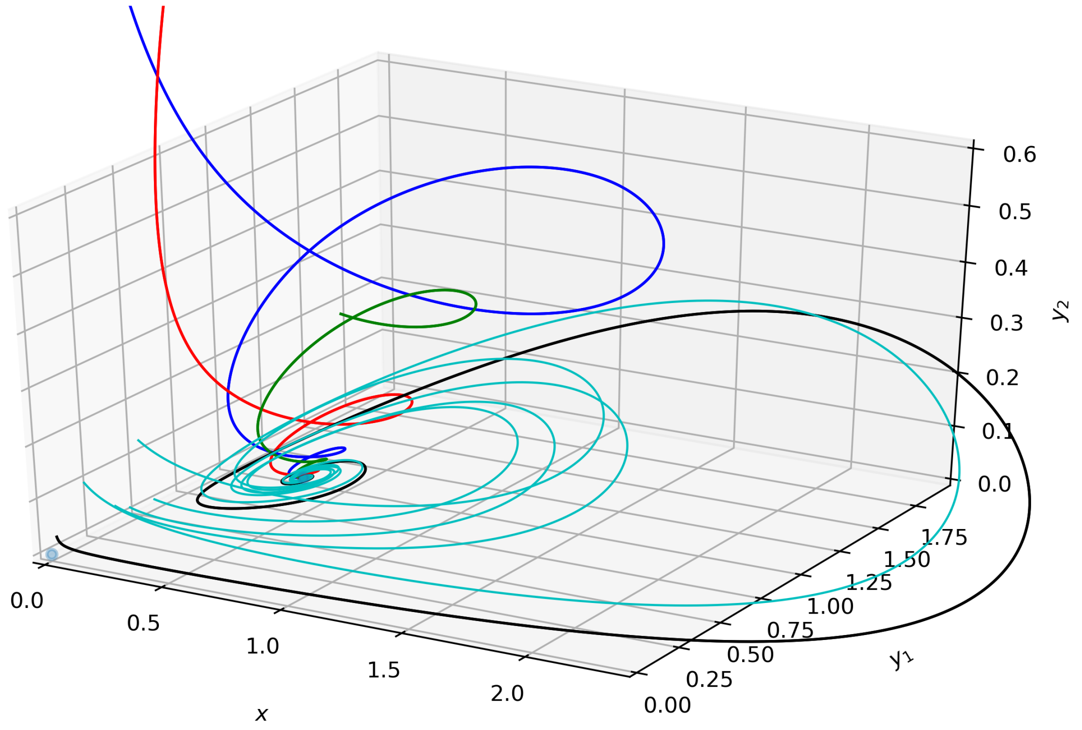
Parameter values	Stable equilibrium point(s) that the system converge to	Status of the system	Figure(s) that demonstrate it
$s = 0$	$EP_5 = (8.118, 0.415, 0.74)$	Coexistence of all three species	Figure 8.24
$s = 0.5$	$EP_5 = (8.395, 0.347, 0.90)$	Coexistence of all three species	Figure 8.32
$s = 3$	$EP_5 = (9.575, 0.055, 1.592)$	Coexistence of all three species	
$s = 4$	$EP_3 = (9.804, 0., 1.781)$	Extinction of the first predator	
$s = 10$	$EP_3 = (9.888, 0.0, 2.545)$	Extinction of first predator	Figure 8.33

Table 8.16: Effect of s at on global stability of the coexistence equilibrium point.

From the previous table we conclude that increasing the value of s ,the species will lose their coexistence due to the fact that the first predator will go extinct.

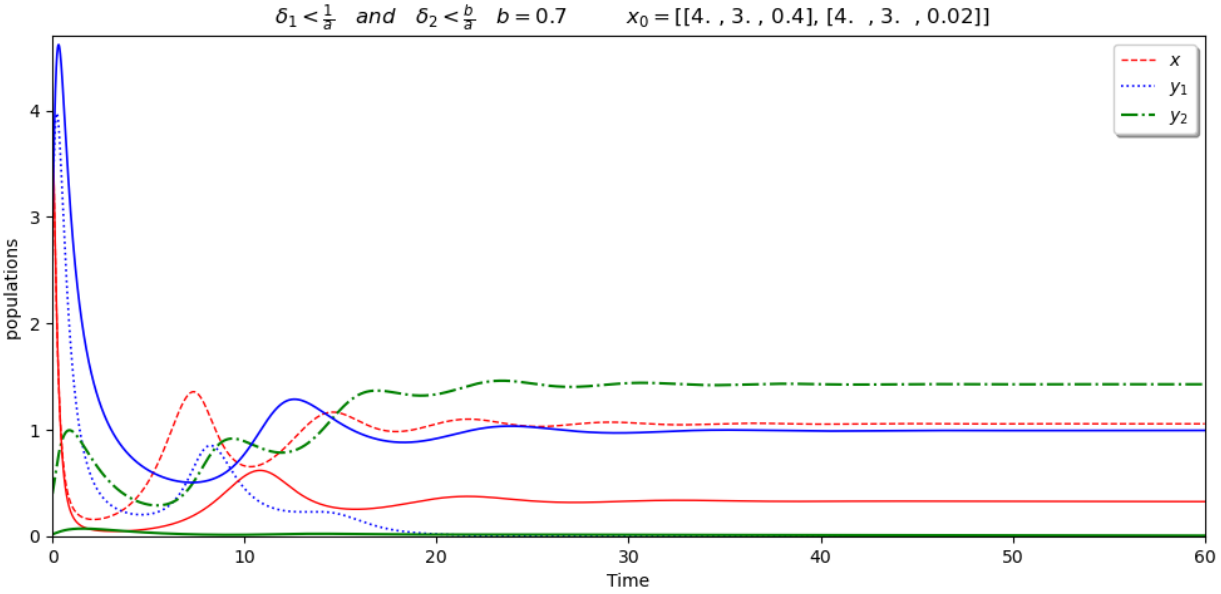


(a) Time series for $b = 0.1$. The populations converge to the eqm. point $EP_3 = (0.32, 0.99, 0)$.

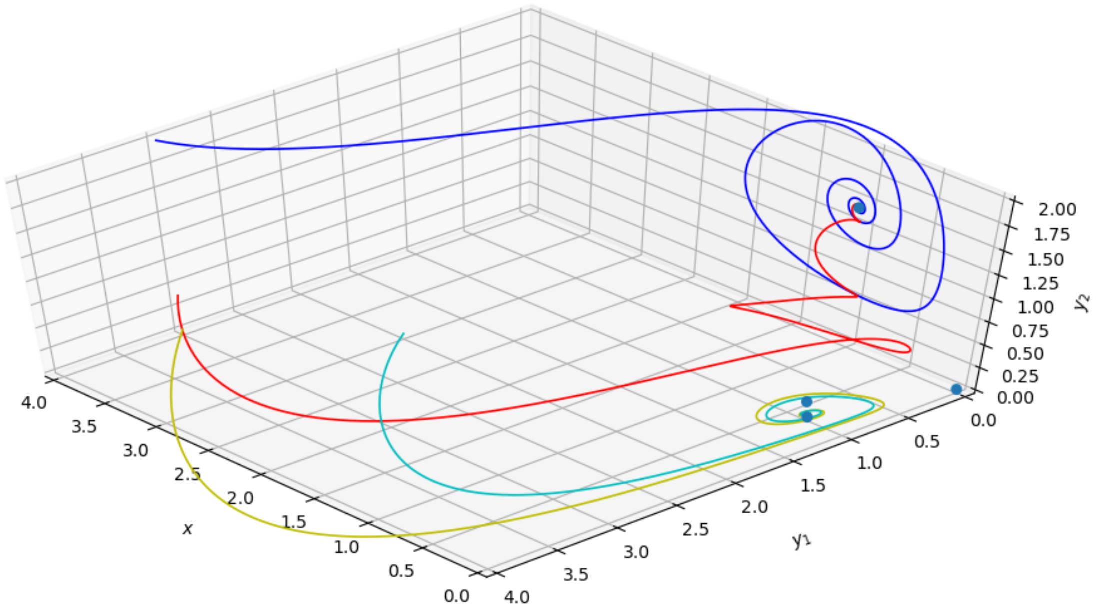


(b) Phase portrait for $b = 0.1$ and several initial conditions. All trajectories converge to the eqm. point $EP_3 = (0.32, 0.99, 0)$.

Figure 8.2: Time series and phase portrait for the system for $b = 0.1$ and other equals to their base values.

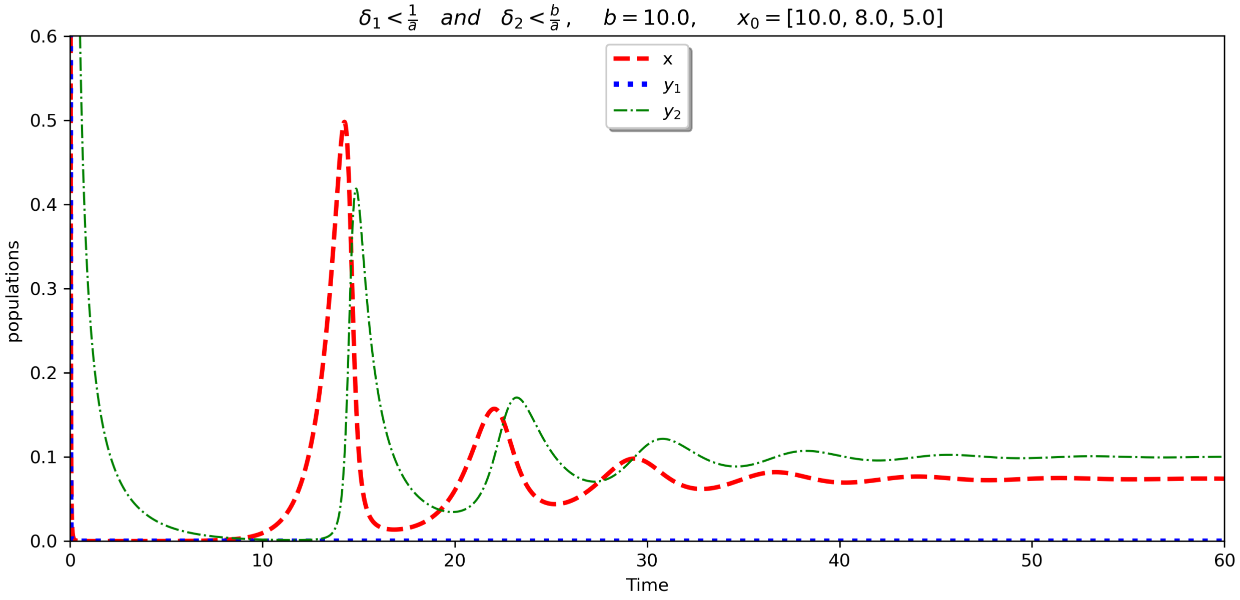


(a) Time series for $b = 0.7$. The solution (continuous lines) that starts from $(4, 3, 0.02)$ converges to $EP_3 = (0.32, 1., 0.)$, while the solution (dashed lines) starts from $(4, 3, 0.4$ converges to $EP_4 = (1.06, 0., 1.43)$.

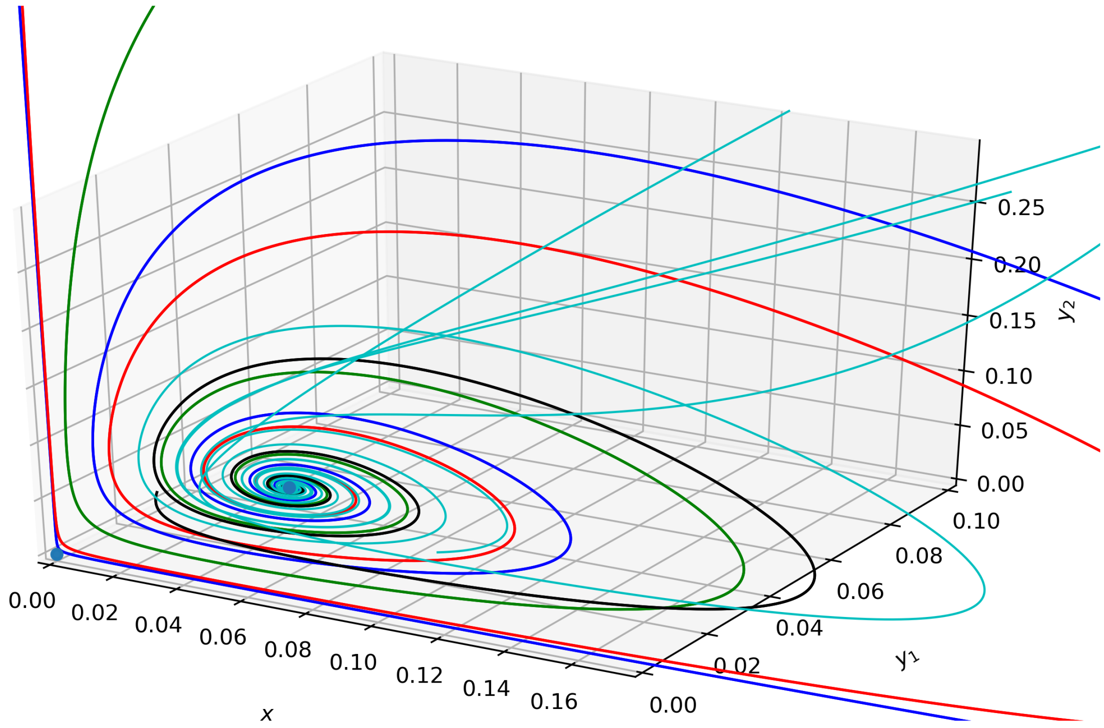


(b) Phase portrait for $b = 0.7$ and several initial conditions. The yellow and blue trajectories represent the continuous and dashed solutions from time series.

Figure 8.3: Time series and phase portrait of the system for $b = 1$ and other equal to their base values.

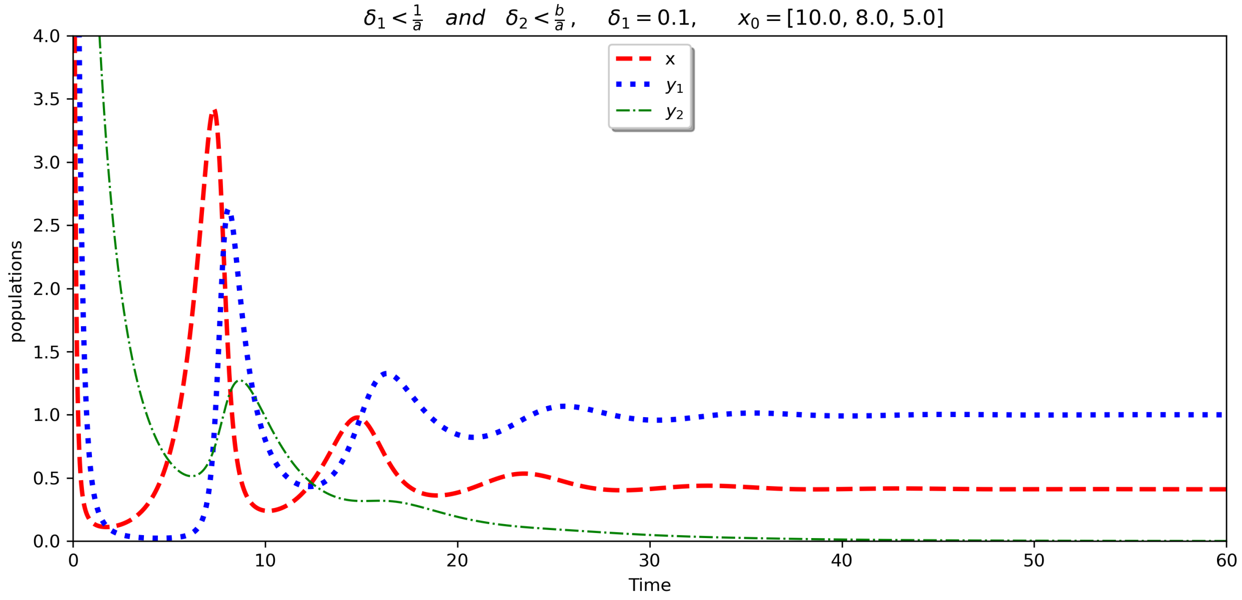


(a) Time series for $b = 10$. The populations converge to $EP_4 = (0.07, 0., 0.1)$.

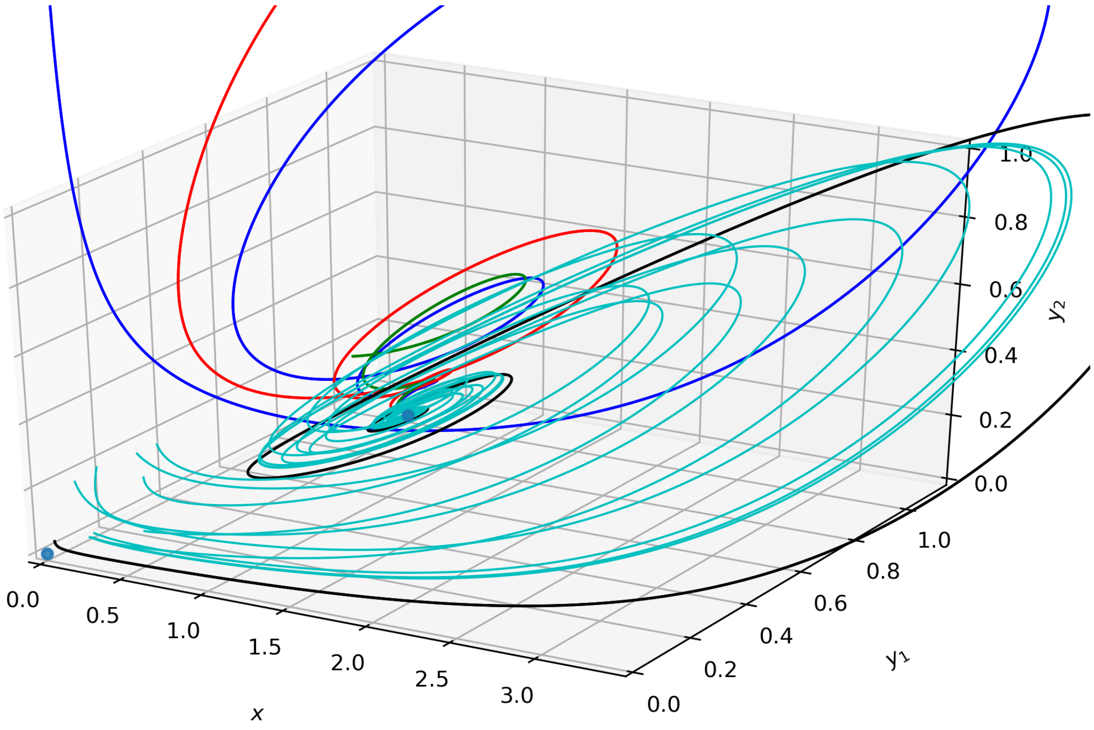


(b) Phase portrait for $b = 10$ several initial conditions. All trajectories converge to $EP_4 = (0.07, 0., 0.1)$.

Figure 8.4: Time series and phase portrait of the system for $b = 10$ and other equal to their base values.

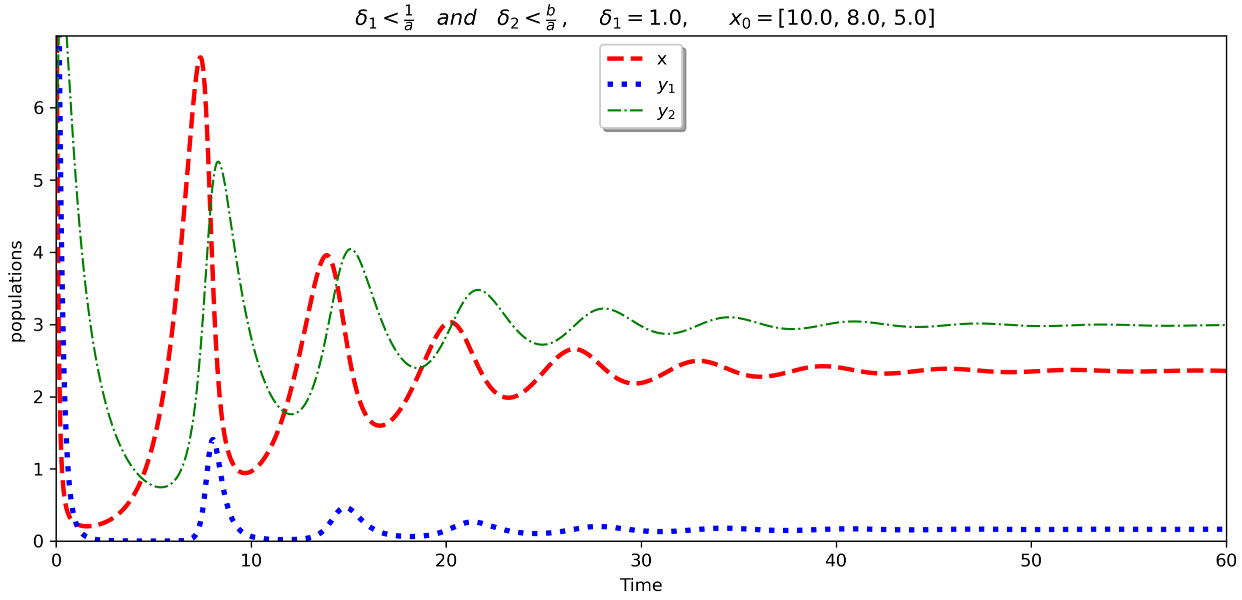


(a) Time series for $\delta_1 = 0.1$. The populations converge to $EP_3 = (0.41, 1., 0.)$.

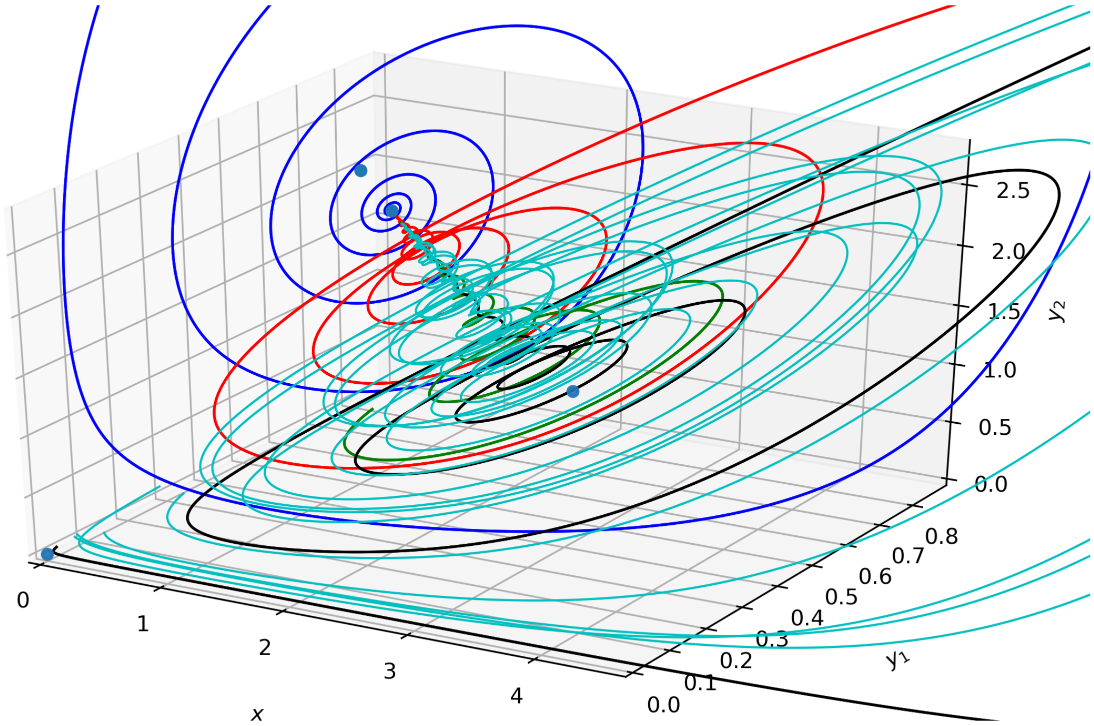


(b) Phase portrait for$\delta_1 = 0.1$ and several initial conditions. All trajectories converge to $EP_3 = (0.41, 1., 0.)$.

Figure 8.5: Time series and phase portrait of the system for $\delta_1 = 0.1$ and other equal to their base values.

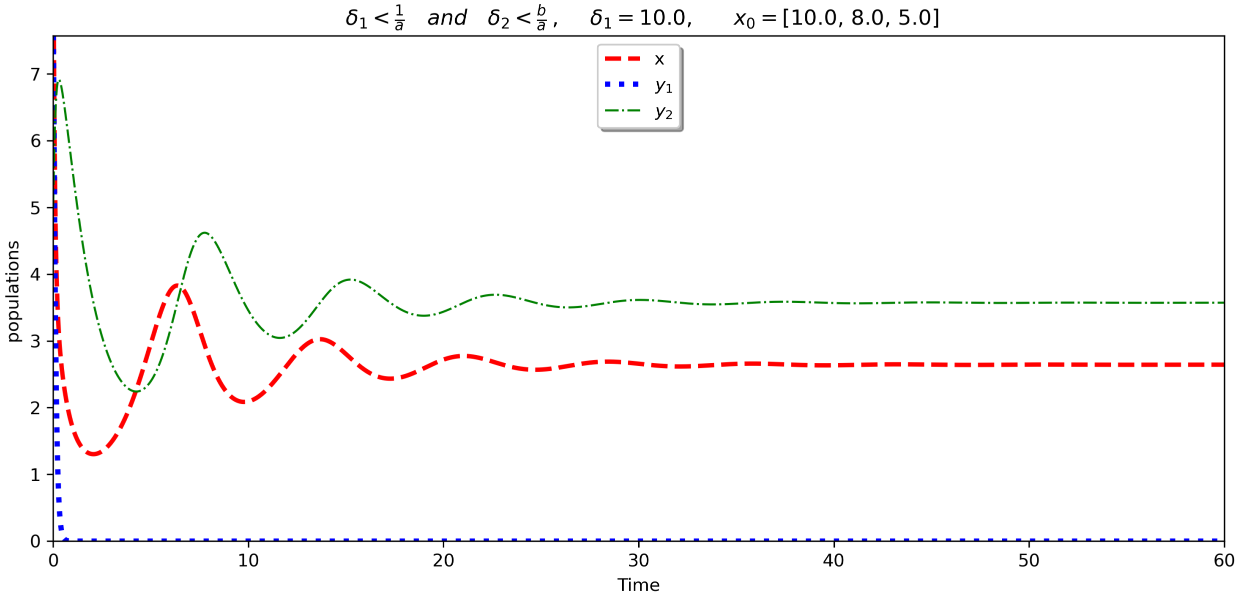


(a) Time series for $\delta_1 = 1$. The populations converge to $EP_5 = (2.35, 0.17, 2.98)$.

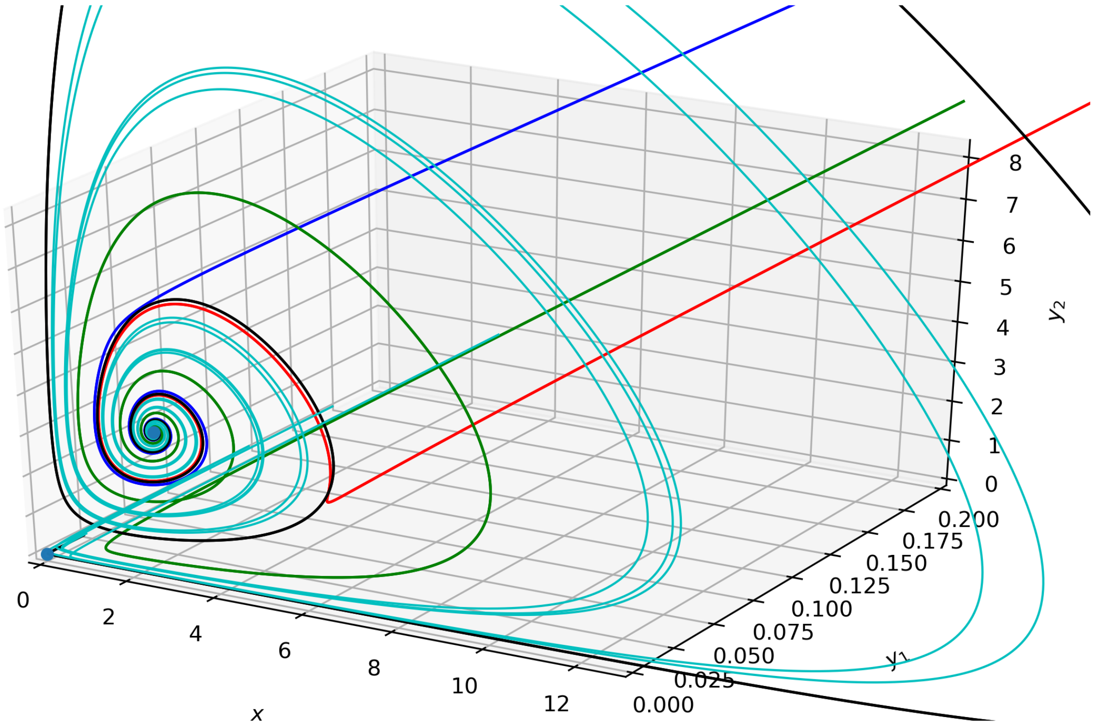


(b) Phase portrait for $\delta_1 = 1$ and several initial conditions. All trajectories converge to $EP_5 = (2.35, 0.17, 2.98)$.

Figure 8.6: Time series and phase portrait of the system for $\delta_1 = 1$ and other equal to their base values.

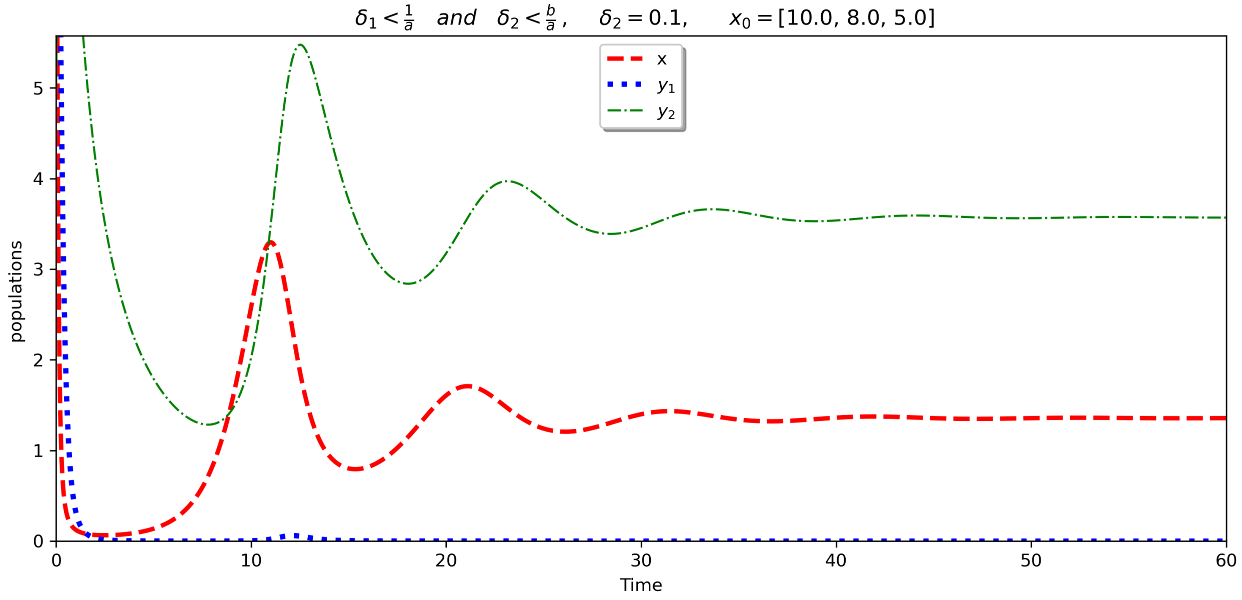


(a) Time series for $\delta_1 = 10$. The populations converge to $EP_4 = (2.64, 0., 3.57)$.

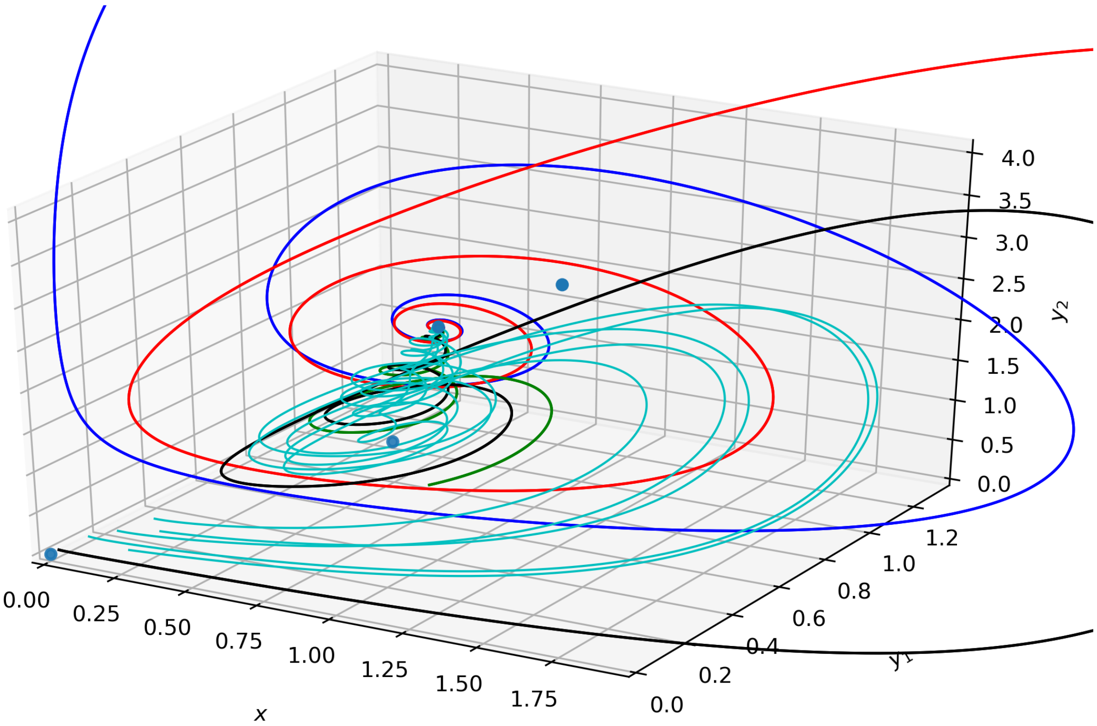


(b) Phase portrait for $\delta_1 = 10$ and several initial conditions. All trajectories converge to $EP_4 = (2.64, 0., 3.57)$.

Figure 8.7: Time series and phase portrait of the system for $\delta_1 = 10$ and other equal to their base values.

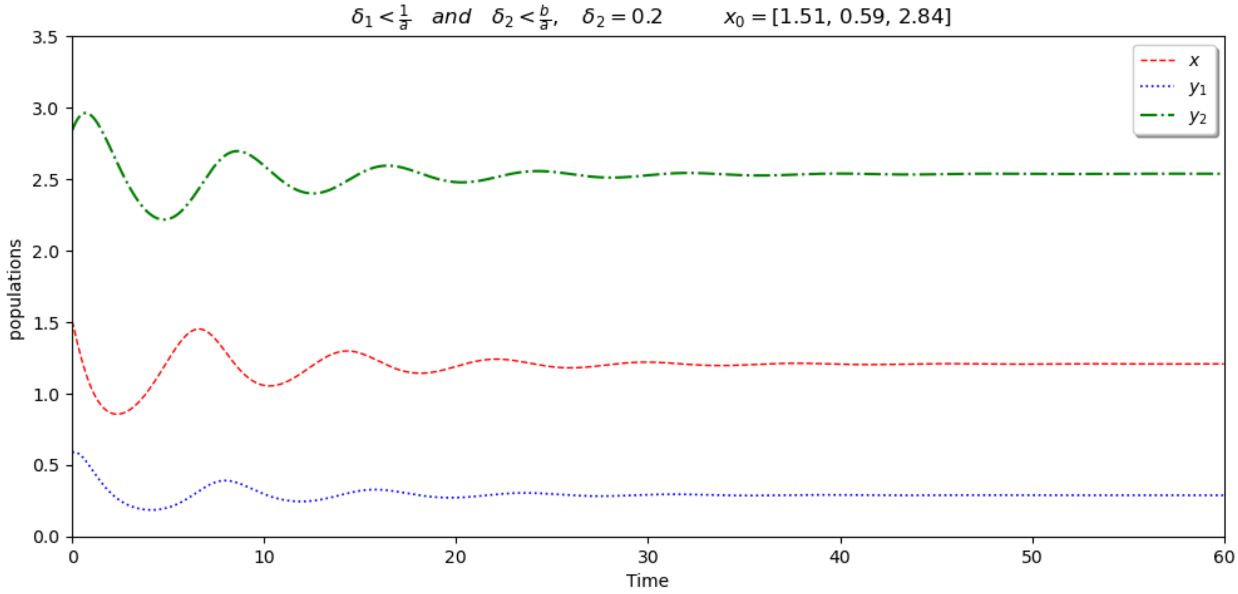


(a) Time series for $\delta_2 = 0.1$. The populations converge to $EP_4 = (1.36, 0., 3.57)$.

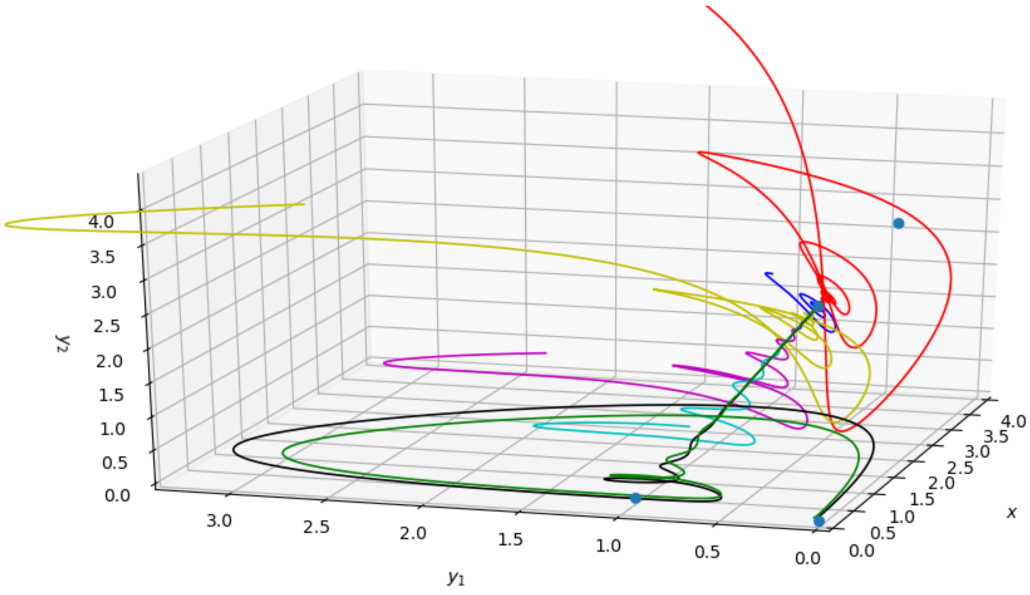


(b) Phase portrait for $\delta_2 = 0.1$ and several initial conditions. All trajectories converge to $EP_4 = (1.36, 0., 3.57)$.

Figure 8.8: Time series and phase portrait of the system for $\delta_2 = 0.1$ and other equal to their base values.

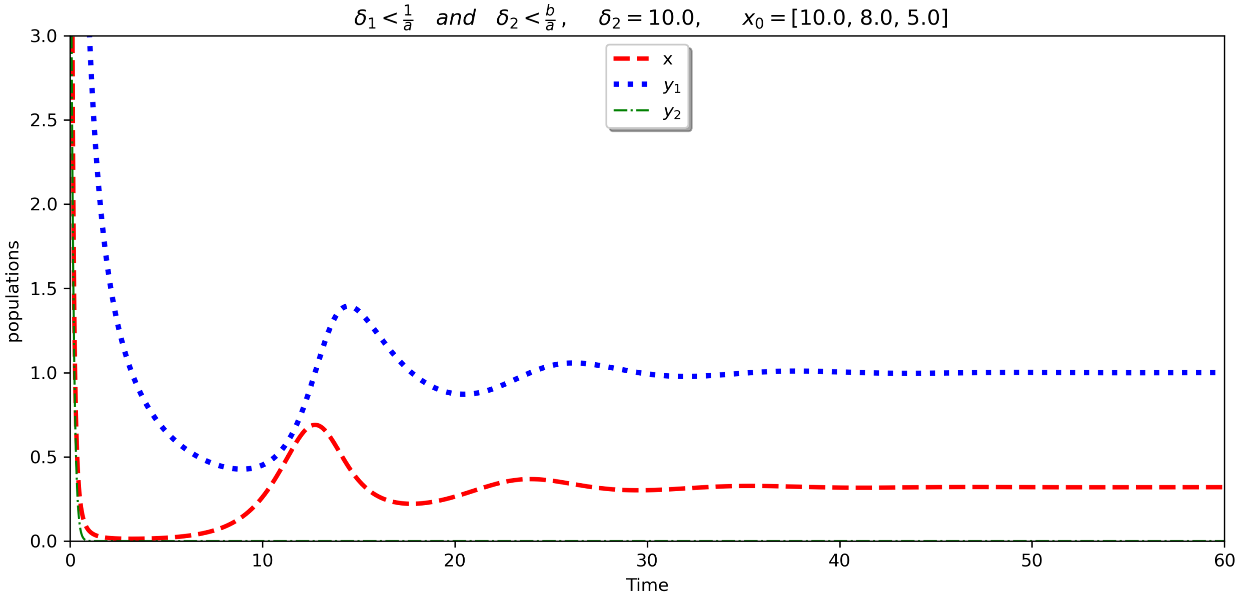


(a) Time series Time series for $\delta_2 = 0.2$. The populations converge to $EP_5 = (1.21, 0.29, 2.54)$.

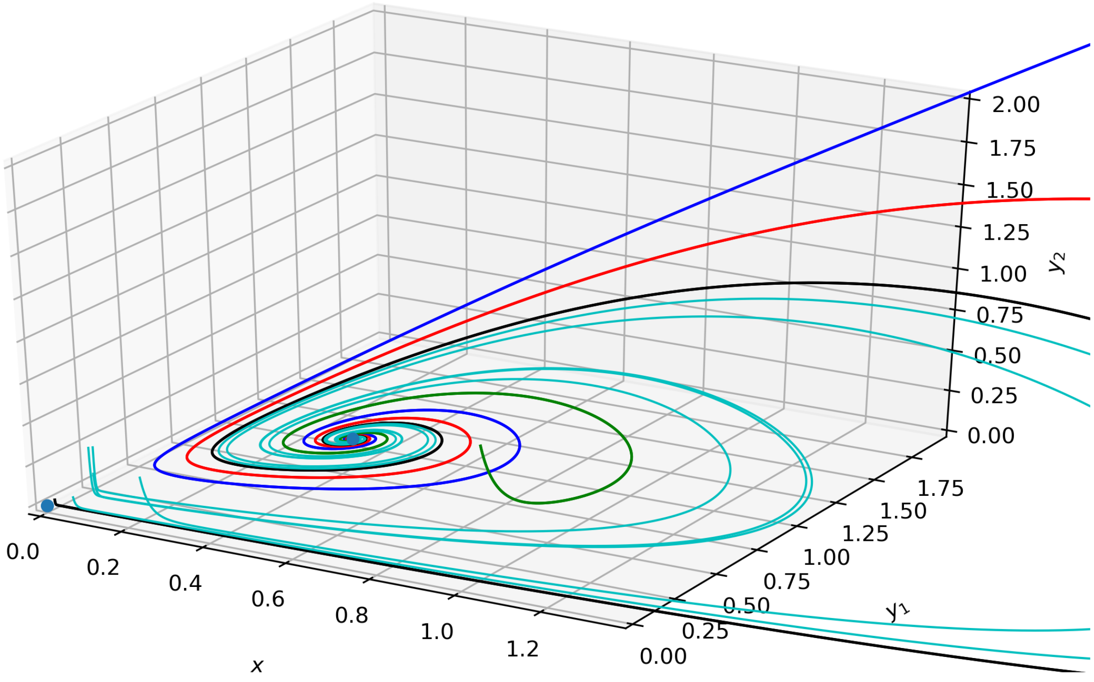


(b) Phase portrait for $\delta_2 = 0.2$ and several initial conditions. All trajectories converge to $EP_5 = (1.21, 0.29, 2.54)$.

Figure 8.9: Time series and phase portrait of the system for $\delta_2 = 0.2$ and other equal to their base values.

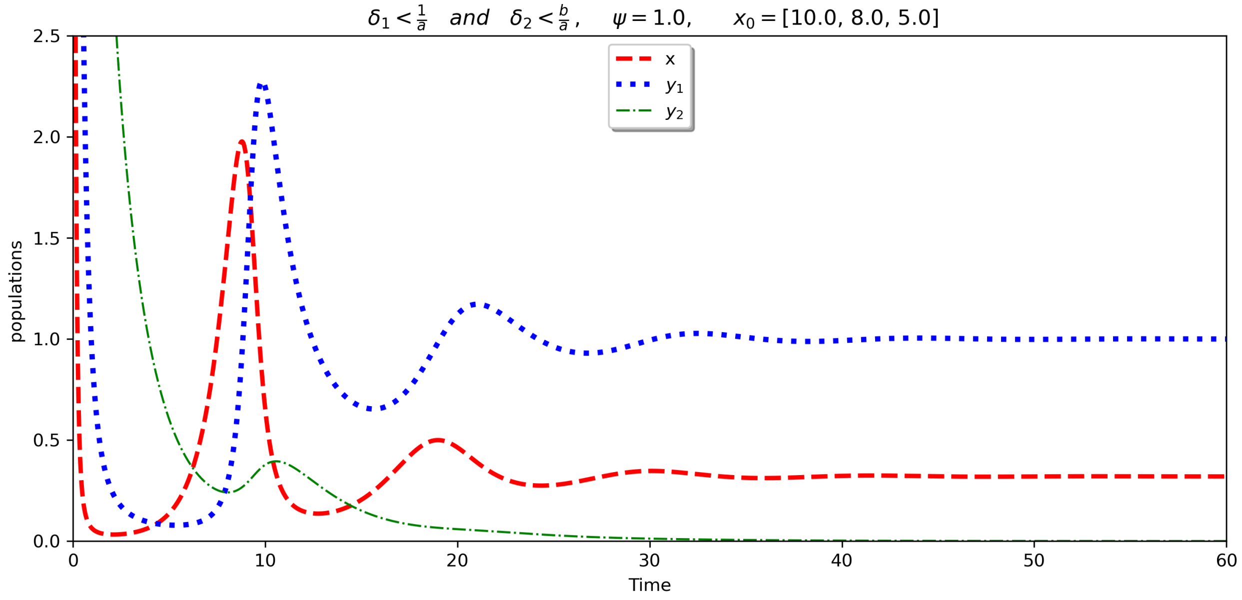


(a) Time series Time series for $\delta_2 = 10$. The populations converge to $EP_3 = (0.32, 1., 0.)$.

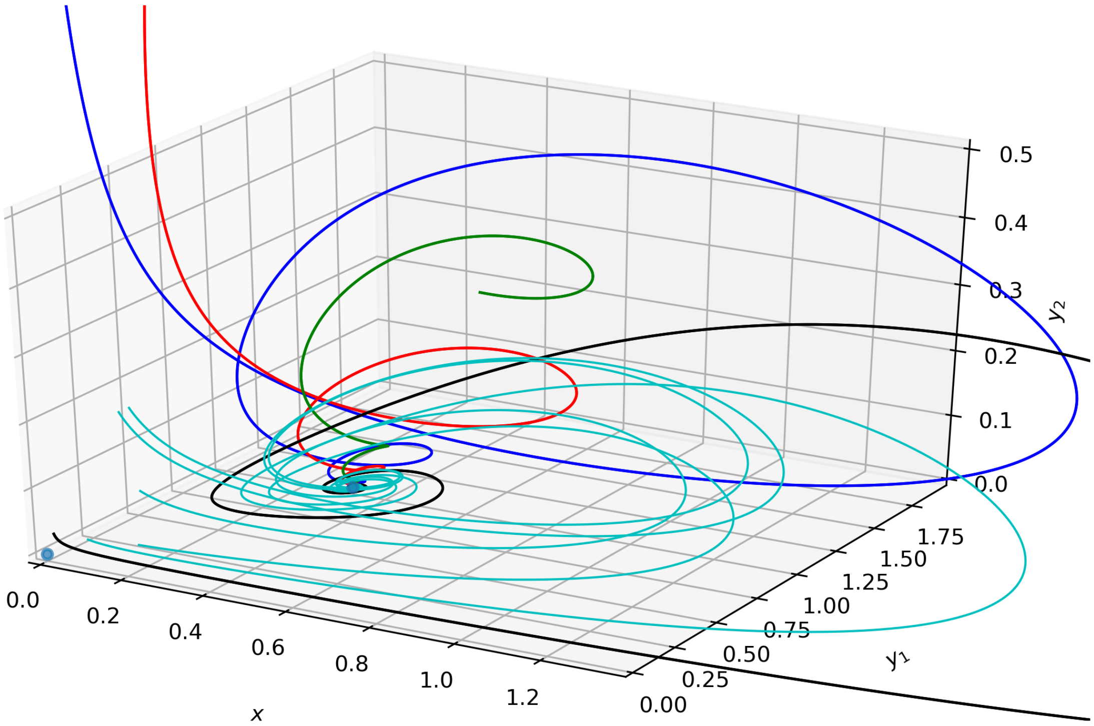


(b) [Phase portrait for $\delta_2 = 10$ and several initial conditions. All trajectories converge to $EP_3 = (0.32, 1., 0.)$.

Figure 8.10: Time series and phase portrait of the system for $\delta_2 = 10$ and other equal to their base values.

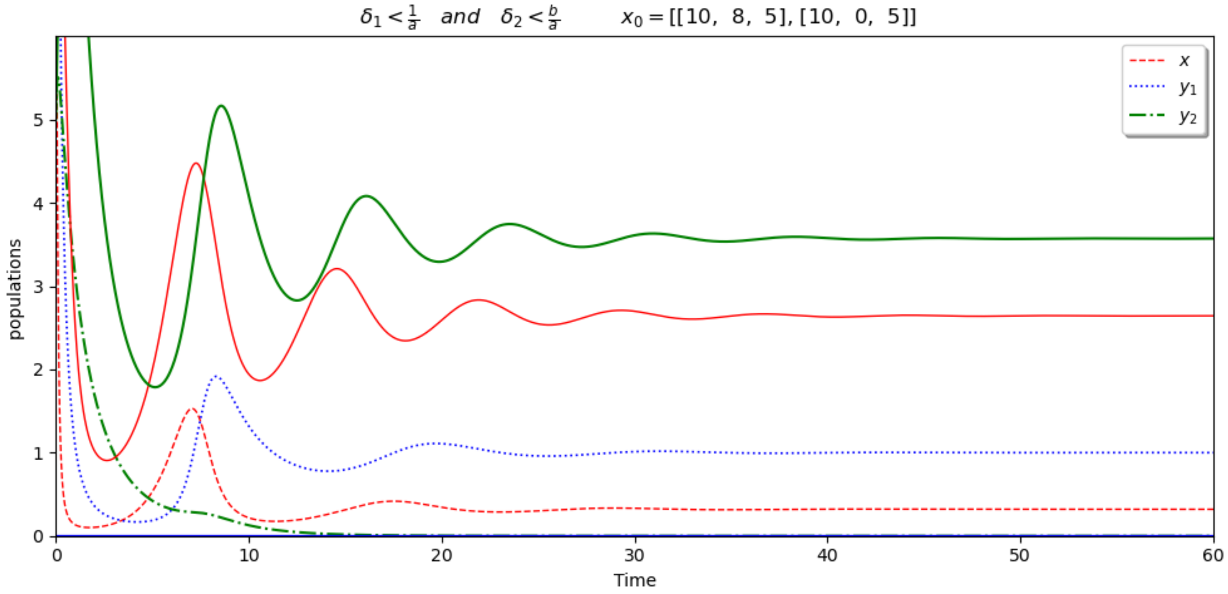


(a) Time series for $\psi = 1$. The populations converge to $EP_3 = (0.32, 1., 0.)$.

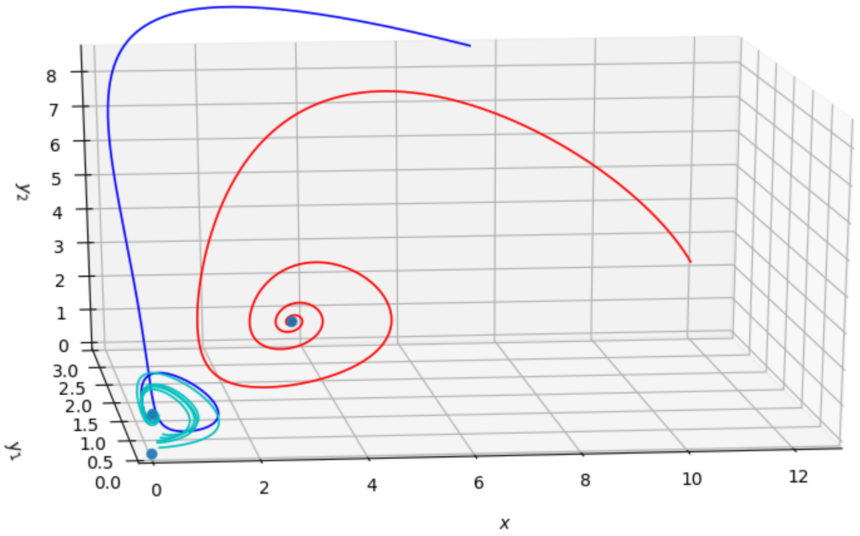


(b) Phase portrait for $\psi = 1$ and different initial condition. All trajectories converge to $EP_3 = (0.32, 1., 0.)$.

Figure 8.11: Time series and phase portrait of the system for $\psi = 1$ and other equal to their base values.

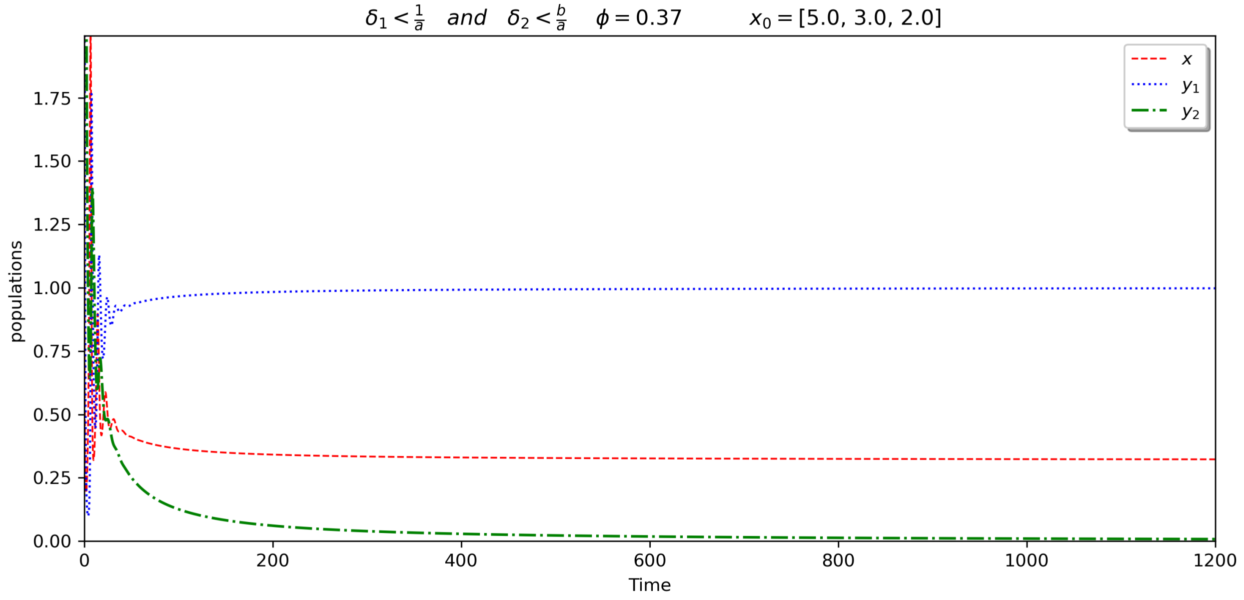


(a) Time series for $\psi = 10$. The dashed lines represent the solution that starts from $(10, 8, 5)$ converges to $EP_3 = (0.32, 1., 0.)$, while the continuous lines represent the solution that starts from $(10, 0, 5)$ converge to the $EP_4 = (2.64, 0., 3.57)$.

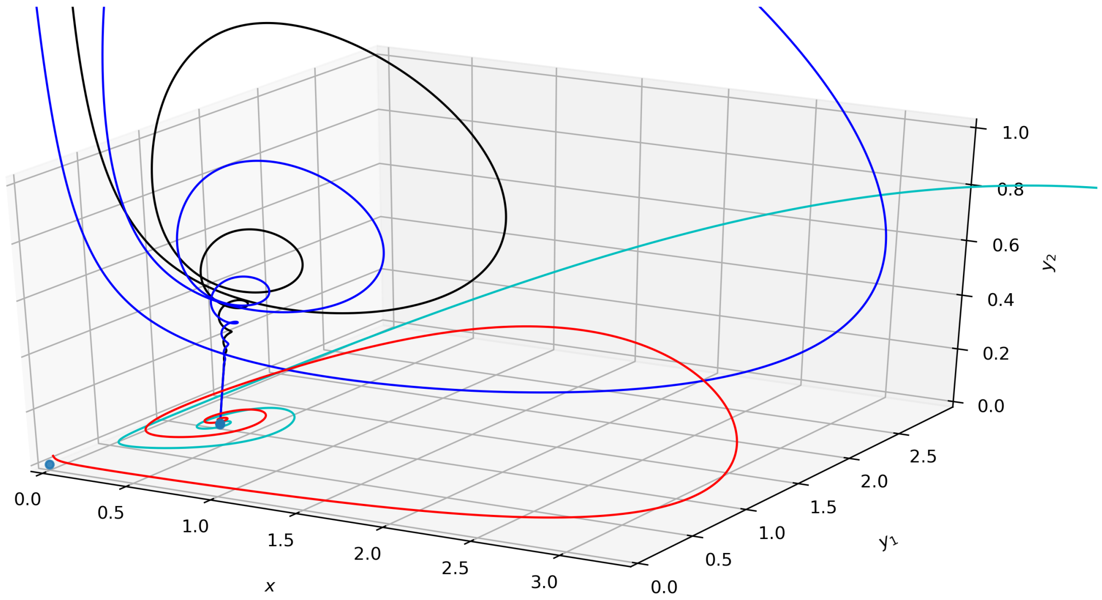


(b) Phase portrait for $\psi = 10$ and two different initial conditions. The blue trajectory represents the dashed solution while the red one represents the continuous line solution from the time series.

Figure 8.12: Time series and phase portrait of the system for $\psi = 10$ and other equal to their base values.

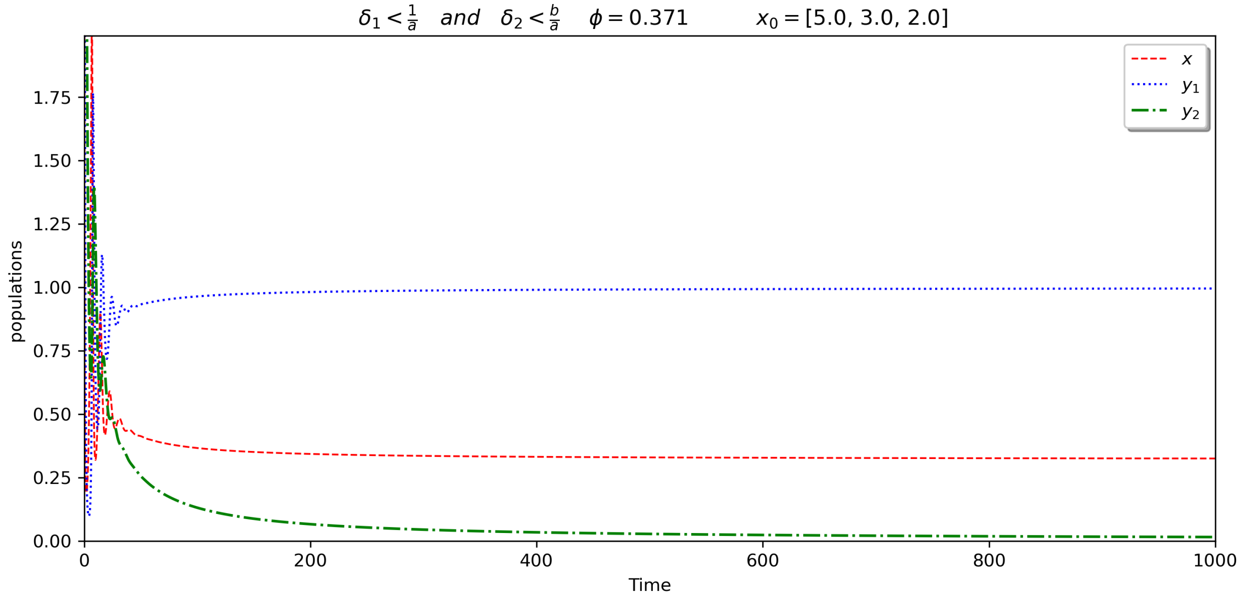


(a) Time series for $\phi = 0.37$. The system converges to $EP_3 = (0.32, 1, 0)$.

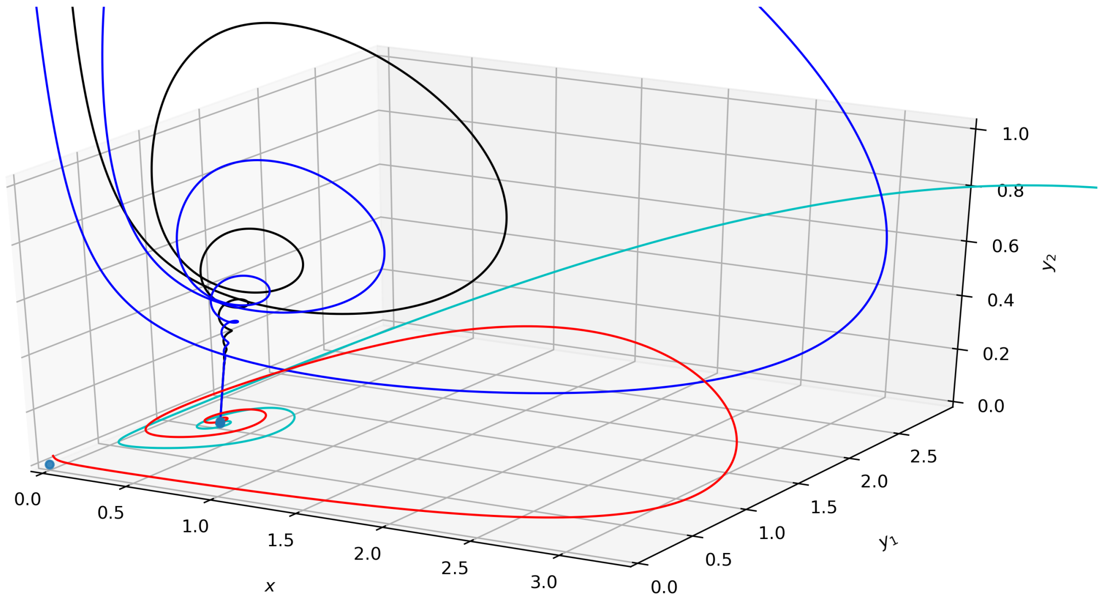


(b) Phase portrait $\phi = 0.37$ and several initial conditions. All trajectories converge to $EP_3 = (0.32, 1, 0)$.

Figure 8.13: Time series and phase portrait of the system for $\phi = 0.37$ and other equal to their base values.

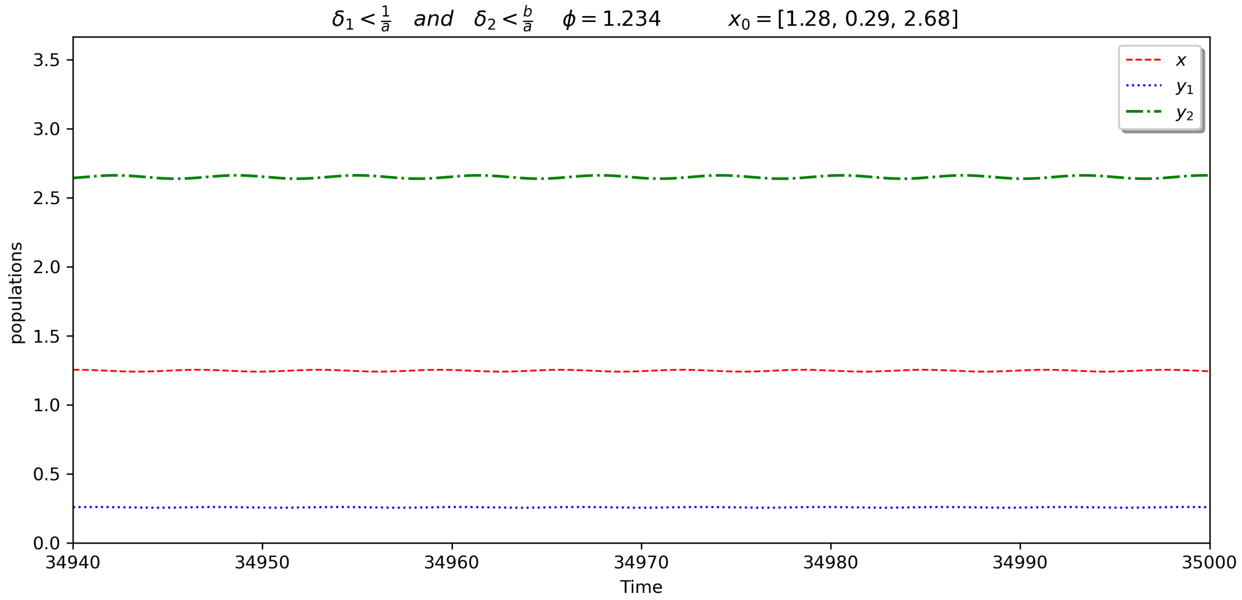


(a) Time series for $\phi = 0.371$. The system converges to $EP_5 = (0.32,\ 1,\ 0.007)$.

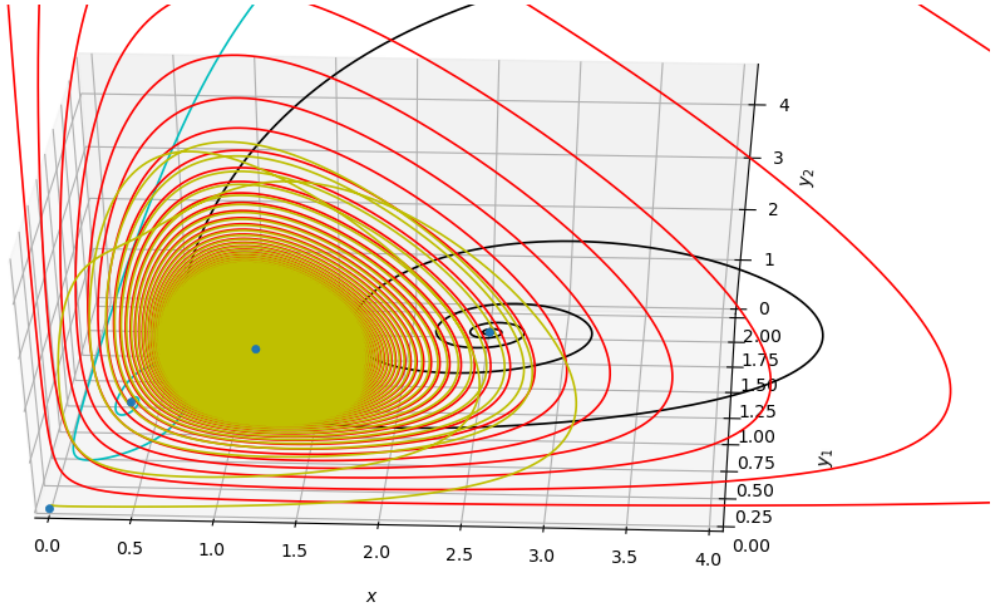


(b) Phase portrait for $\phi = 0.371$. and several initial conditions. All trajectories converge to $EP_5 = (0.32,\ 1,\ 0.007)$.

Figure 8.14: Time series and phase portrait of the system for $\phi = 0.371$ and other equal to their base values.

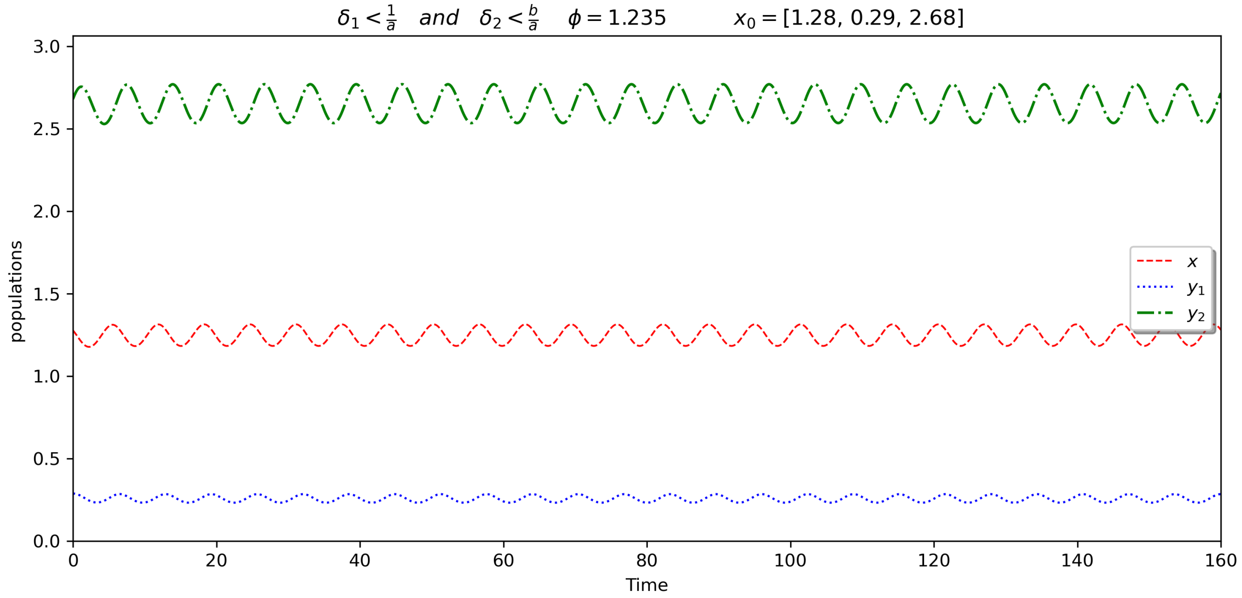


(a) Time series for $\phi = 1.234$. The system converges to $EP_5 = (1.25, 0.26, 2.65)$ but after long oscilatioin.

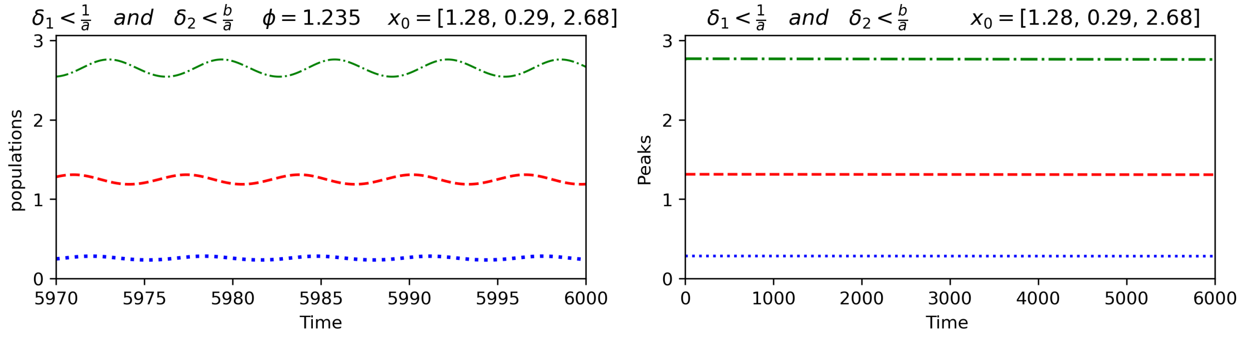


(b) Phase portrait for $\phi = 1.234$ and several initial conditions. The cyan trajectory starts from $(5.32, 6., 0.)$ and converges to $(0.32, 1., 0.)$, the black trajectory starts from $(7.64, 0., 8.57)$ and converges to $(2.64, 0., 3.57)$, and finally the blue, red, and yellow trajectories that start from $(1.28, 0.29, 2.68), (10, 8, 5), (0.01, 0.02, 0.03)$, respectively converge to the interior eqm. point $EP_5 = (1.25, 0.26, 2.65)$ but after long time oscillation because it is a border case between stable eqm. point and periodic solution.

Figure 8.15: Time series and phase portrait of the system for $\phi = 1.234$ and other equal to their base values.



(a) Time series for $\phi = 1.235$. The system is periodic with period 6.39.



(b) peaks of the solution that starts from the initial condition $(1.28, 0.29, 2.68)$.

Figure 8.16: Time series,and peaks of the system for $\phi = 1.235$ and other equal to their base values.

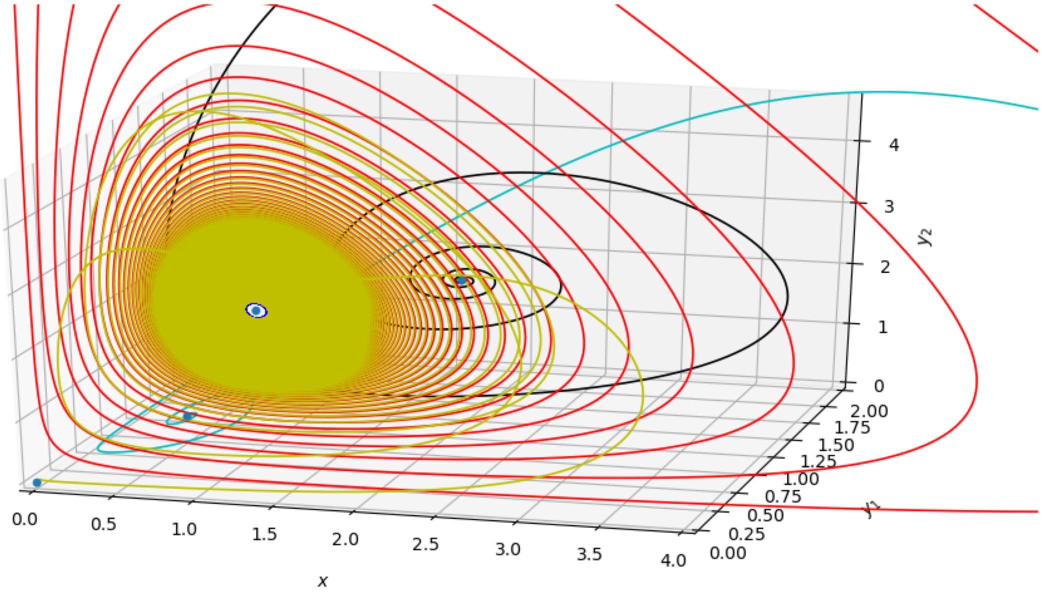
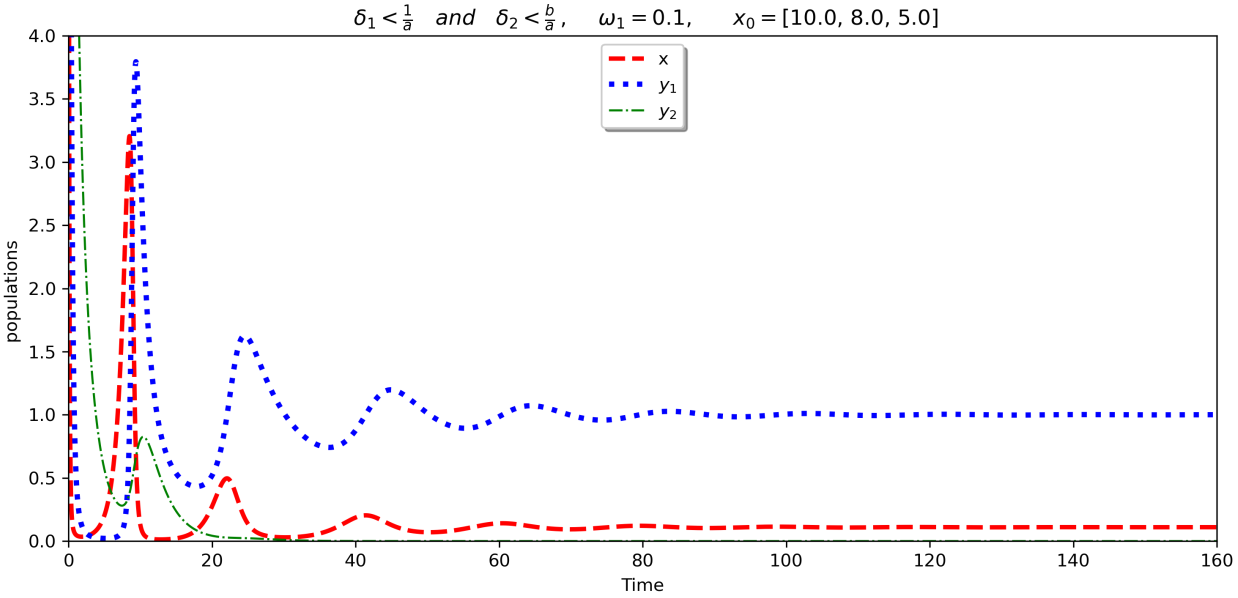
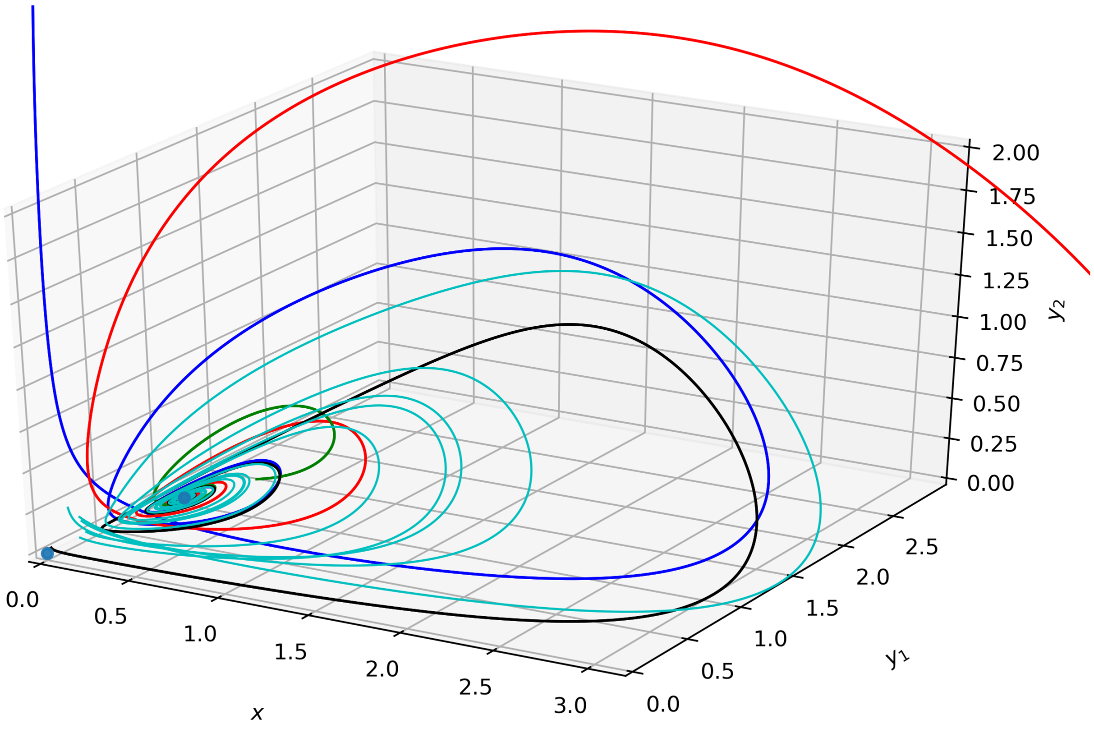


Figure 8.17: Phase portrait for $\phi = 1.235$ and several initial conditions. There is a stable limit cycle between the blue and yellow trajectories. It is stable because the blue trajectory starts near the interior eqm. point at $(1.28, 0.29, 2.68)$ and converges outwardly to the limit cycle, the red and yellow trajectories that start from $(10, 8, 5)$, $(0.01, 0.02, 0.03)$, respectively and converge inwardly toward the limit cycle. The limit cycle has a period of 6.393.

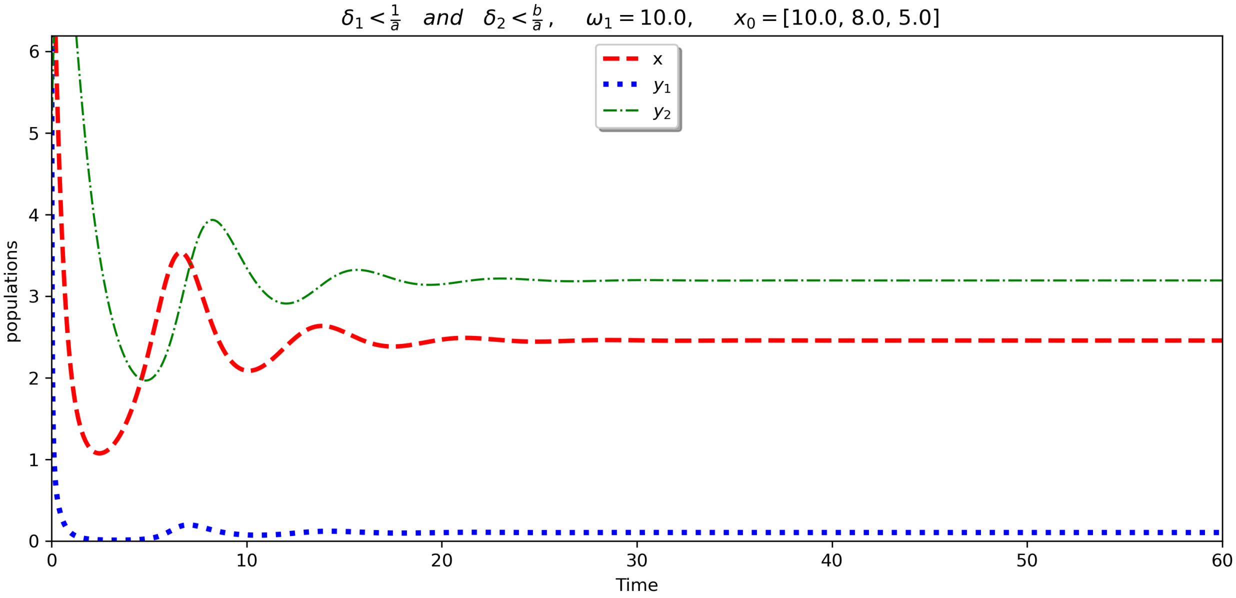


(a) Time series for $\omega_1 = 0.1$. The populations converge to $EP_3 = (0.11, 1., 0.)$.

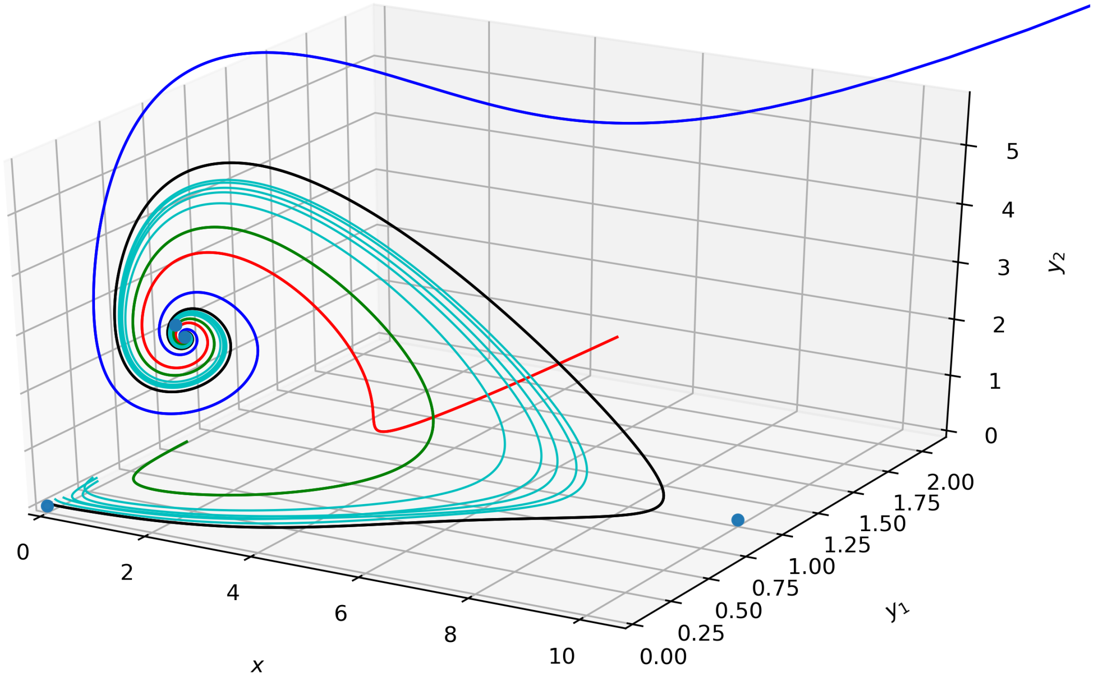


(b) Phase portrait for different initial conditions. All trajectories converge to $EP_3 = (0.11, 1., 0.)$.

Figure 8.18: Time series and phase portrait of the system for $\omega_1 = 0.1$ and other equal to their base values.

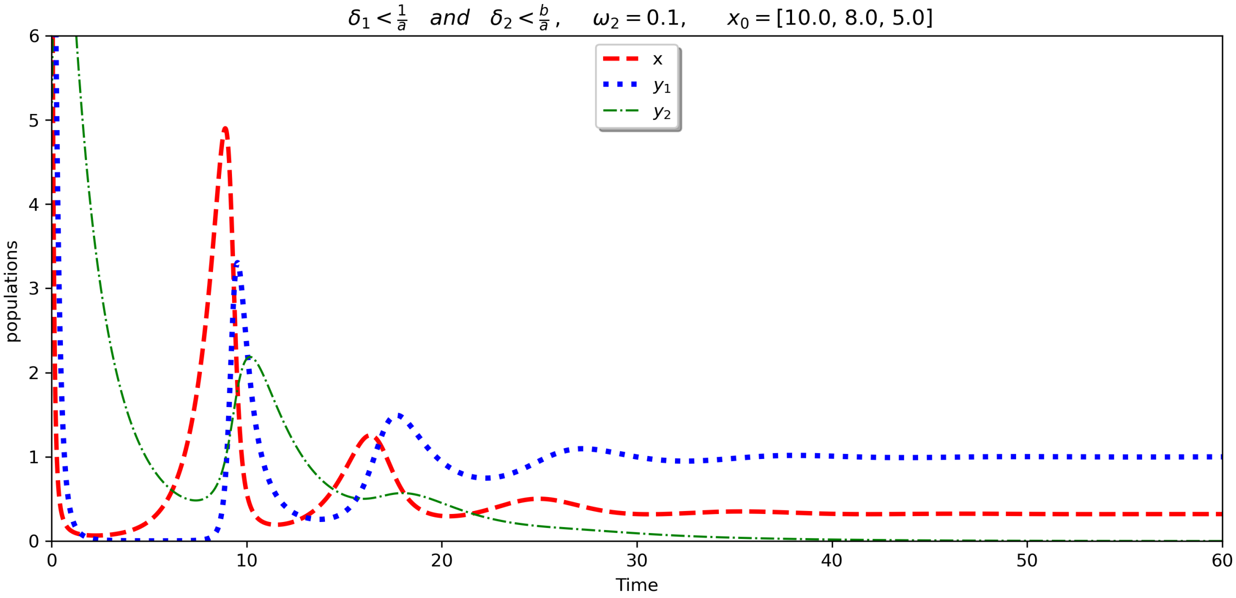


(a) Time series for $\omega_1 = 10$. The populations converge to $EP_5 = (2.46, 0.11, 3.19)$.

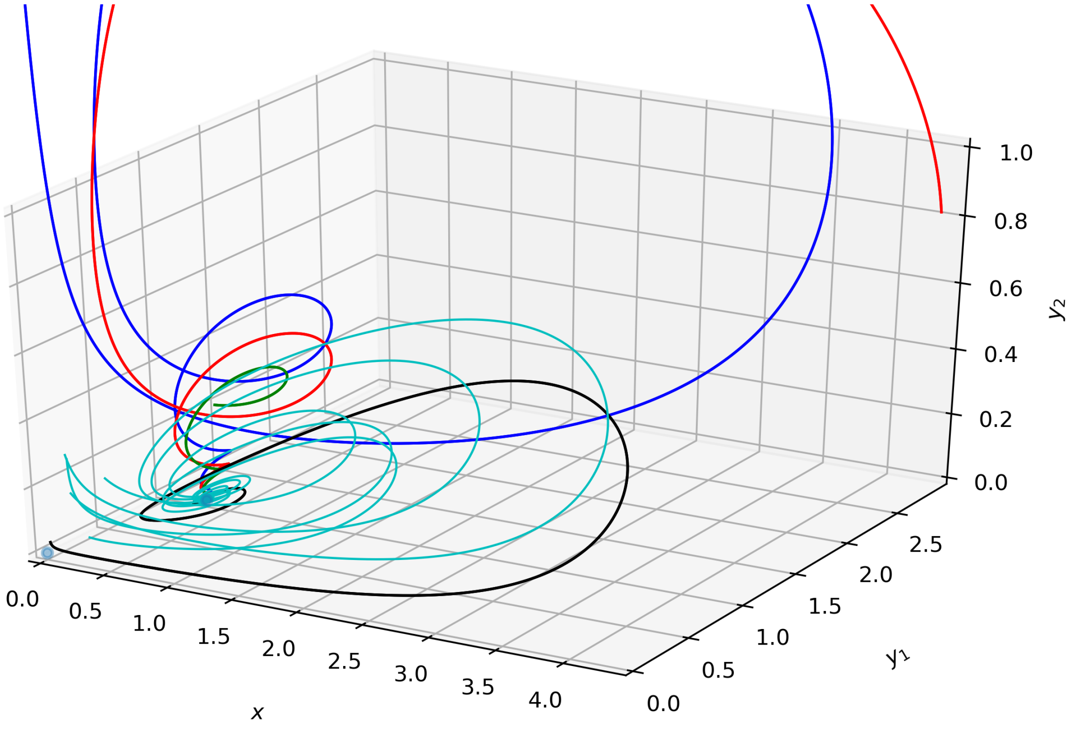


(b) Phase portrait for different initial conditions. All trajectories converge to $EP_5 = (2.46, 0.11, 3.19)$.

Figure 8.19: Time series and phase portrait of the system for $\omega_1 = 10$ and other equal to their base values.

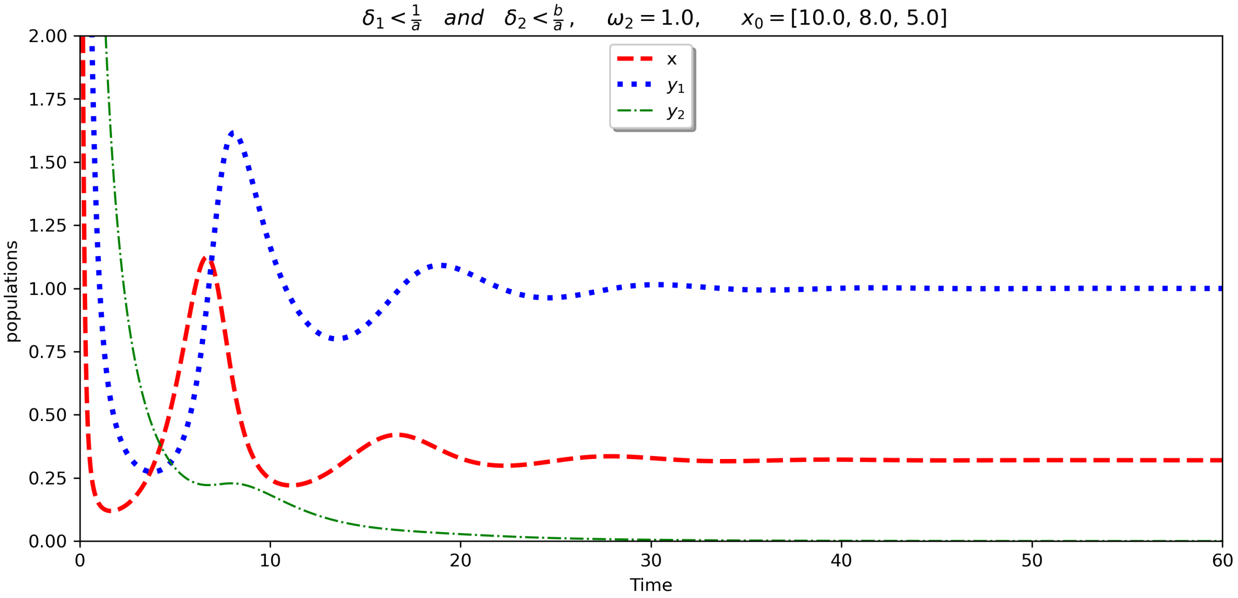


(a) Time series for $\omega_2 = 0.1$. The populations converge to $EP_3 = (0.32, 1., 0.)$.

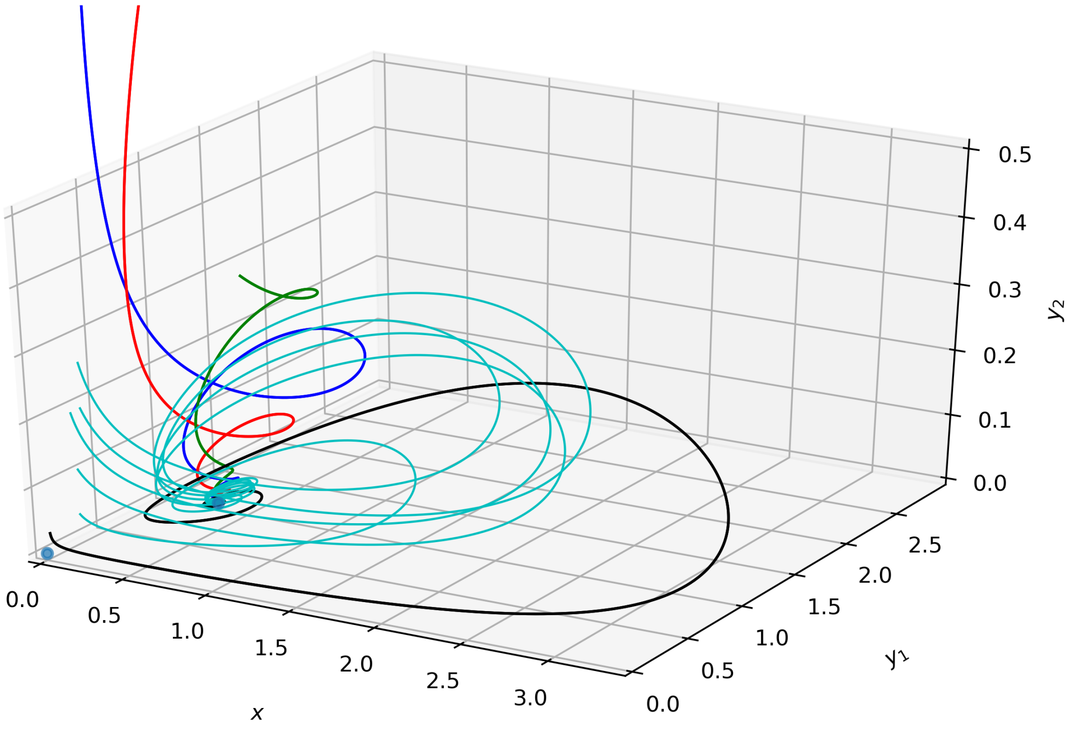


(b) Phase portrait for different initial conditions. All trajectories converge to $EP_3 = (0.32, 1., 0.)$.

Figure 8.20: Time series and phase portrait of the system for $\omega_2 = 0.1$ and other equals to their base values.

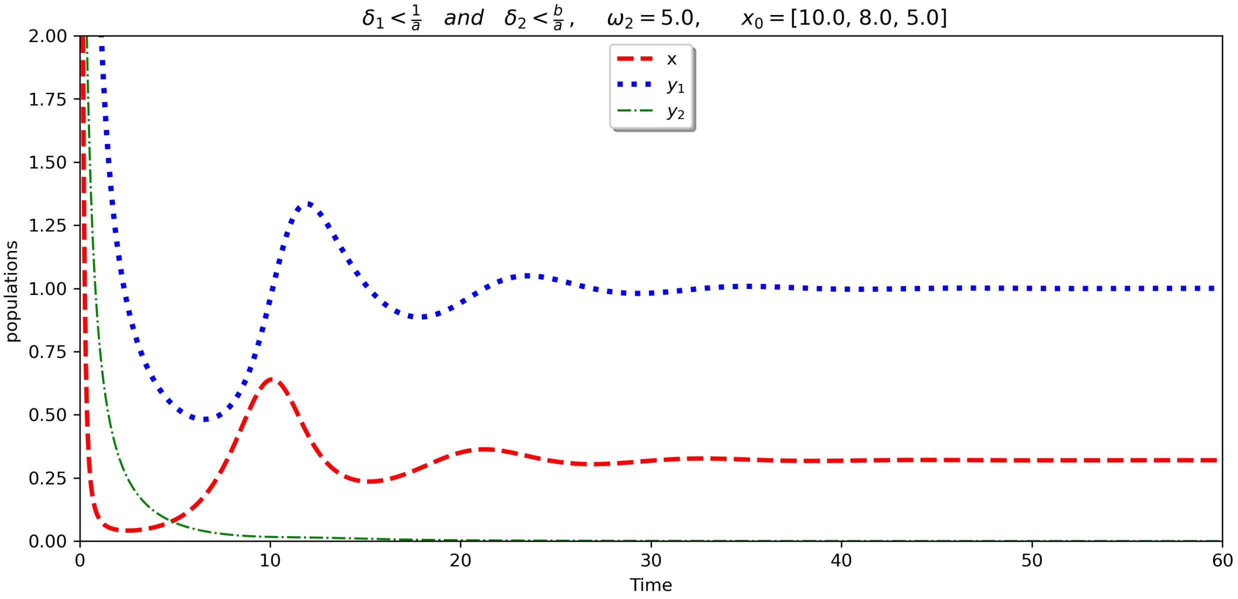


(a) Time series for $\omega_2 = 1$. The populations converge to $EP_3 = (0.32, 1., 0.)$.

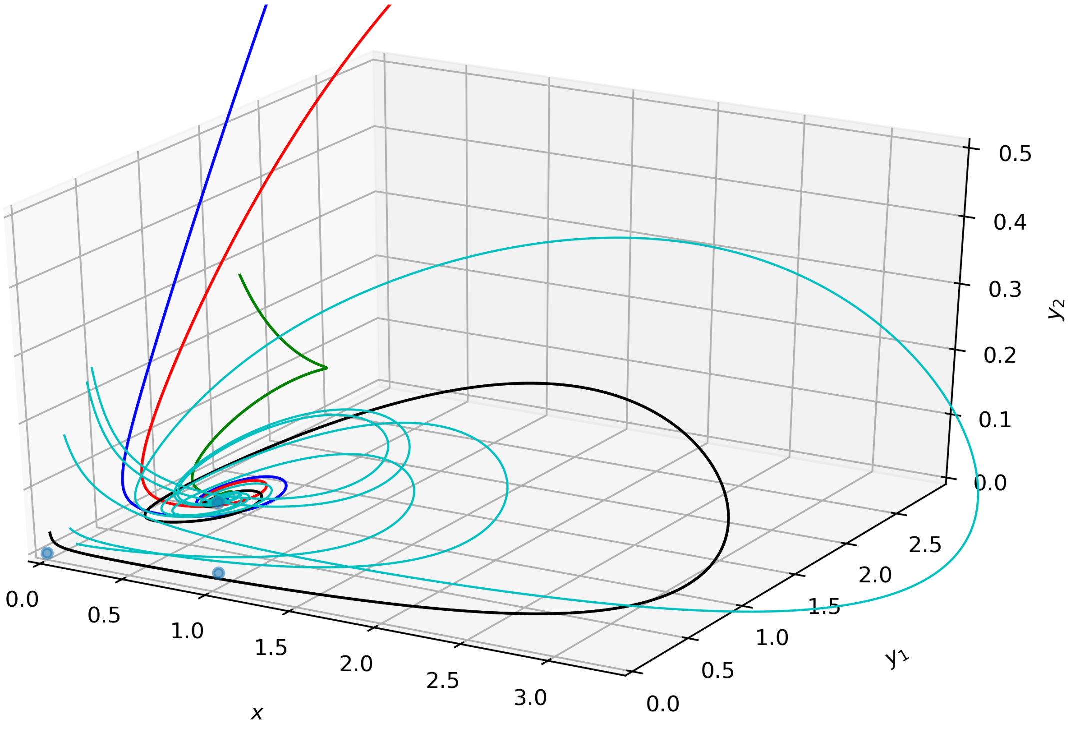


(b) Phase portrait for different initial conditions. All trajectories converge to $EP_3 = (0.32, 1., 0.)$.

Figure 8.21: Time series and phase portrait of the system for $\omega_2 = 1$ and other equal to their base values.

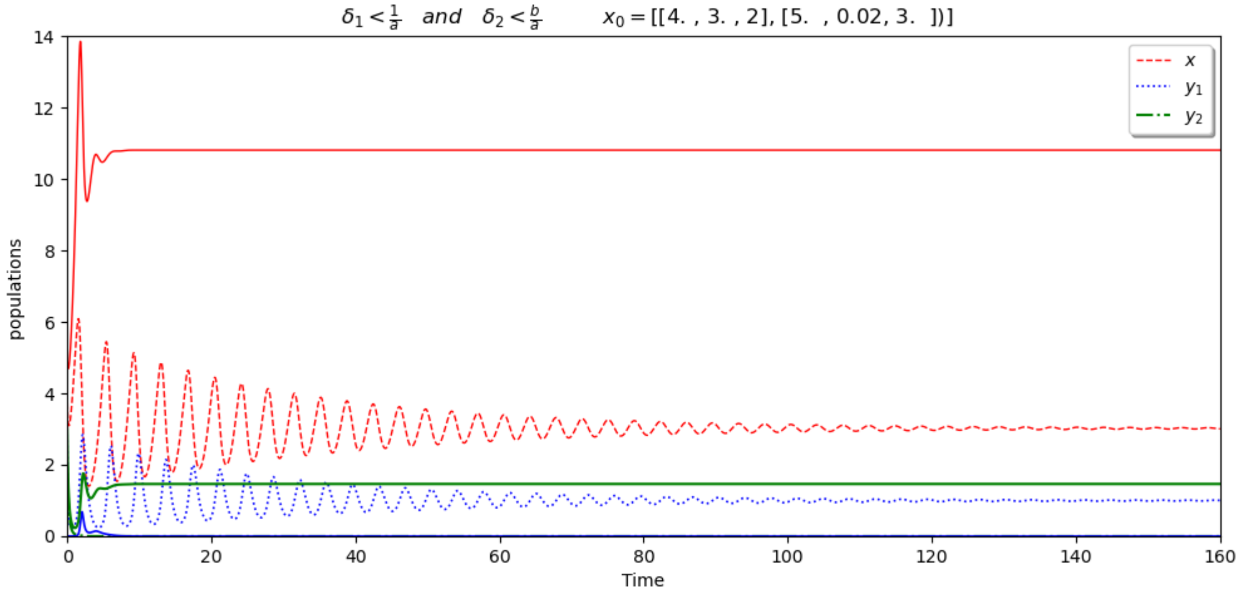


(a) Time series for $\omega_2 = 5$. The populations converge to $EP_3 = (0.32, 1., 0.)$

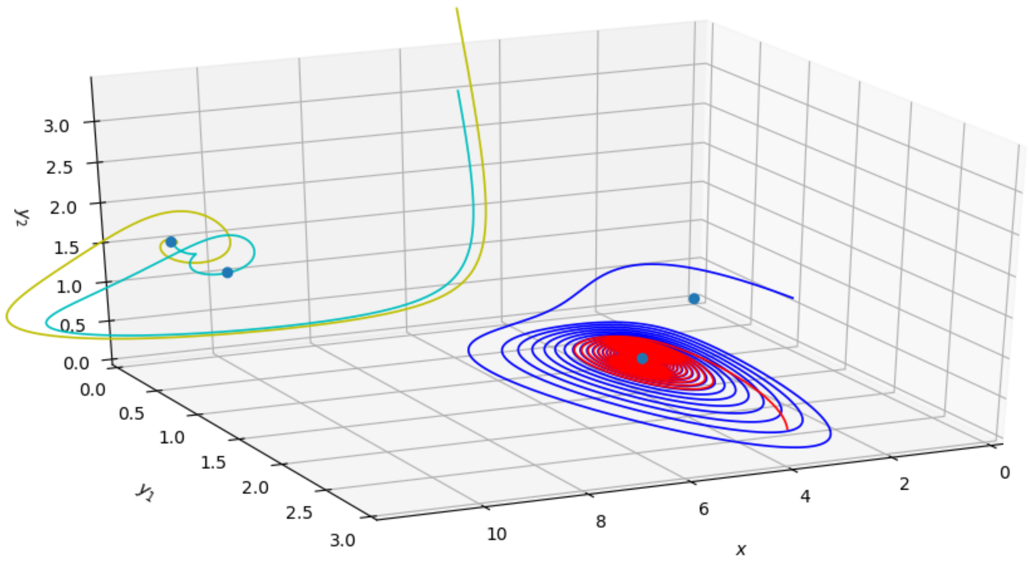


(b) Phase portrait for different initial conditions. All trajectories converge to $EP_3 = (0.32, 1., 0.)$.

Figure 8.22: Time series and phase portrait of the system for $\omega_2 = 5$ and others equal to their base values.

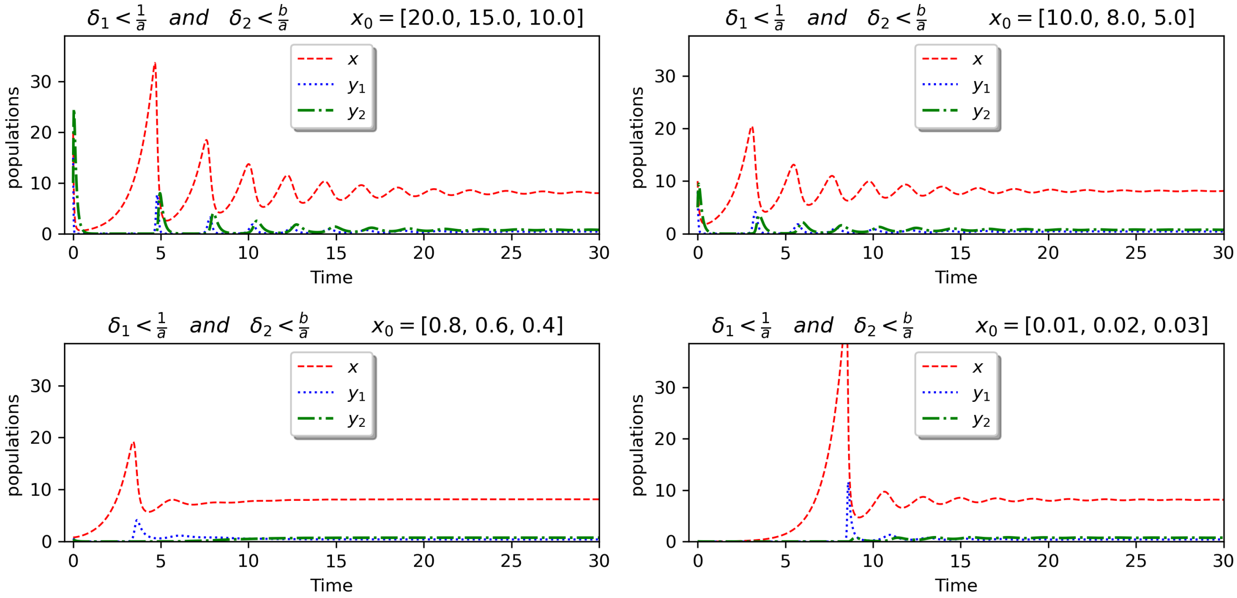


(a) Time series for PM_{34} and two different initial conditions. The dashed solution starts from $(4, 3, 2)$ and converges to $EP_3 = (3.02, 1., 0.)$, while the continuous solution starts from $(5, 0.02, 3)$ and converges to $EP_4 = (10.81, 0., 1.46)$.

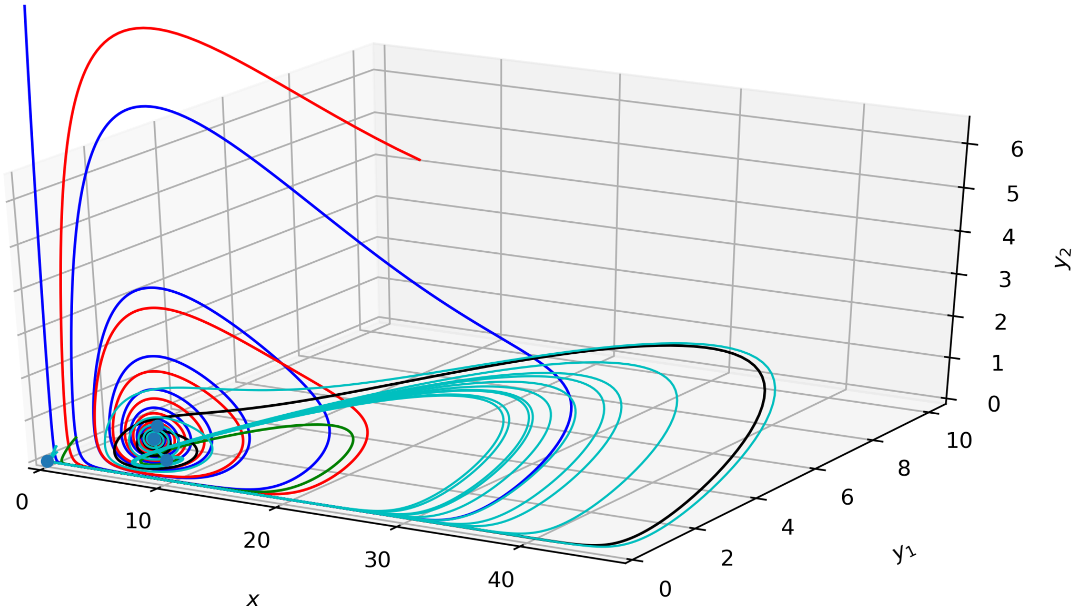


(b) Phase portrait for PM_{34} and several initial conditions. The blue trajectories represent the dashed solution, whereas the cyan trajectory represents the continuous solution from time series.

Figure 8.23: Time series and phase portrait for the system for PM_{34} and two different initial conditions.

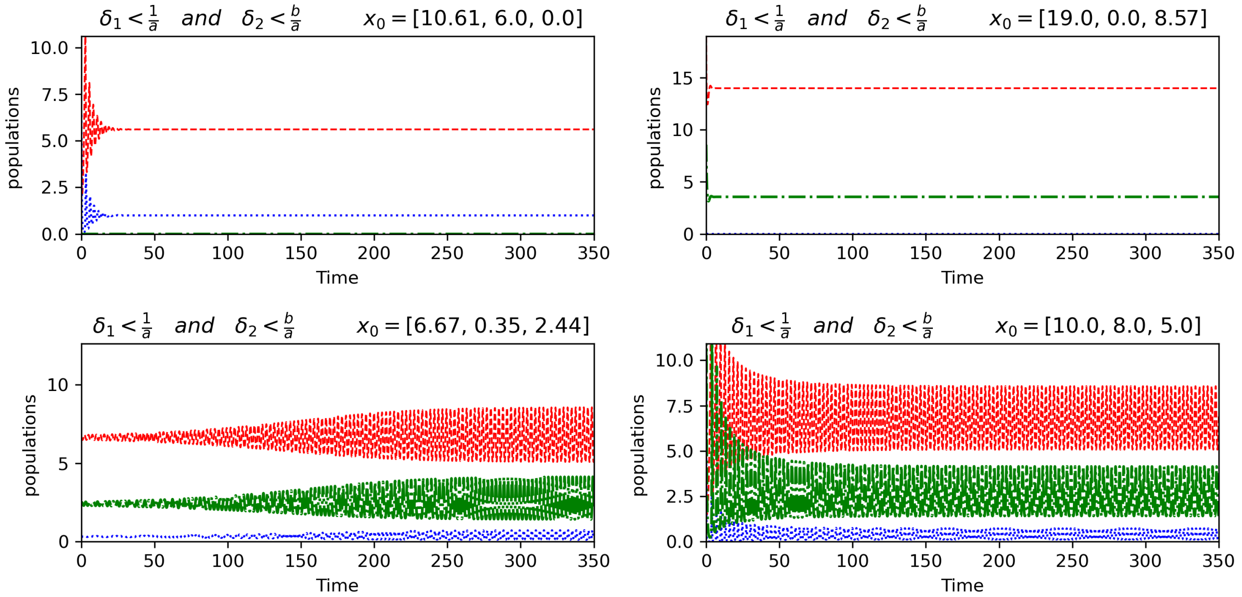


(a) Time series for the parameter set PM_{gs} and four different initial conditions. All solutions converge to the eqm. point $(8.13, 0.41, 0.75)$. Thus, it is globally stable.

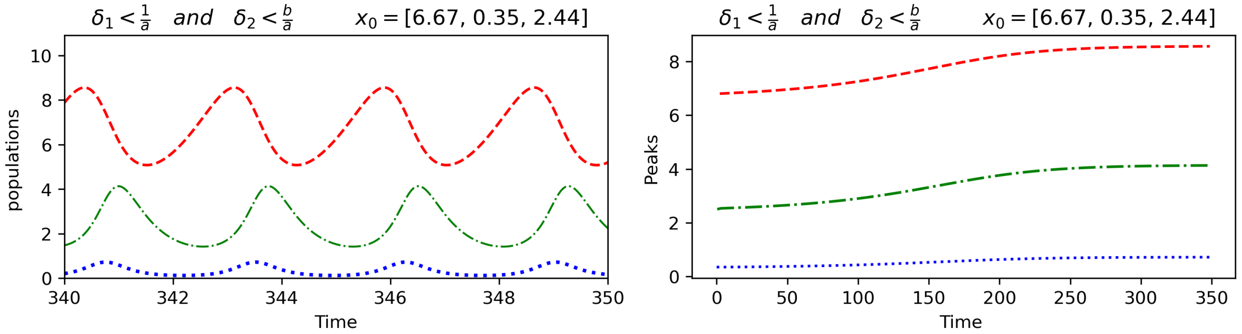


(b) Phase portrait for several initial conditions. The blue, red, green, and black trajectories represent solutions that start from $(20, 15, 10), (10, 8, 5), (0.8, 0.6, 0.4), (0.01, 0.02, 0.03)$, respectively, while the cyan trajectories correspond to solutions that start from random initial conditions between zero and unity. All trajectories converge to the interior equilibrium point $(8.13, 0.41, 0.75)$, showing its global stability.

Figure 8.24: Time series and phase portrait for the system for the parameter set PM_{gs}.

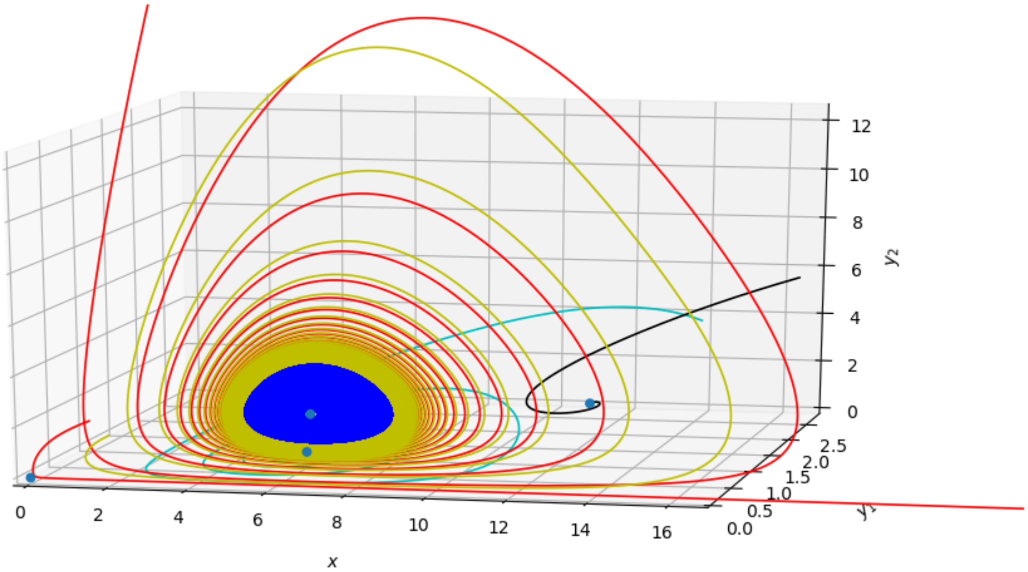


(a) Time series for the parameter set $PM_{periodic_1}$ and four different initial conditions. The colors red, blue, and green represent x, y_1, and y_2 respectively. The solution in the upper left plot converge to $(5.61, 1.0, 0.0)$, upper right to $(14.0, 0.0, 3.57)$. However, the solutions in the two bottom plots are periodic with period 2.76.

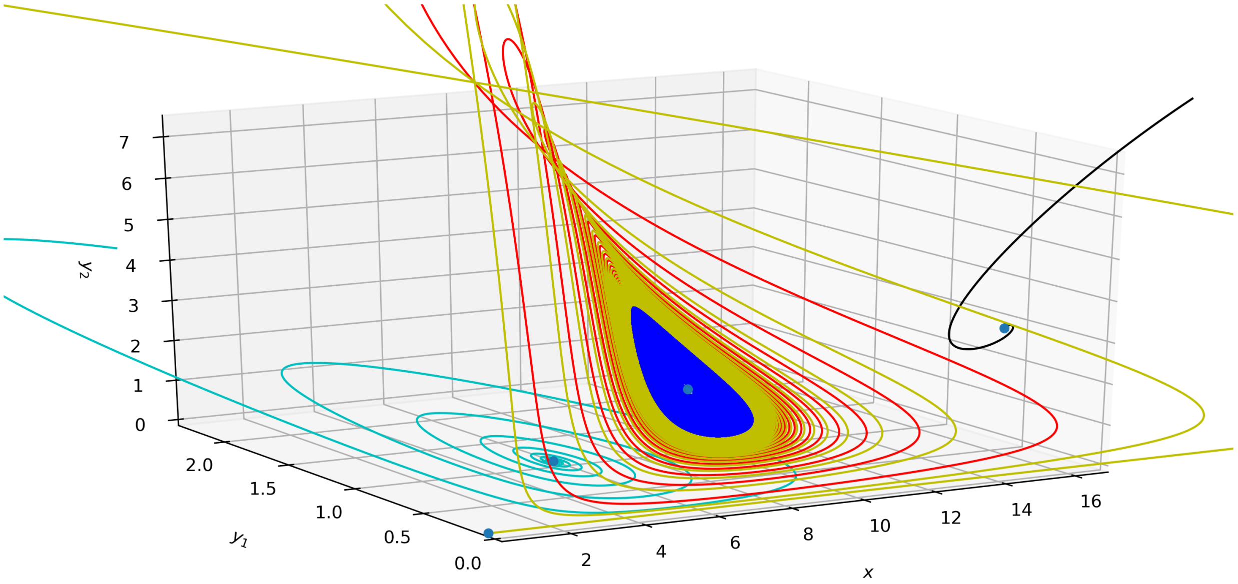


(b) Peaks of the solution that starts from the initial condition $(6.67, 0.35, 2.44)$. All the peaks became constant in finite time (at $t \geq 300$.)

Figure 8.25: Time series and peaks of the system for the parameter set $PM_{periodic_1}$.

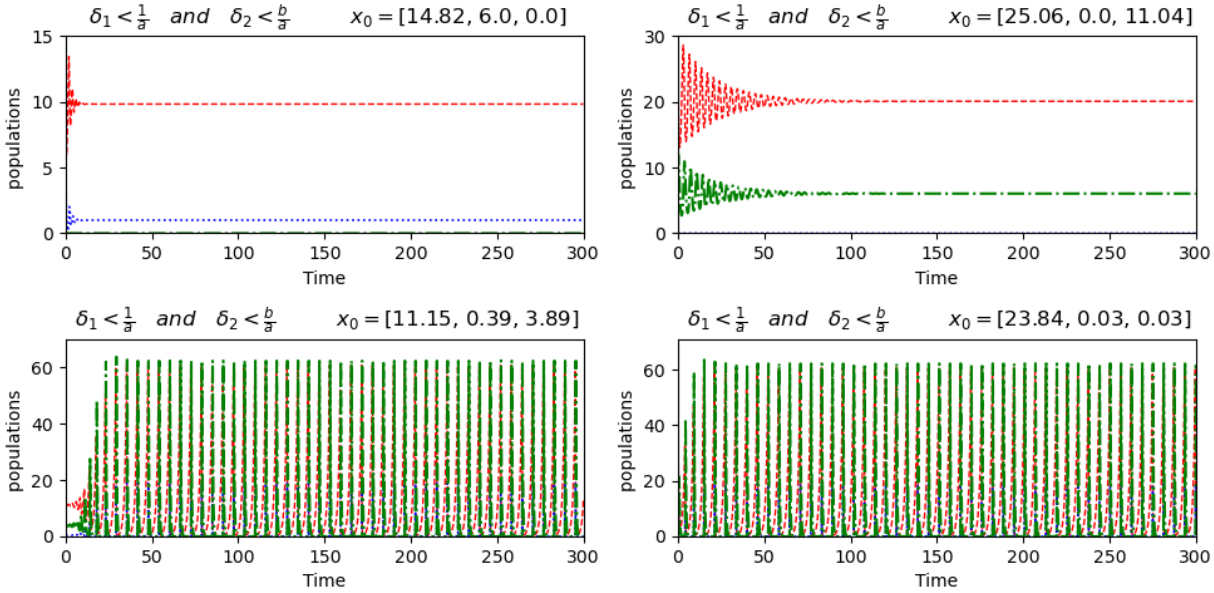


(a) Phase portrait.

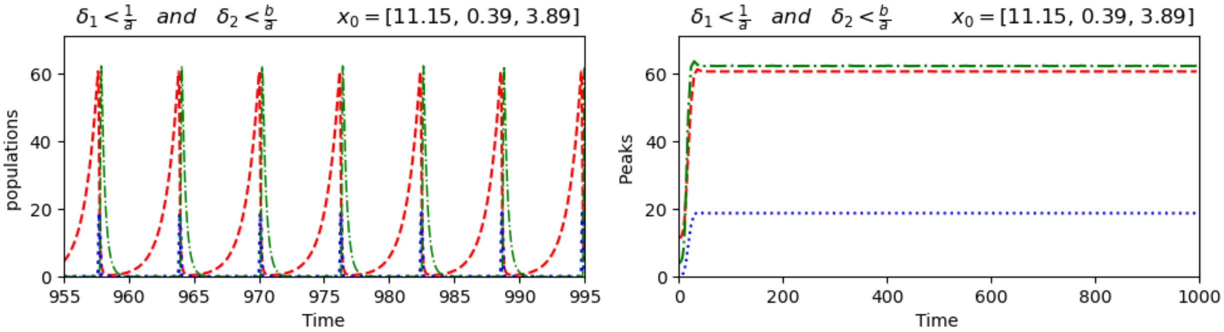


(b) Phase portrait-different view angle.

Figure 8.26: Phase portrait of the system for the parameter set $PM_{periodic_1}$ and five different initial conditions. The cyan, black, blue, and red trajectories represent the solutions in the upper left, upper right, bottom left and bottom right from time series, respectively. The yellow trajectory represents the solution that starts from $(0.01, 0.02, 0.03)$. All trajectories except cyan and black ones converge to the limit cycle, which has a period of 2.76. It is between the blue and yellow trajectories and stable as the blue trajectory converges outwardly to it and the yellow and red trajectories converge inwardly to it.

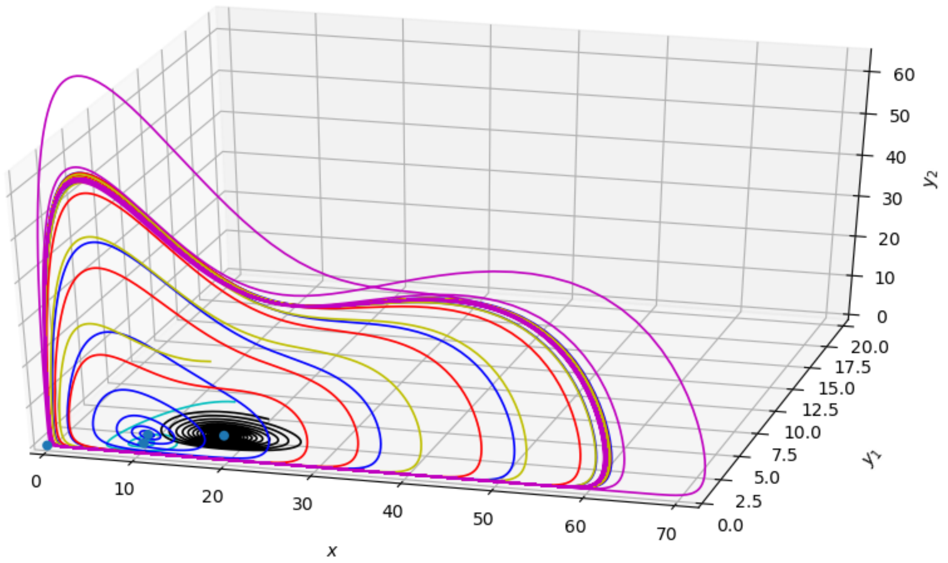


(a) Time series for the parameter set $PM_{periodic_2}$ and four different initial conditions. The colors red, blue, and green represent x, y_1, and y_2 respectively. The solution in the upper left plot converge to $(9.82, 1., 0.)$, upper right to $(20.06, 0., 6.04)$. However the solution in the two bottom plots are peridic with period 6.19.

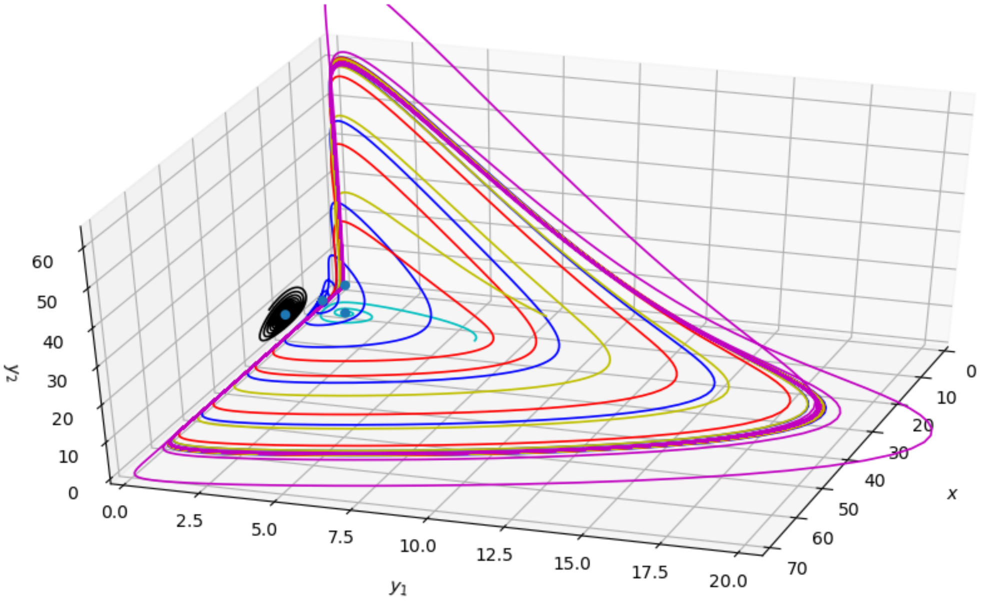


(b) peaks of the solution that starts from the initial conditions $(11.15.0.39, 3.89)$. The peaks became constant after some transient time.

Figure 8.27: Time series and peaks of the system for $PM_{periodic_2}$.

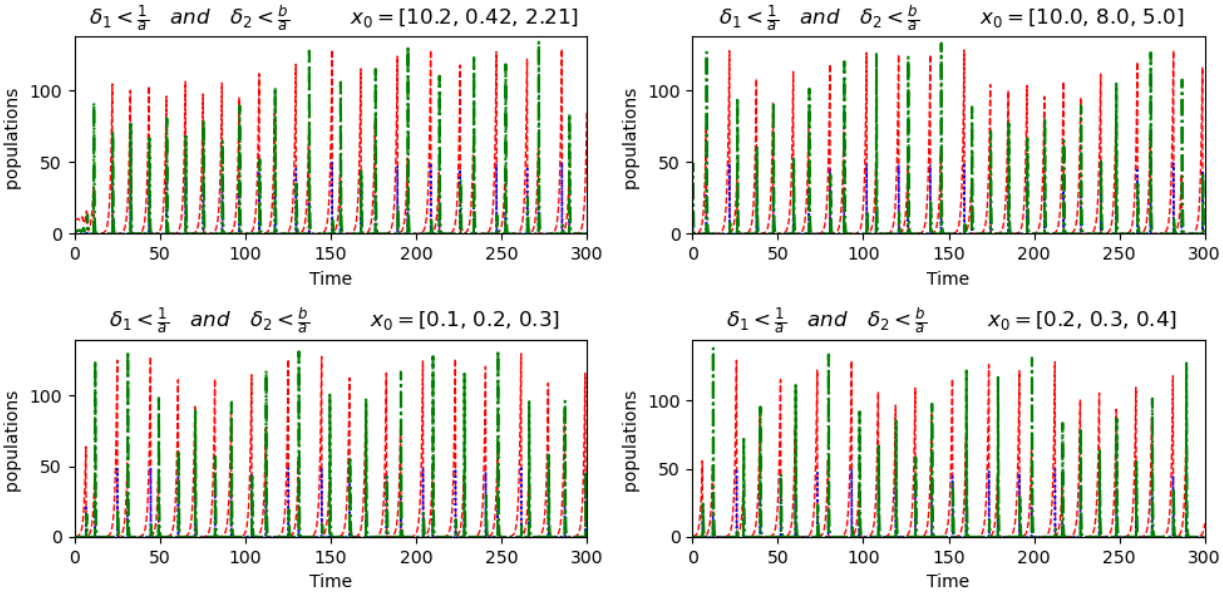


(a) Phase portrait.

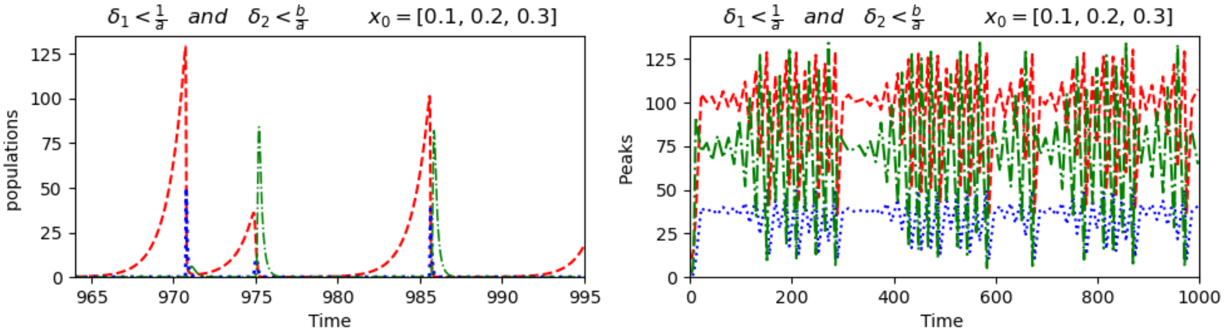


(b) Phase portrait-different view angle.

Figure 8.28: Phase portrait of the system for the parameter set $PM_{periodic_2}$. The cyan, black, blue, and red trajectories represent the solutions in the upper left, upper right, bottom left, and bottom right from time series, respectively. The yellow and magenta trajectories start from $(10., 8., 5.), (60.153, 2.263, 0.04)$. All trajectories except the cyan and black ones converge to the shown attractor.

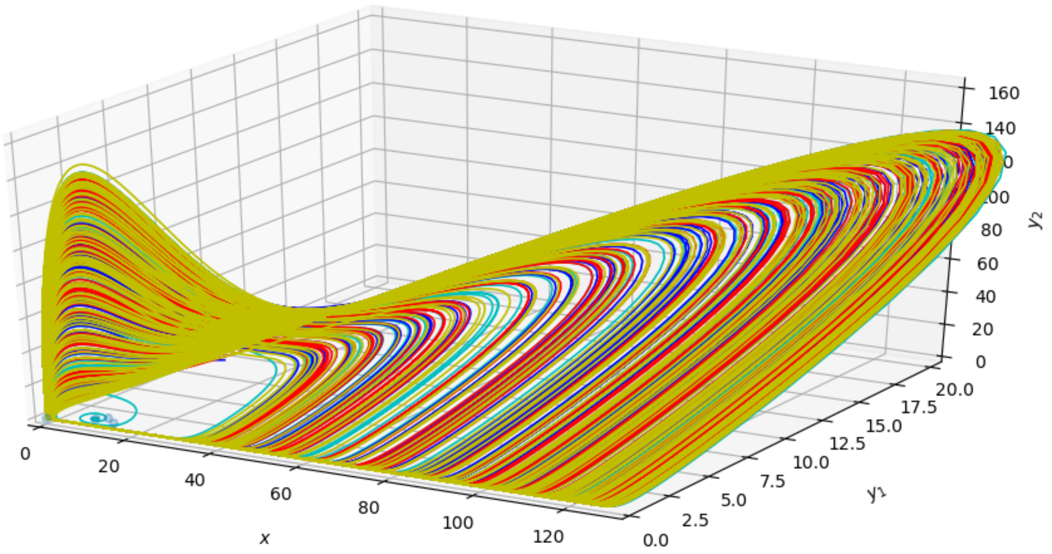


(a) Time series for the parameter set PM_{chaos} and four different initial conditions. All solutions exhibit some kind of cyclic behavior, but they are not periodic.

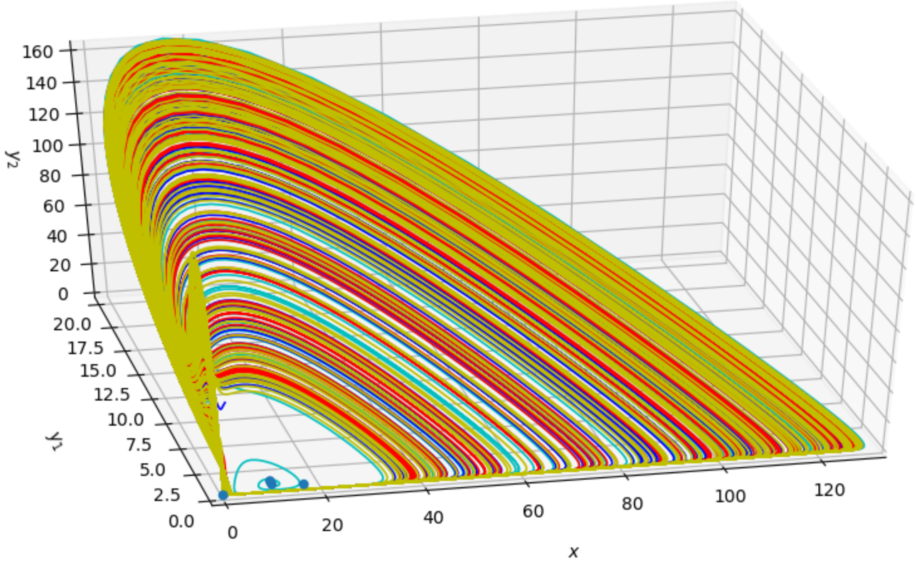


(b) peaks of the solution that starts from the initial conditions $(0.1, 0.2, 0.3)$. As we see, the peaks varies significantly with time.

Figure 8.29: Time series and peaks of the system for the parameter set PM_{chaos}.

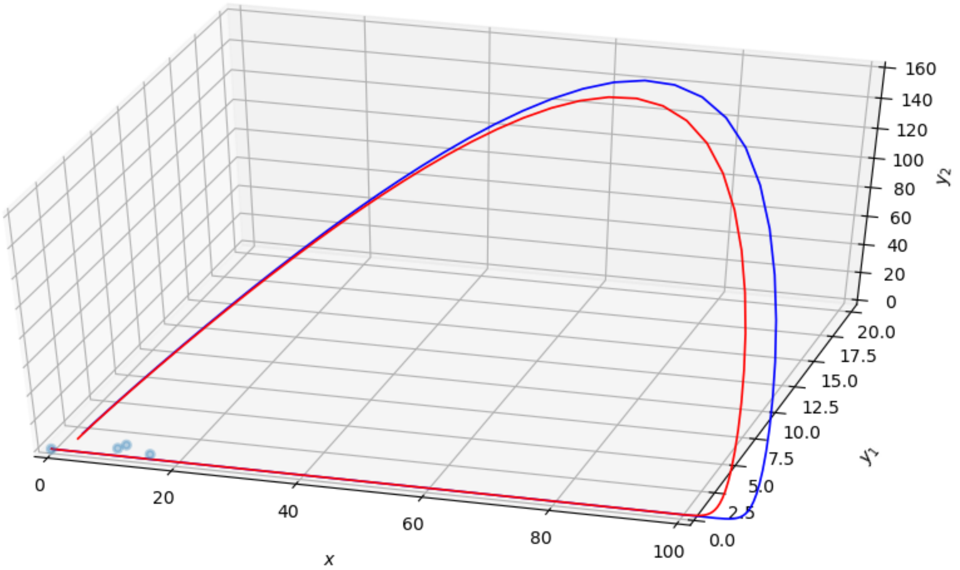


(a) Phase portrait.

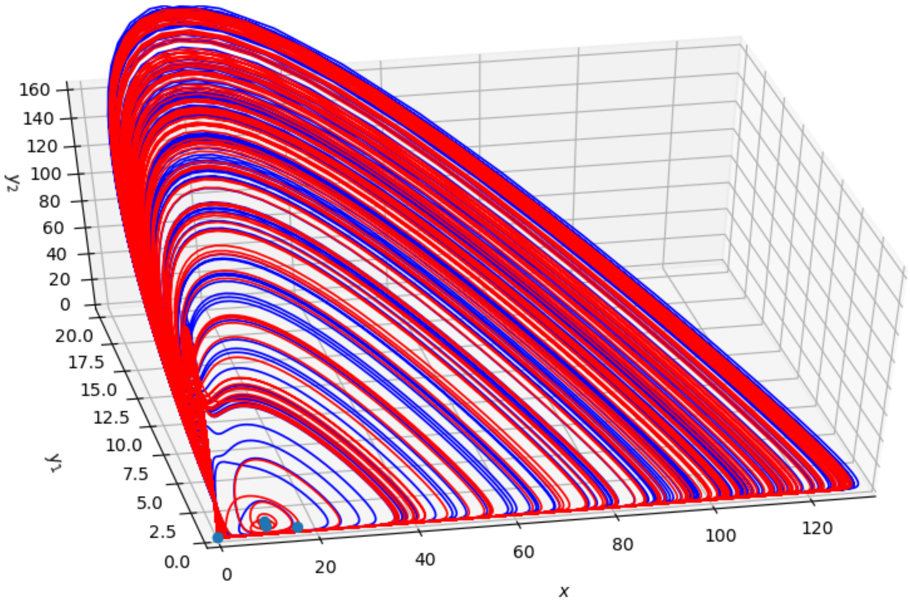


(b) Phase portrait-different angle of view.

Figure 8.30: Phase portrait of the system for PM_{chaos} and four different initial conditions. The cyan, blue, red, and yellow trajectories represent the solutions in the upper left, upper right, bottom left, and bottom right in the time series, respectively. The blue points represent the equilibrium points in the system. All the trajectories converge to the shown strange attractor.

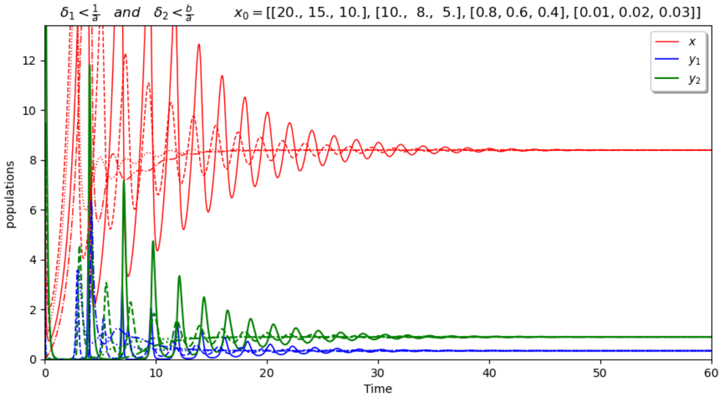


(a) Integration time is 11.

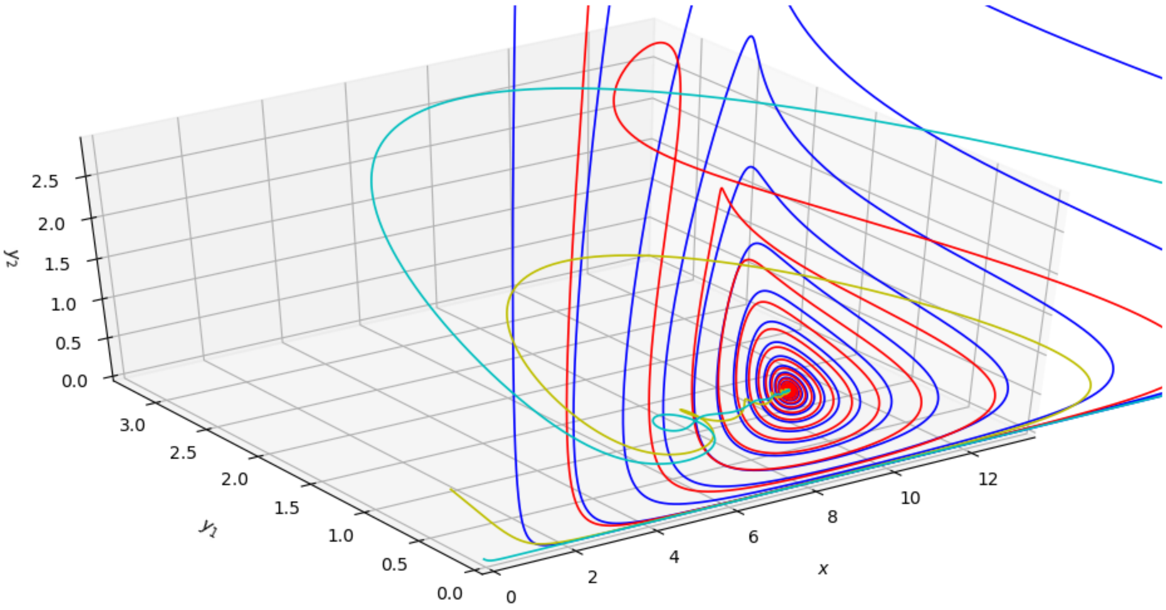


(b) Integration time is 1000.

Figure 8.31: Phase portrait of the system for the parameter set PM_{chaos} and two initial condition with separation 1×10^{-8}. The two trajectories start from $(0.001, 0.002, 0.003)$ with separation 1×10^{-8} and diverge from each other in short time ($t \geq 11$), indicating sensitivity to initial conditions and chaos.

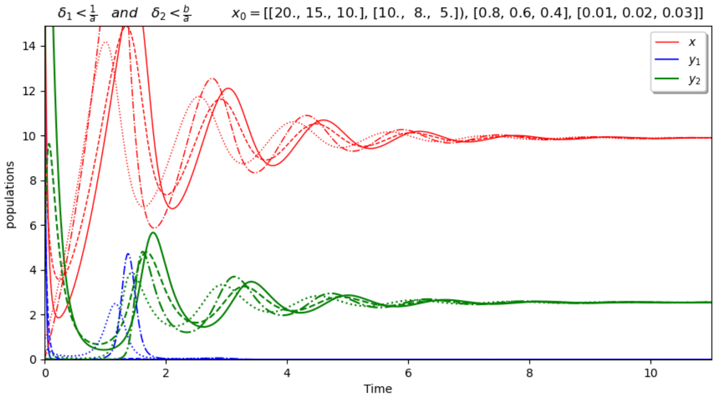


(a) Time series for $s = 0.5$ and four different initial conditions. All solutions converge to $EP_5 = (8.395, 0.347, 0.90)$, showing its global stability.

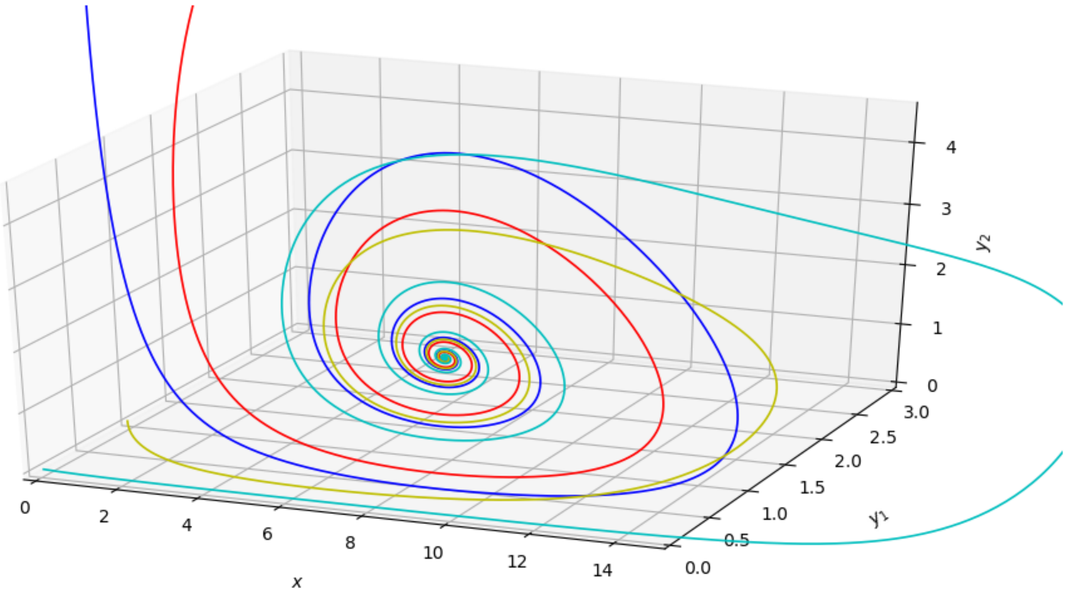


(b) Phase portrait for $s = 0.5$ and four different initial conditions. All trajectories converge to $EP_5 = (8.395, 0.347, 0.90)$, showing its global stability.

Figure 8.32: Time series and phase portrait of the system for PM_{gs} and $s = 0.5$.



(a) Time series for $s = 10$ and four different initial conditions. All solutions converge to $EP_3 = (9.888, 0.0, 2.545)$.



(b) Phase portrait for $s = 10$ and four different initial conditions. All trajectories converge to $EP_3 = (9.888, 0.0, 2.545)$.

Figure 8.33: Time series and phase portrait of the system for PM_{gs} and $s = 10$.

Chapter 9

Summary, conclusion, and further work

9.1 Summary and conclusion

We have made a dynamical model for the interaction between pathogenic bacteria and two other microorganisms that feed on them. The model consists of 3 coupled nonlinear ordinary differential equations. We have performed linear stability analysis of the system and proved that the system has five non-negative equilibrium points where the system can converge to. The first one is the zero-equilibrium point, i.e. none of the species are present and it is always unstable in x direction but stable in y_1 and y_2 directions. Consequently, any perturbation in the bacteria population will lead the system to other equilibrium points. However, if the populations of Daphnia or Protozoa increase, the system will converge back to it. The second one is when only bacteria x is present and it is an unstable saddle point as it is stable in the x direction and unstable in the $y_1 y_2$ plane. This means that if we start with $x > 0, y_1, y_2 = 0$ and the number of bacteria changed, the system will converge to it after some time. The third and fourth equilibrium points are when one of the predators (protozoa y_1 or Daphnia y_2) does not exist and these could be stable or unstable based on some condition on the parameters. The last equilibrium point is the interior equilibrium point where all species are present and is the most important one. This point exists only if certain conditions on parameters are satisfied. Furthermore, it is globally stable if the following condition on parameters $2a\left(4b\omega_1\omega_2 - (b\psi - \phi)^2\right) > 0$ is satisfied. Its global stability means that if we start with initial condition with all species have concentrations greater than zero, the system will converge to it in finite time. Consequently, all species will coexist for an indefinite amount of time and with constant concentrations.

The question that may arise here is, what happens if the system does not have any stable equilibrium points? To where the system will converge given that the system is bounded? To answer these important questions we conducted random sampling for parameters in the system using, for example, the Saltelli sampling method from the sensitivity analysis library [19] and classified the parameter sets that gave stable equilibrium points and the ones for which the system does not have any stable equilibrium points. Our numerical simulation for these parameter sets showed that the system will exhibit a periodic behavior and may converge to a limit cycle (see Figures 8.25,8.26) or will be chaotic (see figures 8.29 and 8.30) with all solutions converge to some kind of strange attractor and it will be difficult to predict the long time behavior of the system due to the sensitive dependence on the initial condition.

Usually, it is not biologically desirable to have a system that exhibits periodic or chaotic behavior. To drive the system from periodicity or chaos, we can use some parameters as a control measure. In the case of periodic behavior, we observed that the parameter ϕ can be used to drive the system into or out of periodicity. Additionally, in the case of a chaotic system, we can use one of the parameters b, δ_2, ψ to drive the system into one of the equilibrium points (see Table 8.15).

It is known that there is a direct flow of bacteria into surface water resources by fecal contamination of some animals either directly or by contaminated water that flows into these water resources in fall or in summer when the snow melts. We have investigated the effect of this on the coexistence of the species for one set of parameters (see Table 8.16). The conclusion was that increasing this effect above a critical value will surprisingly lead to the extinction of the first predator, Protozoa.

Our reduced dimensional system 4.1 had 12 parameters. In order to reduce the number of parameters we nondimensionalized the system using the scaling 4.2 and we obtained the system 4.3 which has 8. To investigate which parameters are the important ones and have the most effect on the model output, we performed a global sensitivity analysis using four different sensitivity analysis methods and for four variables of interest which are the max, mean, median, equilibrium value. As a result, we can arrange the parameter as follows in terms of importance $b, \delta_2, \delta_1, \omega_2, \omega_1, \psi, \phi, a, x_0, y_{1_0}, y_{2_0}$ for all variables of interest except the maximum value for which the sensitive parameters are x_0 for x, y_{1_0} for y_1 and ω_2, b, y_{2_0} for max value of y_2. The parameter b is the most sensitive one, and this is not surprising as it represents the ratio between the food conversion efficiency of the two predators, i.e. their ability to convert the energy gained by preying on bacteria to new predators.

One question which could be of interest to biologists and authorities is, do we have to measure all the species to observe the state of the system or it is enough to only observe one of them? In an attempt to answer this question, we have investigated the observability of the system 4.3 and found out that the system is observable by only observing one of the species. Usually, water quality authorities only measure the concentration of bacteria in surface drinking water resources, and it is valuable to know that this is enough to observe the whole system.

Furthermore, we analyzed the bacteria data from different water resources in Bergen municipality in Norway. For that purpose, we computed the statistics for each individual bacterium and the total number of bacteria along with scatter plots of bacteria as a function of time. From this simple analysis, we found out that there is aperiodicity in the time evolution of the bacteria. There are periods of upspring where the concentration of bacteria is highest and periods where the concentration is low or zero. If we compare Figure 8.29 with Figures 7.2, 7.1, 7.3 , we can see some kind of match between them in terms of existence of spikes and cyclic behavior. Consequently, our model managed to demonstrate the general trend of data for the chaotic parameter set. This may indicate that there is indeed chaos in real life, and predictability of the system is not possible for a long time. However,this needs more detailed study.

In conclusion, our proposed model for the interaction between the three species, Bacteria, Protozoa, and Daphnia gives interesting dynamics. There are two possible scenarios for the interaction between these species, namely extinction of one of the predators or coexistence depending on the initial condition and values of the parameters. In terms of coexistence, the model demonstrates three types,

- coexistence at a stable equilibrium point: i.e. all three species coexist at constant concentrations.
- coexistence in a periodic fashion.
- coexistence in a chaotic fashion, i.e. the cycle repeats but not in the same manner, and it is not possible to predict the future of the system for a long time.

9.2 Further work

There are several concepts and ideas in this project that can be checked, but because of the scope of time, we have not focused on them. One could use Monod prey dependent functional response instead of the logistic equation for bacteria growth and Contois or Arditi-Ginzburg

II which are ratio-dependent instead of Lotka-Volterra linear functional response and check if one can get more interesting dynamics. However, Kesh et al. [28] showed that this is not the case for Lotka-Volterra systems.

We have ignored the idea that some of the bacteria x become resistant to protozoa y_1 digestion and this is important in terms of biology. One suggestion is to multiply the interaction terms xy_1 by a factor $(1 - \Omega)$ where Ω could be a resistance factor or fraction of bacteria that becomes resistant. If $\Omega = 0$, then we come back to our model, but what will be the effect of values of Ω that is greater than zero on the dynamics of the system.

Throughout the project, we ignored the parameter $s(t)$ which represents the flux of bacteria to the water system due to fecal contamination of animals. This could happen all the time, especially in fall by the flux of accumulated water from rain into drinking water resources or in the summer when the snow melts and moves down from the mountain to these resources. Thus, it may be a good idea to make the parameter s time-dependent, not just a constant, and study its effect on the system in terms of stability and sensitivity.

More work needs to be done to analyze bacteria data, for example, curve fitting, periodicity detection, and maybe use the data to perform parameter estimation and to determine the existence of chaos and match the results with chaos from the model.

On important concept in biological systems is controllability to investigate if the system is controllable and choose the proper control function. Is it possible to control the bacteria growth by, for example, increase the concentration of the predators y_1 and y_2 only? Can we reach the total extinction of the bacteria by such a measure? These questions need more detailed work.

Finally, In this work we only analyzed nondimensionalized version of the subsystem 4.1 and it would be interesting to analyze the non-reduced system 2.1 at least numerically and investigate the dynamics.

Bibliography

[1] P. B. Adler, D. Smull, K. H. Beard, R. T. Choi, T. Furniss, A. Kulmatiski, J. M. Meiners, A. T. Tredennick, and K. E. Veblen. Competition and coexistence in plant communities: intraspecific competition is stronger than interspecific competition. *Ecology letters*, 21(9):1319–1329, 2018.

[2] N. Ali, B. Mondal, L. N. Guin, and S. Chakravarty. Dynamics of species in a two preys-one predator model system with help.

[3] R. Arditi and L. R. Ginzburg. *How species interact: altering the standard view on trophic ecology*. Oxford University Press, 2012.

[4] F. Bichai, P. Payment, and B. Barbeau. Protection of waterborne pathogens by higher organisms in drinking water: a review. *Canadian journal of microbiology*, 54(7):509–524, 2008.

[5] R. Blaustein, Y. Pachepsky, R. Hill, D. Shelton, and G. Whelan. Escherichia coli survival in waters: temperature dependence. *Water research*, 47(2):569–578, 2013.

[6] E. Borgonovo. A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92(6):771–784, 2007.

[7] W. E. Boyce. Elementary differential equations and boundary value problems, 2017.

[8] M. N. Byappanahalli, M. B. Nevers, A. Korajkic, Z. R. Staley, and V. J. Harwood. Enterococci in the environment. *Microbiol. Mol. Biol. Rev.*, 76(4):685–706, 2012.

[9] F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental modelling & software*, 22(10):1509–1518, 2007.

[10] D. Contois. Kinetics of bacterial growth: relationship between population density and specific growth rate of continuous cultures. *Microbiology*, 21(1):40–50, 1959.

[11] G. Datseris. Dynamicalsystems.jl: A julia software library for chaos and nonlinear dynamics. *Journal of Open Source Software*, 3(23):598, mar 2018.

[12] J. Dawes and M. Souza. A derivation of holling's type i, ii and iii functional responses in predator–prey systems. *Journal of theoretical biology*, 327:11–22, 2013.

[13] R. C. Dorf and R. H. Bishop. *Modern control systems.* Pearson, 2011.

[14] D. Ebert. *Ecology, epidemiology, and evolution of parasitism in Daphnia.* National Library of Medicine, 2005.

[15] M. Elettreby. Two-prey one-predator model. *Chaos, Solitons and Fractals*, 39(5):2018–2027, 2009.

[16] K. Geist, U. Parlitz, and W. Lauterborn. Comparison of different methods for computing lyapunov exponents. *Progress of theoretical physics*, 83(5):875–893, 1990.

[17] G. Glen and K. Isaacs. Estimating sobol sensitivity indices using correlations. *Environmental Modelling & Software*, 37:157–166, 2012.

[18] V. Govorukhin. Calculation lyapunov exponents for ode. `https://www.mathworks.com/matlabcentral/fileexchange/4628-calculation-lyapunov-exponents-for-ode`,, 2020.

[19] J. Herman and W. Usher. SALib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software*, 2(9), jan 2017.

[20] S. Hsu and S. P. Hubbell. Two predators competing for two prey species: an analysis of macarthur's model. *Mathematical Biosciences*, 47(3-4):143–171, 1979.

[21] A. O. S. I. V. I. . E. B. Johnsen, G.H. Vassdragsundersøkelser i nedbørfeltene til vannverkene på byfjellene i bergen sommeren 2000. `https://www.radgivende-biologer.no/rapporter/ar-2001/vassdragsundersokelser-i-nedborfeltene-til-vannverkene-pa-byfjellene-i-bergen-sommeren-2000`, 2001.

[22] G. H. . E. B. Johnsen. Vannkvalitet i jordalsvassdraget i 2016. `https://www.radgivende-biologer.no/rapporter/ar-2016/vannkvalitet-i-jordalsvassdraget-i-2016/`, 2020.

[23] K. A. Johnson and R. S. Goody. The original michaelis constant: translation of the 1913 michaelis–menten paper. *Biochemistry*, 50(39):8264–8269, 2011.

[24] C. Jost, O. Arino, and R. Arditi. About deterministic extinction in ratio-dependent predator–prey models. *Bulletin of Mathematical Biology*, 61(1):19–32, 1999.

[25] K. Jürgens, J. M. Gasol, R. Massana, and C. Pedrós-Alió. Control of heterotrophic bacteria and protozoans by daphnia pulex in the epilimnion of lake cisó. *Archiv fur Hydrobiologie*, 131:55–55, 1994.

[26] H. Kantz. A robust method to estimate the maximal lyapunov exponent of a time series. *Physics letters A*, 185(1):77–87, 1994.

[27] T. Kar and A. Batabyal. Persistence and stability of a two prey one predator system. *International Journal of Engineering, Science and Technology*, 2(2):174–190, 2010.

[28] D. Kesh, A. Sarkar, and A. Roy. Persistence of two prey–one predator system with ratio-dependent predator influence. *Mathematical methods in the applied sciences*, 23(4):347–356, 2000.

[29] L. K. Korhonen and P. Martikalnon. Survival of escherichia coli and campylobacter jejuni in untreated and filtered lake water. *Journal of Applied Bacteriology*, 71(4):379–382, 1991.

[30] S. S. Lee. Analysis of dynamics in two-prey, one-predator model: Effect of the remained carcass (theory of biomathematics and its applications iii). 2007.

[31] M. Manca, C. Callieri, and A. Cattaneo. Daphnia and ciliates: who is the prey? *Journal of Limnology*, 66(2):170–173, 2007.

[32] C. D. Meyer. *Matrix analysis and applied linear algebra*, volume 71. Siam, 2000.

[33] R. Milo, P. Jorgensen, U. Moran, G. Weber, and M. Springer. Bionumbers–the database of key numbers in molecular and cell biology. *Nucleic acids research*, 38(Database issue):D750–D753, Jan 2010. 19854939[pmid].

[34] J. Monod. The growth of bacterial cultures. *Annual review of microbiology*, 3(1):371–394, 1949.

[35] M. D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[36] J. D. Murray. Mathematical biology i: An introduction, volume i, 2003.

[37] K. Ogata and Y. Yang. *Modern control engineering*, volume 4. Prentice hall India, 2002.

[38] S. P. Otto and T. Day. *A biologist's guide to mathematical modeling in ecology and evolution.* Princeton University Press, 2011.

[39] D. Pal, P. Santra, and G. Mahapatra. Dynamical behavior of three species predator–prey system with mutual support between non refuge prey. *Ecological Genetics and Genomics*, 3-5:1–6, 2017.

[40] E. Plischke. An effective algorithm for computing global sensitivity indices (easi). *Reliability Engineering & System Safety*, 95(4):354–360, 2010.

[41] E. Plischke, E. Borgonovo, and C. L. Smith. Global sensitivity measures from given data. *Eur. J. Oper. Res.*, 226:536–550, 2013.

[42] K. S. Reddy, M. Srinivas, and C. P. Kumar. Dynamics of a two prey and one predator system with time interruption and random fluctuations. *Pacific Science Review A: Natural Science and Engineering*, 18(2):150–156, 2016.

[43] E. J. Routh. *A treatise on the stability of a given state of motion: particularly steady motion.* Macmillan and Company, 1877.

[44] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.

[45] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity analysis in practice: a guide to assessing scientific models*, volume 1. Wiley Online Library, 2004.

[46] B. Samuel. Medical microbiology. *The University of Texas Medical Branch at Galveston–Tx, USA*, 1996.

[47] P. A. Saunders, K. G. Porter, and B. E. Taylor. Population dynamics of daphnia spp. and implications for trophic interactions in small, monomictic lake. *Journal of Plankton Research*, 21(10), 1999.

[48] M. Schallenberg, P. Bremer, S. Henkel, A. Launhardt, and C. Burns. Survival of campylobacter jejuni in water: effect of grazing by the freshwater crustacean daphnia carinata (cladocera). *Appl. Environ. Microbiol.*, 71(9):5085–5088, 2005.

[49] W. J. Snelling, J. E. Moore, J. P. McKenna, D. M. Lecky, and J. S. Dooley. Bacterial–protozoa interactions; an update on the role these phenomena play towards human illness. *Microbes and Infection*, 8(2):578–587, 2006.

[50] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[51] J. C. Sprott. *Elegant chaos: algebraically simple chaotic flows.* World Scientific, 2010.

[52] J. C. Sprott. Numerical calculation of largest lyapunov exponent. `http://sprott.physics.wisc.edu/chaos/lyapexp.htm`, 2020.

[53] J. C. Sprott and J. C. Sprott. *Chaos and time-series analysis*, volume 69. Citeseer, 2003.

[54] G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, 2016.

[55] S. H. Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* CRC press, 2018.

[56] S. L. Strom and T. A. Morello. Comparative growth rates and yields of ciliates and heterotrophic dinoflagellates. *Journal of Plankton Research*, 20(3):571–584, 1998.

[57] S. Tarantola, D. Gatelli, and T. A. Mara. Random balance designs for the estimation of first order global sensitivity indices. *Reliability Engineering & System Safety*, 91(6):717–727, 2006.

[58] C. Thomas, H. Gibson, D. Hill, and M. Mabey. Campylobacter epidemiology: an aquatic perspective. *Journal of Applied Microbiology*, 85(S1):168S–177S, 1998.

[59] J.-Y. Tissot and C. Prieur. Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliability Engineering & System Safety*, 107:205–213, 2012.

[60] V. Volterra. Variations and fluctuations of the number of individuals in animal species living together. *ICES Journal of Marine Science*, 3(1):3–51, 1928.

[61] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[62] X. Zhou and H. Lin. *Local Sensitivity Analysis*, pages 1130–1131. Springer International Publishing, Cham, 2017.

Appendices

Appendix A

Derivation of the model,nondimensionalization, and GSA for the dimensional model

A.1 Derivation of the model

In this section, we present the detailed derivation of the model 2.1.

$$
\begin{aligned}
\frac{dx_1(t)}{dt} =& r_1x_1 - \sigma_{11}x_1x_1 - a_1x_1x_2x_3x_4 - f_{11}x_1y_1 - f_{12}x_1y_2 + s_1(t),\\
\frac{dx_2(t)}{dt} =& r_2x_2 - \sigma_{22}x_2x_2 - a_2x_1x_2x_3x_4 - f_{21}x_2y_1 - f_{22}x_2y_2 + s_2(t),\\
\frac{dx_3(t)}{dt} =& r_3x_3 - \sigma_{33}x_3x_3 - a_3x_1x_2x_3x_4 - f_{31}x_3y_1 - f_{32}x_3y_2 + s_3(t),\\
\frac{dx_4(t)}{dt} =& r_4x_4 - \sigma_{44}x_4x_4 - a_4x_1x_2x_3x_4 - f_{41}x_4y_1 - f_{42}x_4y_2 + s_4(t),\\
\frac{dy_1(t)}{dt} =& - \epsilon_1y_1 - \omega_{11}y_1^2 - \omega_{12}y_1y_2 - g_{12}y_1y_2 + \beta_{11}f_{11}y_1x_1\\
&+ \beta_{21}f_{21}y_1x_2 + \beta_{31}f_{31}y_1x_3 + \beta_{41}f_{41}y_1x_4,\\
\frac{dy_2(t)}{dt} =& - \epsilon_2y_2 - \omega_{22}y_2^2 - \omega_{21}y_2y_1 + \gamma_{12}g_{12}y_2y_1 + \beta_{12}f_{12}y_2x_1\\
&+ \beta_{22}f_{22}y_2x_2 + \beta_{32}f_{32}y_2x_3 + \beta_{42}f_{42}y_2x_4,
\end{aligned}
$$

with initial conditions $N(0) > 0, x_{1..4}(0) > 0, y_1(0) > 0, y_2(0) > 0$.

Putting the competition term as a product of prey species doesn't seem to be logical because if one species is not existent then the competition term cancels out in all other equation, in addition, each prey species compete differently with each of other prey species with a different rate.

$$
\begin{aligned}
\frac{dx_1(t)}{dt} =& r_1x_1 - \sigma_{11}x_1x_1 - \sigma_{12}x_1x_2 - \sigma_{13}x_1x_3 - \sigma_{14}x_1x_4 - f_{11}x_1y_1 - f_{12}x_1y_2 + s_1(t), \\
\frac{dx_2(t)}{dt} =& r_2x_2 - \sigma_{22}x_2x_2 - \sigma_{21}x_2x_1 - \sigma_{23}x_2x_3 - \sigma_{24}x_2x_4 - f_{21}x_2y_1 - f_{22}x_2y_2 + s_2(t), \\
\frac{dx_3(t)}{dt} =& r_3x_3 - \sigma_{33}x_3x_3 - \sigma_{31}x_3x_1 - \sigma_{32}x_3x_2 - \sigma_{34}x_3x_4 - f_{31}x_3y_1 - f_{32}x_3y_2 + s_3(t), \\
\frac{dx_4(t)}{dt} =& r_4x_4 - \sigma_{44}x_4x_4 - \sigma_{41}x_4x_1 - \sigma_{42}x_4x_2 - \sigma_{43}x_4x_3 - f_{41}x_4y_1 - f_{42}x_4y_2 + s_4(t), \\
\frac{dy_1(t)}{dt} =& - \epsilon_1y_1 - \omega_{11}y_1^2 - \omega_{12}y_1y_2 - g_{12}y_1y_2 + \beta_{11}f_{11}y_1x_1 \\
&+ \beta_{21}f_{21}y_1x_2 + \beta_{31}f_{31}y_1x_3 + \beta_{41}f_{41}y_1x_4, \\
\frac{dy_2(t)}{dt} =& - \epsilon_2y_2 - \omega_{22}y_2^2 - \omega_{21}y_2y_1 + \gamma_{12}g_{12}y_2y_1 + \beta_{12}f_{12}y_2x_1 \\
&+ \beta_{22}f_{22}y_2x_2 + \beta_{32}f_{32}y_2x_3 + \beta_{42}f_{42}y_2x_4,
\end{aligned}
$$

Taking r_ix_i as a common factor, putting together y_1y_2 terms in the predators equations, and defining $\phi_{12} = g_{12} + \omega_{12}, \quad \psi_{12} = \gamma_{12}g_{12} - \omega_{21}$

$$
\begin{aligned}
\frac{dx_1(t)}{dt} =& r_1x_1\left(1 - \frac{\sigma_{11}x_1 + \sigma_{12}x_2 + \sigma_{13}x_3 + \sigma_{14}x_4}{r_1}\right) - f_{11}x_1y_1 - f_{12}x_1y_2 + s_1(t),\ x_1(0) > 0 \\
\frac{dx_2(t)}{dt} =& r_2x_2\left(1 - \frac{\sigma_{22}x_2 + \sigma_{21}x_1 + \sigma_{23}x_3 + \sigma_{24}x_4}{r_2}\right) - f_{21}x_2y_1 - f_{22}x_2y_2 + s_2(t),\ x_2(0) > 0 \\
\frac{dx_3(t)}{dt} =& r_3x_3\left(1 - \frac{\sigma_{33}x_3 + \sigma_{31}x_1 + \sigma_{32}x_2 + \sigma_{34}x_4}{r_3}\right) - f_{31}x_3y_1 - f_{32}x_3y_2 + s_3(t),\ x_3(0) > 0 \\
\frac{dx_4(t)}{dt} =& r_4x_4\left(1 - \frac{\sigma_{44}x_4 + \sigma_{41}x_1 + \sigma_{42}x_2 + \sigma_{43}x_3}{r_4}\right) - f_{41}x_4y_1 - f_{42}x_4y_2 + s_4(t),\ x_4(0) > 0 \\
\frac{dy_1(t)}{dt} =& y_1\left[-\epsilon_1 - \omega_{11}y_1 - \psi_{12}y_2 + \beta_{11}f_{11}x_1 + \beta_{21}f_{21}x_2 + \beta_{31}f_{31}x_3 + \beta_{41}f_{41}x_4\right],\ y_1(0) > 0 \\
\frac{dy_2(t)}{dt} =& y_2\left[-\epsilon_2 + \phi_{12}y_1 - \omega_{22}y_2 + \beta_{12}f_{12}x_1 + \beta_{22}f_{22}x_2 + \beta_{32}f_{32}x_3 + \beta_{42}f_{42}x_4\right],\ y_2(0) > 0
\end{aligned}
$$

Dividing the fraction in the parentheses by σ_{ii} in pry equations and the carrying capacity for prey species as $K_i = \frac{r_i}{\sigma_{ii}}$ we get:

$$\begin{aligned}
\frac{dx_1(t)}{dt} =& x_1 \left[r_1 \left(1 - \frac{x_1 + \frac{\sigma_{12}}{\sigma_{11}} x_2 + \frac{\sigma_{13}}{\sigma_{11}} x_3 + \frac{\sigma_{14}}{\sigma_{11}} x_3}{K_1} \right) - f_{11} y_1 - f_{12} y_2 \right] + s_1(t),\ x_1(0) > 0 \\
\frac{dx_2(t)}{dt} =& x_2 \left[r_2 \left(1 - \frac{x_2 + \frac{\sigma_{21}}{\sigma_{22}} x_1 + \frac{\sigma_{23}}{\sigma_{22}} x_3 + \frac{\sigma_{24}}{\sigma_{22}} x_4}{K_2} \right) - f_{21} y_1 - f_{22} y_2 \right] + s_2(t),\ x_2(0) > 0 \\
\frac{dx_3(t)}{dt} =& x_3 \left[r_3 \left(1 - \frac{x_3 + \frac{\sigma_{31}}{\sigma_{33}} x_1 + \frac{\sigma_{32}}{\sigma_{33}} x_2 + \frac{\sigma_{34}}{\sigma_{33}} x_4}{K_3} \right) - f_{31} y_1 - f_{32} y_2 \right] + s_3(t),\ x_3(0) > 0 \\
\frac{dx_4(t)}{dt} =& x_4 \left[r_4 \left(1 - \frac{x_4 + \frac{\sigma_{41}}{\sigma_{44}} x_1 + \frac{\sigma_{42}}{\sigma_{44}} x_2 + \frac{\sigma_{43}}{\sigma_{44}} x_3}{K_4} \right) - f_{41} y_1 - f_{42} y_2 \right] + s_4(t),\ x_4(0) > 0 \\
\frac{dy_1(t)}{dt} =& y_1 \left[-\epsilon_1 - \omega_{11} y_1 - \psi_{12} y_2 + \beta_{11} f_{11} x_1 + \beta_{21} f_{21} x_2 + \beta_{31} f_{31} x_3 + \beta_{41} f_{41} x_4 \right],\ y_1(0) > 0 \\
\frac{dy_2(t)}{dt} =& y_2 \left[-\epsilon_2 + \phi_{12} y_1 - \omega_{22} y_2 + \beta_{12} f_{12} x_1 + \beta_{22} f_{22} x_2 + \beta_{32} f_{32} x_3 + \beta_{42} f_{42} x_4 \right],\ y_2(0) > 0
\end{aligned}$$

Let us define a common environmental carrying capacity for the prey species as $K = \eta_{ii} K_i$. This is reasonable as they prey on the same type of nutrients. Thus, we multiply the fraction in the parentheses in the prey equations by η_{ii} and define $\eta_{ik} = \frac{\sigma_{ik}}{\sigma_{ii}} \eta_{ii}$ and we obtain the following

and

$$\begin{aligned}
\frac{dx_1(t)}{dt} =& x_1 \left[r_1 \left(1 - \frac{\eta_{11} x_1 + \eta_{12} x_2 + \eta_{13} x_3 + \eta_{14} x_4}{K} \right) - f_{11} y_1 - f_{12} y_2 \right] + s_1(t),\ x_1(0) > 0 \\
\frac{dx_2(t)}{dt} =& x_2 \left[r_2 \left(1 - \frac{\eta_{22} x_2 + \eta_{21} x_1 + \eta_{23} x_3 + \eta_{24} x_4}{K} \right) - f_{21} y_1 - f_{22} y_2 \right] + s_2(t),\ x_2(0) > 0 \\
\frac{dx_3(t)}{dt} =& x_3 \left[r_3 \left(1 - \frac{\eta_{33} x_3 + \eta_{31} x_1 + \eta_{32} x_2 + \eta_{34} x_4}{K} \right) - f_{31} y_1 - f_{32} y_2 \right] + s_3(t),\ x_3(0) > 0 \\
\frac{dx_4(t)}{dt} =& x_4 \left[r_4 \left(1 - \frac{\eta_{44} x_4 + \eta_{41} x_1 + \eta_{42} x_2 + \eta_{43} x_3}{K} \right) - f_{41} y_1 - f_{42} y_2 \right] + s_4(t),\ x_4(0) > 0 \\
\frac{dy_1(t)}{dt} =& y_1 \left[-\epsilon_1 - \omega_{11} y_1 - \psi_{12} y_2 + \beta_{11} f_{11} x_1 + \beta_{21} f_{21} x_2 + \beta_{31} f_{31} x_3 + \beta_{41} f_{41} x_4 \right],\ y_1(0) > 0 \\
\frac{dy_2(t)}{dt} =& y_2 \left[-\epsilon_2 + \phi_{12} y_1 - \omega_{22} y_2 + \beta_{12} f_{12} x_1 + \beta_{22} f_{22} x_2 + \beta_{32} f_{32} x_3 + \beta_{42} f_{42} x_4 \right],\ y_2(0) > 0
\end{aligned}$$

If we consider the nutrients as an independent variable, then the system became :

$$
\begin{aligned}
\frac{dN(t)}{dt} =& R(t) - c_1x_1 - c_2x_2 - c_3x_3 - c_4x_4 \\
\frac{dx_1(t)}{dt} =& x_1 \left[r_1 \left(1 - \frac{\eta_{11}x_1 + \eta_{12}x_2 + \eta_{13}x_3 + \eta_{14}x_4}{K} \right) - f_{11}y_1 - f_{12}y_2 \right] + s_1(t),\, x_1(0) > 0 \\
\frac{dx_2(t)}{dt} =& x_2 \left[r_2 \left(1 - \frac{\eta_{22}x_2 + \eta_{21}x_1 + \eta_{23}x_3 + \eta_{24}x_4}{K} \right) - f_{21}y_1 - f_{22}y_2 \right] + s_2(t),\, x_2(0) > 0 \\
\frac{dx_3(t)}{dt} =& x_3 \left[r_3 \left(1 - \frac{\eta_{33}x_3 + \eta_{31}x_1 + \eta_{32}x_2 + \eta_{34}x_4}{K} \right) - f_{31}y_1 - f_{32}y_2 \right] + s_3(t),\, x_3(0) > 0 \\
\frac{dx_4(t)}{dt} =& x_4 \left[r_4 \left(1 - \frac{\eta_{44}x_4 + \eta_{41}x_1 + \eta_{42}x_2 + \eta_{43}x_3}{K} \right) - f_{41}y_1 - f_{42}y_2 \right] + s_4(t),\, x_4(0) > 0 \\
\frac{dy_1(t)}{dt} =& y_1 \left[-\epsilon_1 - \omega_{11}y_1 - \psi_{12}y_2 + \beta_{11}f_{11}x_1 + \beta_{21}f_{21}x_2 + \beta_{31}f_{31}x_3 + \beta_{41}f_{41}x_4 \right],\, y_1(0) > 0 \\
\frac{dy_2(t)}{dt} =& y_2 \left[-\epsilon_2 + \phi_{12}y_1 - \omega_{22}y_2 + \beta_{12}f_{12}x_1 + \beta_{22}f_{22}x_2 + \beta_{32}f_{32}x_3 + \beta_{42}f_{42}x_4 \right],\, y_2(0) > 0
\end{aligned}
$$

The carrying capacity is usually a function of several variables as nutrients, area or volume, water, and perdition, but lets us assume that the carrying capacity is just a function of the nutrients, so we can write $K = K(N) = mN$ where m is proportionality constant, furthermore, let us take m=1 then system became:

$$
\begin{aligned}
\frac{dN(t)}{dt} =& R(t) - c_1x_1 - c_2x_2 - c_3x_3 - c_4x_4 \\
\frac{dx_1(t)}{dt} =& x_1 \left[r_1 \left(1 - \frac{\eta_1x_1 + \eta_2x_2 + \eta_3x_3 + \eta_4x_4}{N(t)} \right) - f_{11}y_1 - f_{12}y_2 \right] + s_1(t),\, x_1(0) > 0 \\
\frac{dx_2(t)}{dt} =& x_2 \left[r_2 \left(1 - \frac{\eta_1x_1 + \eta_2x_2 + \eta_3x_3 + \eta_4x_4}{N(t)} \right) - f_{21}y_1 - f_{22}y_2 \right] + s_2(t),\, x_2(0) > 0 \\
\frac{dx_3(t)}{dt} =& x_3 \left[r_3 \left(1 - \frac{\eta_1x_1 + \eta_2x_2 + \eta_3x_3 + \eta_4x_4}{N(t)} \right) - f_{31}y_1 - f_{32}y_2 \right] + s_3(t),\, x_3(0) > 0 \\
\frac{dx_4(t)}{dt} =& x_4 \left[r_4 \left(1 - \frac{\eta_4x_4 + \eta_1x_1 + \eta_2x_2 + \eta_3x_3}{N(t)} \right) - f_{41}y_1 - f_{42}y_2 \right] + s_4(t),\, x_4(0) > 0 \\
\frac{dy_1(t)}{dt} =& y_1 \left[-\epsilon_1 - \omega_{11}y_1 - \psi_{12}y_2 + \beta_{11}f_{11}x_1 + \beta_{21}f_{21}x_2 + \beta_{31}f_{31}x_3 + \beta_{41}f_{41}x_4 \right],\, y_1(0) > 0 \\
\frac{dy_2(t)}{dt} =& y_2 \left[-\epsilon_2 + \phi_{12}y_1 - \omega_{22}y_2 + \beta_{12}f_{12}x_1 + \beta_{22}f_{22}x_2 + \beta_{32}f_{32}x_3 + \beta_{42}f_{42}x_4 \right],\, y_2(0) > 0
\end{aligned}
$$

A.2 Nondimensionalization

The purpose of nondimensionalization is to reduce the number of parameters in the system to ease the analysis. Additionally it is useful tool to determine the parameter combinations that dominate. Thus we need to choose scaling to keep as many terms as possible. Let us first expand the dimensional equations as follows,

$$\begin{aligned}
\frac{dx(t)}{dt} &= \left(-f_1xy_1 - f_2xy_2 + rx - \frac{rx^2}{k}\right)\\
\frac{dy_1(t)}{dt} &= \left(\beta_1f_1xy_1 - \epsilon_1y_1 - \omega_{11}y_1^2 - \psi_{12}y_1y_2\right)\\
\frac{dy_2(t)}{dt} &= \left(\beta_2f_2xy_2 - \epsilon_2y_2 - \omega_{22}y_2^2 + \phi_{12}y_1y_2\right)
\end{aligned}$$

Define the scaling

$$x = Xz_1, \quad y_1 = Y_1z_2 \quad y_2 = Y_2z_3$$

and substitute,

$$\begin{aligned}
\frac{X}{T}\frac{dz_1}{\bar{t}} &= -\frac{X^2rz_1^2}{k} - XY_1f_1z_1z_2 - XY_2f_2z_1z_3 + Xrz_1,\\
\frac{Y_1}{T}\frac{dz_2}{\bar{t}} &= XY_1\beta_1f_1z_1z_2 - Y_1^2\omega_{11}z_2^2 - Y_1Y_2\psi_{12}z_2z_3 - Y_1\epsilon_1z_2,\\
\frac{Y_2}{T}\frac{dz_3}{\bar{t}} &= XY_2\beta_2f_2z_1z_3 + Y_1Y_2\phi_{12}z_2z_3 - Y_2^2\omega_{22}z_3^2 - Y_2\epsilon_2z_3
\end{aligned}$$

Multiplying both sides of the first equation by $\frac{X}{T}$, the second one by $\frac{Y_1}{T}$ and the third one by $\frac{Y_2}{T}$, we obtain,

$$\begin{aligned}
\frac{dz_1}{d\bar{t}} &= -\frac{TXrz_1^2}{k} - TY_1f_1z_1z_2 - TY_2f_2z_1z_3 + Trz_1,\\
\frac{dz_2}{d\bar{t}} &= TX\beta_1f_1z_1z_2 - TY_1\omega_{11}z_2^2 - TY_2\psi_{12}z_2z_3 - T\epsilon_1z_2,\\
\frac{dz_3}{d\bar{t}} &= TX\beta_2f_2z_1z_3 + TY_1\phi_{12}z_2z_3 - TY_2\omega_{22}z_3^2 - T\epsilon_2z_3
\end{aligned}$$

Balance the coupled terms,

$$\begin{aligned}
&TY_1f_1z_1z_2 \sim TX\beta_1f_1z_1z_2 \Rightarrow Y_1 \sim \beta_1X\\
&TY_2f_2z_1z_3 \sim TX\beta_2f_2z_1z_3 \Rightarrow Y_2 \sim \beta_2X\\
&TY_2\psi_{12}z_2z_3 \sim TY_1\phi_{12}z_2z_3 \Rightarrow Y_2\psi_{12} \sim Y_1\phi_{12}
\end{aligned}$$

Taking $T = 1/r$ and substituting for this and for $Y_1 \sim \beta_1 X, Y_2 \sim \beta_2 X$, we obtain

$$\begin{aligned}
\frac{dz_1}{d\bar{t}} &= -\frac{X\beta_1 f_1 z_1 z_2}{r} - \frac{X\beta_2 f_2 z_1 z_3}{r} - \frac{X z_1^2}{k} + z_1, \\
\frac{dz_2}{d\bar{t}} &= \frac{X\beta_1 f_1 z_1 z_2}{r} - \frac{X\beta_1 \omega_{11} z_2^2}{r} - \frac{X\beta_2 \psi_{12} z_2 z_3}{r} - \frac{\epsilon_1 z_2}{r}, \\
\frac{dz_3}{d\bar{t}} &= \frac{X\beta_1 \phi_{12} z_2 z_3}{r} + \frac{X\beta_2 f_2 z_1 z_3}{r} - \frac{X\beta_2 \omega_{22} z_3^2}{r} - \frac{\epsilon_2 z_3}{r}
\end{aligned}$$

If we choose $X = \frac{r}{\beta_1 f_1}$ we will obtain,

$$\begin{aligned}
\frac{dz_1}{d\bar{t}} &= - z_1 z_2 + z_1 - \frac{\beta_2 f_2 z_1 z_3}{\beta_1 f_1} - \frac{r z_1^2}{\beta_1 f_1 k}, \\
\frac{dz_2}{d\bar{t}} &= -\frac{\epsilon_1 z_2}{r} + z_1 z_2 - \frac{\omega_{11} z_2^2}{f_1} - \frac{\beta_2 \psi_{12} z_2 z_3}{\beta_1 f_1}, \\
\frac{dz_3}{d\bar{t}} &= -\frac{\epsilon_2 z_3}{r} + \frac{\phi_{12} z_2 z_3}{f_1} + \frac{\beta_2 f_2 z_1 z_3}{\beta_1 f_1} - \frac{\beta_2 \omega_{22} z_3^2}{\beta_1 f_1}
\end{aligned}$$

Finally, we define new parameters as follows and substitute,

$$\begin{aligned}
a &= \frac{r}{\beta_1 f_1 k}, \quad b = \frac{\beta_2 f_2}{\beta_1 f_1}, \quad \omega_1 = \frac{\omega_{11}}{f_1}, \quad \omega_2 = \frac{\omega_{22}}{f_2}, \\
\delta_1 &= \frac{\epsilon_1}{r}, \quad \delta_2 = \frac{\epsilon_2}{r}, \quad \psi = \frac{\psi_{12}}{f_2}, \quad \phi = \frac{\phi_{12}}{f_1}
\end{aligned}$$

then, obtain our non-dimensional system 4.3. Note that here we used z_1, z_2, z_3 as non-dimensional variables instead of $\bar{x}, \bar{y_1}, \bar{y_2}$ for convenience.

A.3 Sensitivity analysis of the dimensional model

Here we will present sensitivity analysis results of the max value , mean value or equilibrium value (end value) for the reduced dimensional model 4.1.

A.3.1 Maximum value of x, y_1 and y_2 respectively

Max	x			y_1			y_1		
	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$
k	-0.0	-0.0	0.000	0.0	-0.0	0.000	0.000	0.0	0.000
r	-0.0	-0.0	0.001	-0.0	0.0	0.000	0.000	0.0	0.000
f_1	0.0	0.0	0.000	0.0	0.0	0.000	0.000	-0.0	0.000
f_2	0.0	-0.0	0.000	-0.0	-0.0	0.000	0.000	-0.0	0.000
β_1	0.0	-0.0	0.000	0.0	-0.0	0.000	0.000	-0.0	0.000
β_2	0.0	0.0	0.001	0.0	-0.0	0.000	0.000	-0.0	0.001
ϵ_1	-0.0	0.0	0.001	0.0	0.0	0.001	0.000	-0.0	0.000
ϵ_2	-0.0	-0.0	0.000	0.0	-0.0	0.000	-0.000	0.0	0.001
ψ_{12}	-0.0	0.0	0.001	-0.0	0.0	0.001	0.000	0.0	0.000
ϕ_{12}	-0.0	-0.0	0.001	0.0	0.0	0.001	-0.000	-0.0	0.001
ω_{11}	-0.0	-0.0	0.000	-0.0	-0.0	0.000	-0.000	0.0	0.001
ω_{22}	-0.0	0.0	0.000	-0.0	0.0	0.001	-0.000	-0.0	0.000
x_0	1.0	1.0	1.000	0.0	-0.0	0.000	-0.000	0.0	0.000
y_{1_0}	0.0	-0.0	0.000	1.0	1.0	1.000	-0.000	-0.0	0.000
y_{2_0}	0.0	0.0	0.001	0.0	0.0	0.000	0.999	1.0	1.000
$\sum$	1.0	1.0	1.006	1.0	1.0	1.004	0.999	1.0	1.004

Table A.1: Main Effect sensitivity index for all methods for Max value of all variables

Max	x			y_1			y_1		
	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}
k	0.0	153.002	0.019	0.0	5.101	0.020	0.00000	8.210	0.018
r	0.0	85.951	0.021	0.0	3.942	0.020	0.00000	6.391	0.019
f_1	0.0	142.531	0.020	0.0	11.081	0.020	0.00000	19.556	0.021
f_2	0.0	166.142	0.021	0.0	12.866	0.020	0.00000	16.332	0.020
β_1	0.0	79.185	0.021	0.0	13.206	0.019	0.00000	10.979	0.021
β_2	0.0	82.828	0.020	0.0	7.797	0.020	0.00000	14.493	0.022
ϵ_1	0.0	16.437	0.019	0.0	3.903	0.020	0.00000	8.340	0.020
ϵ_2	0.0	26.688	0.020	0.0	1.930	0.019	0.00000	5.878	0.020
ψ_{12}	0.0	32.098	0.019	0.0	16.717	0.021	0.00000	50.652	0.020
ϕ_{12}	0.0	11.893	0.020	0.0	14.028	0.021	0.00000	32.618	0.020
ω_{11}	0.0	18.891	0.019	0.0	22.963	0.019	0.00000	42.847	0.021
ω_{22}	0.0	67.723	0.019	0.0	7.303	0.021	0.00000	52.394	0.020
x_0	1.0	3831.494	0.958	0.0	10.348	0.019	0.00000	11.586	0.019
y_{10}	0.0	8.415	0.020	1.0	3978.292	0.958	0.00000	11.561	0.020
y_{20}	0.0	8.576	0.021	0.0	5.285	0.019	0.99885	3923.924	0.958
$\sum$	1.0	4731.854	1.237	1.0	4114.762	1.236	0.99885	4215.761	1.239

Table A.2: Total or Interaction Effect sensitivity index for all methods for Max value of all variables

A.3.2 Mean value of x, y_1 and y_2 respectively

Mean	x			y_1			y_1		
	S_{1Sobol}	$S_{1RbdFast}$	S_{1DMIM}	S_{1Sobol}	$S_{1RbdFast}$	S_{1DMIM}	S_{1Sobol}	$S_{1RbdFast}$	S_{1DMIM}
k	0.000	0.000	0.001	0.000	-0.000	0.000	0.000	0.000	0.001
r	0.001	0.001	0.002	0.052	0.050	0.050	0.068	0.071	0.072
f_1	0.025	0.013	0.014	0.003	0.001	0.001	0.082	0.046	0.047
f_2	0.017	0.028	0.035	0.166	0.143	0.148	0.083	0.093	0.095
β_1	0.002	0.006	0.007	0.026	0.031	0.031	0.012	0.015	0.015
β_2	0.003	0.002	0.002	0.053	0.051	0.053	0.006	0.009	0.010
ϵ_1	0.000	0.000	0.001	0.003	0.004	0.004	0.001	0.001	0.001
ϵ_2	0.000	-0.000	0.000	0.010	0.010	0.010	0.003	0.004	0.004
ψ_{12}	0.000	0.000	0.001	0.001	0.004	0.004	0.002	0.001	0.002
ϕ_{12}	-0.000	-0.000	0.000	0.006	0.005	0.005	-0.001	0.003	0.003
ω_{11}	-0.000	0.000	0.001	0.000	0.001	0.001	0.000	0.000	0.001
ω_{22}	0.001	0.000	0.001	0.015	0.012	0.012	0.004	0.011	0.011
x_0	0.000	-0.000	0.000	-0.000	0.000	0.000	-0.000	0.000	0.001
y_{10}	-0.000	-0.000	0.000	0.036	0.033	0.033	-0.000	-0.000	0.001
y_{20}	0.000	-0.000	0.001	-0.000	-0.000	0.000	0.003	0.004	0.004
$\sum$	0.049	0.050	0.066	0.371	0.345	0.352	0.264	0.258	0.268

Table A.3: Main Effect sensitivity index for all methods for Mean value of all variables

Mean		x			y_1			y_1	
	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}
k	0.092	226.378	0.296	0.003	1.919	0.039	0.01506	10.467	0.017
r	0.164	137.053	0.606	0.252	2.875	0.146	0.26760	13.309	0.170
f_1	0.803	283.822	0.678	0.400	4.717	0.103	0.63442	13.467	0.102
f_2	1.195	354.369	0.434	0.634	3.748	0.126	0.73424	21.705	0.171
β_1	0.403	112.307	0.507	0.163	2.291	0.093	0.23445	7.628	0.053
β_2	0.117	93.772	0.400	0.214	1.785	0.096	0.07263	8.367	0.063
ϵ_1	0.001	13.600	0.360	0.019	0.524	0.053	0.00396	0.454	0.028
ϵ_2	0.001	18.195	0.314	0.039	0.549	0.066	0.02420	1.859	0.049
ψ_{12}	0.107	59.678	0.365	0.049	2.307	0.053	0.05501	4.118	0.022
ϕ_{12}	0.001	16.425	0.329	0.050	2.419	0.049	0.02822	3.107	0.036
ω_{11}	0.004	29.690	0.367	0.013	3.697	0.042	0.00313	1.829	0.020
ω_{22}	0.068	97.751	0.409	0.108	1.625	0.063	0.24549	18.464	0.037
x_0	0.000	6.438	0.342	0.000	0.018	0.040	0.00002	0.054	0.016
y_{1_0}	0.000	2.855	0.363	0.036	0.517	0.164	0.00001	0.091	0.018
y_{2_0}	0.000	3.025	0.370	0.000	0.014	0.039	0.00366	0.681	0.059
$\sum$	2.956	1455.358	6.140	1.980	29.005	1.172	2.32210	105.600	0.861

Table A.4: Total or Interaction Effect sensitivity index for all methods for Mean value of all variables

A.3.3 Median value of x, y_1 and y_2 respectively

Median	x			y_1			y_1		
	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$
k	0.000	0.000	0.001	-0.000	-0.000	0.000	0.000	0.000	0.001
r	0.001	0.001	0.002	0.052	0.049	0.050	0.068	0.071	0.072
f_1	0.025	0.013	0.014	0.001	0.001	0.001	0.083	0.047	0.047
f_2	0.017	0.028	0.035	0.173	0.149	0.155	0.084	0.092	0.095
β_1	0.002	0.006	0.007	0.027	0.030	0.031	0.013	0.015	0.016
β_2	0.003	0.002	0.002	0.050	0.049	0.050	0.006	0.010	0.010
ϵ_1	0.000	0.000	0.000	0.003	0.004	0.004	0.001	0.001	0.001
ϵ_2	0.000	-0.000	0.000	0.010	0.010	0.011	0.003	0.004	0.004
ψ_{12}	0.000	0.000	0.001	-0.000	0.004	0.004	0.002	0.002	0.002
ϕ_{12}	-0.000	-0.000	0.000	0.008	0.006	0.006	-0.001	0.003	0.003
ω_{11}	-0.000	0.000	0.001	-0.000	0.001	0.001	0.000	0.000	0.001
ω_{22}	0.001	0.000	0.001	0.016	0.013	0.014	0.003	0.010	0.010
x_0	0.000	-0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.001
y_{1_0}	-0.000	-0.000	0.000	-0.000	-0.000	0.000	-0.000	-0.000	0.001
y_{2_0}	-0.000	-0.000	0.001	0.000	0.000	0.000	0.000	-0.000	0.000
$\sum$	0.049	0.050	0.065	0.340	0.316	0.327	0.263	0.255	0.264

Table A.5: Main Effect sensitivity index for all methods for Median value of all variables

Median	x			y_1			y_1		
	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}	$S_{T\,Sobol}$	$\mu^{\star}{}_{Morris}$	δ_{DMIM}
k	0.092	226.049	0.310	0.003	1.927	0.171	0.01511	10.452	0.090
r	0.164	137.575	0.843	0.262	2.865	0.388	0.26870	13.298	0.551
f_1	0.803	282.266	1.544	0.416	4.714	0.196	0.63751	13.447	0.142
f_2	1.194	353.143	3.028	0.662	3.743	0.217	0.73702	21.721	0.234
β_1	0.403	112.115	0.660	0.168	2.266	0.234	0.23637	7.647	0.098
β_2	0.117	93.430	0.449	0.219	1.755	0.190	0.07428	8.364	0.124
ϵ_1	0.001	13.226	0.390	0.021	0.509	0.185	0.00427	0.436	0.093
ϵ_2	0.001	17.800	0.330	0.043	0.539	0.192	0.02449	1.811	0.101
ψ_{12}	0.107	59.402	0.394	0.051	2.262	0.182	0.05621	4.061	0.092
ϕ_{12}	0.001	15.765	0.355	0.059	2.394	0.179	0.02833	3.027	0.094
ω_{11}	0.004	29.060	0.409	0.017	3.682	0.175	0.00377	1.783	0.092
ω_{22}	0.068	97.381	0.448	0.115	1.601	0.181	0.24562	18.371	0.086
x_0	0.000	4.486	0.380	0.000	0.006	0.170	0.00001	0.016	0.091
y_{1_0}	0.000	2.078	0.404	0.000	0.005	0.170	0.00000	0.072	0.090
y_{2_0}	0.000	2.526	0.417	0.000	0.006	0.171	0.00000	0.176	0.092
$\sum$	2.955	1446.302	10.361	2.036	28.274	3.001	2.33169	104.682	2.070

Table A.6: Total or Interaction Effect sensitivity index for all methods for Median value of all variables

A.3.4 Eqm. value of $x\,y_1$ and y_2 respectively

Eqm.	x			y_1			y_1		
	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$	$S_{1_{Sobol}}$	$S_{1_{RbdFast}}$	$S_{1_{DMIM}}$
k	0.000	0.000	0.001	0.000	-0.000	0.000	0.000	0.000	0.001
r	0.001	0.001	0.002	0.050	0.048	0.049	0.069	0.070	0.071
f_1	0.025	0.013	0.014	0.007	0.001	0.001	0.084	0.046	0.046
f_2	0.017	0.028	0.035	0.165	0.138	0.144	0.082	0.091	0.093
β_1	0.002	0.006	0.007	0.028	0.030	0.030	0.012	0.015	0.015
β_2	0.003	0.002	0.002	0.059	0.052	0.053	0.005	0.009	0.009
ϵ_1	0.000	0.000	0.001	0.004	0.004	0.004	0.001	0.001	0.002
ϵ_2	0.000	-0.000	0.000	0.011	0.010	0.010	0.003	0.004	0.004
ψ_{12}	0.000	0.000	0.001	0.004	0.004	0.004	0.002	0.001	0.002
ϕ_{12}	-0.000	-0.000	0.000	0.006	0.005	0.005	0.001	0.003	0.003
ω_{11}	0.000	0.000	0.001	0.001	0.001	0.002	-0.001	0.000	0.001
ω_{22}	0.001	0.000	0.001	0.014	0.011	0.011	0.002	0.011	0.011
x_0	-0.000	-0.000	0.000	-0.001	0.000	0.000	-0.001	0.000	0.001
y_{1_0}	0.000	-0.000	0.000	0.000	-0.000	0.000	-0.001	-0.000	0.001
y_{2_0}	-0.000	-0.000	0.001	-0.000	0.000	0.000	-0.001	-0.000	0.000
$\sum$	0.049	0.050	0.066	0.349	0.304	0.313	0.261	0.251	0.260

Table A.7: Main Effect sensitivity index for all methods for Eqm. value of all variables

Eqm.	x			y_1			y_1		
	$S_{T Sobol}$	${\mu^\star}_{Morris}$	δ_{DMIM}	$S_{T Sobol}$	${\mu^\star}_{Morris}$	δ_{DMIM}	$S_{T Sobol}$	${\mu^\star}_{Morris}$	δ_{DMIM}
k	0.095	229.348	0.308	0.031	1.975	0.167	0.02363	10.488	0.090
r	0.165	140.435	0.754	0.288	2.912	0.360	0.27522	13.350	0.525
f_1	0.805	286.532	1.539	0.462	4.746	0.197	0.63829	13.460	0.141
f_2	1.194	357.210	2.870	0.678	3.807	0.181	0.73400	21.792	0.221
β_1	0.403	115.783	0.653	0.196	2.309	0.231	0.24018	7.706	0.097
β_2	0.117	96.778	0.454	0.287	1.800	0.184	0.08377	8.582	0.119
ϵ_1	0.001	16.401	0.393	0.047	0.551	0.181	0.00903	0.529	0.092
ϵ_2	0.002	21.374	0.331	0.097	0.584	0.188	0.03002	1.888	0.100
ψ_{12}	0.108	61.939	0.387	0.078	2.302	0.178	0.06388	4.188	0.091
ϕ_{12}	0.001	19.228	0.347	0.125	2.443	0.175	0.04326	3.097	0.093
ω_{11}	0.004	31.877	0.414	0.050	3.711	0.169	0.00894	1.820	0.091
ω_{22}	0.068	100.106	0.439	0.163	1.660	0.177	0.24909	18.453	0.085
x_0	0.001	6.516	0.380	0.062	0.054	0.165	0.00538	0.045	0.090
y_{1_0}	0.000	5.296	0.397	0.035	0.045	0.166	0.00417	0.135	0.090
y_{2_0}	0.003	5.374	0.417	0.018	0.060	0.167	0.00209	0.269	0.092
$\sum$	2.967	1494.197	10.083	2.617	28.959	2.886	2.41095	105.802	2.017

Table A.8: Total or Interaction Effect sensitivity index for all methods for Eqm. value of all variables

	δ_{DMIM}	S_{1DMIM}	δ_{DMIM}	S_{1DMIM}
k	0.058	0.001	0.097	0.001
r	0.132	0.044	0.140	0.050
f_1	0.131	0.011	0.127	0.006
f_2	0.154	0.022	0.185	0.016
β_1	0.126	0.007	0.121	0.004
β_2	0.118	0.005	0.124	0.003
ϵ_1	0.082	0.003	0.114	0.002
ϵ_2	0.075	0.003	0.111	0.003
ψ_{12}	0.058	0.002	0.103	0.002
ϕ_{12}	0.056	0.000	0.103	0.000
ω_{11}	0.063	0.002	0.106	0.002
ω_{22}	0.073	0.027	0.091	0.046
x_0	0.056	0.001	0.101	0.001
y_{10}	0.054	0.001	0.102	0.001
y_{20}	0.055	0.000	0.100	0.000
$\sum$	1.291	0.129	1.725	0.137

Table A.9: Sensitivity of median and eqm values of y_1 after log transformation with DMIM method

A.3.5 Summary

S_1,Variable of interest	Max	Mean	Median	Eqm
x	x_0	f_1, f_2	f_1, f_2	f_1, f_2
y_1	y_{10}	f_2	f_2	f_2
y_2	y_{20}	f_2	f_2	f_1, f_2
S_T,Variable of interest	Max	Mean	Median	Eqm
x	x_0	f_2, f_1	f_1, f_2	f_2, f_1, r
y_1	y_{10}	f_2, f_1, y_{10}	f_2	f_2
y_2	y_{20}	f_2	f_2	f_2, r

Table A.10: The most sensitive parameter

Appendix B

Python codes used in the thesis

Here we present python codes used in this work. All codes are in Jupyter notebook form. Furthermore, the codes are not compiled but they are functional.

B.1 Analysis of the model

Her we present the python code formed via Jupyter notebook used in the stability analysis of the model 4.3.

Importing packages

In [1]:

```python
from sympy.interactive import printing
printing.init_printing(use_latex='mathjax')
from sympy import *
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
```

Defining the symbols

In [2]:

```python
x,y1,y2=sp.symbols('x,y1,y2',real=True,nonnegative=True)
a,b,omega1,omega2,s=sp.symbols('a,b,omega1,omega2,s',real=True,positive=True)
delta1,delta2,psi,phi=sp.symbols('delta1,delta2,psi,phi',real=True,positive=True)
x_d,y1_d,y2_d=sp.symbols('x_d,y1_d,y2_d')



U0,U1,U2,U3,U4,U5,U6,U7,U8,U9=sp.symbols('U0,U1,U2,U3,U4,U5,U6,U7,U8,U9')
mu0,mu1,mu2,mu3,mu4,m5,mu6,mu7,mu8,mu9=sp.symbols('mu0,mu1,mu2,mu3,mu4,m5,mu6,mu7,mu8,mu9')
E0,E1,E2,E3=sp.symbols('E0,E1,E2,E3')
W0,W1,W2,W3,W4,W5,W6,W7,W8,W9=sp.symbols('W0,W1,W2,W3,W4,W5,W6,W7,W8,W9')

L0,L1,L2,L3,L4=sp.symbols('L0,L1,L2,L3,L4')
A1,A2,A3,A4=sp.symbols('A1,A2,A3,A4')


k,r,f1,f2,s = sp.symbols('k,r,f1,f2,s', positive=True)
epsilon1,epsilon2,beta1,beta2= sp.symbols('epsilon1,epsilon2,beta1,beta2', positive=True)
omega11,omega22,phi12,psi12 = sp.symbols('omega11,omega22,phi12,psi12', positive=True)
```

In [3]:

```python
dxdt=x*(1-a*x-y1-b*y2)
dy1dt=y1*(-delta1-omega1*y1-psi*b*y2+x)
dy2dt=y2*(-delta2-omega2*b*y2+phi*y1+b*x)
F=Matrix([[dxdt,dy1dt,dy2dt]])
```

Check if the nondimensionalization is correct

In [ ]:

```
pm=[
    (x, (beta1*f1*x_d)/r), (y1,(f1*y1_d)/r),  (y2,(beta1*f1*y2_d)/(r*beta
2)),

    (a,r/(beta1*f1*k)), (b,(beta2*f2)/(beta1*f1)), (delta1, epsilon1/r),

    (delta2, epsilon2/r), (psi, psi12/f2),(phi, phi12/f1),

    (omega1, omega11/f1), (omega2,(omega22)/f2)]

F1dim=(simplify(F[0].subs(pm))*(r**2/(beta1*f1))).expand()
F2dim=(simplify(F[1].subs(pm))*(f1/r**2)**-1).expand()
F3dim=(simplify(F[2].subs(pm))*(beta2*r**2/(beta1*f1))).expand()
```

In [ ]:

```
[F1dim  ,F2dim  ,F3dim ]
```

Equilibrium points (EP)

The system has 5 nonnegative EPs

In [ ]:

```
F_1=(1-a*x-y1-b*y2)
F_2=(-delta1-omega1*y1-psi*b*y2+x)
F_3=(-delta2-omega2*b*y2+phi*y1+b*x)

equ = ((x, F_1), (y1, F_2), (y2, F_3))

from itertools import product
for eqs in product(*equ):
    sol = solve(eqs, [x, y1, y2], simplify=True)
    #pprint(sol)
    #print(sol)
    print('ep=',sol)
```

In [5]:

```
ep1={y2: 0, y1: 0, x: 0}
EP1=Matrix(list(ep1.values()))
ep2={x: 1/a, y1: 0, y2: 0}
EP2=Matrix(list(ep2.values()))
ep3={x: (delta1 + omega1)/(a*omega1 + 1), y1: (-a*delta1 + 1)/(a*omega1 +
1), y2: 0}
EP3=Matrix(list(ep3.values()))

ep4={x: (delta2 + omega2)/(a*omega2 + b), y1: 0, y2: (-a*delta2 + b)/(b*(
a*omega2 + b))}
EP4=Matrix(list(ep4.values()))

ep5={x: (delta1*(omega2 + phi) + delta2*(omega1 - psi) + omega1*omega2 +
phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     y1: (-b*psi - delta1*(a*omega2 + b) + delta2*(a*psi + 1) + omega2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     y2: (b*omega1 - delta1*(a*phi - b) - delta2*(a*omega1 + 1) + phi)/
     (b*(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi))}
EP5=Matrix(list(ep5.values()))

ep=[ep1,ep2,ep3,ep4,ep5]
EP_all=[EP1,EP2,EP3,EP4,EP5]
```

Simplifying the interiopr equilibrium point EP_5

In [6]:

```
E0s=(delta1*(omega2 + phi) + delta2*(omega1 - psi) + omega1*omega2 + phi*
psi)
E1s=(-b*psi - delta1*(a*omega2 + b) + delta2*(a*psi + 1) + omega2)
E2s=(b*omega1 - delta1*(a*phi - b) - delta2*(a*omega1 + 1) + phi)
U0s=(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)
EP5p=simplify(EP5.subs([(U0s,U0),(E0s,E0),(E1s,E1),(E2s,E2)]))
```

In [ ]:

```
EP5p
```

Check f the solution satisfys the equations

In [ ]:

```
for i in ep:
    print(checksol(equs,i))
```

In [ ]:

```
for i in equs:
    print(simplify(i.subs([(x,EP5[0]),(y1,EP5[1]),(y2,EP5[2])])))
```

Jacobian matrix and correspounding eigenvalues at each EP

Jacobian matrix of the subsystem

In [ ]:

```
X=Matrix([x,y1,y2])
J=F.jacobian(X)
tr_j=simplify(Trace(J)) # Trace of the jacobian matrix
```

Evaluating Jacobiab matrix at each eqm. point

First equilibrium point

In [ ]:

```
J1=J.subs([(x,EP1[0]),(y1,EP1[1]),(y2,EP1[2])])
EV1=J1.eigenvals()
lambda1=Matrix(list(EV1.keys()))
V1=J1s.eigenvects()
```

In [ ]:

```
lambda1
```

In [ ]:

```
V1
```

Second equilibrium point

In [ ]:

```
J2=J.subs([(x,EP2[0]),(y1,EP2[1]),(y2,EP2[2])])
EV2=J2.eigenvals()
lambda2=Matrix(list(EV2.keys()))
Vec2=J2.eigenvects()
```

In [ ]:

```
EV2
```

In [ ]:

```
lambda2
```

In [ ]:

```
Vec2
```

Third equilibrium point

In [ ]:

```
J3=J.subs([(x,EP3[0]),(y1,EP3[1]),(y2,EP3[2])])
J3s=simplify(J3)
EV3=J3s.eigenvals()
ev3=Matrix(list(EV3.keys()))
```

In [ ]:

```
ev3
```

In [ ]:

```
mu0s=simplify((a**2*delta1**2*omega1**2 + 2*a**2*delta1**2*omega1 + a**2*
delta1**2 + \
               2*a**2*delta1*omega1**2 + 2*a**2*delta1*omega1 + a**2*omeg
a1**2 + 4*a*delta1**2 \
               - 2*a*delta1*omega1**2 + 2*a*delta1*omega1 - 2*a*omega1**2
- 4*delta1 + omega1**2 - 4*omega1))

mu0ss=a*delta1*omega1*(a*(delta1*(omega1+2) +2*(omega1+1))+\
      2*(1-omega1))+a*delta1**2*(a+4)+(omega1**2)*(a-1)**2 -4*(delta1+ome
ga1)
```

In [ ]:

```
ev3p=ev3.subs([(E2s.expand(),E2),(mu0s,mu0)])
```

Fourth equilibrium point

In [ ]:

```
J4=J.subs([(x,EP4[0]),(y1,EP4[1]),(y2,EP4[2])])
J4s=simplify(J4)
EV4=J4s.eigenvals()
ev4=Matrix(list(EV4.keys()))
ev4s=simplify(ev4)
```

In [ ]:

```
ev4s
```

In [ ]:

```
mu1s=simplify((a**2*delta2**2*omega2**2 + 2*a**2*delta2**2*omega2+ a**2*d
elta2**2 + 2*a**2*delta2*omega2**2 \
               + 2*a**2*delta2*omega2 + a**2*omega2**2 + 4*a*b*delta2**2-
2*a*b*delta2*omega2**2 \
               + 2*a*b*delta2*omega2 - 2*a*b*omega2**2 - 4*b**2*delta2 +
b**2*omega2**2 - 4*b**2*omega2))
```

In [ ]:

```
ev4p=ev4s.subs([(mu1s,mu1),(E1s.expand(),E1)])
```

Fifth equilibrium point

In [ ]:

```
J5=J.subs([(x,EP5[0]),(y1,EP5[1]),(y2,EP5[2])])
J5s=simplify(J5)
J5ss=simplify(J5s.expand())
```

In [ ]:

```
U1s=(-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1*om
ega2 - phi*psi)
U2s=(a*delta1*omega2 - a*delta2*psi + b*delta1 + b*psi - delta2 - omega2)
U3s=(a*delta1*phi + a*delta2*omega1 - b*delta1 - b*omega1 + delta2 - phi)
```

In [ ]:

```
J5p=J5ss.subs([(U0s,U0),(U1s,U1),(U2s,U2),(U3s,U3),])
```

In [ ]:

```
J5p
```

In [ ]:

```
EV5p=J5p.eigenvals()
ev5p=Matrix(list(EV5p.keys()))
```

In [ ]:

```
A1s=simplify((-U1*a - U2*omega1 - U3*omega2))

A2s=simplify((U1*U2*a*omega1 + U1*U2 + U1*U3*a*omega2 + U1*U3*b + U2*U3*o
mega1*omega2 + U2*U3*phi*psi))

A3s= simplify((-U1*U2*U3*a*omega1*omega2 - U1*U2*U3*a*phi*psi - U1*U2*U3*
b*omega1\
               + U1*U2*U3*b*psi - U1*U2*U3*omega2 - U1*U2*U3*phi))
```

In [ ]:

```
ev5pp=ev5p.subs([(A1s,A1),(A2s,A2),(A3s.expand(),A3),])
```

In [ ]:

```
ev5pp
```

In [ ]:

```
mu3s=(A1**3/U0**3 - 9*A1*A2/(2*U0**3) + 27*A3/(2*U0**3) + sqrt(-4*(A1**2/
U0**2 - 3*A2/U0**2)**3\
     + (2*A1**3/U0**3 - 9*A1*A2/U0**3 + 27*A3/U0**3)**2)/2)
```

In [ ]:

```
simplify(mu3s)
```

In [ ]:

```
ev5ps=ev5pp.subs([(mu3s,mu3)])
```

In [ ]:

```
ev5ps
```

GLobal stability via Lyapunov function

Global stability of EP_5

In [ ]:

```
p1,p2,p3=sp.symbols('p1,p2,p3')
x_e,y1_e,y2_e=sp.symbols('x_e,y1_e,y2_e')

dXdt=Matrix([dxdt,dy1dt,dy2dt]) #system
A=Matrix([[a,1,b],[-1,omega1,psi*b],[-b,-phi,omega2*b]])
r=Matrix([1,-delta1,-delta2])
X_bar=diag(x,y1,y2)

#Lyapunov function
V=p1*(x-x_e-x_e*ln(x/x_e))+p2*(y1-y1_e-y1_e*ln(y1/y1_e))+p3*(y2-y2_e-y2_e
*ln(y2/y2_e))
D=Matrix([[p1,0,0],[0,p2,0],[0,0,p3]])
M=simplify(D*A+A.transpose()*D)
X1=Matrix([[x-x_e],[y1-y1_e],[y2-y2_e]])
FF=-(1/2)*X1.transpose()*(M*X1)
```

Derivative of the lyapunov function

In [ ]:

```
dVdt=(diff(V,x))*F[0]+(diff(V,y1))*F[1]+(diff(V,y2))*F[2]
```

In [ ]:

```
FF.expand()
```

In [ ]:

```
dVdt.expand()
```

In [ ]:

```
dVdt.expand()==FF.expand() #They will be equal after manipualtion
```

Find charachteristic polynomial

not helpful in this case

In [ ]:

```
M.charpoly()
```

Find leading principal minors

In [ ]:

```
Ms=simplify(M.subs([(p1,1),(p2,1),(p3,1)]))
det_Ms=simplify(sp.det(Ms))
M1=M.minor_submatrix(2, 2)
M1s=simplify(M1.subs([(p1,1),(p2,1),(p3,1)]))
det_M1=simplify(det(M1s))
```

In [ ]:

```
det_Ms
```

In [ ]:

```
det_M1
```

So EP_5 is globally stable if $2a\left(-b^2\psi^2+4b\omega_1\omega_2+2b\phi\psi-\phi^2\right)>0$

In [ ]:

```
%%latex
$EP_{2}$ is globally stable if it is locallaly stable
```

Global stability of EP2

In [ ]:

```
H1=1/x
dH1F1dx=diff(H1*F1,x).subs([(y1,0),(y2,0)])
```

In [ ]:

```
dH1F1dx
```

Global stability of EP3

In [ ]:

```
H2=1/(x*y1)
dH2F1dx=diff(H2*F1,x).subs([(y2,0)])
dH2F2dy1=diff(H2*F2,y1).subs([(y2,0)])
```

In [ ]:

```
BD2=dH2F1dx+dH2F2dy1
```

In [ ]:

```
BD2
```

DB2 <0 for all values of x, y so there is no limit cycle in interior of xy1-plane i.e EP3 is globally stable there.

Global stability of EP4

In [ ]:

```
H3=1/(x*y2)
dH3F1dx=diff(H3*F1,x).subs([(y1,0)])
dH3F3dy2=diff(H3*F3,y2).subs([(y1,0)])
```

In [ ]:

```
BD3=dH3F1dx+dH3F3dy2
```

In [ ]:

```
BD3
```

$BD_3 < 0$ for all values of x, y so there is no limit cycle in interior of xy1-plane i.e EP3 is globally stable there.

B.2 Sensitivity analysis

Her we present the python code formed via Jupyter notebook used in the sensitivity analysis section.

Import packages

In [ ]:

```
import numpy as np
from numpy import *
from cmath import*
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import scipy as scp
from scipy.integrate import solve_ivp, odeint
import pandas as pd
import SALib
from SALib.sample import saltelli,morris,latin,ff,finite_diff
from SALib.sample import morris
from SALib.analyze import sobol,dgsm,rbd_fast,delta,ff,dgsm
from SALib.analyze import morris
from SALib.plotting.bar import plot as barplot
from SALib.plotting.morris import *
```

Define the model inputs

We choose the intervals for a, b, δ_1, δ_2 such that we are in the regime
$\delta_1 < 1/a, \quad \delta_2 < b/a$

In [ ]:

```
problem = {
    'num_vars': 11,
    'names': ['$a$','$b$','$\\delta_{1}$', '$\\delta_{2}$','$\psi$','$\ph
i$',
              '$\\omega_{1}$','$\\omega_{2}$','$x_{0}$','${y_{1}}_{0}$',
'${y_{2}}_{0}$'],
    'bounds': [  [1e-05,1e-03]
               , [0.1,10]
               , [0.1,10]
               , [0.1,10]
               , [0.1,10]
               , [0.0001,5]
               , [0.001,10]
               , [0.001,5]
                ,[0,10000],
                 [0,1000],
                 [0,100]
                          ]
}
```

Model ,functions and solvers

In [ ]:

```
def DM_bacteria(t,X,pm):
    x, y1, y2 = X
    a, b,delta1,delta2,psi,phi,omega1,omega2=pm
    dxdt=x*(1-a*x-y1-b*y2)
    dy1dt=y1*(-delta1-omega1*y1-psi*b*y2+x)
    dy2dt=y2*(-delta2-omega2*b*y2+phi*y1+b*x)
    return [dxdt,dy1dt,dy2dt]
def DM_jac(t,X,pm):
    x, y1,y2 = X
    a, b,delta1,delta2,psi,phi,omega1,omega2=pm
    J=np.array([[-2*a*x - b*y2 - y1 + 1, -x, -b*x],
                [y1, -b*psi*y2 - delta1 - 2*omega1*y1 + x, -b*psi*y1],
                [b*y2, phi*y2, -2*b*omega2*y2 + b*x - delta2 + phi*y1]])
    return J
```

In [ ]:

```
#Solver that can be used
SOLVERS=['RK45','RK23','DOP853','BDF','Radau','LSODA']
def run_model(param_values,tMax,solvers,RelTol,AbsTol):
    loop_n = param_values.shape[0]

    x_res = np.empty((loop_n, 4))
    y1_res = np.empty((loop_n, 4))
    y2_res = np.empty((loop_n, 4))

    t = [0, tMax]
    t_eval_pts = np.linspace(0, tMax, 25*tMax)
    for i in range(loop_n):
        a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20,*_ = param_va
lues[i]
        X0 = [x0, y10, y20]
        pm=np.array ([ a, b,delta1,delta2,psi,phi,omega1,omega2])
        #hmax=0.01
        #,max_step=0.01

        sol=solve_ivp(DM_bacteria,t,X0,method= solvers,t_eval= t_eval_pts
,args=(pm,),jac=DM_jac,rtol=RelTol,atol=AbsTol);

        x, y1, y2 = sol.y
        x_res[i, :] = amax(x), mean(x), median(x), x[-1]
        y1_res[i, :] = amax(y1), mean(y1), median(y1), y1[-1]
        y2_res[i, :] = amax(y2), mean(y2), median(y2), y2[-1]
    return x_res, y1_res, y2_res
```

function to calculate max values from boundness for each set of parameters

In [ ]:

```
 def MaxBound(param_values):
        a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = param_value
s
        X0 = [x0, y10, y20]
        x_m=np.amax(np.array([x0,1/a]))
        y1_m=np.amax(np.array([y10,(x_m-delta1)/omega1]))
        y2_m=np.amax(np.array([y20,(b*x_m+phi*y1_m-delta2)/(b*omega2)]))
        MaxBound=np.array([x_m,y1_m,y2_m])
        return MaxBound
```

Function to find eqm. points and check their stability for a given set of parameters

In [ ]:

```
def EqmValues(param_values):
    a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = param_values
    EP1=np.array([0,0,0])

    EP2=np.array([1/a, 0, 0])

    EP3=np.array([(delta1 + omega1)/(a*omega1 + 1), (-a*delta1 + 1)/(a*om
ega1 + 1), 0])

    EP4=np.array([(delta2 + omega2)/(a*omega2 + b), 0, (-a*delta2 + b)/(b
*(a*omega2 + b))])

    EP5=np.array([(delta1*(omega2 + phi) + delta2*(omega1 - psi) + omega1
*omega2 + phi*psi)/
                  (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega
2 + phi),

                  (-b*psi - delta1*(a*omega2 + b) + delta2*(a*psi + 1) +
omega2)/
                  (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega
2 + phi),

                  (b*omega1 - delta1*(a*phi - b) - delta2*(a*omega1 + 1)
+ phi)/
                  (b*(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + om
ega2 + phi))])
    EP=np.stack(np.array([EP1,EP2,EP3,EP4,EP5]))


    J1=np.array([[1, 0, 0], [0, -delta1, 0], [0, 0, -delta2]])

    J2=np.array([[-1, -1/a, -b/a], [0, -delta1 + 1/a, 0], [0, 0, -delta2
+ b/a]])

    J3=np.array([
    [-a*(delta1 + omega1)/(a*omega1 + 1), -(delta1 + omega1)/(a*omega1 +
1),-b*(delta1 + omega1)/(a*omega1 + 1)],
    [(-a*delta1 + 1)/(a*omega1 + 1),omega1*(a*delta1 - 1)/(a*omega1 + 1),
b*psi*(a*delta1 - 1)/(a*omega1 + 1)],
    [0, 0, (b*(delta1 + omega1) - delta2*(a*omega1 + 1) - phi*(a*delta1 -
1))/(a*omega1 + 1)]])

    J4=np.array([
    [-a*(delta2 + omega2)/(a*omega2 + b), -(delta2 + omega2)/(a*omega2 +
b),-b*(delta2 + omega2)/(a*omega2 + b)],
    [0, (-delta1*(a*omega2 + b) + delta2 + omega2 + psi*(a*delta2 - b))/(
a*omega2 + b), 0],
    [(-a*delta2 + b)/(a*omega2 + b), -phi*(a*delta2 - b)/(b*(a*omega2 + b
)), omega2*(a*delta2 - b)/(a*omega2 + b)]])

    J5=np.array([
    [a*(-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1
```

```
*omega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     (-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1*o
mega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     b*(-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1
*omega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)],


    [(-a*delta1*omega2 + a*delta2*psi - b*delta1 - b*psi + delta2 + omega
2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     omega1*(a*delta1*omega2 - a*delta2*psi + b*delta1 + b*psi - delta2 -
omega2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     b*psi*(a*delta1*omega2 - a*delta2*psi + b*delta1 + b*psi - delta2 -
omega2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)],


    [(-a*delta1*phi - a*delta2*omega1 + b*delta1 + b*omega1 - delta2 + ph
i)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     phi*(-a*delta1*phi - a*delta2*omega1 + b*delta1 + b*omega1 - delta2
+ phi)/
     (b*(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)),

    omega2*(a*delta1*phi + a*delta2*omega1 - b*delta1 - b*omega1 + delta2
- phi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)]
                ])
    lamda1,ev1=np.linalg.eig(J1)
    lamda2,ev2=np.linalg.eig(J2)
    lamda3,ev3=np.linalg.eig(J3)
    lamda4,ev4=np.linalg.eig(J4)
    lamda5,ev5=np.linalg.eig(J5)
    Lambda=np.stack(np.array([lamda1,lamda2,lamda3,lamda4,lamda5])) # Use
of linear algebra
    EV=np.stack(np.array([ev1,ev2,ev3,ev4,ev5]))
    return EP,Lambda<0
```

Function to find stable eqms and index of parameter sets for which all eqms are unstable

We will use this to get statistics for positive stable Eqm. values to compare model output at eqm. with them.If the ODE solver does't give same values we need to find them and run model again with differnet parameters to get reasonalbe results.

In [ ]:

```
def SatbleEQMs(param_values):
    StableEP3=[]
    StableEP4=[]
    StableEP5=[]
    UnstableEP=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if L[4].all()==True and Eqm[4][0]>0 and Eqm[4][1]>0 and Eqm[4][2]
>0:
            StableEP5.append(Eqm[4])
        elif L[3].all()==True and all(n > 0 for n in Eqm[3]):
            StableEP4.append(Eqm[3])
        elif L[2].all()==True  and all(n > 0 for n in Eqm[2]):
            StableEP3.append(Eqm[2])
        else:
             UnstableEP.append(i)

    return  [np.stack(StableEP3) , np.stack(StableEP4), np.stack(StableEP
5), np.stack(UnstableEP)]
```

Find Outliers

1) via IQR test

In [2]:

```
def FindOutliers(dataset):
    Outliers=[]
    DS_sorted = np.sort(dataset)
    Q1 = np.quantile(DS_sorted,0.25)
    Q3 = np.quantile(DS_sorted,0.75)

   # This should be identical to `scipy.stats.iqr(y1_median)`
    dataset_iqr = Q3 - Q1
    lower_bound = Q1 - 1.5 *dataset_iqr
    upper_bound = Q3 + 1.5 *dataset_iqr
    Outliers_data=dataset [(dataset < lower_bound)|(dataset >  upper_boun
d)]
    Outliers.append( Outliers_data)
    DataLower =dataset [dataset < lower_bound]
    DataUpper= DS_sorted[dataset > upper_bound]
    Outliers_position=np.where((dataset <lower_bound)  |(dataset >upper_b
ound) )
    return   Outliers_position, DataLower, DataUpper,lower_bound, upper_b
ound
```

2) via z scores

In [3]:

```
def outliers_z_score(data):
    threshold = 3
    mean_data = np.mean(data)
    stdev_data = np.std(data)
    z_scores = [(y - mean_data) / stdev_data for y in data]
    return np.where(np.abs(z_scores) > threshold),data[np.abs(z_scores) >
threshold]
```

Function for finding max,min,median,eqm values for each set of parameters

In [4]:

```
def SolSystemStat(param_values,tMax,solvers,RelTol,AbsTol):
    t = [0, tMax]
    t_eval_pts = np.linspace(0, tMax, 25*tMax)
    a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20,*_ = param_values
    X0 = [x0, y10, y20]
    pm=np.array ([ a, b,delta1,delta2,psi,phi,omega1,omega2])
    #hmax=0.01
    #,max_step=0.01
    sol=solve_ivp(DM_bacteria,t,X0,method= solvers,t_eval= t_eval_pts,arg
s=(pm,),jac=DM_jac,rtol=RelTol,atol=AbsTol);

    x, y1, y2 = sol.y
    Xstat = np.array([amax(x), mean(x), median(x), x[tMax-500:tMax].max
()])
    Y1stat=np.array([amax(y1), mean(y1), median(y1), y1[tMax-500:tMax].ma
x()])
    Y2stat = np.array([amax(y2), mean(y2), median(y2), y2[tMax-500:tMax].
max()])
    MaxEnd=np.amax([x[tMax-500:tMax].max(),y1[tMax-500:tMax].max(),y2[tMa
x-500:tMax].max()])
    return  Xstat,Y1stat, Y2stat,x,y1,y2,MaxEnd
```

Functions for ploting solution and running the model for unreasonable model output

In [ ]:

```
#Solvers that can be used
def SolPlot(param_values,tMax,solvers,RelTol,AbsTol):
    E,L=EqmValues(param_values)
    MB=MaxBound(param_values)
    a,b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = param_values
    X0 = [x0, y10, y20]
    t = [0, tMax]
    t_eval_pts = np.linspace(0, tMax, 25*tMax)
    a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20,*_ = param_values
    X0 = [x0, y10, y20]
    pm=np.array ([ a, b,delta1,delta2,psi,phi,omega1,omega2])
    #hmax=0.01
    #,max_step=0.01
    sol=solve_ivp(DM_bacteria,t,X0,method= solvers,t_eval= t_eval_pts,arg
s=(pm,),jac=DM_jac,rtol=RelTol,atol=AbsTol);
    x,y1,y2=sol.y
    Xstat = np.array([amax(x), mean(x), median(x), x[-1]])
    Y1stat=np.array([amax(y1), mean(y1), median(y1), y1[-1]])
    Y2stat = np.array([amax(y2), mean(y2), median(y2), y2[-1]])
    MinXY12=([np.min(x),np.min(y1),np.min(y2)])
    MaxEnd=np.amax([x[tMax-500:tMax].max(),y1[tMax-500:tMax].max(),y2[tMa
x-500:tMax].max()])
    fig=plt.figure(figsize=(15,5))
    plt.xlabel('Time  (day)')
    plt.ylabel('populations (num/ml)')
    plt.title(r'$\delta_{1}<\frac{1}{a}\quad  and\quad \delta_{2}<\frac
{b}{a}\quad $'+'      '+'$x_{0}=$' +str(X0))
    plt.plot(t_eval_pts,x,'r--' ,linewidth=2.5, markersize=1.5)
    plt.plot(t_eval_pts,y1,'b:', linewidth=3, markersize=0.5)
    plt.plot(t_eval_pts,y2,'g-.', linewidth=3.5, markersize=0.2)
    # set the limits
    plt.xlim([tMax-50, tMax])
    plt.ylim([0,MaxEnd])
    plt.legend( ('x','$y_{1}$','$y_{2}$'),loc='upper center', shadow=True
)
    plt.show()

    return E,L,Xstat, Y1stat,Y2stat,MinXY12,MB,sol['message']
```

Title for plotting

In [ ]:

```
TITLE=['Sensitivity of Maximum value of $x$'     ,'Sensitivity of Mean va
lue of $x$'     ,
       'Sensitivity of Median value of $x$'    ,'Sensitivity of Equilibri
um value of $x$',
       'Sensitivity of Maximum value of $y_{1}$' ,'Sensitivity of Mean va
lue of $y_{1}$' ,
       'Sensitivity of Median value of $y_{1}$','Sensitivity of Equilibri
um value of $y_{1}$',
       'Sensitivity of Maximum value of $y_{2}$' , 'Sensitivity of Mean v
alue of $y_{2}$',
       'Sensitivity of Median value of $y_{2}$','Sensitivity of Equilibri
um value of $y_{2}$']
```

SOBOL METHOD

Generate Sampels

In [ ]:

```
num_par=11
N1=1000*num_par

N1*(2*num_par+2)

param_values1 = saltelli.sample(problem, 30000,calc_second_order=False)
print(param_values1.shape)
```

Statistics for parameter sampling

In [ ]:

```
pd.DataFrame(param_values1).describe()
```

Run the model

In [ ]:

```
%%time
xSobol, y1Sobol, y2Sobol=run_model(param_values1,364,SOLVERS[0],1e-09,1e-
09)
```

In [ ]:

```
param_values1=np.loadtxt('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'
                         +"ParamValues1SobolNewNonDimV1.txt", delimiter=',')
```

In [ ]:

```
param_values1.shape
```

Save parameter values and model output to txt file for later use

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"ParamValues1SobolNewNonDimV1.txt", "w+") as f:

    np.savetxt(f,param_values1, delimiter=',',header='a,b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20,x0,y10,y20')



with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"xSobolNewNonDimV1.txt", "w+") as f:
    np.savetxt(f,xSobol, delimiter=',')


with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"y1SobolNewNonDimV1.txt", "w+") as f:
    np.savetxt(f,y1Sobol, delimiter=',')

with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"y2SobolNewNonDimV1.txt", "w+") as f:
    np.savetxt(f,y2Sobol, delimiter=',')
```

Loading the results after saving them as text files

In [ ]:

```
xSobol, y1Sobol, y2Sobol=[
np.loadtxt('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"xSobolNewNonDimV1.txt", delimiter=','),
np.loadtxt('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"y1SobolNewNonDimV1.txt", delimiter=','),
np.loadtxt('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+"y2SobolNewNonDimV1.txt", delimiter=',')]
```

Analysis of errors in model output

I)Get statistics for model output to check positivity and boundness

In [ ]:

```
xdfSobol=pd.DataFrame(xSobol[:,:]).describe()
y1dfSobol=pd.DataFrame(y1Sobol[:,:]).describe()
y2dfSobol=pd.DataFrame(y2Sobol[:,:]).describe()
print(xdfSobol.to_markdown())
print(y1dfSobol.to_markdown())
print(y2dfSobol.to_markdown())
```

In [ ]:

```
EP3psSobol,EP4psSobol,EP5psSobol,usEPsSobol=SatbleEQMs(param_values1)
```

In [ ]:

```

EP3psSobol_stat=pd.DataFrame(EP3psSobol).describe()
EP3psSobol_stat.rename(columns={0: 'x', 1: '$y_{1}$',2:'$y_{2}$ '},inplace=True)
EP4psSobol_stat=pd.DataFrame(EP4psSobol).describe()
EP4psSobol_stat.rename(columns={0: 'x', 1: '$y_{1}$',2:'$y_{2}$ '},inplace=True)
EP5psSobol_stat=pd.DataFrame(EP5psSobol).describe()
EP5psSobol_stat.rename(columns={0: 'x', 1: '$y_{1}$',2:'$y_{2}$ '},inplace=True)
UsEPsIndexSobol_stat=pd.DataFrame(EPusSobol).describe()
UsEPsIndexSobol_stat.rename(columns={0: 'Index of parameter that gives unstable Eqm. points'},inplace=True)
```

In [ ]:

```
EP3psSobol_stat
```

In [ ]:

```
EP4psSobol_stat
```

In [ ]:

```
EP5psSobol_stat
```

In [ ]:

```
UsEPsIndexSobol_stat
```

In [ ]:

```
usEPsSobol
```

Max of x, y_1, y_2 of the third, fourth and fifth eqm. points

In [ ]:

```
XmaxEP345Sobol=np.amax(np.array([EP3psSobol[:,0].max(),EP4psSobol[:,0].max(),EP5psSobol[:,0].max()]))
Y1maxEP345Sobol=np.amax(np.array([EP3psSobol[:,1].max(),EP4psSobol[:,1].max(),EP5psSobol[:,1].max()]))
Y2maxEP345Sobol=np.amax(np.array([EP3psSobol[:,2].max(),EP4psSobol[:,2].max(),EP5psSobol[:,2].max()]))
print(XmaxEP345Sobol,Y1maxEP345Sobol,Y2maxEP345Sobol)
```

Statistics for max values from boundness

In [ ]:

```
MB1=np.empty((param_values1.shape[0],3))
for i in range(param_values1.shape[0]):
       MB1[i,:] =MaxBound(param_values1[i])
```

In [ ]:

```
MBdfSobol=pd.DataFrame(MB1[:,:]).describe()
MBdfSobol.rename(columns={0: 'Max $x$', 1: 'Max $y_1$',2: 'Max $y_2$'},inplace=True)
```

Get index of unreasonable model output to modify them

1)Get index of the output that is less than zero

In [ ]:

```
np.where(xSobol<0) # Same for other variables
y1Sobol[xSobol<0]
```

1)Get index of the output that is greater than max value at eqm or from boundedness

In [ ]:

```
np.where(xSobol[:,:]>MB1[:,0].max())    #Same for other variables
np.where(xSobol[:,-1]>XmaxEP345Sobol)[0]
```

In [ ]:

```
#Store index of parameter that gives wrong model output in this
WrongOutput=[]
```

Run the model for the parameter set that give unreasnable model output

In [ ]:

```
for i in WrongOutput[]:
    XX=SolPlot(param_values1[i],2000,SOLVERS[0],1e-7,1e-7)
    print('EP=',XX[0],'Lambda=',XX[1],
          'Max , min ,median ,eqm. of x=',XX[2],
          'Max , min ,median ,eqm. of y1=',XX[3],
          'Max , min ,median ,eqm. of y2=',XX[4],
          'Min of x,y1,y2=',XX[5],
          'MaxBound=',XX[6],
          'SolStatus=',XX[7] ,
          'index of the parameter=',i,sep='\n\n')
    print('Inital values=',param_values1[i][-4:-1])
```

Last Workabout

Set eqm values from solvers to eqm. values from analytical solution for stable eqm. points

In [ ]:

```
for i in range(param_values1.shape[0]):
        Eqm,L=EqmValues(param_values1[i])
        if L[4].all()==True and Eqm[4][0]>0 and Eqm[4][1]>0 and Eqm[4][2]
>0:
            xSobol[i][-1],y1Sobol[i][-1],y2Sobol[i][-1]=Eqm[4]

        elif L[3].all()==True and all(n > 0 for n in Eqm[3]):
            xSobol[i][-1],y1Sobol[i][-1],y2Sobol[i][-1]=Eqm[3]

        elif L[2].all()==True  and all(n > 0 for n in Eqm[2]):
            xSobol[i][-1],y1Sobol[i][-1],y2Sobol[i][-1]=Eqm[2]
```

Run analysis

Here we run the analysis and plot the results as well

In [ ]:

```
Total_Sobol=[]
First_Sobol=[]
SOBOL = [i for i in xSobol.T] + [i for i in y1Sobol.T] + [i for i in y2So
bol.T]
for i, j in zip(SOBOL, range(12)):
    SI_Sobol=sobol.analyze(problem, i,calc_second_order=False,num_resampl
es=1000)
    total, first =SI_Sobol.to_df()
    #Renaming the columns names to put them in math notation
    totalNew=total.rename(columns={'ST': '$S_{T}$','ST_conf':'${S_{T}}_{c
onf}$'})
    firstNew=first.rename(columns={'S1': '$S_{1}$','S1_conf':'${S_{1}}_{c
onf}$'})
    Total_Sobol.append(totalNew)
    First_Sobol.append(firstNew)
    barplot(total)
    #plt.xlim([-1, 100])
    plt.ylim([0, 1.2])
    plt.title(TITLE[j] )
    plt.gcf().set_size_inches(10, 4.8)
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+TITLE
[j]+''+'NewNonDimModelTotalSobolV1.png',
                dpi=300, bbox_inches='tight')
    barplot(first)
    #plt.xlim([-1, 100])
    plt.ylim([0, 1])
    plt.title(TITLE[j] )
    plt.gcf().set_size_inches(10, 4.8)
    plt.savefig('NonDimFigures\ModelNewNondim\ModelNewNonDimSobol/'+TITLE
[j]+''+'NewNonDimModelFirstSobolV1.png',
                dpi=300, bbox_inches='tight')
    plt.show();plt.close()
```

Save Sensitivity results as dataframe into txt file

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+
          "Total_SobolNewNonDim.txt", "w+") as txt_file:
    print (Total_Sobol, file=txt_file)
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+
          "First_SobolNewNonDim.txt", "w+") as txt_file:
    print (First_Sobol, file=txt_file)
```

Convet dataframe to latex table

In [ ]:

```
STx_Max,STx_Mean,STx_Median,STx_End,STy1_Max,STy1_Mean,STy1_Median,STy1_E
nd,
STy2_Max,STy2_Mean,STy2_Median,STy2_End=Total_Sobol
S1x_Max,S1x_Mean,S1x_Median,S1x_End,S1y1_Max,S1y1_Mean,S1y1_Median,S1y1_E
nd,
S1y2_Max,S1y2_Mean,S1y2_Median,S1y2_End=First_Sobol
```

The whole table

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+
          "NewNonDimModelTotal_Sobolv1.txt", "w+") as latex_file:
    for i, j in zip(Total_Sobol, range(12)):
        print (TITLE[j], file=latex_file)
        print(i.to_latex(index=True,escape=False), file=latex_file)
```

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'
          +"NewNonDimModelTotal_Sobolv1Sorted.txt", "w+") as latex_file:
    for i, j in zip(Total_Sobol, range(12)):
        print (TITLE[j], file=latex_file)
        print(i.round(4).sort_values(by=['$S_{T}$'], ascending=False).to_
latex(index=True,escape=False), file=latex_file)
```

Morris method

Generate Samples

In [ ]:

```
N2=100*num_par
N2*(num_par+1)
param_values2 =SALib.sample.morris.sample(problem, 15000)
print(param_values2.shape)
```

Run the model

In [ ]:

```
%%time
xMorris, y1Morris, y2Morris=run_model(param_values2,364,SOLVERS[0],1e-09,
1e-09)
```

Run sensitivty analysis and plot results

In [ ]:

```
MORRIS = [i for i in xMorris.T] + [i for i in y1Morris.T] + [i for i in y
2Morris.T]
RESULTSMorris=[]

for i, j in zip(MORRIS, range(12)):
    SI=morris.analyze(problem,param_values2, i)
    SI_DF=SI.to_df()
    #Renaming columns names to get mathematical notation
    SI_DFnew=SI_DF.rename(columns={'mu': '$\mu$','mu_star':'$\mu^{\star}
$','sigma':'$\sigma$',
                                   'mu_star_conf':'$\mu^{\star}_{conf}$'
})
    #Plotting in two differint wayys
    fig, ax1 = plt.subplots(1)
    plt.gcf().set_size_inches(6.4, 4.8)
    SALib.plotting.morris.horizontal_bar_plot(ax1,SI , {'color':'r'}, sor
tby='mu_star', unit='')
    #plt.xlim([-1, 100])
    #plt.ylim([-2, 0.7])
    plt.title(TITLE[j] )
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimMorris/'+TITL
E[j]+''
                +'NewNonDimModelMorris.png', dpi=300, bbox_inches='tight'
)

    barplot(SI_DFnew.drop(['$\mu^{\star}_{conf}$'],axis=1))
    #plt.xlim([-1, 100])
    #plt.ylim([-2, 0.7])
    plt.title(TITLE[j] )
    plt.gcf().set_size_inches(8, 4.8)
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimMorris/'+TITL
E[j]+''
                +'NewNonDimModelMorris1.png', dpi=300, bbox_inches='tigh
t')
    plt.show();plt.close()

    RESULTSMorris.append(SI_DFnew)
```

In [ ]:

```
Sixm_Max,Sixm_Mean,Sixm_Median,Sixm_End,Siy1m_Max,Siy1m_Mean,
Siy1m_Median,Siy1m_End,Siy2m_Max,Siy2m_Mean,Siy2m_Median,Siy2m_End=RESULT
SMorris
```

rbd_fast method

Generate Samples

Number of model runs = number of samples

In [ ]:

```
param_values3 =SALib.sample.latin.sample(problem, 100000)
```

In [ ]:

```
%%time
xRbdFast, y1RbdFast, y2RbdFast=run_model(param_values3,364,SOLVERS[0],1e-
09,1e-09)
```

Run analysis

In [ ]:

```
RbdFast = [i for i in xRbdFast.T] + [i for i in y1RbdFast.T] + [i for i i
n y2RbdFast.T]
RESULTSRbdFast=[]

for i, j in zip(RbdFast, range(12)):
    Si_RbdFast= rbd_fast.analyze(problem,param_values3, i)
    Sidf_RbdFast=Si_RbdFast.to_df()
    #Renaming the columns names to put them in math notation
    Sidf_RbdFastNew=Sidf_RbdFast.rename(columns={'S1': '$S_{1}$'})
    RESULTSRbdFast.append( Sidf_RbdFastNew)
    barplot( Sidf_RbdFastNew)
    #plt.xlim([-1, 100])
    #plt.ylim([-0.5, 0.7])
    plt.title(TITLE[j] )
    #plt.gcf().set_size_inches(20, 4)
    #Save file in a specific folder and name from title
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimRbdFast/'+TIT
LE[j]+''+
                'NewNonDimModelRbdFast.png', dpi=300, bbox_inches='tight'
)
    plt.show
```

Delta method

We can use samples and model output from rbd_fast method as both uses the same sampling method

Run model

param_values4 =SALib.sample.latin.sample(problem, 60000)
print(param_values4.shape)%%time xDelta, y1Delta, y2Delta=run_model(param_values4)

In [ ]:

```
xDelta,y1Delta, y2Delta=xRbdFast,y1RbdFast,y2RbdFast
param_values4=param_values3
```

Run analysis

In [ ]:

```
%%time

DELTA=[i for i in xDelta.T] + [i for i in y1Delta.T] + [i for i in y2Delt
a.T]
RESULTSDelta=[]
for i, j in zip(DELTA, range(12)):
    Si_Delta= delta.analyze(problem,param_values4, i,num_resamples=1000)
    Sidf_Delta= Si_Delta.to_df()
    #Renaming the columns names to put them in math notation
    Sidf_DeltaNew=Sidf_Delta.rename(columns={'delta':'$\delta$','delta_co
nf':'$\delta_{conf}$',
                                             'S1': '$S_{1}$','S1_conf':
'${S_{1}}_{conf}$'})
    RESULTSDelta.append( Sidf_DeltaNew)
    barplot( Sidf_DeltaNew.drop(['$\delta_{conf}$','${S_{1}}_{conf}$'],ax
is=1))
    #plt.xlim([-1, 100])
    #plt.ylim([-0.5, 0.7])
    plt.title(TITLE[j] )
    #plt.gcf().set_size_inches(20, 4)
    #Save file in a specific folder and name from title
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimDelta/'+TITLE
[j]+''+
                'NewNonDimModelDeltat.png', dpi=300, bbox_inches='tight')
```

Run analysis for median and eqm. value of y_1 after log and square root transformation

In [ ]:

```
y1Delta[y1Delta<=0]
```

In [ ]:

```
y1DeltaC=y1Delta.copy()
#y1DeltaC[y1DeltaC<=0] = np.nextafter(0, 1)
#y1DeltaC[y1DeltaC<0]=0
y1DeltaC=y1DeltaC+np.nextafter(0, 1)
```

Run Si analysis after log transforming

In [ ]:

```
DELTA1=[ y1DeltaC.T[2],y1DeltaC.T[3]]
title=['Sensitivity of Median value of $y_1$ after log transformation',
       'Sensitivity of Equilibrium value of $y_1$ after log transformatio
n']
RESULTSDelta1=[]
for i, j in zip(DELTA1, range(2)):
    Si_Delta= delta.analyze(problem,param_values4, np.log(i),num_resample
s=1000)
    Sidf_Delta= Si_Delta.to_df()
    #Renaming the columns names to put them in math notation
    Sidf_DeltaNew=Sidf_Delta.rename(columns={'delta':'$\delta$','delta_co
nf':'$\delta_{conf}$',
                                              'S1': '$S_{1}$','S1_conf':
'${S_{1}}_{conf}$'})
    RESULTSDelta1.append( Sidf_DeltaNew)
    barplot( Sidf_Delta)
    #plt.xlim([-1, 100])
    #plt.ylim([-0.5, 0.7])
    plt.title(TITLE[j] )
    #plt.gcf().set_size_inches(20, 4)
    #Save file in a specific folder and name from title
    plt.savefig('NonDimFigures\ModelNewNonDim\ModelNewNonDimDelta/'+TITLE
[j]+''+
                'NewNonDimModelDeltatLog.png', dpi=300, bbox_inches='tigh
t')
    plt.show
```

Analyzing and setting results together

Comparison between Sobol ,RBD-FAST and DMIM methods for S_1

In [ ]:

```
S1Max_all=   pd.concat([First_Sobol[0]['$S_{1}$'].round(4),
                        RESULTSRbdFast[0].round(4) ['$S_{1}$'],RESULTSDel
ta[0].round(4) ['$S_{1}$'] ,
                        First_Sobol[4]['$S_{1}$'].round(4),RESULTSRbdFast
[4].round(4) ['$S_{1}$'],
                        RESULTSDelta[4].round(4) ['$S_{1}$'],First_Sobol[
8] ['$S_{1}$'].round(4),
                        RESULTSRbdFast[8].round(4) ['$S_{1}$'],RESULTSDel
ta[8].round(4) ['$S_{1}$']],axis=1,
                       keys=['${S_{1}}_{Sobol}$' , '${S_{1}}_{RbdFast}$'
,'${S_{1}}_{DMIM}$ '
                             ,'${S_{1}}_{Sobol}$', '${S_{1}}_{RbdFast}$'
,'${S_{1}}_{DMIM}$ '
                             ,'${S_{1}}_{Sobol}$ ' , '${S_{1}}_{RbdFast}
$', '${S_{1}}_{DMIM}$ '])

S1Mean_all=  pd.concat([First_Sobol[1]['$S_{1}$'].round(4),RESULTSRbdFast
[1].round(4) ['$S_{1}$'],
                        RESULTSDelta[1].round(4) ['$S_{1}$'] ,First_Sobol
[5]['$S_{1}$'].round(4),
                        RESULTSRbdFast[5].round(4) ['$S_{1}$'],RESULTSDel
ta[5].round(4) ['$S_{1}$'],
                        First_Sobol[9] ['$S_{1}$'].round(4),RESULTSRbdFas
t[9].round(4) ['$S_{1}$'],
                        RESULTSDelta[9].round(4) ['$S_{1}$']],axis=1,
                       keys=['${S_{1}}_{Sobol}$' , '${S_{1}}_{RbdFast}$'
,'${S_{1}}_{DMIM}$ ' ,
                             '${S_{1}}_{Sobol}$', '${S_{1}}_{RbdFast}$' ,
'${S_{1}}_{DMIM}$ '   ,
                             '${S_{1}}_{Sobol}$ ' , '${S_{1}}_{RbdFast}$'
, '${S_{1}}_{DMIM}$ '])

S1Median_all=pd.concat([First_Sobol[2]['$S_{1}$'].round(4),RESULTSRbdFast
[2].round(4) ['$S_{1}$'],
                        RESULTSDelta[2].round(4) ['$S_{1}$'] ,First_Sobol
[6]['$S_{1}$'].round(4),
                        RESULTSRbdFast[6].round(4) ['$S_{1}$'],RESULTSDel
ta[6].round(4) ['$S_{1}$'],
                        First_Sobol[10]['$S_{1}$'].round(4),RESULTSRbdFas
t[10].round(4) ['$S_{1}$'],
                        RESULTSDelta[10].round(4) ['$S_{1}$']],axis=1,key
s=
                       ['${S_{1}}_{Sobol}$' , '${S_{1}}_{RbdFast}$' ,'${S
_{1}}_{DMIM}$ ' ,

                        '${S_{1}}_{Sobol}$', '${S_{1}}_{RbdFast}$' ,'${S_
{1}}_{DMIM}$ '  ,

                        '${S_{1}}_{Sobol}$ ' , '${S_{1}}_{RbdFast}$',
'${S_{1}}_{DMIM}$ '])

S1End_all=   pd.concat([
```

```
    First_Sobol[3]['$S_{1}$'].round(4),RESULTSRbdFast[3].round(4) ['$S_
{1}$'],
    RESULTSDelta[3].round(3) ['$S_{1}$'] ,First_Sobol[7]['$S_{1}$'].round
(4),RESULTSRbdFast[7].round(4) ['$S_{1}$'],
    RESULTSDelta[7].round(4) ['$S_{1}$'],First_Sobol[11]['$S_{1}$'].round
(4),RESULTSRbdFast[11].round(4) ['$S_{1}$'],
    RESULTSDelta[11].round(4) ['$S_{1}$']],axis=1,
    keys=['${S_{1}}_{Sobol}$' , '${S_{1}}_{RbdFast}$' ,'${S_{1}}_{DMIM}$
 ' ,'${S_{1}}_{Sobol}$',
          '${S_{1}}_{RbdFast}$' ,'${S_{1}}_{DMIM}$ '  ,'${S_{1}}_{Sobol}$
' , '${S_{1}}_{RbdFast}$', '${S_{1}}_{DMIM}$ '])
```

Comparison between S_T Sobol and $\mu^\star$ Morris and δ from DMIM

In [ ]:

```
STMax_all=pd.concat([
Total_Sobol[0]['$S_{T}$'].round(4),RESULTSMorris[0].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[0].round(4) ['$\delta$'] ,
Total_Sobol[4]['$S_{T}$'].round(4),RESULTSMorris[4].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[4].round(4) ['$\delta$'],
Total_Sobol[8]['$S_{T}$'].round(4), RESULTSMorris[8].round(4) ['$\mu^{\st
ar}$'],  RESULTSDelta[8].round(4) ['$\delta$']],
axis=1,   keys=['${S_{T}}_{Sobol}$', '${\mu^{\star}}_{Morris}$','${\delt
a}_{DMIM}$ ','${S_{T}}_{Sobol}$',
                '${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ ','${S_{T}}
_{Sobol}$', '${\mu^{\star}}_{Morris}$',
                '${\delta}_{DMIM}$ '])

STMean_all=  pd.concat([
Total_Sobol[1]['$S_{T}$'].round(4),RESULTSMorris[1].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[1].round(4) ['$\delta$'] ,
Total_Sobol[5]['$S_{T}$'].round(4),RESULTSMorris[5].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[5].round(4) ['$\delta$'],
Total_Sobol[9]['$S_{T}$'].round(4), RESULTSMorris[9].round(4) ['$\mu^{\st
ar}$'],  RESULTSDelta[9].round(4) ['$\delta$']],
axis=1,   keys=['${S_{T}}_{Sobol}$', '${\mu^{\star}}_{Morris}$','${\delt
a}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ '])

STMedian_all=pd.concat([
Total_Sobol[2]['$S_{T}$'].round(4),RESULTSMorris[2].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[2].round(4) ['$\delta$'] ,
Total_Sobol[6]['$S_{T}$'].round(4),RESULTSMorris[6].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[6].round(4) ['$\delta$'],
Total_Sobol[10]['$S_{T}$'].round(4),RESULTSMorris[10].round(4)['$\mu^{\st
ar}$'],RESULTSDelta[10].round(4) ['$\delta$']],
axis=1,    keys=['${S_{T}}_{Sobol}$', '${\mu^{\star}}_{Morris}$','${\delt
a}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ '])

STEnd_all=   pd.concat([
Total_Sobol[3]['$S_{T}$'].round(4),RESULTSMorris[3].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[3].round(4) ['$\delta$'] ,
Total_Sobol[7]['$S_{T}$'].round(4),RESULTSMorris[7].round(4) ['$\mu^{\sta
r}$'],RESULTSDelta[7].round(4) ['$\delta$'],
Total_Sobol[11]['$S_{T}$'].round(4),RESULTSMorris[11].round(4)['$\mu^{\st
ar}$'], RESULTSDelta[11].round(4) ['$\delta$']],
axis=1,   keys=['${S_{T}}_{Sobol}$', '${\mu^{\star}}_{Morris}$','${\delt
a}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ ','${S_{T}}_{Sobol}$',
'${\mu^{\star}}_{Morris}$','${\delta}_{DMIM}$ '])
```

Add sum to all tables and set negative values to zero

In [ ]:

```
SiTables=[S1Max_all,S1Mean_all,S1Median_all,S1End_all,STMax_all,STMean_al
l,STMedian_all,STEnd_all]
for i in SiTables:
    i.drop(['$\sum$'],inplace=True)
    i.loc['$\sum$'] =i.clip(lower=0).sum(axis=0)
```

In [ ]:

```
AllTablesTitle=['Main Effect sensitivity index for all methods for Max va
lue of all variables',

                'Main Effect sensitivity index for all methods for Mean v
alue of all variables',

                'Main Effect sensitivity index for all methods for Median
value of all variables',

                'Main Effect sensitivity index for all methods for Eqm. v
alue of all variables',

                'Total or Interaction Effect sensitivity index for all me
thods for Max value of all variables',

                'Total or Interaction Effect sensitivity index for all me
thods for Mean value of all variables',

                'Total or Interaction Effect sensitivity index for all me
thods for Median value of all variables',

                'Total or Interaction Effect sensitivity index for all me
thods for Eqm. value of all variables'


               ]


AllTablesTitleShort=['MainEffetctMax',
                     'MainEffetctMean' ,
                     'MainEffetctMedian',
                     'MainEffetctEqm',

                     'TotalInteractionsEffectMax',
                     'TotalInteractionsEffectMean',
                     'TotalInteractionsEffectMedian',
                     'TotalInteractionsEffectEqm',
                              ]
```

Save summary tables as latex files

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimAllTablesCompared/'
+
          "MainTotalInteractionEffectAllNewNonDim.txt", "w+") as latex_fi
le:
    for i,j in zip(SiTables,range(8)):
        print(AllTablesTitle[j], file=latex_file)
        print(i.to_latex(index=True,escape=False), file=latex_file)
```

Save summary tables as excel files

In [ ]:

```
for i,j in zip(SiTables,range(8)):
    i.to_excel(r'C:\Users\farha\Documents\Python_new\NonDimFigures\ModelN
ewNonDim\ModelNewNonDimAllTablesCompared/'+
               AllTablesTitleShort[j]+'NewNonDim.xlsx',index = True, head
er=True)
```

Save with use of rank

In [ ]:

```
with open('NonDimFigures\ModelNewNonDim\ModelNewNonDimAllTablesCompared/'
+
          "MainTotalInteractionEffectAllNewNonDimRank.txt", "w+") as late
x_file:
    for i,j in zip(SiTables,range(8)):
        print(AllTablesTitle[j], file=latex_file)
        print(i.rank(method='average',ascending=False).to_latex(index=Tru
e,escape=False), file=latex_file)
```

B.3 Numerical simulation

Her we present the python code formed via Jupyter notebook used in the numerical simulation section.

Importing necessary packages

In [16]:

```
%matplotlib inline
#uncomment the next three lines if you want interactive polts
#%matplotlib notebook
#from matplotlib import use
#use("Qt5Agg")
import numpy as np
from numpy import *
from cmath import*

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.signal import find_peaks
from scipy import signal
import scipy as scp
from scipy.integrate import solve_ivp, odeint
import pandas as pd
from scipy import linalg as LA
import SALib
from SALib.sample import saltelli,morris,latin,ff,finite_diff
from SALib.sample import morris
from SALib.analyze import sobol,dgsm,rbd_fast,delta,ff,dgsm
from SALib.analyze import morris
from SALib.plotting.bar import plot as barplot
from SALib.plotting.morris import *
import nolds
```

Import the parameter set from saltelli (Sobol) sampling method

In [17]:

```
param_values1=np.loadtxt('NonDimFigures\ModelNewNonDim\ModelNewNonDimSobol/'+
                         "ParamValues1SobolNewNonDimV1.txt", delimiter=',')
```

Get nondimensional parameter from dimensional parameter

to check that both Dim and NonDim models give the same dynamics

In [18]:

```
def DimToNonDim(pDim):
    k,r,f1,f2,beta1, beta2, epsilon1,epsilon2,psi12,phi12,omega11,omega22
,x0,y10,y20=pDim
    pNonDim=np.array([r/(beta1*f1*k),

            (beta2*f2)/(beta1*f1),

            epsilon1/r,

            epsilon2/r,

            psi12/f2,

            phi12/f1,

            omega11/f1,

            (omega22)/f2,

            (beta1*f1*x0)/r,

            (f1*y10)/r,

            (r*beta2*y20)/(beta1*f1)])


    return pNonDim
```

Model ,functions and solvers

In [19]:

```
def DM_bacteria(t,X,pm):
    x, y1, y2 = X
    a, b,delta1,delta2,psi,phi,omega1,omega2=pm
    dxdt=x*(1-a*x-y1-b*y2)
    dy1dt=y1*(-delta1-omega1*y1-psi*b*y2+x)
    dy2dt=y2*(-delta2-omega2*b*y2+phi*y1+b*x)
    return [dxdt,dy1dt,dy2dt]
def DM_jac(t,X,pm):
    x, y1,y2 = X
    a, b,delta1,delta2,psi,phi,omega1,omega2=pm
    J=np.array([[-2*a*x - b*y2 - y1 + 1, -x, -b*x],
                [y1, -b*psi*y2 - delta1 - 2*omega1*y1 + x, -b*psi*y1],
                [b*y2, phi*y2, -2*b*omega2*y2 + b*x - delta2 + phi*y1]])
    return J
```

```
In [20]:
```

```
#Solver that can be used
SOLVERS=['RK45','RK23','DOP853','BDF','Radau','LSODA']
```

Function to find eqm. points and check their stability for a given set of parameters

In [21]:

```
def EqmValues(param_values):
    a, b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = param_values
    EP1=np.array([0,0,0])

    EP2=np.array([1/a, 0, 0])

    EP3=np.array([(delta1 + omega1)/(a*omega1 + 1), (-a*delta1 + 1)/(a*om
ega1 + 1), 0])

    EP4=np.array([(delta2 + omega2)/(a*omega2 + b), 0, (-a*delta2 + b)/(b
*(a*omega2 + b))])

    EP5=np.array([(delta1*(omega2 + phi) + delta2*(omega1 - psi) + omega1
*omega2 + phi*psi)/
                  (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega
2 + phi),

                  (-b*psi - delta1*(a*omega2 + b) + delta2*(a*psi + 1) +
omega2)/
                  (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega
2 + phi),

                  (b*omega1 - delta1*(a*phi - b) - delta2*(a*omega1 + 1)
+ phi)/
                  (b*(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + om
ega2 + phi))])
    EP=np.stack(np.array([EP1,EP2,EP3,EP4,EP5]))


    J1=np.array([[1, 0, 0], [0, -delta1, 0], [0, 0, -delta2]])

    J2=np.array([[-1, -1/a, -b/a], [0, -delta1 + 1/a, 0], [0, 0, -delta2
+ b/a]])

    J3=np.array([
    [-a*(delta1 + omega1)/(a*omega1 + 1), -(delta1 + omega1)/(a*omega1 +
1),-b*(delta1 + omega1)/(a*omega1 + 1)],
    [(-a*delta1 + 1)/(a*omega1 + 1),omega1*(a*delta1 - 1)/(a*omega1 + 1),
b*psi*(a*delta1 - 1)/(a*omega1 + 1)],
    [0, 0, (b*(delta1 + omega1) - delta2*(a*omega1 + 1) - phi*(a*delta1 -
1))/(a*omega1 + 1)]])

    J4=np.array([
    [-a*(delta2 + omega2)/(a*omega2 + b), -(delta2 + omega2)/(a*omega2 +
b),-b*(delta2 + omega2)/(a*omega2 + b)],
    [0, (-delta1*(a*omega2 + b) + delta2 + omega2 + psi*(a*delta2 - b))/(
a*omega2 + b), 0],
    [(-a*delta2 + b)/(a*omega2 + b), -phi*(a*delta2 - b)/(b*(a*omega2 + b
)), omega2*(a*delta2 - b)/(a*omega2 + b)]])

    J5=np.array([
    [a*(-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1
```

```
*omega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     (-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1*o
mega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     b*(-delta1*omega2 - delta1*phi - delta2*omega1 + delta2*psi - omega1
*omega2 - phi*psi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)],


    [(-a*delta1*omega2 + a*delta2*psi - b*delta1 - b*psi + delta2 + omega
2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     omega1*(a*delta1*omega2 - a*delta2*psi + b*delta1 + b*psi - delta2 -
omega2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     b*psi*(a*delta1*omega2 - a*delta2*psi + b*delta1 + b*psi - delta2 -
omega2)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)],


    [(-a*delta1*phi - a*delta2*omega1 + b*delta1 + b*omega1 - delta2 + ph
i)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi),

     phi*(-a*delta1*phi - a*delta2*omega1 + b*delta1 + b*omega1 - delta2
+ phi)/
     (b*(a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)),

    omega2*(a*delta1*phi + a*delta2*omega1 - b*delta1 - b*omega1 + delta2
- phi)/
     (a*omega1*omega2 + a*phi*psi + b*omega1 - b*psi + omega2 + phi)]
                ])
    lamda1,ev1=np.linalg.eig(J1)
    lamda2,ev2=np.linalg.eig(J2)
    lamda3,ev3=np.linalg.eig(J3)
    lamda4,ev4=np.linalg.eig(J4)
    lamda5,ev5=np.linalg.eig(J5)
    Lambda=np.stack(np.array([lamda1,lamda2,lamda3,lamda4,lamda5])) # Use
of linear algebra
    EV=np.stack(np.array([ev1,ev2,ev3,ev4,ev5]))
    return EP,Lambda<0
```

Function to find stable eqms and index of parameter sets for which all eqms are unstable

We will use this to get statistics for positive stable Eqm. values to compare model output at eqm. with them.If the ODE solver does't give same values we need to find them and run model again with differnet parameters to get reasonalbe results.

In [22]:

```
def SatbleEQMs(param_values):
    StableEP3=[]
    StableEP4=[]
    StableEP5=[]
    UnstableEP=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if L[4].all()==True and Eqm[4][0]>0 and Eqm[4][1]>0 and Eqm[4][2]
>0:
            StableEP5.append(Eqm[4])
        elif L[3].all()==True and all(n > 0 for n in Eqm[3]):
            StableEP4.append(Eqm[3])
        elif L[2].all()==True  and all(n > 0 for n in Eqm[2]):
            StableEP3.append(Eqm[2])
        else:
             UnstableEP.append(i)

    return  [np.stack(StableEP3) , np.stack(StableEP4), np.stack(StableEP
5), np.stack(UnstableEP)]
```

Parameter sets used in simulation

In [23]:

```
# Base parameter set
pm_base=np.array([7.18e-05, 0.28, 0.01, 0.46   ,
       1.56, 0.21, 0.31, 0.28,
       278.52, 592.59, 3.75])
#Psrsmeter sets used in study the effect of each parameters
pmbioAll=np.array([
    [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59,
3.75],

    [7.18e-05, 0.1, 0.01, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.
75],
    [7.18e-05, 1,   0.01, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.
75],
    [7.18e-05, 10,  0.01, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.
75],


    [7.18e-05, 0.28, 0.1, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.
75],
    [7.18e-05, 0.28, 1, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.75
],
    [7.18e-05, 0.28, 10, 0.46 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.7
5],


     [7.18e-05, 0.28, 0.01, 0.1 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 1 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.7
5],
     [7.18e-05, 0.28, 0.01, 10 ,1.56, 0.21, 0.31, 0.28,278.52, 592.59, 3.
75],

     [7.18e-05, 0.28, 0.01, 0.46 ,0.1, 0.21, 0.31, 0.28,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1   , 0.21, 0.31, 0.28,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,10 , 0.21, 0.31, 0.28,278.52, 592.59,
3.75],


     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.37 , 0.31, 0.28,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.371      , 0.31, 0.28,278.52, 5
92.59, 3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 1.234      , 0.31, 0.28,278.52, 5
92.59, 3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 1.235      , 0.31, 0.28,278.52, 5
92.59, 3.75],

     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 0.1 , 0.28,   278.52, 592.5
9, 3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 1     , 0.28,   278.52, 59
```

```
2.59, 3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 10    , 0.28,   278.52, 59
2.59, 3.75],



     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 0.31, 0.1 ,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 0.31, 1   ,278.52, 592.59,
3.75],
     [7.18e-05, 0.28, 0.01, 0.46 ,1.56, 0.21, 0.31, 5   ,278.52, 592.59,
3.75]])

#Parameter set that gives chaotic behavoir
chaotic=np.array([3.99256592e-04, 2.80065918e-01, 9.78851318e+00, 4.33094
482e+00,
       1.59216309e-01, 4.10463217e+00, 7.37010376e-01, 1.67592651e-01,
       4.92309570e+03, 7.03787720e+02, 9.29061279e+01])
#Parameter set that gives periodic behavoir
periodic=np.array([
    [6.22254333e-04, 2.77951050e-01, 5.19471741e+00, 1.81878357e+00,
       1.95896912e+00, 4.23143017e+00, 4.14471832e-01, 2.08935788e+00,
       9.27642822e+03, 1.40179749e+01, 8.68908386e+01],
     [1.52433304e-06, 2.20304719e-01, 2.19023890e+01, 5.75428250e+00,
       2.35503628e-01, 1.38396945e+00, 3.52600725e-01, 1.21520420e-01,
       2.91293087e+05, 1.14919364e+04, 8.40371055e+00]])

#Parameter set  that gives globally stable interior equilibrium poin
gsEP5All=np.array([6.94580078e-05, 7.91259766e-01, 5.57690430e+00, 7.5878
4180e+00,
       3.10190430e+00, 2.99076284e+00, 1.75375439e+00, 1.05959473e-01,
       3.33496094e+03, 4.64891113e+02, 5.39892578e+00])


#Parameter set that gives two stable equilibrium points EP3 ,EP4
StableEP34=np.array(
    [5.03549805e-04, 6.84912109e-01, 2.95688477e+00, 5.52856445e+00,
       9.10571289e+00, 5.05460986e-01, 6.44702148e-02, 1.82924756e+00,
       5.21972656e+03, 5.49767090e+02, 1.43505859e+00])
```

Save stability information to a text file for parameter sets used in simulation

In [ ]:

```
with open('NumericalSimulation/'+"ParameterValus-(StableEP34All).txt", "w
+") as txt_file:
    for i,j in zip(pmbioAll,range(100)):
        eqm,ev=EqmValues(param_values1[i])

        print ('Parameter'+str(j)+'=',param_values1[i],'Equilibrium Valu
e'+str(j)+ '=',
               eqm,'EV'+str(j)+'=',ev,'Stability(EigenValues)'+str(j)+'='
,ev<0,sep='\n', file=txt_file)
```

Function to find parameter sets for which the condition for global stability of interior eqm. point is sataisfied

In [25]:

```
def GlobalStabilityEP5(param_values):
    GS=[]
    for i in range(param_values.shape[0]):
        a,b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = param_values
[i]
        GSep5=2*a*(4*b*omega1*omega2 - (b*psi - phi)**2)
        Eqm,L=EqmValues(param_values[i])
        if  Eqm[4][0]>0 and Eqm[4][1]>0 and Eqm[4][2]>0 and GSep5>0:
            GS.append(i)

    return GS
```

In [26]:

```
Cond=GlobalStabilityEP5(param_values1)
```

Function to find parameter sets for which both EP_3 and EP_4 are stable.

In [27]:

```
def SatbleEP34(param_values):
    StableEP34=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if L[3].all()==True and all(n > 0 for n in Eqm[3]) and L[2].all()
==True  and all(n > 0 for n in Eqm[2]):
            StableEP34.append(i)
    return   StableEP34
```

Function to find parameter sets for which both EP_3 and EP_5 are stable.

However ther is not any one

In [41]:

```
def SatbleEP35(param_values):
    StableEP35=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if all(n >= 0 for n in Eqm[2]) and L[2].all()==True \
        and Eqm[4][0]>0 and Eqm[4][1]>0 and Eqm[4][2]>0 and L[4].all()==T
rue :
            StableEP35.append(i)
    return   StableEP35
```

Function to find parameter sets for which both EP_4 and EP_5 are stable.

However ther is not any one

In [42]:

```
def SatbleEP45(param_values):
    StableEP45=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if all(n >= 0 for n in Eqm[3]) and L[3].all()==True and Eqm[4][0]
>0 \
        and Eqm[4][1]>0 and Eqm[4][2]>0 and L[4].all()==True   :
            StableEP45.append(i)
    return   StableEP45
```

Function to find parameter sets for which both EP_3, EP_4 ,and EP_5 are stable.

However ther is not any one

In [43]:

```
def SatbleEP345(param_values):
    StableEP345=[]
    for i in range(param_values.shape[0]):
        Eqm,L=EqmValues(param_values[i])
        if L[2].all()==True and all(n > 0 for n in Eqm[2]) and L[3].all()
==True \
        and all(n > 0 for n in Eqm[3]) and L[4].all()==True  and all(n >
0 for n in Eqm[4]):
            StableEP345.append(i)
    return   StableEP345
```

Function for ploting time series, peaks, and phase portrait

In [ ]:

```
#Solvers that can be used
colors1=['c','k','b','r','y','m','g']
def SolPlot4(pm,tMax,solvers,RelTol,AbsTol):
    a,b,delta1,delta2,psi,phi,omega1,omega2,x0,y10,y20 = pm
    #find equilibrium points from analytical solution
    E,L=EqmValues(pm) # initial values
    InitVals=np.array([[E[2][0]+5,E[2][1]+5,E[2][2]],
                       [E[3][0]+5,E[3][1],E[3][2]+5],
                        list(E[4]+0.03),
                       [10,8,5],
                       [0.01,0.02,0.03],
                      ])
    t=[0,tMax]
    t_eval_pts=np.linspace(0,tMax,300*tMax)
    sol= [solve_ivp(DM_bacteria,t,init,method= solvers,t_eval= t_eval_pts
,\
                    args=(pm[0:8],),jac=DM_jac,rtol=RelTol,atol=AbsTol) f
or init in InitVals];
# first plot Time series

    fig1=plt.figure(figsize=(10,5))
    for k in range(len(InitVals[0:4])):
        plt.subplot(2,2,k+1);
        xsol,y1sol,y2sol=sol[k].y
        print([ xsol.min(), y1sol.min(),y2sol.min()])
        eqms=[xsol[-1],y1sol[-1],y2sol[-1]]
        plt.xlabel('Time')
        plt.ylabel('populations ')
        plt.title(r' $\delta_{1}<\frac{1}{a}\quad  and\quad \delta_{2}<\f
rac{b}{a}\quad $'+'      '+'$x_{0}=$'
                  +str(list(np.round(InitVals[k],2))))
        plt.plot(t_eval_pts,xsol,'r--' ,linewidth=1, markersize=1.5)
        plt.plot(t_eval_pts,y1sol,'b:', linewidth=1.2, markersize=0.5)
        plt.plot(t_eval_pts,y2sol,'g-.', linewidth=1.5, markersize=0.2)
        # set the limits
        plt.xlim([0, tMax])
        #plt.ylim([0,  MaxEnd])
        plt.legend( ('$x$','$y_{1}$','$y_{2}$'), shadow=True)
        #if you want horizontal legend use
        #plt.legend( ('$x$','$y_{1}$','$y_{2}$'), shadow=True, loc= 2, mo
de = "expand", ncol = 3 )
        plt.tight_layout();
        plt.savefig('NumericalSimulation\TimeSeries-(chaotic2).png', dpi=
300, bbox_inches='tight')

#second plot peaks
    # Calculate the peaks for the third inirial condition and plotting
    xsolPeak,y1solPeak,y2solPeak=sol[2].y
    fig2=plt.figure(figsize=(10,5))
    plt.subplot(2,2,1);
    plt.xlabel('Time')
    plt.ylabel('populations ')
```

```
    plt.title(r' $\delta_{1}<\frac{1}{a}\quad  and\quad \delta_{2}<\frac
{b}{a}\quad $'+'      '
              +'$x_{0}=$' +str(list(np.round(InitVals[0],3))))
    plt.plot(t_eval_pts,xsolPeak,'r--' ,linewidth=1.5, markersize=1)
    plt.plot(t_eval_pts,y1solPeak,'b:', linewidth=2, markersize=0.5)
    plt.plot(t_eval_pts,y2solPeak,'g-.', linewidth=1.1, markersize=0.2)
    plt.xlim([tMax-50, tMax])
    #plt.ylim([0,    100])
    plt.legend( ('$x$','$y_{1}$','$y_{2}$'), shadow=True)
    plt.tight_layout()

    plt.subplot(2,2,2)

    Xpeaks, XPropDic = find_peaks(xsolPeak, distance=2,height=0.1)
    Y1peaks, _ = find_peaks(y1solPeak, distance=2,height=0.1)
    Y2peaks, _ = find_peaks(y2solPeak, distance=2,height=0.1)
    plt.plot(t_eval_pts[Xpeaks] ,xsolPeak[Xpeaks],'r--')
    plt.plot(t_eval_pts[Y1peaks], y1solPeak[Y1peaks],'b:' )
    plt.plot(t_eval_pts[Y2peaks], y2solPeak[Y2peaks], 'g-.')
    plt.xlabel('Time ')
    plt.ylabel('Peaks ')
    plt.xlim([0, tMax])
    #plt.ylim([0,  100])
    plt.legend( ('$x$','$y_{1}$','$y_{2}$'), shadow=True)
    plt.title(r' $\delta_{1}<\frac{1}{a}\quad  and\quad \delta_{2}<\frac
{b}{a}\quad $'+'      '
              +'$x_{0}=$' +str(list(np.round(InitVals[0],3))))
    plt.tight_layout()
    #plt.savefig('NumericalSimulation\Timeseries-(chaotic2Peaks).png', dp
i=300, bbox_inches='tight')

#Third plot Phase portrait
    fig3=plt.figure(figsize=(10,5))
    ax = plt.axes(projection="3d")
    for k,j in zip(range(len(InitVals)),colors1):
        x,y1,y2=sol[k].y
        ax.plot(x,y1,y2,j, lw=1.2)
    #plot equilibrium points
    ax.scatter(E[:,0],E[:,1],E[:,2],'c', lw=1)
    ax.set_xlim([0, np.mean( x)+70])
    ax.set_ylim([0,  np.mean( y1)+10])
    ax.set_zlim([0,  np.mean( y2)+100])
    ax.set_xlabel('$x$ ',labelpad=20)
    ax.set_ylabel('$y_{1}$ ',labelpad=20)
    ax.set_zlabel('$y_{2}$ ')
    #Change angle of view
    #ax.view_init(20,-90)
    #plt.draw()
    plt.tight_layout()
    #plt.savefig('NumericalSimulation\PhasePortrait-(chaotic2).png', dpi=
300, bbox_inches='tight')
    plt.show(); plt.close()

    return [t_eval_pts[Xpeaks][-10:],
```

```
            xsolPeak[Xpeaks][-10:],
            t_eval_pts[Y1peaks][-10:],
            y1solPeak[Y1peaks][-10:],
            t_eval_pts[Y2peaks][-10:],
            y2solPeak[Y2peaks][-10:],

            np.ediff1d(t_eval_pts[Xpeaks])[-10:],
            np.ediff1d(t_eval_pts[Y1peaks])[-10:],
            np.ediff1d(t_eval_pts[Y2peaks])[-10:],

            E,L,
            [sol[n].y[:,-1] for n in  range(InitVals.shape[0])]

            ]
```

In [ ]:

```
for psi in np.arange(0.3,0.4,0.01):
    SOL= SolPlot4([3.99256592e-04, 2.80065918e-01, 9.78851318e+00, 4.3309
4482e+00,
       psi,  4.10463217e+00, 7.37010376e-01, 1.67592651e-01,
       4.92309570e+03, 7.03787720e+02, 9.29061279e+01],1000,'RK45',1e-9,1
e-9)
    print('TidXpeaks=',SOL[0],'Xpeaks=',SOL[1],
      'TidY1peaks=',SOL[2],'Y1peaks=',SOL[3],
      'TidY2peaks=',SOL[4],'Y2peaks=',SOL[5],
       'Tx=',SOL[6], 'Ty1=',SOL[7],'Ty2=',SOL[8],

      'EqmPoints=',SOL[9],'Stability=',SOL[10],'XX=',SOL[11],sep='\n\n')
    print(psi)
```

Calculating Largest Lyapunov Exponent (LLE) by Sprott method

Refernce http://sprott.physics.wisc.edu/chaos/lyapexp.htm (http://sprott.physics.wisc.edu/chaos/lyapexp.htm)

In [ ]:

```
def LLE(pm,IC,IS,NumItr,Ntr,func,JAC,solvers,RelTol,AbsTol):
    #Ic: initial condition used
    # IS: Initial separation
    # pm: Set of parameter used (vector or list)
    # solvers: ODE solver used
    # RelTol, AbsTol: relative error and absolute error respectively
    # NumItr: the number of iterations
    #Ntr: number of transient to ignore to guarantee that we are on the a
ttractor.
    # func: function that calculates the right hand side of the system.
    # JAC: function that calculates the Jacobian matrix (used in ODE solv
er) (matrix form)
    x,y1,y2= IC
    x1,y11,y21=[x+IS,y1+IS,y2+IS]
    lam=np.zeros(NumItr)
    lle=np.zeros(NumItr)
    for i in range(0,NumItr):
        InitVals=np.array([[x, y1, y2],
                       [x1,y11,y21]])
        t0=0
        tMax=1
        t=[t0,tMax]
        t_eval_pts=np.linspace(t0,tMax,2*tMax)

        sol= [solve_ivp(func,t,init,method= solvers,t_eval= t_eval_pts,
              args=(pm,),jac=JAC,rtol=RelTol,atol=AbsTol) for init in Ini
tVals];
        X, Y1, Y2 = sol[0].y
        X1, Y11, Y21 = sol[1].y

        d0=LA.norm( InitVals[1]- InitVals[0])
        d1=LA.norm(sol[1].y[:,-1]-sol[0].y[:,-1])
        lam[i]=np.log(abs(d1/d0))

       #reinitialization of the trajectories after each iteration
        #such that their seperation remains as IS
        x1=X[-1]+(X1[-1]-X[-1])*(d0/d1)
        y11=Y1[-1]+(Y11[-1]-Y1[-1])*(d0/d1)
        y21=Y2[-1]+(Y21[-1]-Y2[-1])*(d0/d1)
        x=X[-1]
        y1=Y1[-1]
        y2=Y2[-1]
        t0=tMax
        tMax=tMax+1
        lle[i]=np.mean(lam) # save the mean after each iteration to check
convergence
    LLE=np.mean(lam[Ntr:NumItr]) # we sum from Ntrto be on the attractor
    #plot the last 300 lle
    plt.figure(figsize=(10,5))
    plt.plot(range(0,NumItr),lle)
    plt.xlabel('Time')
    plt.ylabel('Largest Lyapunov Exponent')
```

```
    plt.xlim(len(lam)-300,len(lam))
    plt.show
    return lle[-1],LLE
```

In [ ]:

```

```

Calculate the trace of the Jacobian matrix for an orbit

to check if the system is dissapative (If the trace is negative, then the system is dissapative). Also to compare it with the sum of lyapunov exponents

In [ ]:

```
def MeanTraceJac(pm,IC,tMax,func,JAC,solvers,RelTol,AbsTol):
    t=[0,tMax]
    t_eval_pts=np.linspace(0,tMax,300*tMax)
    init=IC
    sol= solve_ivp(func,t,init,method= solvers,t_eval= t_eval_pts,args=(p
m,),jac=JAC,rtol=RelTol,atol=AbsTol)
    x,y1,y2=sol.y
    TrJAll=[]
    for i,j,k in zip(x,y1,y2):
        x=i
        y1=j
        y2=k
        TrJ=np.trace(JAC(0,[x,y1,y2],pm))
        TrJAll.append(TrJ)
    TrJMean=np.mean(np.array(TrJAll))
    return TrJMean, TrJAll[-10:]
```

Lyapunov exponent for the chaotic paraneter set

In [ ]:

```
LE= LLE(chaotic, [10.2   ,  0.416,   2.207],1e-08,100000,1000,DM_bacteria,D
M_jac,'Radau',1e-9,1e-9)
```

In [ ]:

```
LE
```

Lyapunov exponent for the periodic paraneter set periodic 1 and 2

In [ ]:

```
LEperiodic1= LLE(periodic[0][0:8], [6.67, 0.35, 2.44],1e-08,100000,1000,D
M_bacteria,DM_jac,'RK45',1e-8,1e-8)
```

In [ ]:

```
LEperiodic1
```

In [ ]:

```
LEperiodic2= LLE(periodic[1][0:8], [11.155,  0.391,  3.893],1e-08,100000,
1000,DM_bacteria,DM_jac,'Radau',1e-9,1e-9)
```

In [ ]:

```
LEperiodic2
```

Test with Lorenz System

In [ ]:

```
def Lorenz(t,X,pm):
    x,y,z=X
    SIGMA,R,BETA=pm
    f1=SIGMA*(y-x)
    f2=-x*z+R*x-y
    f3=x*y-BETA*z
    return(f1,f2,f3)

def Lorenz_Jac(t,X, pm):
    SIGMA,R,BETA=pm
    LzJac=np.array([
        [-SIGMA, SIGMA,     0],
        [R-z,    -1,    -x],
        [y,     x, -BETA]])

    return LzJac
```

In [ ]:

```
LE_Lz1=LLE([10,28,8/3], [0  ,  1,  0],1e-08,100000,1000,Lorenz,Lorenz_Jac
,'RK45',1e-8,1e-8)
```

In [ ]:

```
LE_Lz1
```

In [ ]:

```
LLE_Lz2= LLE([16,45.29,4], [0  ,  1,  0],1e-08,100000,1000,Lorenz,Lorenz_Jac,'RK45',1e-8,1e-8)
```

In [ ]:

```
LLE_Lz2
```

Refernce values for lorenz systems

Reference values for t=10 000 (Lorenz):

InitialCondition=[0 , 1, 0]

$\sigma = 10, R = 28, \beta = 8/3$

$L_1 = 0.9056, L_2 = 0, LE_3 = -14.5723$

Reference values for t=10 000 (Lorenz):

InitialCondition= not known

$\sigma = 16, R = 45.29, \beta = 4$

$\lambda_1 = 2.16, \lambda_2 = 0, \lambda_3 = -32.4$ base 2 logaritm

$\lambda_1 = 1.5, \lambda_2 = 0, \lambda_3 = -22.46$ base e logaritm

Calculated values by the function LLE

First set of parameters of Lorenz

LLE=0.9064129395130333, t=100000,IC=[0 , 1, 0]

Second set of parameters of Lorenz

LLE=1.4902486394652161,t=100000,IC=[0 , 1, 0]

In [ ]:

```

```

Life is a chaotic system, and we can't predict it for more than a few seconds. So enjoy your life with being caring and sharing.

Farhan Omar