Oscillating Chemical Reactions

Searching for Periodic Behaviour in the Brusselator and Oregonator Models

by

Hanne Rachel Buit

to obtain the degree of Bachelor of Science

at the Delft University of Technology.

Student number:	5403111	
Project duration:	22nd of April, 2024 – 28th of June, 2024	
Thesis committee:	Ir. Z.G.J. Gromotka,	TU Delft, supervisor
	Dr.ir. D. den Ouden-van der Horst,	TU Delft

Delft
University of
Technology

Layman’s Abstract

This thesis explores how certain chemical reactions can show repeating patterns, called oscillations, by using two well-known models: the Brusselator and the Oregonator. First, we checked and confirmed previous findings on the Brusselator model, showing that under certain conditions, the chemicals involved can cycle in a stable, repeating way. We then extended the model to include a setup where chemicals are continuously added, making it more realistic. Using computer simulations, we found the specific conditions needed for the oscillations to occur. These varying conditions include, for example, the speed of the chemical reactions or the starting concentration of certain molecules. Lastly, we looked at the Oregonator model, proving again that repeating cycles can happen and identifying the exact conditions for these cycles. In all cases, we discovered the conditions under which the chemical reactions would display these periodic patterns.

Abstract

This thesis aims to identify the parameter combinations necessary for the Brusselator and Oregonator models to exhibit oscillating behavior. Initially, the Brusselator model is reviewed, reproducing the results from [1]. Using Bendixson's Criterion and the Poincaré-Bendixson Theorem, the existence of stable limit cycles is proven, and the Hopf-Bifurcation locations are derived analytically. The Brusselator model is then extended from a batch reactor to a Continuous Stirred-Tank Reactor to include the inflow of two components previously considered constant. An extensive eigenvalue investigation in Matlab is conducted to determine the parameter combinations that induce oscillating behavior, with Hopf-Bifurcation locations presented in a three-dimensional plot. Lastly, the Oregonator model is introduced and analysed. The existence of limit cycles is proven using similar methods, and an eigenvalue analysis yields the analytic expressions for the Hopf-Bifurcation locations. Periodic solutions are found in all three models, and the necessary parameter combinations are identified.

Nomenclature

Abbreviations

BZ	Belousov-Zhabotinsky
CSTR	Continuous stirred-tank reactor

Symbols

$[A]$, A	Concentration of molecule A [mol L^{-1}] or [M]
v_i	Reaction rate of a chemical reaction i[mol $L^{-1}s^{-1}$] or [$M\ s^{-1}$]
k_i	Rate constant of a chemical reaction i [variable]
f	Stoichiometric coefficient [-]
t	Time [s]
μ	Varying parameter [-]
∇	Divergence [-]
Q	In- and outflow rate [s^{-1}]
T	Trace of a Jacobian matrix [-]
D	Determinant of a Jacobian matrix [-]
S_i	Sub-determinant of a Jacobian matrix [-]

Contents

1

Introduction

In 1980, the two chemists B. P. Belousov and A. M. Zhabotinsky won the prestigious Lenin prize for the discovery of oscillating chemical reactions. This oscillating behaviour in chemical reactions was long thought to be impossible because of the second law of thermodynamics, but Belousov and Zhabotinsky turned the tide and convinced the world of the idea of oscillating chemical systems. Out of their research a very complex theoretical chemical model surfaced: the Belousov-Zhabotinsky reaction mechanism. To simplify the analysis of this mechanism, the essence of this system of reactions was captured in the *Oregonator* model, a system of five chemical reactions displaying oscillating behaviour. Another theoretical chemical system introduced around that time is the *Brusselator* model, which comprises only four reactions capable of generating oscillating behaviour.

The Brusselator and Oregonator models are both very abstract versions of oscillating chemical reactions, making them theoretical models. The aim of this thesis is to first establish both the models as such and look for oscillating behaviour. This will be done by transforming the models into systems of differential equations and evaluating the dynamics of these systems. This is done using the theoretical works of bifurcation theory, the analysis of the behaviour of such a dynamic system. With this theory it is possible to find the necessary conditions for oscillating behaviour. The models take place in a batch reactor, essentially a tank where every component is put into. In practice, it is more realistic to evaluate these chemical systems in another form of reactor: the continuous stirred-tank reactor (CSTR). In this reactor it is possible to add and remove components at all times, essentially creating an inflow and outflow of fluids. The Brusselator will be modelled in a CSTR after having been modelled in a batch reactor.

The objective of this thesis is to look for oscillating behaviour in two different theoretical chemical oscillators, while enhancing the realism of the models. In each chapter the model will be introduced with its assumptions, followed by the creation of a system of differential equations. The system will be analysed and an eigenvalue investigation will be conducted to find the conditions under which the system exhibits oscillating behaviour. When possible, a proof is carried out to demonstrate the existence of periodic behaviour in the models.

This report is structured as follows. First, in Chapter 2 an introduction on chemical oscillators is given, as well as a short introduction on bifurcation theory with an emphasis on Hopf-Bifurcations. Secondly, in Chapter 3 the Brusselator model is introduced and a proof is given that there exists oscillating behaviour. An extension of the model is given in Chapter 4, where the chemical reactions take place in the CSTR. This is a more realistic set up of the Brusselator model. In Chapter 5 another theoretical oscillator is introduced and analysed: the Oregonator. The same structure of proof is done on the Oregonator. Lastly, in Chapter 6 conclusions and recommendations for further research are given.

2

Background

Before we can dive into the mathematical modelling, there is some background information needed. To model the behaviour of chemical reactions, we will use differential equations. In this chapter you will find a short explanation on how to transform chemical reactions into a system of differential equation. You will also find a short introduction to bifurcation theory, the analysis of the behaviour of a dynamic system.

2.1. From Chemical Reactions to Differential Equations

A chemical reaction describes how reactants are transformed into products. It transcribes the ratio of needed reactants to generate a certain amount of products. The ratios are described in moles, a unit of measuring the amount of a substance containing a fixed number of particles. You can compare this to the idea of a dozen. In a dozen fit exactly twelve pieces, no matter the size or the weight of these pieces. In the chemical reaction stated below (the combustion of methane), the molar ratio tells us that if we have 1 mole of CH_4, we need to add 2 moles of O_2 to create 1 mole of CO_2 and 2 moles of H_2O.

$$\mathrm{CH_4} + 2\,\mathrm{O_2} \longrightarrow \mathrm{CO_2} + 2\,\mathrm{H_2O} \tag{2.1}$$

These numbers stating the molar ratio are called the *stoichiometric coefficients*. Stoichiometry describes the ratio between the reactants and the products of a chemical reaction. This relationship is based on the law of conservation of mass.

Definition 2.1 (Law of Conservation of Mass) *In a closed system mass is neither created or destroyed, thus the total mass of the reactants equals the total mass of the products [2].*

Because of this law, stoichiometric coefficients will always be positive integers. Stoichiometry states that the number of atoms of a substance A should be the same on both sides of the reaction. The composition in which the atoms exist can change (i.e. C-atoms first exist in CH_4 and then in CO_2).

When modelling a chemical system with differential equations, you model the change in concentration of the molecules in time. The concentration is the amount of moles of a substance per reaction volume, (mol/L). Meaning that if the volume of the reaction stays the same throughout your model process, you measure the components in moles. As the differential equations depend on time, we need some way to know how fast the chemical reaction takes place. For this we use a characteristic of chemical reactions: the rate constant k, which quantifies the speed of the reaction in relation to the concentration of reactants. This rate constant k is used in the reaction rate v, the speed of the reaction. The reaction rate v describes at what speed the reactants are transformed into products (mol/L/s) for a given reaction. How the reaction rate is set up and is determined, depends on the chemical reaction it is a characteristic of.

To model the concentration, we need to know how fast the concentration changes in time. The law of mass action states the following:

Definition 2.2 (Law of Mass Action) *The rate of a chemical reaction is proportional to the concentrations of the reactants.*

Meaning that the reaction rate not only gives us the speed of the reaction but also the speed at which the concentrations of the reaction components change. Now that we know how the reaction rate and the concentrations relate to each other, we can look at how to determine the reaction rate.

In general, $v = k[A]^x[B]^y$... where k is the rate constant, [A] and [B] are the concentrations of the reactants, and x and y the stoichiometric coefficients. The unit of rate constant k depends on the chemical reaction. Lets say we want to have the reaction rate of reaction (2.1). We have two reactants with different stoichiometric coefficients. Suppose the rate constant of this reaction is $\hat{k}$. Then $v = \hat{k}[CH_4][O_2]^2$. The unit of concentrations is mol/L, which is also noted as M (the molar concentration or molarity). The unit of $\hat{k}$ depends on the other concentrations in the reaction rate expression. In this case $\hat{k}$ has to navigate three molar concentrations per time unit, resulting in $[M^{-2}s^{-1}]$.

It is possible for chemical reactions to be reversible, meaning that there is not only a rate constant from reactant to product, but also the other way around. The reaction then has k as rate constant for the forward production and k_r for the backward production.

2.1.1. Chemical Oscillators

First some background on the type of model we are going to look at. These models are called chemical oscillators, meaning that the concentrations of the components in the model behave in an oscillating way. This oscillating behaviour is also called periodic behaviour.

It was long thought that it was not possible for chemical reactions to showcase oscillating behaviour. To explain why, we first need to state the second law of thermodynamics. For this law we need the definition of entropy:

Definition 2.3 (Entropy) *Entropy is the degree of randomness in a system.*

In a chemical system you can see the entropy of the system as how random the components in your system are distributed. The entropy is a sort of measurement for the chaos in a system. In the figure below you first see two components, blue and red, in a reversible reaction. There is not much randomness here yet. Once the components start mixing and transforming partly into the purple component, the randomness starts increasing. As the two components mix, more chaos ensues. Everything is mixed together, nothing is sorted anymore: ultimate chaos. This is the entropy of the system.

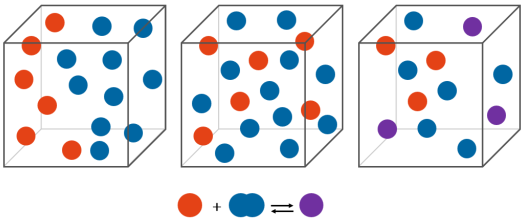


Figure 2.1: A chemical reaction approaching its equilibrium state.

The second law of thermodynamics tells us that the chronological order of the images in Figure 2.1 is possible, but the other way around is not.

Definition 2.4 (Second Law of Thermodynamics) *The total entropy of a system never decreases in any process. It either increases or stays constant.*

This law states that the entropy of a system *cannot* become smaller. So, if you have achieved a certain level of randomness you can't reverse this. For a chemical reaction this means that once the components have reached an equilibrium state, you can't turn back. Periodic behaviour of concentrations of chemical components would entail that the reaction deviates from its equilibrium state and thus has a decrease in entropy. This should not be possible.

In 1917, two chemists W. C. Bray and A. L. Caulkins discovered periodic production of oxygen under very specific conditions. They cautiously proceeded their research as to how this was possible, with the second law of thermodynamics in the back of their mind. Their discovery was met with a lot of scepticism and their result was questioned for years. In 1951 another researcher, B.P. Belousov, discovered a second oscillating reaction. He too was questioned. A.M. Zhabotinsky heard of his work and began studying the newfound reaction.

He discovered that oscillating behaviour *is* possible in a system of chemical reactions. Instead of just one reaction, a cascade of different simple reactions makes up a system of reactions. In this system it is possible that the concentrations of some intermediate reactions oscillate. In 1964 Zhabotinsky published a groundbreaking paper, after which the idea of a chemical oscillator was finally accepted. Unfortunately, Belousov died 10 years before getting awarded the Lenin prize in 1980, the Nobel Prize of the Soviet Union at the time [3]. But their work paid of, there are now multiple great examples of chemical oscillators.

One of the most famous chemical oscillators now is the Belousov-Zhabotinsky (BZ) reaction itself. It is a family of reactions describing this oscillating behaviour. The BZ mechanism is very complex and thus very hard to analyse, it contains over 18 different steps. To simplify the system, the FKN mechanism was introduced by Field and Noyes [4]. This mechanism still constist of 11 reactions. To simplify the mechanism even more, the Oregonator was introduced. Its name is a blend of Oregon and oscillator, as it was found at a university in Oregon.

The BZ reaction is an autocatalytic reaction [5]. Meaning that one of the reaction products is also a catalyst for this reaction. A catalyst is a substance added to a chemical reaction that makes is proceed faster, without being consumed itself. To see this autocatalytic behaviour, we can use the Oregonator. The Oregonator is characterized by the following equations [6]:

$$\begin{aligned} \mathrm{A}+\mathrm{Y} &\xrightarrow{\mathrm{k}_1} \mathrm{X}+\mathrm{P}, \\ \mathrm{X}+\mathrm{Y} &\xrightarrow{\mathrm{k}_2} 2\,\mathrm{P}, \\ \mathrm{A}+\mathrm{X} &\xrightarrow{\mathrm{k}_3} 2\mathrm{X}+2\mathrm{Z}, \\ 2\mathrm{X} &\xrightarrow{\mathrm{k}_4} \mathrm{A}+\mathrm{P}, \\ \mathrm{B}+\mathrm{Z} &\xrightarrow{\mathrm{k}_c} \frac{1}{2} f\,\mathrm{Y}. \end{aligned} \tag{2.2}$$

A, B, P, X, Y and Z are all different components here. There are five reactions and thus five rate constants. In the fifth reaction something interesting is going on, f is a yet undetermined stoichiometric coefficient. You can see the autocatalytic characteristic in the third reaction, where component X is a reactant and a product of the reaction. X is used in the reaction but not consumed. As X is produced in an earlier reaction of this system, we have a component that is produced and used as a catalyst in the system.

Lastly, there is also the Brusselator. The name is a blend of Brussels and oscillator, as this reaction was discovered at a university in Brussels. The Brusselator is a very simple version of system still showcasing oscillating behaviour, also making it a very unrealistic model. The Brusselator is another great example of an autocatalytic system. It is characterized by the following equations [1]:

$$\begin{aligned} \mathrm{A} &\xrightarrow{\mathrm{k_1}} \mathrm{X}, \\ \mathrm{B}+\mathrm{X} &\xrightarrow{\mathrm{k_2}} \mathrm{Y}+\mathrm{D}, \\ 2\mathrm{X}+\mathrm{Y} &\xrightarrow{\mathrm{k_3}} 3\mathrm{X}, \\ \mathrm{X} &\xrightarrow{\mathrm{k_4}} \mathrm{E}. \end{aligned} \tag{2.3}$$

Here we have A, B, D, E, X and Y as components, with four rate coefficients (differing from those in the Oregonator mechanism). In this system you see the use of self-produced component X as a catalyst in the third reaction, the Brusselator is indeed an autocatalytic system.

2.1.2. Transforming

Let's transform the Brusselator, as shown in (2.3), into a system of differential equations. Before converting the reactions, we have to decide which components we would like to keep track of. All other components are then assumed to be constants or to be redundant (e.g. too small to be of influence or present in such an abundance that a change in concentration does not have an effect). We will keep track of components X and Y, resulting in a system of two differential equations.

Before looking at a specific component yet, let's first determine the reaction rate for all four reactions, where v_i belongs to reaction number i, with here $i = 1,2,3,4$.

Reaction i	Reaction rate v_i	Rate coefficient k_i
$\mathrm{A} \xrightarrow{\mathrm{k_1}} \mathrm{X}$	$v_1 = k_1 A$	$k_1 : [s^{-1}]$
$\mathrm{B}+\mathrm{X} \xrightarrow{\mathrm{k_2}} \mathrm{Y}+\mathrm{D}$	$v_2 = k_2 BX$	$k_2 : [M^{-1}s^{-1}]$
$2\mathrm{X}+\mathrm{Y} \xrightarrow{\mathrm{k_3}} 3\mathrm{X}$	$v_3 = k_3 X^2 Y$	$k_3 : [M^{-2}s^{-1}]$
$\mathrm{X} \xrightarrow{\mathrm{k_4}} \mathrm{E}$	$v_4 = k_4 X$	$k_4 : [s^{-1}]$

Table 2.1: Reaction rates for all four reactions of the Brusselator.

When modelling the concentration of X, we need to look at all factors that influence X and take the system as a whole. We look at all factors decreasing and increasing the concentration of X. First, X is produced proportional to A, an increase in X substance. Then X is used to produce other components, entailing a decrease of X. The third reaction effectively produces X and the fourth reaction uses X to make E. The X concentration is then modelled as $\frac{dX}{dt}$ = total increase - total decrease.

Combining the reaction rates correctly and doing a similar thing for component Y, yields the following system of differential equations:

$$\begin{aligned} \frac{dX}{dt} &= k_1 A - k_2 BX + k_3 X^2 Y - k_4 X, \\ \frac{dY}{dt} &= k_2 BX - k_3 X^2 Y. \end{aligned}$$

There is quite some similarity between the two equations, as the two components interact with each other. As X produces Y and Y produces X, common terms have opposing signs.

2.2. Introduction to Bifurcation Theory

A system of differential equations describes the change in time of multiple variables at once. If there is no change in all variables at a certain point, the system 'stands' still. This is called an equilibrium point, where the differential equations are equal to zero. It is possible to have no equilibrium points, have one, or have multiple. An equilibrium point is sometimes also called a fixed point.

Definition 2.5 (Equilibrium Point) *Consider the n-dimensional system* $\dot{\boldsymbol{x}} = F(\boldsymbol{x}, \mu)$ *with* $\mu \in \mathbb{R}$ *and* $\boldsymbol{x} \in \mathbb{R}^n$*. Then* $\boldsymbol{x} = \boldsymbol{x}(\mu)$ *is an equilibrium point if* $F(\boldsymbol{x}(\mu), \mu)) = \boldsymbol{0}$*. [7]*

An equilibrium point can be attracting, meaning that the differential equations will eventually arrive at this point of no change. The solutions (a sort of walk starting from a certain point in the plane of the variables) starting near this point will move to the equilibrium point, it attracts the solutions. This is called a stable point. An equilibrium point can also be repelling, meaning that the solutions starting near the point will move away from the point. This is an unstable point, as all solutions that start there will immediately leave there.

If you have a non-linear system of differential equations it is very hard to see whether the system will approach the equilibrium point or if it will leave the point. To make this easier, we make use of the Jacobian matrix. The Jacobian matrix is the change of the system at every point.

Definition 2.6 (Jacobian Matrix) *The Jacobian matrix* $\boldsymbol{J}$ *is an m x n matrix with entries* $\boldsymbol{J}_{i,j} = \frac{\delta f_i}{\delta x_j}$*, yielding the following matrix:*

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

This matrix is the best linear approximation of the non-linear system. If we fill in the equilibrium point into the Jacobian matrix of the system, we get a good linear approximation of how the system behaves around this point. The Jacobian predicts the behaviour close to an equilibrium point of a non-linear system. To get this prediction, we need to compute the eigenvalues of the Jacobian evaluated at an equilibrium point.

Definition 2.7 (Eigenvalue) *Suppose n x n matrix A and nonzero vector* $\boldsymbol{u}$*. Then the scalars* λ_i*,* $i = 1, \ldots, n$*, are the solutions to* $A\boldsymbol{u} = \lambda \boldsymbol{u}$*.* λ_i *can be either real or complex.*

These eigenvalues of the Jacobian tell us how the solutions close to the equilibrium point behave, whether they grow (repelling point) or decay (attracting point) after time. Positive eigenvalues tell us it's an repelling equilibrium point and the solutions leave the point. Negative eigenvalues imply an attracting equilibrium point where the solutions move to the point.

A system of differential equations has a certain amount of fixed points, with the Jacobian matrix we can establish the stability of these points. Something interesting occurs when this system of equations depends on a parameter. When varying this parameter, fixed points can be destroyed or created, or the stability of them can change. If such a thing happens, this is called a bifurcation. We call the parameter values at which this occurs the bifurcation points. There are several types of bifurcations classified, where we are especially interested in the Hopf-Bifurcation. A Hopf-Bifurcation happens if a parameter is changed and as a result the stability of the system changes (from stable to unstable, or the other way around) and a periodic solution arises. In practice, a Hopf-bifurcation occurs if the eigenvalues of the system at this point cross the imaginary axis. Meaning that when the parameter varies, either $\mathrm{Re}\{\lambda\}$ goes from being negative to positive (crossing the imaginary axis from left to right) or $\mathrm{Re}\{\lambda\}$ goes from being positive to negative (crossing the imaginary axis from right to left).

Definition 2.8 (Periodic solution) *A solution $x(t)$ with $t \in \mathbb{R}$ of a system of differential equations is a periodic solution if there exists a $T \neq 0$ such that $x(t+T) = x(t)$, $T \neq 0$.*

Definition 2.9 (Hopf-Bifurcation) *A system of differential equations can encounter a Hopf-Bifurcation if the eigenvalues of the Jacobian, evaluated at an equilibrium point of said system, cross the imaginary axis. Meaning that the eigenvalues are of the form $\lambda_{1,2} = \pm bi$, where $b \neq 0$.*

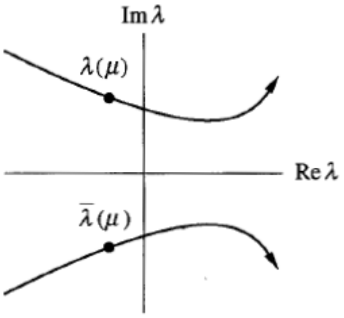


Figure 2.2: Eigenvalues crossing the imaginary axis from left to right. *Adapted from 'Nonlinear dynamics and chaos' by Strogatz, 1994, p. 251* [8].

To visualize the changes of the fixed point(s), you can make a bifurcation diagram. Here you plot your differential equation variables against the varying parameter. In the case of a Hopf-bifurcation it is often hard to visualize. A solution for this would be to make different plots for different stages, as done in the figure below. Where parameter μ is varied, resulting in two different cases.

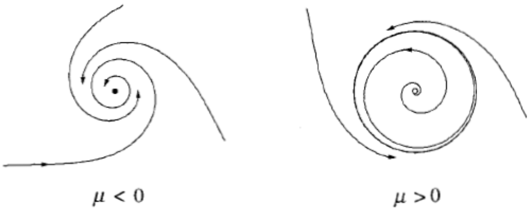


Figure 2.3: Behaviour of a system for different cases of μ values. *Adapted from 'Nonlinear dynamics and chaos' by Strogatz, 1994, p. 250* [8].

As you can see in the figure above, the equilibrium point seems to vanish. It did not vanish, but it did change from a stable to an unstable point. When $\mu < 0$, we have a stable equilibrium point, meaning that it attracts. The long term solution of the system will approach this point and eventually reach it. But then, when $\mu > 0$, the stabel equilibrium point seems to have disappeared and turned into a stable limit-cycle. The equilibrium point is now unstable and repels the solutions, with an attracting cycle around it. This cycle stems from the equilibrium point and grows bigger until a certain amplitude is reached. The cycle is attracting, meaning that on the outside and inside everything approaches the cycle. We have a stable cycle. This is called a supercritical Hopf-Bifurcation. If an unstable limit cycle surrounds a stable equilibrium point, it is a subcritical Hopf-Bifurcation.

A simple example of a system with a Hopf-bifurcation are the following equations:

$$\frac{dy_1}{dt} = -y_2 + y_1(\mu - y_1^2 - y_2^2),$$
$$\frac{dy_2}{dt} = y_1 + y_2(\mu - y_1^2 - y_2^2).$$

This system has an equilibrium point at $(y_1, y_2) = (0, 0)$, which yields the following Jacobian matrix, evaluated at the equilibrium point:

$$J_{eq} = \begin{bmatrix} \mu & -1 \\ 1 & \mu \end{bmatrix}.$$

Yielding that $\lambda_(1,2) = -\mu \pm i$. First, we have $\mu < 0$, where we get a stable solution, already looking quite spirally. The eigenvalues cross the imaginary axis if $\mu = 0$. When $\mu = 0$, the eigenvalues are purely imaginary, yielding a centre as solution. In Figure 2.4 you can see that the center solution does not reach the equilibrium point we had before. Then, when $\mu > 0$, the equilibrium point becomes unstable. Meaning that we have found a bifurcation, the behaviour of the system has changed. Now the cycles arise in the phase plots, as visible in Figure 2.4. These cycles are stable surrounding an unstable equilibrium point, meaning that we have supercritical Hopf-Bifurcations.

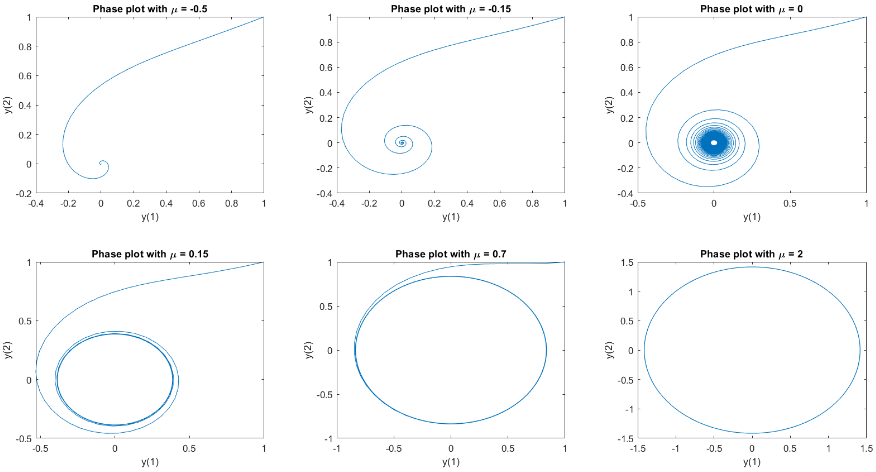


Figure 2.4: Behaviour of a system for different cases of μ values.

3

The Brusselator

In this chapter, we will create a model based on the Brusselator. This is a system of four chemical reactions, as shown below. Here, A, B, X, Y, D and E are all concentrations [mol m^{-3}] of certain components. k_1, k_2, k_3 and k_4 are the rate coefficients, determining the speed of the reaction. The dimension of rate coefficient k_i depends on reaction i. As there is only a rate coefficient from reactants to products, this means that the reactions are irreversible. Once a product has been made, there is no way back using the same reaction. It's worth noting that in the second reaction, Y is generated from component X, while in the third reaction, X is produced from Y. This reciprocal production could lead to oscillations.

$$\begin{aligned} A &\xrightarrow{k_1} X \\ B + X &\xrightarrow{k_2} Y + D \\ 2X + Y &\xrightarrow{k_3} 3X \\ X &\xrightarrow{k_4} E \end{aligned}$$

The Brusselator is a model in itself, as it is a theoretical model for autocatalytic chemical reactions. A chemical reaction is autocatalytic if one of the reaction products is also a catalyst for this same reaction. Meaning that this reaction product helps to increase the speed of the reaction, while remaining intact. You can see this in the Brusselator in the third reaction, where the component X is produced as well as used as reactant, while remaining intact.

In practise, these chemical reactions have to take place somewhere. This is done in a reactor, an enclosed space in which the reactions take place. In this chapter we will model the Brusselator in a batch reactor. In Figure 3.1 you see a schematic diagram of a batch reactor. In this reactor there is no possibility of putting in more chemicals or retrieving already made products until all reactions are finished. The reactor has a stirrer, which mixes all the present molecules. Because of the stirring in the reactor the components are distributed evenly over the reactor.

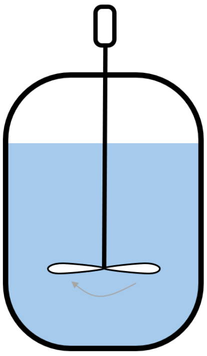


Figure 3.1: A batch reactor modelled.

3.1. The Model

To be able to see how the Brusselator behaves, we need a way to express the model and a way to implement it. We will model the Brusselator using a system of differential equations. But before we can start we need to establish some assumptions.

3.1.1. Assumptions

The following assumptions are made:

- A, B, X, Y, D, E are in $\mathbb{R}_{\geq 0}$. As A, B, X, Y, D and E are all concentrations, they cannot be negative and have continuous values.
- A and B are constants. In practise this means that there is an abundance present of A and B, making the change in concentration of A and B almost zero.
- D and E are not of influence. Because D and E are only products of the reactions and not used further, we do not consider them in the model.
- $k_1 = k_2 = k_3 = k_4 = 1$. Numerically we will not look at the influence of different k_i values, but assume they are all equal to 1. In practise this means that every reaction has the same rate, meaning that all reactions produce their products at the same speed. In Section 3.2 we will drop this assumption and work with nondimensional parameters instead.
- All components are homogeneously distributed over the reaction volume. Because of this we do not have to consider where a reaction takes places and we can eliminate the dimension of space. The stirrer in the reactor distributes all components evenly, making the concentration of all molecules the same over the reactor.
- The reaction volume is constant. The batch reactor is a closed system, there is no in- or outflow. Because of the law of Conservation of Mass, as stated in Definition 2.1, the reaction volume will stay constant.

3.1.2. System of Differential Equations

Now that we have established our assumptions, we can start to model the Brusselator. First, we need to convert the chemical system into a system of differential equations. In section 2.1 you can see how to derive a set of differential equations based on chemical reactions. This method results in the following system of differential equations:

$$\begin{aligned} \frac{dX}{dt} &= k_1 A - k_2 BX + k_3 X^2 Y - k_4 X, \\ \frac{dY}{dt} &= k_2 BX - k_3 X^2 Y. \end{aligned} \tag{3.1}$$

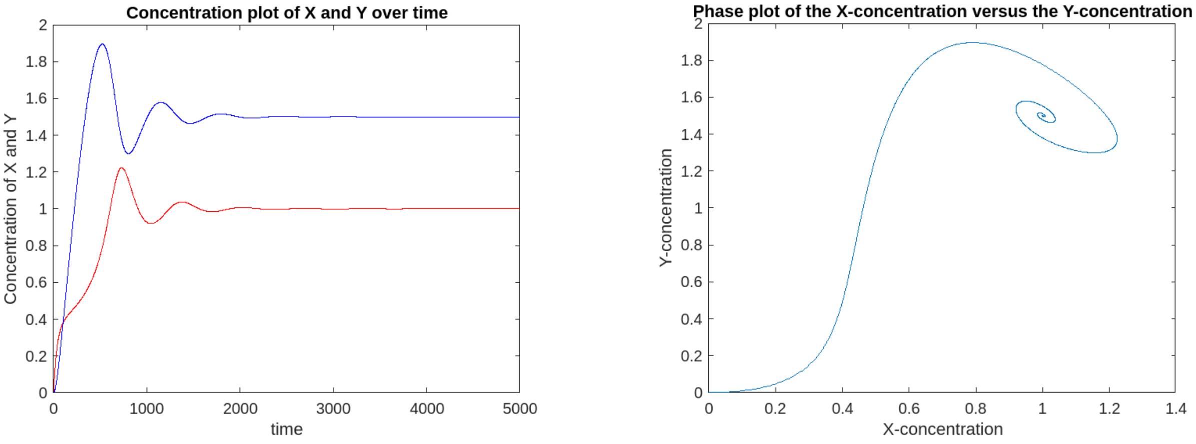
3.1.3. Behaviour

To see how the concentration changes in time, a numerical method is implemented in Matlab. The Matlab code can be found in appendix A.1. Here we set A and B as constants and see what happens with X and Y. We also set the starting conditions as X(0) = 0 and Y(0) = 0. We also assume $k_1 = k_2 = k_3 = k_4 = 1$ for the numerical method.

The numerical approximation of the model is done with the Forward Euler method, the simplest way of computing the concentrations at the next time step. We set time step $\Delta t = 0.01$ and use predetermined values for concentrations A and B. Meaning that with initial concentrations for x_0 and y_0 the implementation is the following:

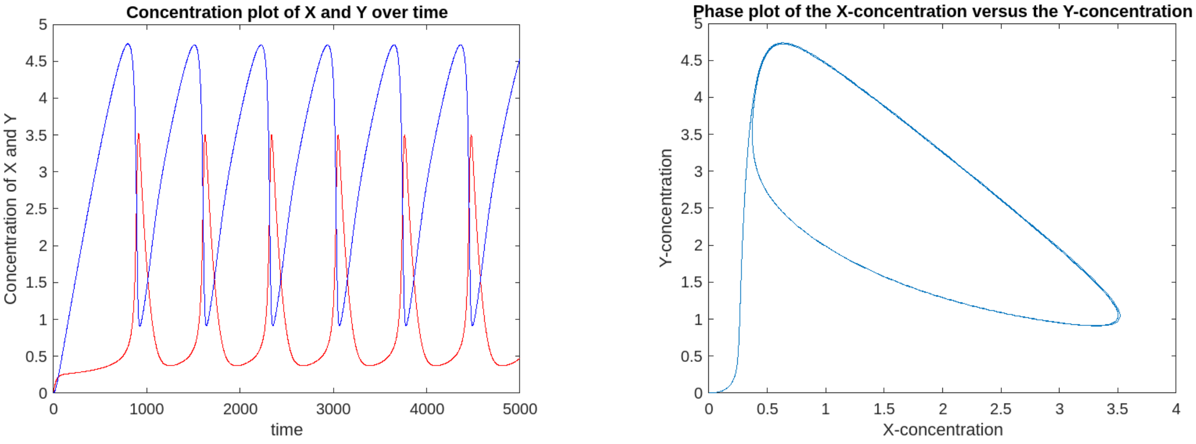
$$\begin{aligned} x_{n+1} &= x_n + \Delta t(k_1 A - k_2 B x_n + k_3 x_n^2 y_n - k_4 x_n), \\ y_{n+1} &= y_n + \Delta t(k_2 B x_n - k_3 x_n^2 y_n). \end{aligned}$$

From article [1] we know that if A = 1 and B = 3, the X and Y concentrations will be oscillating and thus creating a periodic solution. Here we see the concentrations of X and Y over time in one plot and the phase plot of X and Y in the other plot, for 2 different cases; the first case has A = 1 and B = 1.5. The second case has A = 1 and B = 3.



(a) Concentration plot with A = 1 and B = 1.5.

(b) Phase plot with A = 1 and B = 1.5.

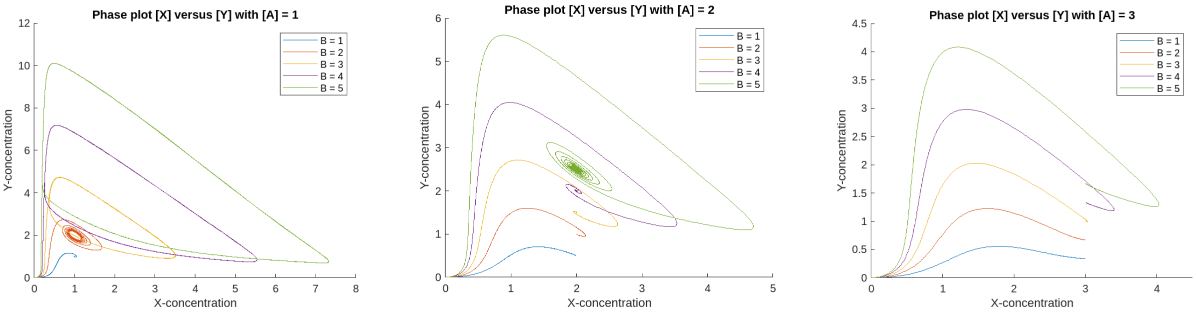


(c) Concentration plot with A = 1 and B = 3.

(d) Phase plot with A = 1 and B = 3.

Figure 3.2: Plots for two different cases.

The specific case of A = 1 and B = 3 shows indeed that there is a periodic solution. The concentration plot is oscillating very clearly and the phase plot shows an inexhaustive cycle. An interesting question arises: when do these cycles appear? If we plot phase plots for different combinations of A and B concentrations we see that such a cycle is quite rare, as they only seem to arise in the case we specified earlier.



(a) Phaseplot with fixed A = 1.

(b) Phaseplot with fixed A = 2.

(c) Phaseplot with fixed A = 3.

Figure 3.3: Phaseplots of X and Y with varying B concentrations and a fixed A concentration.

To answer the question of when certain behaviour arises, we need to analyse the system more rigorously.

3.2. Analysis

Before we can analyse the model, we would like to simplify it. After nondimensionalization, details to be read in [1], we end up with the following simplified system:

$$\begin{aligned} \frac{dX}{dt} &= \dot{X} = Q - (P+1)X + X^2 Y, \qquad & Q &= \sqrt{\frac{k_1^2 k_3}{k_4^3}} A, \\ \frac{dY}{dt} &= \dot{Y} = PX - X^2 Y, & P &= \frac{k_2}{k_4} B. \end{aligned} \tag{3.2}$$

We now have reduced the amount of parameters from six (k_1, k_2, k_3, k_4, A and B) to two parameters (P and Q). To analyse the stability of the model, we first determine where the fixed points are. When equating the differential equations to zero, we find one fixed point: $(X, Y) = (Q, \frac{P}{Q})$. To say something about the stability, we need the eigenvalues of the Jacobian matrix, evaluated at the found equilibrium point. The Jacobian of equation (3.2) is given by:

$$J = \begin{bmatrix} -(P+1) + 2XY & X^2 \\ P - 2XY & -X^2 \end{bmatrix}.$$

Now, we evaluate the Jacobian at the found equilibrium point $(X, Y) = (Q, \frac{P}{Q})$, this results in:

$$J_{eq} = \begin{bmatrix} P-1 & Q^2 \\ -P & -Q^2 \end{bmatrix}. \tag{3.3}$$

To get the eigenvalues of this Jacobian, we can use the following expression for the characteristic polynomial of a 2 x 2 matrix J:

$$p(\lambda) = Det(J) - \lambda * Trace(J) + \lambda^2.$$

If we solve $p(\lambda) = 0$, we get the following expression for the eigenvalues:

$$\lambda_{1,2} = \frac{1}{2}(Trace(J) \pm \sqrt{\Delta}), \quad \text{where} \quad \Delta = Trace(J)^2 - 4 * Det(J). \tag{3.4}$$

From this expression of the eigenvalues we can derive a general way to determine the stability of a fixed point and a way to determine a Hopf-Bifurcation possibility. We can conclude the following two conditions:

- $\lambda_{1,2} \in \mathrm{C}$ if $\Delta < 0$; in expression (3.4) you see a square root. If the value in the square root is negative, we get complex eigenvalues.
- a fixed point is stable if $Trace(J_{eq}) < 0$; If the eigenvalues of the Jacobian, evaluated at a specific fixed point, are all negative, the fixed point is stable.

This is especially interesting when analysing the behaviour of the whole system. After nondimensionalisation, we only have P and Q as parameters left. Now it would be interesting to see the changes of the system when we vary the values of P and Q. Looking at the system like this, is done in *bifurcation theory*. For some background information, see section 2.2. To analyse the system using expression (3.4), we need the trace and the determinant of the Jacobian evaluated at the equilibrium point ((3.3)). We have $Trace(J_{eq}) = P - 1 - Q^2$ and $Det(J_{eq}) = Q^2$.

3.3. Proof of Oscillating Behaviour

Before trying to find oscillating behaviour, one can confirm that this behaviour is possible. For this we will introduce Bendixson's Criterion. This criterion tells you when periodic behaviour *cannot* occur. Because we have varying parameters in our system, this can be very interesting. The criterion can tell us for which parameter values there is a possibility of oscillating behaviour. Bendixson's Criterion is stated as follows:

Theorem 3.1 (Bendixson's Criterion) *Consider the system* $\dot{x} = g(x,y)$, $\dot{y} = h(x,y)$ *in* $\mathbb{R}^2$. *Suppose D is a simply connected subset of* $\mathbb{R}^2$ *and (g,h) is continuously differentiable in D. If* $div(g,h) = \nabla(g,h) = \frac{dg}{dx} + \frac{dh}{dy}$ *changes sign in region D or if* $\nabla(g,h) = 0$, *only then can the system have periodic solutions. [9]*

The Bendixson's Criterion uses the divergence of a system. The divergence represents the flux of a system, the flux of a system can be described as the movement of the system at every point. If a solution approaches an attracting equilibrium point, the flux towards this point is positive: more movement is going towards the point than there is movement leaving the point. For an unstable equilibrium point this would be the other way around. If the divergence of a system on a certain domain D is equal to zero, there are two possibilities: either there is no movement going in or out of the domain, every solution that is in the domain stays there, meaning that there should be a limit cycle. Or exactly the amount leaving the domain is also the amount of movement going into the domain. With the Bendixson's Criterion you can conclude that if the divergence is equal to zero either of these cases is true. Which is why you can only dismiss regions for which this is not true and cannot conclude a definite case of limit cycles, as there is always still the option of having the same flux into the domain as out of it.

Let us start applying the Bendixson's Criterion. Let's consider our nondimensionalised system again:

$$\begin{aligned} \frac{dx}{dt} &= \dot{x} = Q - (P+1)x + x^2 y, \\ \frac{dy}{dt} &= \dot{y} = Px - x^2 y. \end{aligned} \tag{3.5}$$

We will apply Bendixson's theorem to see whether there is a possibility of a periodic solution and if so, for which P and Q values. We will do this for D(x,y) = $\mathbb{R}_{\geq 0}$. We first need the divergence of the dynamic system. Let $g(x,y) = \dot{x}$ and $h(x,y) = \dot{y}$, then:

$$\nabla(g,h) = \frac{dg}{dx} + \frac{dh}{dy} = -(P+1) + 2xy - x^2. \tag{3.6}$$

Parameter Q is not represented in $\nabla(g,h)$, meaning that we cannot say anything about the values of Q with the Bendixson's Criterion. If $\nabla(g,h) = 0$, we know that there is a possibility for oscillating behaviour. Unfortunately, this is not the case. Now we need to investigate whether there occurs a sign change in $\nabla(g,h)$ in (3.6). Because we do not want to fix the P-value, we will have to do this in a more general way.

Let $\nabla(g,h) = -(P+1) + 2xy - x^2 = f(x,y)$. Where $f(x,y) = 0$, we know there could be a sign change around this set of points. Equating $f(x,y)$ to zero yields the following:

$$f(x,y) = 0 \rightarrow l : y = \frac{x^2 + P + 1}{2x}.$$

In Figure 3.4 you see line *l* where $f(x,y) = 0$ with P-value 10. Because x an y represent concentrations in the model, we only look at the first quadrant. We now need to show that the points crossing *l* switch signs. We'll show that the points under the line are all negative and all the points above the line are positive.

Because $x, y, P \geq 0$, we know that $(x, y) = (0,0)$ will always be under line l. Filling in this point into (3.6) tells us that $f(0,0) = -P-1$. This value is always negative. So, the points under l have negative values.

Now, to show that the value above line l are positive, we construct a new line h. This line is line l, but shifted two to the right and two upwards. Meaning that line h is always above line k.

$$y = \frac{(x-2)^2+P+1}{2(x-2)} + 2 \rightarrow h : y = \frac{x^2+P+1}{2x-4}$$

As h also depends on parameter P we can use this line to show that the points above line k are positive. Let's look on the line where $x = 4$, then $h : y(4) = 4 + \frac{1}{4}(P+1)$. We now have coordinates still depending on P to fill into equation (3.6), this yields: $f(4, 4+\frac{1}{4}(P+1)) = P+1$. This value is always positive, proving that points above line k have positive values.

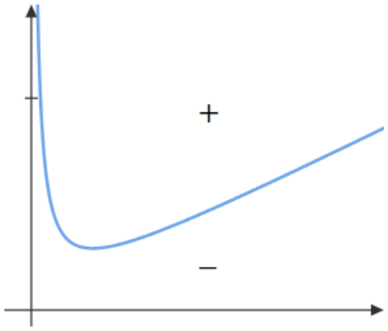


Figure 3.4: f(x,y) = 0 with P = 10.

We have now shown that there always is a sign change in $\mathbb{R}^2_{\geq 0}$. From Bendixson's Criterion we can conclude that a periodic solution is *possible*.

Now that we know oscillating behaviour is a possibility, we can prove that it exists. To do this, we will use the Poincaré-Bendixson theorem. The idea of this theorem is to 'trap' your system of equations in a certain region and check if the solution of the system stays in this *trapping region*. If you have oscillating behaviour, you will encounter a limit cycle. If such a cycle is in the trapping region, it cannot leave the trapping region. Important to note is that there cannot be an equilibrium point in this trapping region. If an attracting equilibrium point would be in the trapping region and you show that all solutions in the trapping region stay there, you have shown that the attracting equilibrium point has attracted all solutions. This would not proof the existence of a stable limit cycle.

Theorem 3.2 (Poincaré-Bendixson) *Suppose that:*

1. *R is a closed, bounded subset of the plane;*
2. $\dot{x} = \boldsymbol{f}(\boldsymbol{x})$ *is a continuously differentiable vector field on an open set containing R;*
3. *R does not contain any fixed points, and*
4. *There exists a trajectory C that is 'confined' in R; C starts in R and stays in R for all future time.*

Then, either C is a closed orbit, or it spirals toward a closed orbit as $t \rightarrow \infty$. *In either case, R contains a closed orbit. [8]*

To prove that there exists a confined trajectory C, we will construct this trapping region R. We will show that the vector field on the boundary of R always points inward, ensuring that all trajectories in R are confined. If we remove any fixed points from region R, attracting solutions, we ensure that there is a closed orbit in R.

Consider again our nondimensionalised system (3.5). To apply the Poincaré-Bendixson theorem, we first need to construct the trapping region R. We do this by determining the nullclines, as defined below, for system (3.5):

Definition 3.3 (Nullclines for a 2d System) *A nullcline is a line of points where the vector field changes direction along one vector component, the places where the slope of that directional vector component is equal to zero. In a two dimensional system of two differential equations:*

$$\frac{dx}{dt} = g(x,y), \quad \frac{dy}{dt} = h(x,y)$$

We define two nullclines:

- ***x-nullcline:*** *the set of points where* $\dot{x} = 0$*. The vector field is vertical on this set of points.*
- ***y-nullcline:*** *the set of points where* $\dot{y} = 0$*. The vector field is horizontal on this set of points.*

The nullclines tell us where the vector field changes direction and gives us some directions of the vector field. Let us determine the expressions of the nullclines of system (3.5).

$$\dot{x} = 0 \rightarrow y = \frac{-Q + (P+1)x}{x^2} \tag{3.7}$$

$$\dot{y} = 0 \rightarrow y = \frac{P}{x} \tag{3.8}$$

On the x-nullcline (3.7) we have that the slope in the x direction is zero, meaning that the vector field is only vertical on this line. For the y-nullcline (3.8) we only have horizontal vectors in the vector field. All this information together gives us the following figure:

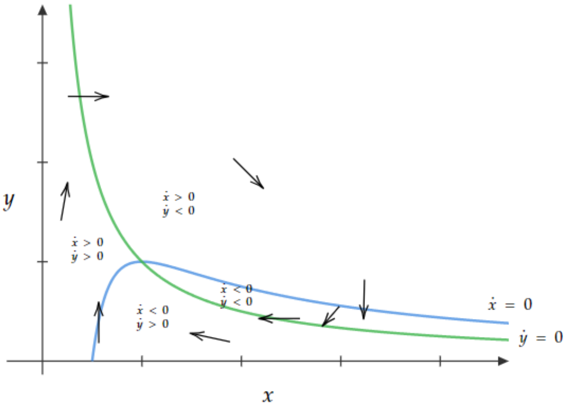


Figure 3.5: Nullclines of the system.

You can already see some hints to a trapping region in the vectors on the nullclines. Now consider the orange dashed line as in Figure 3.6 as a possible option for a trapping region. To show that this is true, we need to show that all vector field vectors on the boundary of this region point inward. We already have some information. In Figure 3.5 we firstly have two vertical and two horizontal vectors. Secondly, we know how the derivatives behave in the four areas divided by the nullclines. For example, under the green line ($\dot{y} = 0$) and above the blue line ($\dot{x} = 0$) we know that $\dot{x} \geq 0$ and $\dot{y} \geq 0$, meaning that the direction of the vectors here point upward right. For all four different areas this gives us the direction of the vector field.

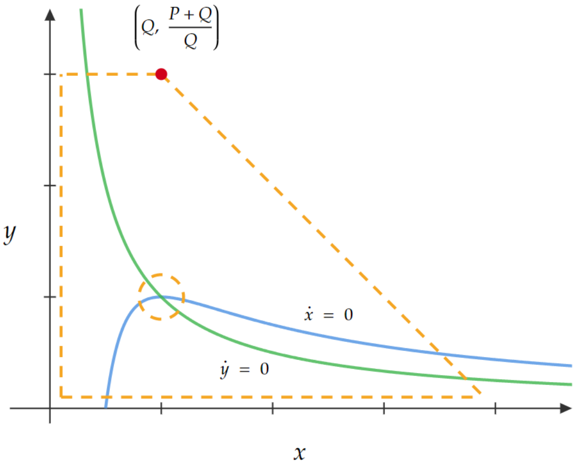


Figure 3.6: Trapping region of the system.

We know that all vectors point into the trapping region, except for the diagonal line from $(Q, \frac{P+Q}{Q})$ to the x-axis with slope -1. We want to show that the vector field here points inward too. For this, consider x very large. Then $\dot{X}$ and $\dot{Y}$ from (3.5) become $\dot{x} \approx x^2 y$ and $\dot{y} \approx -x^2 y$, with the direction $\frac{\dot{y}}{\dot{x}} = -1$. So, for large x we know that the vector field is almost parallel to the diagonal line of the trapping region. To actually know how parallel this is, we compare $\dot{x}$ and $-\dot{y}$. This to see the difference between the two equations, so we get: $\dot{x} - (-\dot{y}) = Q - (P+1)x + x^2 y + Px - x^2 y = Q - x$. Meaning that $-\dot{y} > \dot{x}$ if $x > Q$. For the whole of the diagonal line we have that $x > Q$, meaning that $-\dot{y} > \dot{x}$ and we thus have that $\frac{\dot{y}}{\dot{x}}$ is more negative than -1. Implying that the vector field points inward. Thus, we have constructed a trapping region!

One of the conditions of the Poincaré-Bendixson theorem is that there is no fixed point in the trapping region. Let there be a very small hole in the trapping region at the fixed point $(x, y) = (Q, \frac{P}{Q})$. If we have that this fixed point is an unstable point, repelling the solution, we have satisfied all conditions for the Poincaré-Bendixson theorem:

- All boundaries of the trapping region have inward pointing vectors, making sure that all trajectories stay in the region.

- There are no fixed points in region R.

The only question left, is when this fixed point is unstable and thus when the Poincaré-Bendixson theorem is applicable.

From earlier analysis of the model we know that for the equilibrium point $(x, y) = (Q, \frac{P}{Q})$ we have $\lambda_{1,2} = \frac{1}{2}(Trace(J) \pm \sqrt{\Delta})$ from (3.4) for the Jacobian evaluated at this equilibrium point. The equilibrium point is unstable if $Trace(J) > 0$. Our fixed point is a repeller if $Trace(J) = P - 1 - Q^2 > 0$. The dividing line for this is $P = 1 + Q^2$, giving us the line in the figure on the right. Under this line we have a stable fixed point, where the theorem is not satisfied and we do not have limit cycles. Above the line we have satisfied all conditions of the theorem, thus proving that there are stable limit cycles here. We have proved the existence of oscillating behaviour in our model!

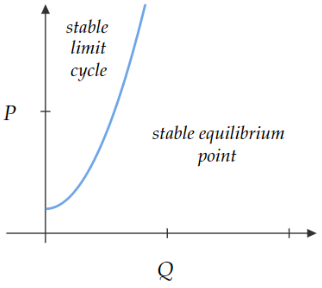
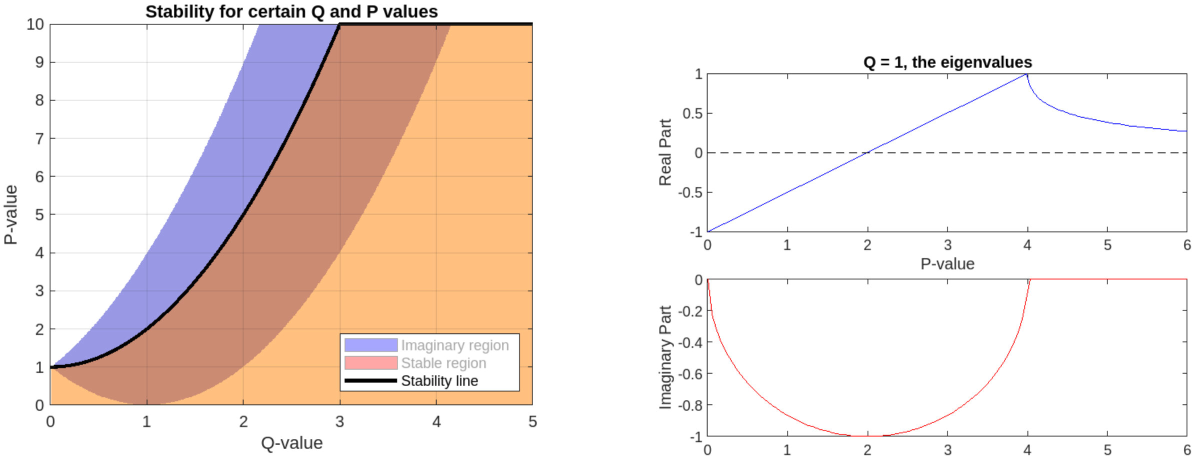


Figure 3.7: P versus Q, location of stable limit cycles.

3.4. Bifurcation Search

In figure 3.8a, P and Q are varied to see what happens to the eigenvalues of the system. This is done using the two conditions that followed out of expression 3.4. The blue region shows where the eigenvalues are complex applying the first condition, the orange region shows where the fixed point is stable, using the second condition. There is some overlap between the regions, meaning that there are stable fixed points with complex eigenvalues. The black line is interesting to look at. This line is plotted where $Trace(J) = 0$ (1) and $\Delta < 0$ (2). In practise this translates to $P - 1 - Q^2 = 0$ and $(P - 1 - Q^2)^2 - 4Q^2 < 0$. If $P - 1 - Q^2 = 0$, the second expression becomes $-4Q^2 < 0$. This always holds. Meaning that the black line is plotted where $P - 1 - Q^2 = 0$. On the black line, the eigenvalues are purely imaginary. And thus there should be a Hopf bifurcation around there. According to this plot, the eigenvalues are purely imaginary at (Q,P) = (1,2).



(a) Coloured regions for complex eigenvalues and for stable eigenvalues and a line stating a Hopf bifurcation for various P and Q values.

(b) The real and complex part of the eigenvalues of the model at its equilibrium point, with fixed Q = 1 and a varying P value.

Figure 3.8: Combined stability and eigenvalue plots.

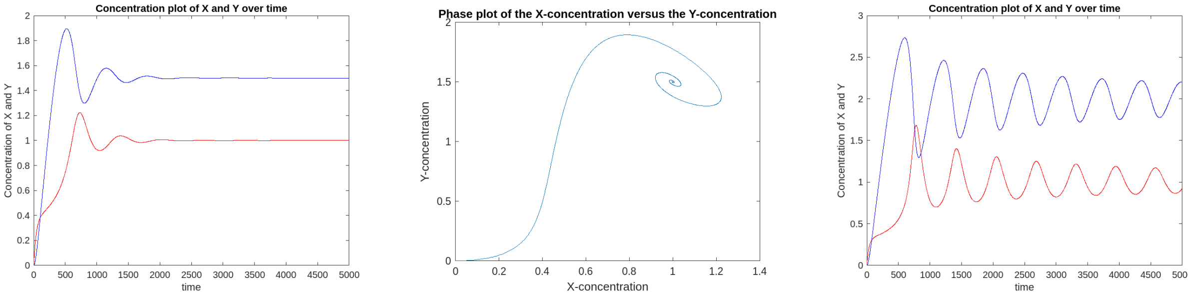
Let's zoom in on this case of (Q, P) = (1,2). To do this, we fix Q = 1 and vary P from 0 to 6. In figure 3.8b we can see that indeed at P = 2, the eigenvalues only have a complex part.

3.4.1. Oscillating Behaviour

We are looking for oscillating behaviour in the Brusselator model. While we verified one certain case of oscillating behaviour, we did not have a general idea of the behaviour of the model. By analysing the system of differential equations we have now found that a Hopf bifurcation occurs when $P - 1 - Q^2 = 0$. We found a Hopf bifurcation at (Q,P) = (1,2). In the figure below we see the phase plot of the system at a fixed value of Q

= 1, while P takes on the values 1.5, 2 and 3.

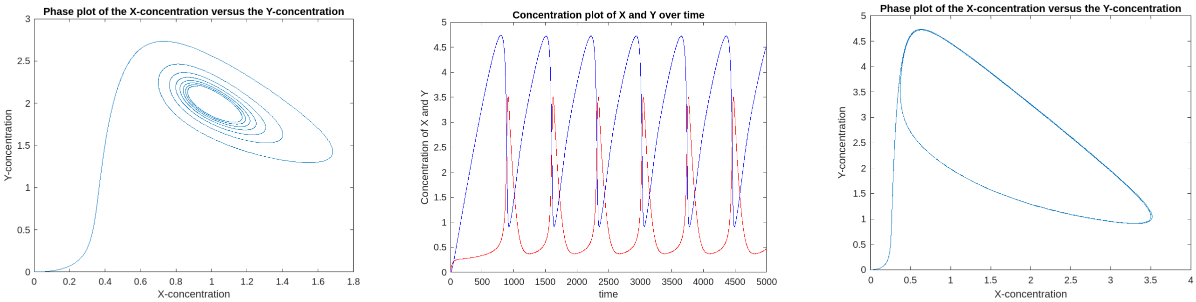
The equilibrium point of the system depends on the values of the parameters, so the equilibrium points moves as parameters are varied. For P = 1.5 we see a stable equilibrium point. For P = 2, we know we have a Hopf bifurcation and thus purely complex eigenvalues and thus a centre. Our bifurcation analysis is done on a linearised system, which is a good approximation of the actual non-linear system we have. For P = 2 we would expect there to be only a center, but because we also have a non-linear part in the system, the solution already behaves like a limit cycle. Lastly, when B moves even further along to P = 3, we see a cycle. So we have found oscillating behaviour and we now know when it occurs.



(a) Concentration plot with Q = 1 and P = 1.5.

(b) Phase plot with Q = 1 and P = 1.5.

(c) Concentration plot with Q = 1 and P = 2.



(d) Phase plot with Q = 1 and P = 2.

(e) Concentration plot with Q = 1 and P = 3.

(f) Phase plot with Q = 1 and P = 3.

Figure 3.9: Concentration and phase plots for Q = 1 and P varying from P = 1.5, 2, 3. The red lines in the concentration plots represent the concentration of X, the blue line the concentration of Y.

4

Continuous Stirred-Tank Reactor Model

In Chapter 3 we looked at the behaviour of the Brusselator. The model that was analysed is a fairly simple model, not entirely a realistic way of modelling the behaviour of such a system. We assumed that the model was executed in a batch reactor, where we then assumed that there was a constant amount of components A and B. As a result of this assumption we only tracked the concentration of X and Y in the time. In this chapter we will expand the Brusselator model by changing the type of reactor the reactions take place in. Instead of a batch reactor, a continuous stirred tank reactor will be used. This reactor type has a constant inflow of products, so the concentrations of A and B are not constant anymore. We will try to find oscillating behaviour in this newly updated model.

4.1. Continuous Stirred-Tank Reactor

A continuously stirred-tank reactor (CSTR) is a common used reactor for chemical reactions. In this reactor, the fluids present are continuously stirred. They are mixed sufficiently fast, that it doesn't matter where you measure the composition in the reactor. Everywhere in the reactor the components are in the exact same composition. It is also possible to let components flow in and flow out of the reactor, as visualized in Figure 4.1. The inflow of new reactants influences the composition of the fluids, but does not influence one position in particular. The composition differs in the inflow pipe, as the new substances are added here. But because of the continuing stirring, the inflow is mixed immediately into the already existing fluids. The outflow pipe has the same composition of components as the mixture in the reactor [10]. In this model we will only make use of inflow. There will be an inflow of certain components into the composition with a constant rate. As there is one inflow pipe, we can have only one rate of inflow. We will call the inflow rate Q ($[s^{-1}]$), where Q is constant. All components being brought in the reactor will have an inlet rate of Q multiplied by the initial amount of moles of the component. Because of the inflow and no outflow, we will not have a constant reaction volume.

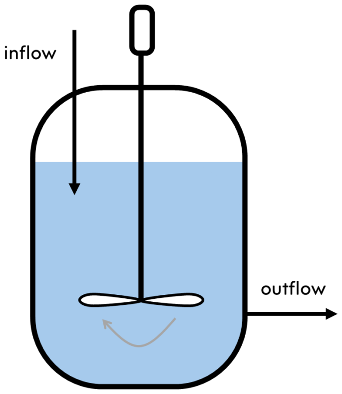


Figure 4.1: A CSTR modelled.

4.2. The Updated Model

The main change in the model is the reactor type, which translates to two changes in the mathematical model. Firstly, we will model all four components (X, Y, A and B) instead of only X and Y. And secondly, we will add an inflow of components A and B with a constant rate. Before we can look at the differential equations, we need to establish the assumptions of the model.

4.2.1. Assumptions

Some assumptions from Section 3.1.1 can be applied again. This, with our newly added assumptions, gives us the following list of assumptions:

- A, B, X, Y, D, E are in $\mathbb{R}_{\geq 0}$. As A, B, X, Y, D and E are all concentrations, they cannot be negative and have continuous values.
- D and E are not of influence. Because D and E are only products of the reactions and not used further, we do not consider them in the model.
- $k_1 = k_2 = k_3 = k_4 = 1$. When looking at the model numerically before analysing it, we will assume all reaction rates are equal to 1.
- All components are homogeneously distributed over the reaction volume. The reactor continuously stirs the components, so we can assume all components are evenly distributed. Meaning that we can eliminate the dimension of space.
- Q in $\mathbb{R}_{\geq 0}$ constant. Q, the inflow rate, is a continuous constant number, dictating the amount of inflow of all components.
- The changes in reaction volume will be ignored.

4.2.2. System of Differential Equations

In Chapter 3 we modelled only the concentrations of X and Y, as we assumed all other components to have a constant concentration. Now, we will model X, Y, A and B. This gives us a four dimensional system. The equations for X and Y remain the same as in (3.1). We will add the equation for A and B using the same method as in Section 2.1. This gives us the following system:

$$\begin{aligned}\frac{dX}{dt} &= k_1A - k_2BX + k_3X^2Y - k_4X,\\ \frac{dY}{dt} &= k_2BX - k_3X^2Y,\\ \frac{dA}{dt} &= -k_1A,\\ \frac{dB}{dt} &= -k_2BX.\end{aligned}$$

We have not yet added the inflow of components A and B. We have already established that the flow rate Q will determine the amount of new reactant. Q is a ratio, based on the initial amount of moles of a component (inflow), in this case, A_0 and B_0. Meaning that for the concentration of inflow of A we have to add QA_0 to the equation, modelling the constant inflow of A into the reactor. We do the same for B.

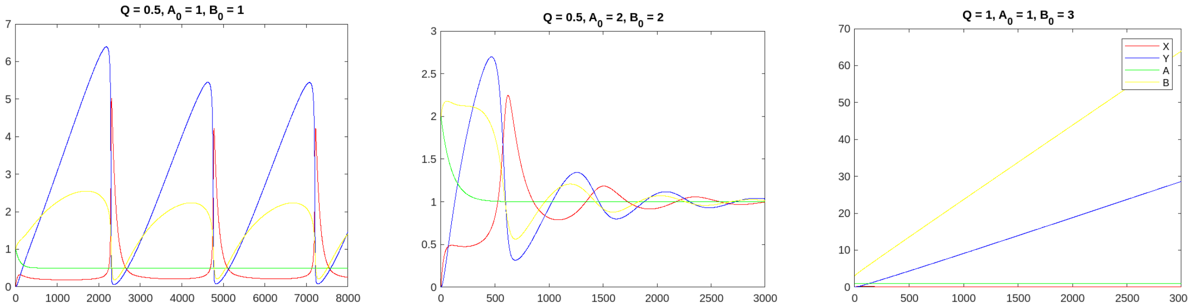
This yields the following system of differential equations:

$$\begin{aligned}
\frac{dX}{dt} &= k_1 A - k_2 B X + k_3 X^2 Y - k_4 X,\\
\frac{dY}{dt} &= k_2 B X - k_3 X^2 Y,\\
\frac{dA}{dt} &= -k_1 A + Q A_0,\\
\frac{dB}{dt} &= -k_2 B X + Q B_0.
\end{aligned} \tag{4.1}$$

4.2.3. Behaviour

To get an initial idea of the model's behaviour, we can numerically approximate the model again in Matlab. Here we use our assumption of $k_i = 1$ for all $i = 1,2,3,4$. We also assume that $X(0) = 0$, $Y(0) = 0$, $A(0) = A_0$, and $B(0) = B_0$, where we can vary A_0 and B_0. We do not yet have an idea about the value of Q. Meaning that we now have three uncertain parameters, A_0, B_0 and Q; and four assumed parameters $k_1 = k_2 = k_3 = k_4 = 1$. Again the Forward Euler method is used, now with four equations instead of two. The equations are of the same form as the equations in Section 3.1.3.

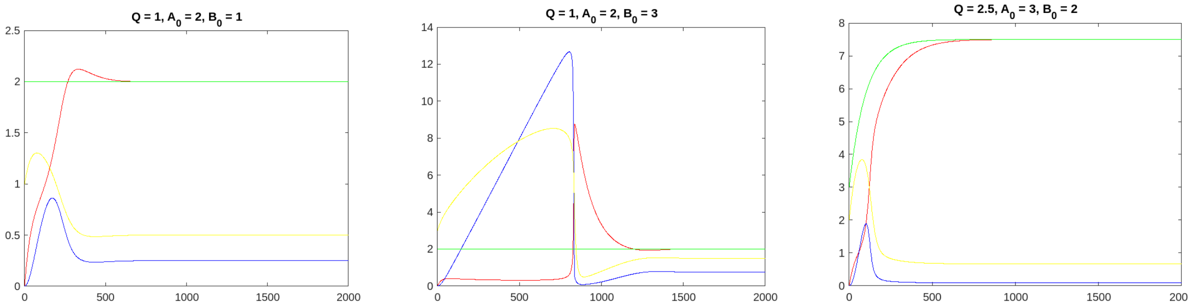
In Chapter 3 we knew where to look, now we will need to find the oscillating behaviour on our own.



(a) Concentration plot with Q = 0.5, A_0 = 1 and B_0 = 1.

(b) Concentration plot with Q = 0.5, A_0 = 2 and B_0 = 2.

(c) Concentration plot with Q = 1, A_0 = 1 and B_0 = 3.



(d) Concentration plot with Q = 1, A_0 = 2 and B_0 = 1.

(e) Concentration plot with Q = 1, A_0 = 2 and B_0 = 3.

(f) Concentration plot with Q = 2.5, A_0 = 3 and B_0 = 2.

Figure 4.2: Concentration plots for different inflow rates and initial values.

In Figure 4.2 you see six different cases with various values for inflow Q and initial conditions for A and B. All these plots look different. The first plot contains oscillating behaviour for Q=0,5, but the plot next to it with the same Q-value does not display the same behaviour. Other parameter value combinations generate divergent behaviour, quickly converged behaviour and a single oscillation. There does not seem to be a logic behind the oscillating behaviour, so we do not yet know where to look.

4.3. Analysis

To get a better idea of where to look for the oscillating behaviour, we'll analyse the model. Because the model is four dimensional now instead of two dimensional, we cannot use the Bendixson's Criterion or the Poincaré-Bendixson theorem. We will have to find the behaviour using other techniques. First, let's simplify the model.

4.3.1. Nondimensionalisation

Right now we have seven undetermined parameters in the model. To reduce this, we can nondimensionalise the system. We first scale the concentrations and the time with the following scaling factors:

$$X^* = \tfrac{X}{X_0}, \quad Y^* = \tfrac{Y}{Y_0}, \quad A^* = \tfrac{A}{A_0}, \quad B^* = \tfrac{B}{B_0}, \quad t^* = \tfrac{t}{t_0}.$$

After scaling and setting $t_0 = \frac{1}{k_4}$, we get the following equations:

$$\begin{aligned} \frac{dX^*}{dt^*} &= \frac{k_1 A_0}{k_4 X_0} A^* - \frac{k_2 B_0}{k_4} B^* X^* - X^* + \frac{k_3 X_0 Y_0}{k_4} X^{*2} Y^*, \\ \frac{dY^*}{dt^*} &= \frac{k_2 B_0 X_0}{k_4 Y_0} B^* X^* - \frac{k_3 X_0^2}{k_4} X^{*2} Y^*, \\ \frac{dA^*}{dt^*} &= \frac{Q}{k_4} - \frac{k_1}{k_4} A^*, \\ \frac{dB^*}{dt^*} &= \frac{Q}{k_4} - \frac{k_2 X_0}{k_4} B^* X^*. \end{aligned} \tag{4.2}$$

To simplify this further, we need an expression for X_0, Y_0, A_0 and B_0. To do this, we will make the term before $X^{*2} Y^*$ equal to 1. This results in an expression for X_0 and Y_0: $\sqrt{\frac{k_4}{k_3}}$. If we then look at the differential equations for X^* and A^* in (4.2), we find that X_0 should be equal to A_0. This to get the $\frac{k_1 A_0}{k_4 X_0} A^*$ from X^* to be equal to the $\frac{k_1}{k_4} A^*$ from A^*. To get an expression for B_0, we equate the term containing B_0 in the equation of X^* to 1: $\frac{k_2 B_0}{k_4}$.

This all yields the following expressions:

$$X_0 = Y_0 = A_0 = \sqrt{\tfrac{k_4}{k_3}}, \quad B_0 = \tfrac{k_4}{k_2}.$$

Substituting these expressions yields the following simplified system:

$$\begin{aligned} \frac{dX}{dt} &= \dot{X} = pA - X - BX + X^2 Y, \\ \frac{dY}{dt} &= \dot{Y} = BX - X^2 Y, \\ \frac{dA}{dt} &= \dot{A} = r - pA, \\ \frac{dB}{dt} &= \dot{B} = r - sBX, \end{aligned} \qquad \begin{aligned} p &= \frac{k_1}{k_4}, \\ r &= \frac{Q}{k_4}, \\ s &= \sqrt{\frac{k_2^2}{k_4 k_3}}. \end{aligned} \tag{4.3}$$

4.3.2. Linearising the System

The system has three parameters now, instead of the seven we had before in (4.1). This is looking very nice. To analyse the system, we first need the Jacobian matrix:

$$J = \begin{bmatrix} 2XY - B - 1 & X^2 & p & -X \\ B - 2XY & -X^2 & 0 & X \\ 0 & 0 & -p & 0 \\ -sB & 0 & 0 & -sX \end{bmatrix}.$$

To be able to say anything about the stability, we need to evaluate the Jacobian at the equilibrium point(s) of the system. When equating all equations to zero, we first get a possibility where X=0. But, when looking at the equation for A, we see that this would mean that r = 0. r can only be 0 if Q = 0. By definition, Q cannot be zero. Thus we disregard this option. One other point is found: $(X, Y, A, B) = (r, \frac{1}{sr}, \frac{r}{p}, \frac{1}{s})$. The Jacobian evaluated at this point is then:

$$J_{eq} = \begin{bmatrix} \frac{1}{s} - 1 & r^2 & p & -r \\ -\frac{1}{s} & -r^2 & 0 & r \\ 0 & 0 & -p & 0 \\ -1 & 0 & 0 & -rs \end{bmatrix}. \tag{4.4}$$

4.3.3. Assessment of Eigenvalues

In Chapter 3 we could express the eigenvalues of the Jacobian in a very convenient way. Expression (3.4) made it easier to analyse the eigenvalues and to find out where the oscillating behaviour takes place. As the Jacobian is a four dimensional matrix now, we need a different formula. For this, we define the following four expressions, where S_2 and S_3 are sort of sub determinants:

$$\begin{aligned} S_1 &= \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = Trace(J_{eq}) = T, \\ S_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_1\lambda_4 + \lambda_2\lambda_3 + \lambda_2\lambda_4 + \lambda_3\lambda_4, \\ S_3 &= \lambda_1\lambda_2\lambda_3 + \lambda_1\lambda_2\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_2\lambda_3\lambda_4, \\ S_4 &= \lambda_1\lambda_2\lambda_3\lambda_4 = Det(J_{eq}) = D. \end{aligned} \tag{4.5}$$

The following proposition is formulated with these expressions, to find where a Hopf bifurcation can arise:

Proposition 4.1 (Hopf Bifurcation in a 4-dimensional System) *A Hopf bifurcation generally arises if and only if*

$$S_2 = \frac{S_3}{T} + \frac{DT}{S_3} \tag{4.6}$$

and S_3 and T have the same sign [11].

The exact proof for this proposition can be read in [11], but we can get a feeling of the idea of the proof easily. The proof relies on using the characteristics of the eigenvalues at a Hop-Bifurcation. Suppose we have eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and λ_4, where λ_1 and λ_2 cross the imaginary axis, giving them the following form: $\lambda_{1,2} = \pm bi$, with $b \in \mathbb{R}$ and $b \neq 0$. Then we know that $\lambda_1 + \lambda_2 = 0$ and $\lambda_1\lambda_2 = b^2 > 0$. We can update equations (4.5) to the following:

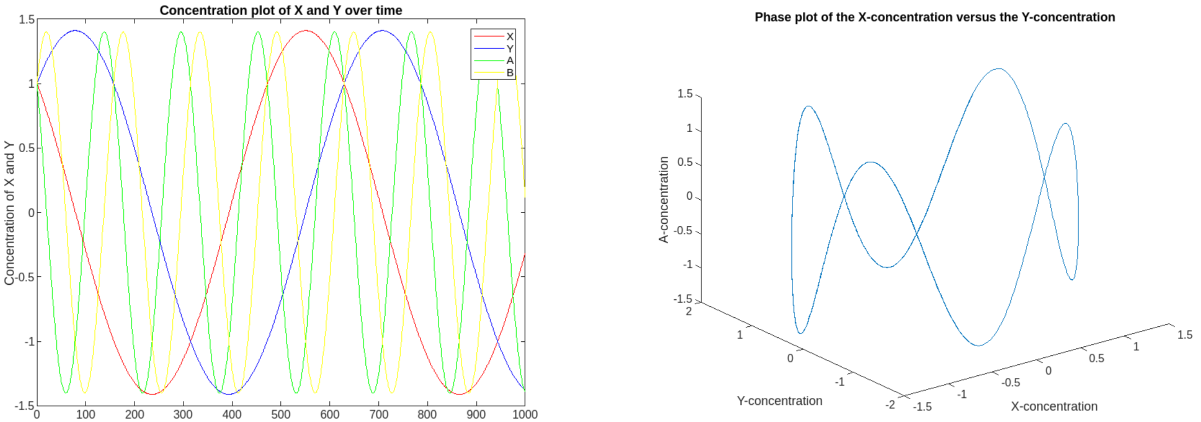
$$\begin{aligned} S_1 &= \lambda_3 + \lambda_4 = T, \\ S_2 &= \lambda_3\lambda_4 + b^2, \\ S_3 &= b^2 T, \\ S_4 &= b^2\lambda_3\lambda_4 = D. \end{aligned}$$

We can rewrite $S_2 = \lambda_3\lambda_4 + b^2$ to $b^2 + \frac{D}{b^2}$. To make this an expression only containing T, S_2, S_3 and D instead of b^2, we rewrite it to the following: $S_2 = \frac{S_3}{T} + \frac{DT}{S_3}$. This expression only depends on the 'sub determinants', meaning that we could do a general analysis of Hopf-Bifurcation locations with undetermined parameters. S_3 and T have to have the same sign to make sure that the fraction $\frac{S_3}{T}$ representing b^2 is always a positive number.

Because we are looking at a four dimensional system now, we get four eigenvalues out of the Jacobian. This means that there is a possibility for two conjugate pairs of complex eigenvalues to cross the imaginary axis at the same time, this would be a double Hopf bifurcation:

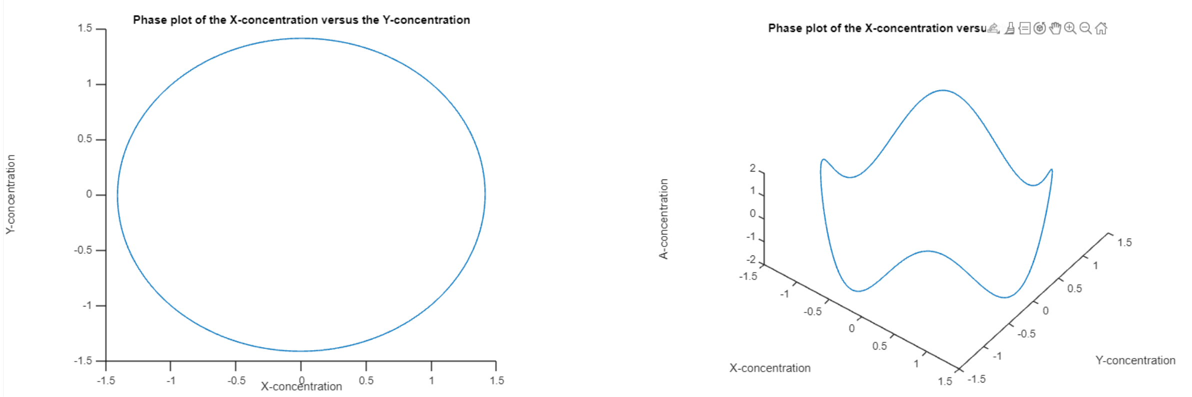
Definition 4.2 (Double-Hopf Bifurcation) *A double Hopf bifurcation occurs in a four dimensional system if* $\lambda_1 = bi$, $\lambda_2 = -bi$, $\lambda_3 = di$ *and* $\lambda_4 = -di$, *with b,d in* $\mathbb{R}$ *[11].*

A double Hopf-Bifurcation means that two Hopf-Bifurcation curves intersect. So, because of certain parameter values, you have two pairs of conjugate eigenvalues crossing the imaginary axis. The cycles these axis-crossing pairs create, intersect with each other. So while one pair of eigenvalues is creating a limit cycle, the other pair of eigenvalues generates another limit cycle at the same time. In the figure below you see the concentration and phase plots of a system that has a case of parameters where there occurs a double Hopf-Bifurcation.



(a) Concentration plot of a double Hopf-Bifurcation. All elements are oscillating.

(b) Phase plot of a double Hopf-Bifurcation, showcasing the concentration of three of the four molecules present. There is a clear limit cycle visible.



(c) The same phase plot as before, but oriented differently. When only looking at the concentration of components X and Y, you see the limit cycle as we've always seen in a single Hopf-Bifurcation.

(d) The same phase plot, now oriented slightly different. Here you can clearly see that there is a stable limit cycle, around which another cycle oscillates. This is the double Hopf-Bifurcation.

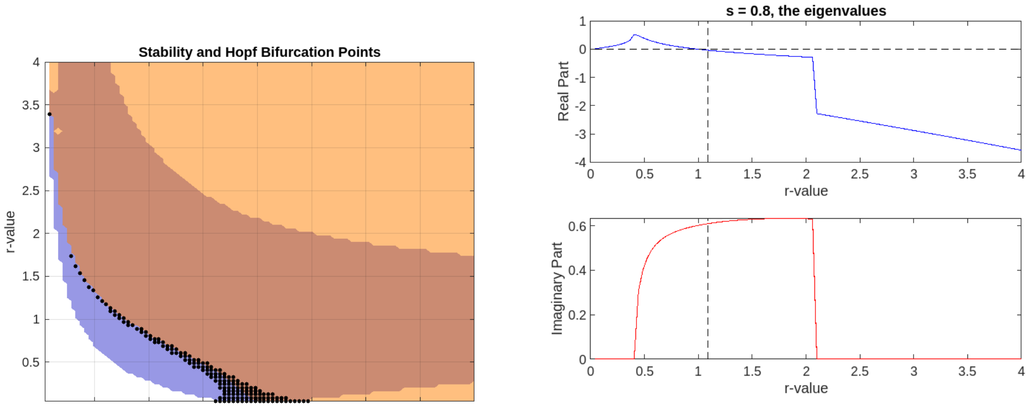
Figure 4.3: In these (differently) oriented phase plots of a system with a double Hopf-Bifurcation you can clearly see how the double Hopf-Bifurcation behaves. In the concentration plot you see that the concentrations of all components are oscillating.

As can be seen in Figure 4.3, a double Hopf-Bifurcation is one limit cycle around which another limit cycle seems to circle. The chance of a double Hopf-Bifurcation is very small, as there has to be a parameter combination for which all four eigenvalues have a real part equal to zero. As can be seen in Jacobian (4.4), one of the eigenvalues will always be $\lambda = -p$. As p is a real number, it is never possible to have eigenvalues as stated in Definition 4.2. Thus there will never occur a double Hopf bifurcation in this system, no matter the parameter changes.

Because of the complexity of the Jacobian (4.4), we get too complex expressions for the eigenvalues to analyse them. Fortunately, Matlab can help us with this.

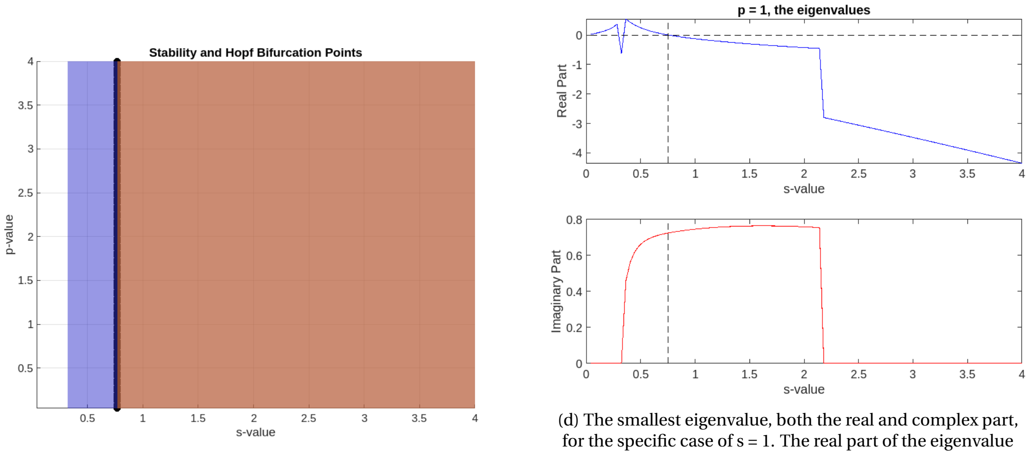
4.4. Bifurcation Search

Looking for oscillating behaviour, we have three parameters we can vary: p, r and s, the parameters we found in Section ((4.3.1)). We want to have two conjugate eigenvalues such that the real part is zero and the imaginary part not equal to zero. Because we have a four dimensional system, there is no easy way to express the eigenvalues like we did in Chapter 3 and to get a nice formula for the Hopf-bifurcation locations. To still be able to analyse the eigenvalues, we use Matlab. You can find the Matlab code in Appendix B.2. The plots in Figure 4.4 are generated using Proposition 4.1. For every combination of p, r and s the Jacobian matrix is set up, the eigenvalues are computed and the proposition is checked. If the eigenvalues comply with the proposition, there is a Hopf-bifurcation possibility and a black dot is put on this s, p, r combination. The stability of the eigenvalues and whether they are complex or not is checked too. Because this analysis is done on a generated grid, we get an estimation of all regions and points. Because of this we cannot draw a perfect line, but place dots on the grid points as close to the correct values as possible. The parameter ranges for s, p and r are visually determined, all necessary information is accessible when the parameters range from 0 to 4. All this information together creates the following figures:



(a) Regions for s versus r with p = 1.

(b) The smallest eigenvalue, both the real and complex part, for the specific case of s = 0.8. The real part of the eigenvalue crosses the imaginary axis at the black dashed vertical line.



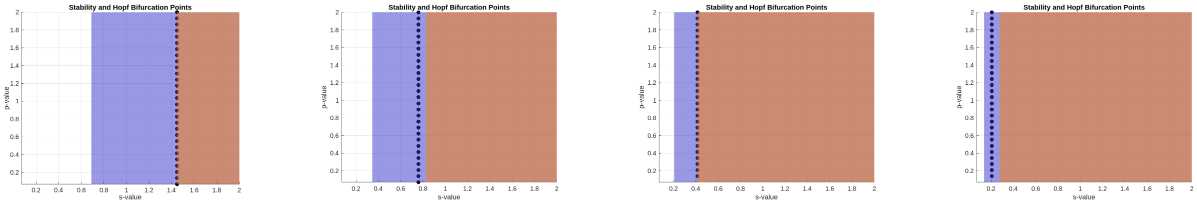
(c) Regions for s versus p with r = 1.

(d) The smallest eigenvalue, both the real and complex part, for the specific case of s = 1. The real part of the eigenvalue crosses the imaginary axis at the black dashed vertical line.

Figure 4.4: In the plots on the left, the orange region shows where the eigenvalues are stable, the blue region shows where (some) eigenvalues are complex. When these two regions overlap, you get a brown colour. The black dots show where the real part of the eigenvalues crosses the imaginary axis. The plots to the right show the eigenvalues for a specific case of the plots on the left.

From these plotted Hopf-Bifurcation locations we can draw some conclusions. The most interesting plot is parameters s versus r, as we can see a curve in this plot. On this black 'line' the real part of the eigenvalues is (almost) zero. Meaning that when eigenvalues cross from above this line to under this line, they cross the imaginary axis and the stability of the equilibrium point shifts from stable to unstable. Thus creating the possibility for a limit cycle. The second plot, parameters s versus p, tells us that the p-value does not matter much for the Hopf-Bifurcation location. For $r = 1$ and $s \approx 0.75$ we have a Hopf-Bifurcation for all p-values. The placement of the black line is backed up by the plots of the eigenvalues in Figure 4.4. The real part of the eigenvalues crosses from positive to negative when there appears a black dot in the region plots.

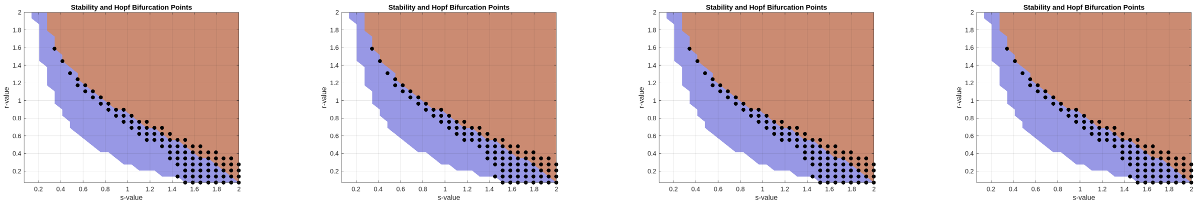
In Figure 4.4 we see the stability regions and Hopf-Bifurcation locations for either a fixed p-value or a fixed r-value. What happens if this parameter is varied? To visualise this, in the figure below we see the region and Hopf location plots with various p-values and r-values. These plots were generated with a much smaller grid, meaning that they are less accurate about the actual Hopf-Bifurcation location placement. Despite the inaccuracy, we can still see how the plots change when the fixed parameter is varied.



(a) r = 0.5 (b) r = 1 (c) r = 1.5 (d) r = 2

Figure 4.5: Stability regions and Hopf-Bifurcation locations on a small grid for parameters s versus p with varying r value.

In Figure 4.5 the r-value is varied to see the effect on the Hopf-Bifurcation locations. When r becomes larger, the Hopf-line shifts to the left. From this plot we can conclude that the p-value really does not influence the Hopf-Bifurcation locations. There is a correlation between r and s. In the figure below we see this verified, where varying the p-value does not change the relation between s and r.



(a) p = 0.5 (b) p = 1 (c) p = 1.5 (d) p = 2

Figure 4.6: Stability regions and Hopf-Bifurcation locations on a small grid for parameters s versus r with varying p value.

Combining all the information of the two dimensional plots, we can create a three dimensional plot containing all Hopf-Bifurcation locations (Figure 4.7). Here we should still see that the p-value does not influence these locations and see a curved relation between parameters s and r.
In Chapter 3 an analytical solution was found as to when the Hopf-Bifurcation takes place. Because the system is four dimensional this has become much harder. In theory it is possible to solve the characteristic polynomial of the eigenvalues generally and extract the information necessary for a Hopf-Bifurcation, but the result would not be interpretable anymore.

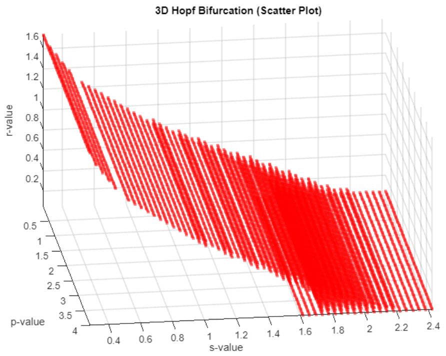
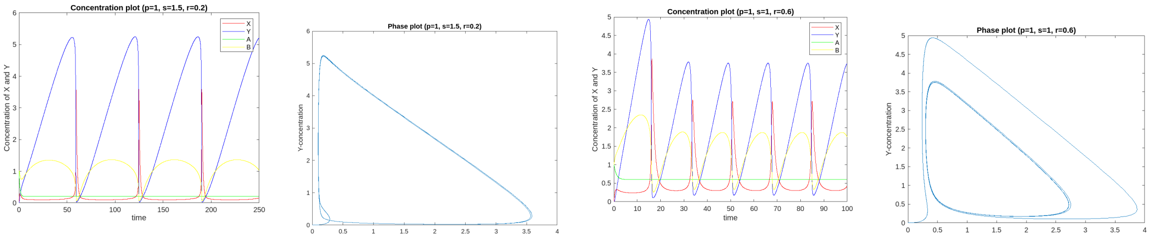


Figure 4.7: Three dimensional space for parameters s, p and r. The red dots represent a Hopf-Bifurcation location.

4.4.1. Oscillating behaviour

Now that we have done extensive research on the locations of oscillating behaviour, we can numerically find the oscillating behaviour again. From Figure 4.7 we can extract the points where this oscillating behaviour takes place. In Section 3.3 we have seen that after the eigenvalues cross the imaginary axis, the oscillating behaviour takes place where the equilibrium point is unstable. Meaning that we have to look in the region where not all of the eigenvalues have negative real part. In Figure 4.4 this would be the side of the black line where the orange region has stopped. Here the eigenvalues are still complex, but not negative anymore. Let's choose some points from Figure 4.7 and plot the different cases of parameters numerically. Two cases of oscillating behaviour would be $(p,s,r) = (1,1.5,0.2)$ and $(p,s,r) = (1,1,0.6)$. One case approaching the Hopf-bifurcation line is $(p,s,r) = (1,0.75,1.2)$. The last case is not oscillating at all: $(p,s,r) = (1,2,1)$.

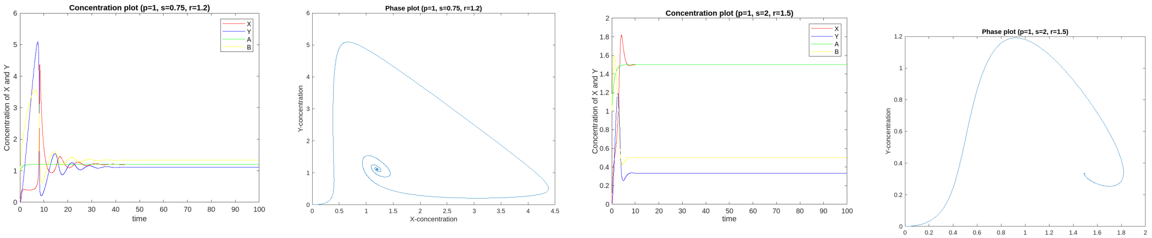


(a) Concentration plot; p = 1, s = 1.5, r = 0.2

(b) Phase plot; p = 1, s = 1.5, r = 0.2

(c) Concentration plot; p = 1, s = 1, r = 0.6

(d) Phase plot; p = 1, s = 1, r = 0.6



(e) Concentration plot; p = 1, s = 0.75, r = 1.2

(f) Phase plot; p = 1, s = 0.75, r = 1.2

(g) Concentration plot; p = 1, s = 2, r = 1.5

(h) Phase plot; p = 1, s = 2, r = 1.5

Figure 4.8: Four cases of parameter values and their concentration and phase plots. The first two cases (a - d) are oscillating. The third cases (e - f) is approaching the Hopf-Bifurcation line. The last case (g - h) is not oscillating at all.

4.5. Modelling Outflow

Instead of only allowing inflow into the reactor, adding outflow would also be interesting. This would mean that at all times at rate Q some fluids are led out of the reactor. In the case of our model we could add outflow to A and B. The inflow and outflow rate is the same value Q, meaning that the inflow is going at the same rate as the outflow. Adding this outflow gives us the following system of equations:

$$\begin{aligned}
\frac{dX}{dt} &= k_1 A - k_2 BX + k_3 X^2 Y - k_4 X,\\
\frac{dY}{dt} &= k_2 BX - k_3 X^2 Y,\\
\frac{dA}{dt} &= -k_1 A + QA_0 - QA,\\
\frac{dB}{dt} &= -k_2 BX + QB_0 - QB.
\end{aligned}$$

Using the exact same steps for nondimensionalising as in Section 4.3.1, we get the updated system:

$$\begin{aligned}\frac{dX}{dt} &= \dot{X} = pA - X - BX + X^2Y,\\ \frac{dY}{dt} &= \dot{Y} = BX - X^2Y,\\ \frac{dA}{dt} &= \dot{A} = r - pA - rA,\\ \frac{dB}{dt} &= \dot{B} = r - sBX - rB.\end{aligned}$$

This system has one equilibrium point: $(X, Y, A, B) = (\frac{r}{p+r}, \frac{p+r}{p+r+ps}, \frac{pr}{p+r}, \frac{o^2+2pr+r^2}{pr^2+p^2r+p^2rs})$. When analysing this system with the same approach as in Section 4.4, there seem to only be stable eigenvalues. Meaning that the real part of the eigenvalues always is negative and there is no possibility for a Hopf-bifurcation to occur. In the figure below you only see an orange region, which indicates that all eigenvalues are real and negative.

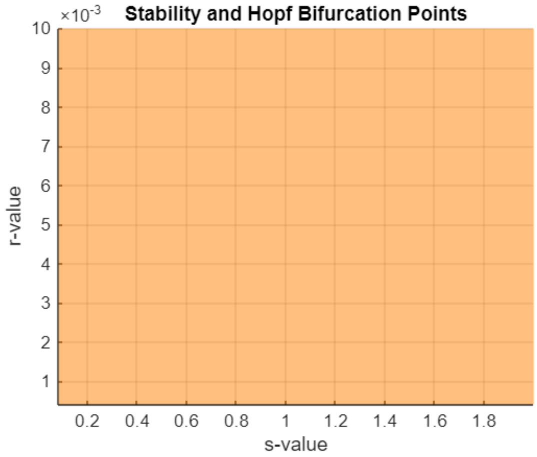


Figure 4.9: Stability region plot for very small r value against s, with fixed p = 1. The orange region indicates that all eigenvalues for all combinations of r and s are stable.

An apparent problem could be $\frac{dA}{dt}$. Molecule A is the initiator of the system; molecule A starts the production of component X, which is half of the reason for the oscillating behaviour. With the newly added outflow to $\frac{dA}{dt}$, the only adding factor of $\frac{dA}{dt}$ is the inflow, now variable r. There seems to not be enough of molecule A left over to initialize the production of X. Even when varying parameter r from 0 to 0.01, there is no unstable eigenvalue in sight. Unfortunately, we do not find any oscillating behaviour when adding outflow to the Brusselator model.

5

Oregonator

In this chapter, we will model another theoretical model describing chemical reactions. The Oregonator is a model describing an oscillating chemical system. It is a simpler form of the Belousov-Zhabotinsky reaction still displaying oscillating behaviour. Six components make up the reactions: A, B, X, Y, Z and P. The following five reactions describe the Oregonator:

$$\begin{aligned} \mathrm{A}+\mathrm{Y} &\xrightarrow{\mathrm{k_1}} \mathrm{X}+\mathrm{P}, \\ \mathrm{X}+\mathrm{Y} &\xrightarrow{\mathrm{k_2}} 2\,\mathrm{P}, \\ \mathrm{A}+\mathrm{X} &\xrightarrow{\mathrm{k_3}} 2\,\mathrm{X}+2\,\mathrm{Z}, \\ 2\mathrm{X} &\xrightarrow{\mathrm{k_4}} \mathrm{A}+\mathrm{P}, \\ \mathrm{B}+\mathrm{Z} &\xrightarrow{\mathrm{k}_c} \frac{1}{2} f\,\mathrm{Y}. \end{aligned} \tag{5.1}$$

The f in the fifth reaction is a yet undetermined stoichiometric coefficient, meaning that $f \in \mathbb{N}_{\geq 0}$. Setting f determines how much of Y is produced by B and Z in the last reaction. In the Brusselator model it was easy to see which reactions of the system cause the oscillating behaviour. Here it is less obvious. First, as initiator of the cascade, Y produces X. Then X leads to Z and more X in the third reaction. The fifth reaction transforms Z into Y again. As these three reactants all produce each other again, we can search for oscillating behaviour.

We will model the Oregonator in a batch reactor in this chapter. For more information on the batch reactor, see Chapter 3.

5.1. Rate Constants

The Belousov-Zhabotinsky reaction is very complicated and contains over 18 different steps. To be able to analyse this system, a simplified version was extracted: the FKN mechanism. The FKN mechanism still contains 11 reactions and 12 different components. So, an even simpler version was created: the Oregonator. Only five reactions and six components left. Because the Oregonator is an extraction of the widely researched FKN mechanism, we have some information about the parameters in the model. Field and Noyes, two chemists, published their findings in [4]. From this we know the following:

$$\begin{aligned} k_1 &= 2M^{-3}s^{-1}[H^+]^2, \\ k_2 &= 3*10^6M^{-2}s^{-1}[H^+], \\ k_3 &= 42M^{-2}s^{-1}[H^+], \\ k_4 &= 3*10^3M^{-1}s^{-1}[H^+]. \end{aligned}$$

Table 5.1: Rate coefficients for the Oregonator, retrieved from the FKN mechanism

With these parameter ranges the Oregonator model is a pretty good approximation of the more complex BZ-reaction mechanism. Here $[H^+]$ is the concentration of H^+ in the FKN mechanism, a constant concentration. From the article on the Oregonator itself [6] we know that realistically for the Oregonator the $[H^+]$ value ranges from 0.1 to 2 M. All parameters ranges include $[H^+] \in (0.1-2)M$, $k_c \in (0.1-10)M^{-1}s^{-1}$, $f = 0,1,2,3$ and $A, B \in (0.01-1)M$. We will use these parameter ranges in the modelling of the Oregonator.

5.2. The Model

To see how the model behaves we will transform the system into differential equations again. We will model the concentrations of components X, Y and Z. Before we can do this, we need some assumptions.

5.2.1. Assumptions

The following assumptions are made:

- A, B, X, Y, Z, P are in $\mathbb{R}_{\geq 0}$. As A, B, X, Y, Z and P are all concentrations, they cannot be negative and have continuous values.
- A and B are constants. There is an abundance of these molecules present, meaning that the change in concentration for A and B is almost zero over time.
- P is not of influence. P is a product of the reaction and is not used further.
- All parameters will be in the ranges as defined in Section 5.1.
- All components are homogeneously distributed over the reaction volume. The batch reactor stirs the reaction, so all molecules are evenly distributed over the reactor. Because of this we can eliminate the dimension of place from the system.
- The reaction volume is constant. The batch reactor is a closed system, there is no in- or outflow. Because of the law of Conservation of Mass, as stated in Definition 2.1, the reaction volume will stay constant.

5.2.2. System of Differential Equations

We first need to convert the chemical reactions into differential equations. We do this in the same way as we did for the Brusselator, which can be read in Section 2.1. From every chemical reaction a reaction rate is derived:

Reaction i	Reaction rate i	Rate constant i
$\mathrm{A+Y \xrightarrow{k_1} X+P}$	$v_1 = k_1 AY$	$k_1 : [M^{-1}s^{-1}]$
$\mathrm{X+Y \xrightarrow{k_2} 2P}$	$v_2 = k_2 XY$	$k_2 : [M^{-1}s^{-1}]$
$\mathrm{A+X \xrightarrow{k_3} 2X+2Z}$	$v_3 = k_3 AX$	$k_3 : [M^{-1}s^{-1}]$
$\mathrm{2X \xrightarrow{k_4} A+P}$	$v_4 = k_4 X^2$	$k_4 : [s^{-1}]$
$\mathrm{B+Z \xrightarrow{k_c} \frac{1}{2} \mathit{f} Y}$	$v_5 = k_c BZ$	$k_c : [M^{-1}s^{-1}]$

Table 5.2: Reaction rates for all five reactions of the Oregonator.

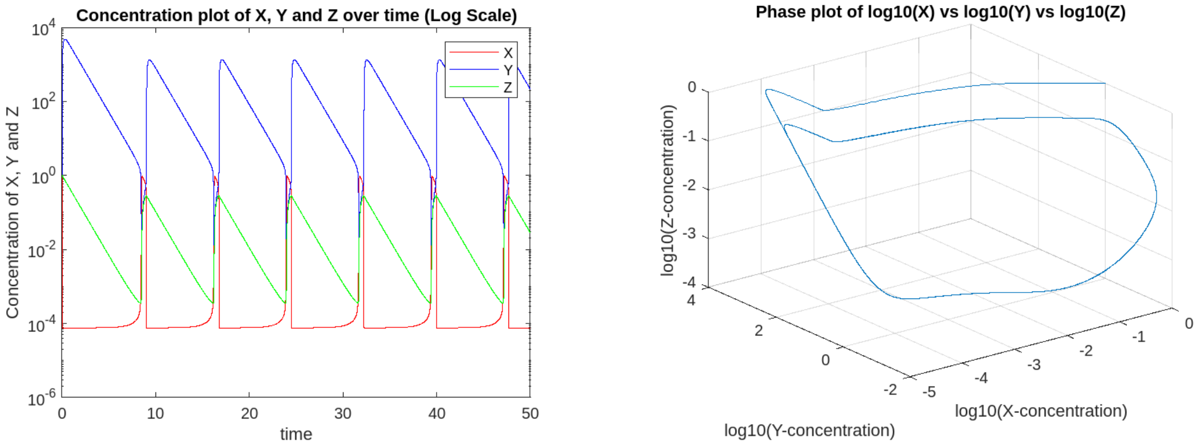
Which yields the following system of differential equations:

$$\begin{aligned}\frac{dX}{dt} &= k_1AY - k_2XY + k_3AX - 2k_4X^2,\\ \frac{dY}{dt} &= -k_1AY - k_2XY + \frac{1}{2}k_cfBZ,\\ \frac{dZ}{dt} &= 2k_3AX - k_cBZ.\end{aligned} \tag{5.2}$$

We assumed component P to not be of influence, which is why P is not represented in this system. The system is quite straightforward, but let's take a closer look at the $-2k_4X^2$ in $\frac{dX}{dt}$ and the $2k_3AX$ in $\frac{dZ}{dt}$. The reaction rates are multiplied by two, this is because either two times the reactant is used ($2X \longrightarrow A+P$) or two products are produced ($A+X \longrightarrow 2X+2Z$).

5.2.3. Behaviour

Let's first look at a numerical approximation of the model. There are eight parameters and three initial conditions to set before being able to get a result. In the article on the Oregonator model [6] parameters are provided for a case of oscillating behaviour. For this we set the following parameter values: $A = 0.06M$, $B = 0.02M$, $[H^+] = 0.8M$, $k_c = 1M^{-1}s{-}1$, $f = 1$ and $X(0) = Y(0) = Z(0) = 1$. With the value of $[H^+]$ and the expressions from Table 5.2 the values for k_1, k_2, k_3 and k_4 can be computed. With these parameters we get the following behaviour:



(a) Concentration plot of the Oregonator.

(b) Phase plot of the Oregonator.

Figure 5.1: Plots of the Oregonator with certain parameter values.

In the concentration plot in Figure 5.1a you can clearly see the periodic behaviour of the concentrations. It is interesting to note that the values are all very small in comparison to what we saw earlier with the Brusselator. In Figure 5.1b you can see that the phase plot contains a three dimensional limit cycle. It starts a little above this cycle, but eventually tends towards the cycle.

5.2.4. Numerical Method

The numerical method used on the Brusselator (Forward Euler method) does not work very well on the Oregonator. We can classify (5.2) as stiff equations: differential equations for which a numerical method is unstable, unless a really small step size is taken. The result of the Forward Euler method for the Oregonator was very unclear, it is an unstable method for this system of equations. To solve this problem, a stiff ordinary differential equations solver is used in Matlab: *ode15s*.

Ode15s is an ordinary differential equations (ode) solver specifically for stiff ode's. There are two major differences towards the Forward Euler method. Firstly, the Forward Euler method is an *explicit* method: calculating the next step based on the current step. Ode15s is an *implicit* method: solving the whole system based on the current position and the next step. For every time step the system has to be solved again. This is computationally costly in comparison to an explicit method. Secondly, ode15s uses an adaptive step size. Instead of always using the same fixed time step like the Forward Euler method, it adjusts the step size based on certain thresholds. The user can set the tolerated relative and absolute error of the system. If the relative error or the absolute error is higher than the set allowed threshold, the time step is taken smaller and the system is solved again for this time step. Only if the error values are below the tolerated error value, the solver continues to the next step. The relative error is the error relative to the magnitude of the solution. The error is estimated by the estimated local truncation error, which is based on the step size. Meaning that if your estimated error is too high, it will become smaller if the step size is smaller.

Ode15s is a trade-off between computational cost (small time steps mean more computations) and accuracy (more time steps result in a better numerical method). The model we are approximating with this method is not very complex and is not very computationally expensive, meaning that we can opt for accuracy.

5.3. Analysis

5.3.1. Nondimensionalisation

We have eight undetermined parameters in our model: five rate coefficients, two constant components and one stoichiometric coefficient (f). To reduce this, we nondimensionalise the system. We first scale the concentrations and the time with the following scaling factors:

$$X^* = \tfrac{X}{X_0}, \quad Y^* = \tfrac{Y}{Y_0}, \quad Z^* = \tfrac{Z}{Z_0}, \quad t^* = \tfrac{t}{t_0}.$$

After scaling and setting $t_0 = \frac{1}{k_c B} = \frac{1}{[M^{-1}s^{-1}][M]} = \frac{1}{[s^{-1}]}$, we get the following equations:

$$\begin{aligned}
\frac{dX^*}{dt^*} &= \frac{Ak_3}{Bk_c}X^* - \frac{2k_4X_0}{Bk_c}X^{*2} - \frac{k_2Y_0}{Bk_c}X^*Y^* + \frac{Ak_1Y_0}{Bk_cX_0}Y^*,\\
\frac{dY^*}{dt^*} &= \frac{fZ_0}{2Y_0}Z^* - \frac{Ak_1}{Bk_c}Y^* - \frac{k_2X_0}{Bk_c}X^*Y^*,\\
\frac{dZ^*}{dt^*} &= \frac{2Ak_3X_0}{Bk_cZ_0}X^* - Z^*.
\end{aligned} \tag{5.3}$$

To simplify this further, we need an expression for X_0, Y_0 and Z_0. Instead of equating the highest order to 1 like before, two other parameters (ϵ and ϵ') are introduced, to be read in [6]. We will use the following expressions for X_0, Y_0 and Z_0:

$$X_0 = \tfrac{Ak_3}{2k_4}, \quad Y_0 = \tfrac{Ak_3}{k_2}, \quad Z_0 = \tfrac{(Ak_3)^2}{Bk_4k_c}.$$

Substituting these expressions yields the following simplified system:

$$\begin{aligned}
\frac{dX}{dt} &= \epsilon\dot{X} = qY - XY + X - X^2, &\qquad q &= \frac{2k_1k_4}{k_2k_3},\\
\frac{dY}{dt} &= \epsilon'\dot{Y} = -qY - XY + fZ, &\qquad \epsilon &= \frac{k_cB}{k_3A},\\
\frac{dZ}{dt} &= \dot{Z} = X - Z, &\qquad \epsilon' &= \frac{2k_ck_4B}{k_2k_3A}.
\end{aligned} \tag{5.4}$$

5.3.2. From a 3-dimensional to a 2-dimensional System

Right now we have a three dimensional system, tracking the concentrations of X, Y and Z. The changes in concentration of Y are all multiplied by ϵ', as derived in (5.4). With the assumed parameter values and the definition of ϵ', ϵ' will become very small compared to the other parameters and variables in the system. The expression of ϵ' with filled in k_2 and k_4 values, becomes:

$$\epsilon' = \frac{2k_c k_4 B}{k_2 k_3 A} = \frac{2k_4}{k_2}\epsilon = \frac{2}{10^3}\epsilon.$$

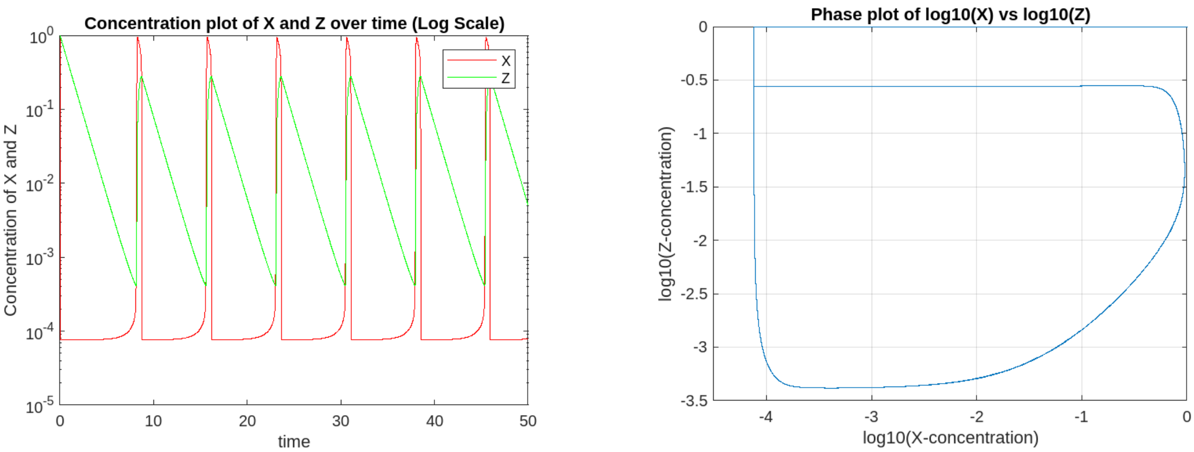
Because ϵ' is relatively small compared to ϵ, we can neglect the influence of Y and assume it is already in steady state:

$$\frac{dY}{dt} = 0 \rightarrow Y_{eq} = \frac{fZ}{X+q}.$$

Substituting the steady state of Y into the nondimensionalised system (5.4) yields the following 2-dimensional system:

$$\begin{aligned} \frac{dX}{dt} &= \dot{X} = \frac{1}{\epsilon}\left(X(1-X) - \frac{fZ(X-q)}{X+q}\right), \\ \frac{dZ}{dt} &= \dot{Z} = X - Z. \end{aligned} \tag{5.5}$$

We will namely analyse the 2-dimensional system in this Chapter. In Section 5.2.3 we looked at a very specific case of parameter values where oscillating behaviour arose. Let's see if it still oscillates with the 2-dimensional system.



(a) 2 dimensional concentration plot of the Oregonator.

(b) 2 dimensional phase plot of the Oregonator.

Figure 5.2: 2 dimensional plots of the Oregonator with certain parameter values.

Because the influence of Y was already very small the 2-dimensional system still oscillates at these parameter values!

5.3.3. Linearising

To analyse the system we will linearise it around its equilibrium points. Let's first get the general Jacobian of the system:

$$J = \begin{bmatrix} \frac{1}{\epsilon}\left(-2X + \frac{fZ}{X+q} - \frac{fZ(X-q)}{(X+q)^2}\right) & \frac{1}{\epsilon}\left(\frac{f(X-q)}{X+q}\right) \\ 1 & -1 \end{bmatrix}.$$

Equating the differential equations generates the following three equilibrium points:

$$\begin{aligned} (x_{eq_1}, z_{eq_1}) &= (0,0), \\ (x_{eq_2}, z_{eq_2}) &= \left(\frac{1}{2}\left(-\sqrt{2q-2f+6fq+f^2+q^2+1} - q - f + 1\right), x_{eq_2}\right), \\ (x_{eq_3}, z_{eq_3}) &= \left(\frac{1}{2}\left(\sqrt{2q-2f+6fq+f^2+q^2+1} - q - f + 1\right), x_{eq_3}\right). \end{aligned}$$

As $Z = X - Z$, we always have that $(x_{eq}, z_{eq}) = (x_{eq}, x_{eq})$. So, there is only one equilibrium point relevant, the last one. A point at the origin and a point with only negative X and Z values are chemically seen not interesting. We will only look at the third point, which exists in the first quadrant and thus is a realistic equilibrium point. The Jacobian evaluated at this point looks like:

$$J_{eq} = \begin{bmatrix} \frac{1}{\epsilon}\left(\frac{f+q-S-fE_1}{E_2} + \frac{fE_1E_3}{E_2^2}\right) & \frac{1}{\epsilon}\left(\frac{-fE_3}{E_2}\right) \\ 1 & -1 \end{bmatrix} \quad \text{with} \quad \begin{aligned} S &= \sqrt{f^2+6fq-2f+q^2+2q+1}, \\ E_1 &= \frac{S-f-q+1}{2}, \\ E_2 &= \frac{S-f+q+1}{2}, \\ E_3 &= \frac{S-f-3q+1}{2}. \end{aligned} \tag{5.6}$$

Because we still have three undetermined parameters we get a very complicated Jacobian.

5.4. Proof of Oscillating Behaviour

Before trying to find the oscillating behaviour, we can prove that the behaviour exists. We will first apply Bendixson's Criterion to see where there is a possibility for oscillating behaviour. We will then apply the Poincaré-Bendixson Theorem on the region where we expect possible oscillating behaviour.

5.4.1. Bendixson's Criterion

We will first apply Bendixson's Criterion to see if it is even possible to have oscillating behaviour in our new model. Recall the criterion as stated in Section 3.3:

Theorem 5.1 (Bendixson's Criterion) *Consider the system $\dot{x} = g(x,y)$, $\dot{y} = h(x,y)$ in $\mathbb{R}^2$. Suppose D is a simply connected subset of $\mathbb{R}^2$ and (g,h) is continuously differentiable in D. If $div(g,h) = \nabla(g,h) = \frac{dg}{dx} + \frac{dh}{dy}$ changes sign in region D or if $\nabla(g,h) = 0$, only then can the system have periodic solutions.*

The Bendixson's Criterion only works on a two dimensional system. Thus we will only apply it to our two dimensional version. Let's consider our nondimensionalised two dimensional system:

$$\begin{aligned} \frac{dx}{dt} &= \dot{x} = \frac{1}{\epsilon}\left(x(1-x) - \frac{fz(x-q)}{x+q}\right), \\ \frac{dz}{dt} &= \dot{z} = x - z. \end{aligned} \tag{5.7}$$

We will apply Bendixson's theorem to see whether there is a possibility of a periodic solution and if so, for which f, ϵ and q values. We will do this for D(X,Z) = $\mathbb{R}^2_{\geq 0}$. We first need the divergence of the dynamic system. Let $g(x,y) = \dot{x}$ and $h(x,y) = \dot{z}$, then:

$$\nabla(g,h) = \frac{dg}{dx} + \frac{dh}{dy} = \frac{1}{\epsilon}\left(\frac{-fz}{x+q} + \frac{fz(x-q)}{(x+q)^2} - 2x + 1 - \epsilon\right). \tag{5.8}$$

If $\nabla(g,h) = 0$, we immediately know that there is a possibility for oscillating behaviour. Unfortunately, this is not the case. Now we need to investigate whether there occurs a sign change in $\nabla(g,h)$ in (5.8). From Section 5.1 we know that $q \in [0, 0.1]$, $f = 0, 1, 2, 3$ and $\epsilon \in (0, \infty)$.

Let $\nabla(g,h) = \hat{f}(x,z)$. To investigate occurrences of sign changes, we need to know where $\hat{f}(x,z) = 0$. Equating $\hat{f}(x,z)$ to zero yields:

$$\hat{f}(x,z) = 0 \rightarrow l : z = \frac{-(2x+\epsilon-1)(q+x)^2}{2qf}.$$

Because of how line l is defined we cannot investigate where $q = 0$ or $f = 0$. All other cases are fine.

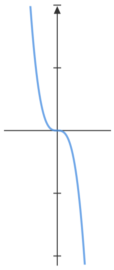


Figure 5.3: Shape of a negative cubic equation.

If we rewrite line l from a fraction to a polynomial we have $l : z = \frac{1}{2qf}(-2x^3 + (1-4q-\epsilon)x^2 + (2q-2q^2-2\epsilon q)x - \epsilon q^2 + q^2)$. This tells us that the general form of line l, which is a negative cubic equation, is of the form in Figure 5.3. We want to show that there is a sign change in the first quadrant for certain parameter values. Because of this cubic shape, we can conclude that if line l crosses the positive z-axis, we can have a sign change. When crossing the positive z-axis, we have $x = 0$. Let's figure out when $z(0) > 0$ and thus crossing the positive z-axis. From the polynomial form of z we can easily see that $z(0) = \frac{1}{f}(q - \epsilon q) > 0$ if $\epsilon < 1$. If $\epsilon \geq 1$, there is no chance to have a sign change in the first quadrant.

We have narrowed down the parameters of looking for a sign change to $q \in (0, 0.1]$, $f = 1, 2, 3$ and $\epsilon \in (0, 1)$. Let's find a point under line l and above line l to show that the points crossing line l change sign. We will show that the points under the line are always positive and the points above the line always negative.

Let $A : (x, z) = (0, 1)$. Let's first prove that point A is always above line l by showing that $z(0) < 1$. $z(0) = \frac{1}{f}(q - \epsilon q) < 1 \rightarrow q < \frac{f}{1-\epsilon}$. As we assumed that $q \in (0, 0.1]$, this q-statement always holds. $(x, z) = (0, 1)$ is always above line l. Let's now fill in point A into (5.8). This yields: $\nabla(x, z) = \nabla(0, 1) = \frac{-2f}{q\epsilon} + \frac{1}{\epsilon} - 1$. This expression is always negative, given our assumed parameter values.

Lastly, let $B : (x, z) = (0, 0)$. We established earlier that $z(0) > 0$ if $\epsilon < 1$. Which is assumed in the parameter values. Meaning that point B is always under line l. $\nabla(x, z) = \nabla(0, 0) = \frac{1}{\epsilon} - 1$, which is always positive.

We have now shown that with parameter assumptions $q \in (0, 0.1]$, $f = 1, 2, 3$ and $\epsilon \in (0, 1)$ we always have a sign change in the first quadrant, meaning that we *could* find oscillating behaviour here.

Let's take a quick look at the cases where $f = 0$ or $q = 0$. Because of how we defined line l, we have not yet concluded anything for these cases.

- If $f = 0$, $\nabla(x, z)$ becomes $h : \frac{1}{\epsilon}(-2x + 1 - \epsilon)$. This is a linear line, meaning that if $h(0) > 0$ there is a sign change. $h(0) > 0$ if $\epsilon < 1$. There is a sign change around line h for our assumed parameter values.
- If $q = 0$, $\nabla(x, z)$ becomes $k : \frac{1}{\epsilon}\left(\frac{-fz}{x} + \frac{fzx}{x^2} - 2x + 1 - \epsilon\right)$. Simplifying line k gives line h, meaning that there is also a sign change for $q = 0$.

5.4.2. Poincaré-Bendixson

Now that we know oscillating behaviour is a possibility, we can prove that it exists. Recall the Poincaré-Bendixson theorem from Section 3.3:

Theorem 5.2 (Poincaré-Bendixson) *Suppose that:*

1. *R is a closed, bounded subset of the plane;*
2. $\dot{x} = \boldsymbol{f}(\boldsymbol{x})$ *is a continuously differentiable vector field on an open set containing R;*
3. *R does not contain any fixed points, and*
4. *There exists a trajectory C that is 'confined' in R; C starts in R and stays in R for all future time.*

Then, either C is a closed orbit, or it spirals toward a closed orbit as $t \rightarrow \infty$. In either case, R contains a closed orbit. [8]

To prove that there exists a confined trajectory C, we will construct this trapping region R. We will show that the vector field on the boundary of R always points inward, ensuring that all trajectories in R are confined. If we then arrange that there are no fixed points in region R, attracting solutions, we ensure that there is a closed orbit in R.

To construct the trapping region R, we want the nullclines of the nondimensionalised system (5.7). These nullclines already provide us with some information about the vector field.

$$\dot{x} = 0 \rightarrow z = \frac{(x+q)(x-x^2)}{f(x-q)} \tag{5.9}$$

$$\dot{z} = 0 \rightarrow z = x \tag{5.10}$$

On the x-nullcline we know that the vector field is vertical, on the z-nullcline we know that the vector field is horizontal. In the figure below you see the nullclines with their direction.

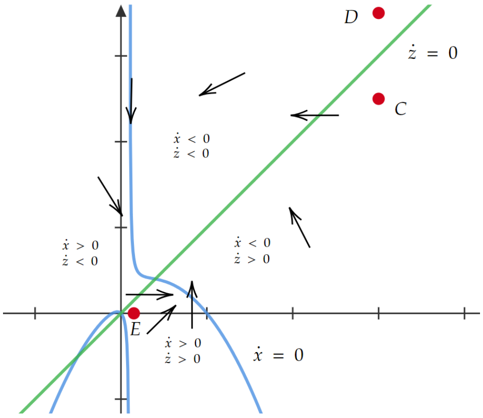


Figure 5.4: Nullclines of the system.

In Figure 5.4 you see that the nullclines cross three times, meaning that there are three equilibrium points. We already saw this when analysing the model. Here you can see clearly that one equilibrium point has negative x and z values and that one equilibrium point is located at the origin. We will only focus on the equilibrium point located in the first quadrant.

We need to figure out how the derivatives behave around the nullclines, the orange dashed lines you see in Figure 5.4. For this we take a point below and above each nullcline and check the sign of the derivative (5.7) in those points. We use point C, D and E from Figure 5.4.

- $C : (x,z) = (5,3)$; $\dot{z} = 2 \rightarrow \dot{z} > 0$ and $\dot{x} \approx -20 - 3f \rightarrow \dot{x} < 0$.
- $D : (x,z) = (5,7)$; $\dot{z} = -2 \rightarrow \dot{z} < 0$.
- $E : (x,z) = (0.1,0)$; $\dot{x} \approx x - x^2 \rightarrow \dot{x} > 0$ for $x \in (0,1)$. As the x-nullcline crosses the x-axis at $x = 1$, this always holds.

Now that we know how the vector field moves, we can start setting up the trapping region. In Figure 5.5 you can see the trapping region drawn by the orange dashed lines. The trapping cannot contain an equilibrium point, we will need to keep this out of it. From Section 5.3 we know that the positive equilibrium point is $(x_{eq}, z_{eq}) = (x_{eq}, x_{eq}) = (\frac{1}{2}(\sqrt{2q - 2f + 6fq + f^2 + q^2 + 1} - q - f + 1), x_{eq})$. Let's eliminate this point from the trapping region. The circle around the equilibrium point is very small, but is drawn somewhat larger for visual purposes. In Figure 5.5 four points are shown: points P, Q, R and S. To make sure that the vector field always moves inward, we need to check the derivatives at these points, as these points are all located at a point of change in direction of the trapping region. For all four points we need to check whether the vector field moves inward here.

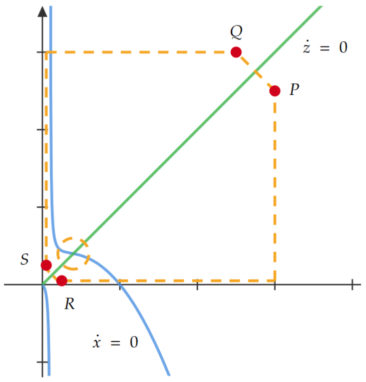


Figure 5.5: Trapping region of the system with points P, Q, R and S.

Let us look at $\dot{x}$ and $\dot{z}$ for the points P and Q. Because we are on the right of the trapping region, we will assume x to be large.

- P: $(x, z) = (x+1, z-1) = (x+1, x-1)$. Let x large, we then get a simplified $\dot{x}$.

$$\dot{z} = 2 \quad \left| \begin{aligned} \dot{x} &= \frac{1}{\epsilon}\left(-x^2 - x - \frac{f(x^2 - qx + q - 1)}{x + q + 1}\right) \\ \dot{x} &\approx \frac{-x^2}{\epsilon} \end{aligned} \right| \quad \frac{\dot{z}}{\dot{x}} = \frac{2\epsilon}{-x^2}$$

- Q: $(x, z) = (x-1, z+1) = (x-1, x+1)$. Let x large.

$$\dot{z} = -2 \quad \left| \begin{aligned} \dot{x} &= \frac{1}{\epsilon}\left(-x^2 + 3x - 2 - \frac{f(x^2 - qx - q - 1)}{x + q - 1}\right) \\ \dot{x} &\approx \frac{-x^2}{\epsilon} \end{aligned} \right| \quad \frac{\dot{z}}{\dot{x}} = \frac{-2\epsilon}{-x^2}$$

Let us now look at point R and S, where point R and S are very close to the axes themselves with a distance of $\hat{\epsilon}$. Let x then approach $\hat{\epsilon}$.

- R: $(x, z) = (x+\hat{\epsilon}, z-\hat{\epsilon}) = (x+\hat{\epsilon}, x-\hat{\epsilon})$. Let $x \to \hat{\epsilon}$, we then get a simplified expression for $\dot{x}$. $\frac{\dot{z}}{\dot{x}}$ becomes slightly larger than 1, as we divide a number by itself plus a little addition.

$$\dot{z} = 2\hat{\epsilon} \quad \left| \begin{aligned} \dot{x} &= \frac{1}{\epsilon}(x - x^2 - 2x\hat{\epsilon} + \hat{\epsilon} - (\hat{\epsilon})^2 - \frac{f(x^2 - qx + q\hat{\epsilon} - (\hat{\epsilon})^2)}{x + \hat{\epsilon} + q} \\ \dot{x} &= \frac{1}{\epsilon}(-4(\hat{\epsilon})^2 + 2\hat{\epsilon}) \end{aligned} \right| \quad \frac{\dot{z}}{\dot{x}} = \frac{2\hat{\epsilon}}{\frac{1}{\epsilon}(-4(\hat{\epsilon})^2 + 2\hat{\epsilon})}$$

- S: $(x, z) = (x-\hat{\epsilon}, z+\hat{\epsilon}) = (x-\hat{\epsilon}, x+\hat{\epsilon})$. Let $x \to \hat{\epsilon}$. $\frac{\dot{z}}{\dot{x}}$ becomes slightly smaller than -1.

$$\dot{z} = -2\hat{\epsilon} \quad \left| \begin{aligned} \dot{x} &= \frac{1}{\epsilon}(x - x^2 + 2x\hat{\epsilon} - \hat{\epsilon} - (\hat{\epsilon})^2 - \frac{f(x^2 - qx - q\hat{\epsilon} - (\hat{\epsilon})^2)}{x - \hat{\epsilon} + q} \\ \dot{x} &= \frac{f}{\epsilon}2\hat{\epsilon} \end{aligned} \right| \quad \frac{\dot{z}}{\dot{x}} = \frac{2\hat{\epsilon}}{\frac{f}{\epsilon}2\hat{\epsilon}}$$

We now have the derivatives in both directions for all four points. The derivatives in x-direction of points P and Q are very large in the negative direction, accompanied with a small z-direction. Meaning that both these points point inward into the trapping region. The derivatives for points R and S are more subtle. $\frac{\dot z}{\dot x}_R$ is slightly larger than 1, meaning that the points follow the diagonal orange dashed line between points S and R, but slightly more inward. $\frac{\dot z}{\dot x}_S$ is slightly smaller than -1, meaning that it points slightly more inward too. To visualise this, the points and its place are shown in Figure 5.6 with the directions of the vector field.

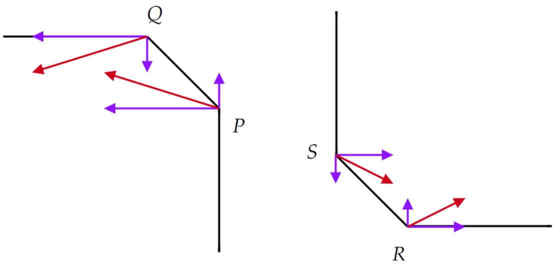


Figure 5.6: Magnified corners of the trapping region to showcase the direction of the vector field at the four points P, Q, R and S. The purple vectors show the separate directions of the x-derivative and the z-derivative. The red vectors show the combined direction of the vector field for said points.

We now know how the vector field behaves on all vertical and horizontal lines, and on the four points we just investigated. We also have to make sure that we stay inward on the diagonal lines we used to make the trapping region. Let the diagonal line between points P and Q be line v and the line between points R and S be line w, as can be seen in Figure 5.7. If we show that the vector field points inward on lines v and w, we have shown that the vector field stays inward on all boundaries of the trapping region we defined. To check this, we need to set up the expressions of lines v and w.

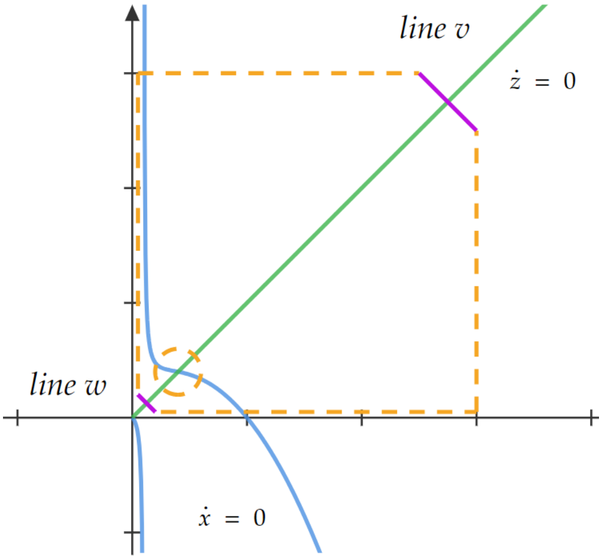


Figure 5.7: Trapping region of the system with lines v and w.

- line v: $(x, -x+\hat{x})$, $x \in [\hat{x}-1, \hat{x}+1]$. For x large we get the following:

$$\dot z = 2x - \hat x \;\left|\; \begin{array}{l} \dot x = \frac{1}{\epsilon}(-x^2 + x + \frac{f(x^2 - x(q+\hat x+\hat x q)}{x+q}) \\ \dot x \approx \frac{-x^2}{\epsilon} \end{array} \;\right|\; \frac{\hat z}{\hat x} = \frac{\epsilon \hat x}{-x^2} = \frac{\epsilon}{-x}$$

- line w: $(x, -x+\hat{\epsilon})$, $x \in [0, \hat{\epsilon}]$. For $x \to \hat{\epsilon}$ we get the following:

$$\begin{array}{r} \dot z = 2x - \hat\epsilon \\ (x=0) \to \dot z = -\hat\epsilon \\ (x=\hat\epsilon) \to \dot z = \hat\epsilon \end{array} \;\left|\; \begin{array}{l} \dot x = \frac{1}{\epsilon}(-x^2 + x + \frac{f(x^2 - x(q+\hat\epsilon+\hat\epsilon q)}{x+q}) \\ \dot x = \frac{1}{\epsilon}(-(\hat\epsilon)^2 + \hat\epsilon) \end{array} \;\right|\; \begin{array}{l} (x=0) \to \frac{\dot z}{\dot x} = \frac{-\hat\epsilon}{\frac{1}{\epsilon}(-\hat\epsilon^2+\hat\epsilon)} \\ (x=\hat\epsilon) \to \frac{\dot z}{\dot x} = \frac{\hat\epsilon}{\frac{1}{\epsilon}(-\hat\epsilon^2+\hat\epsilon)} \end{array}$$

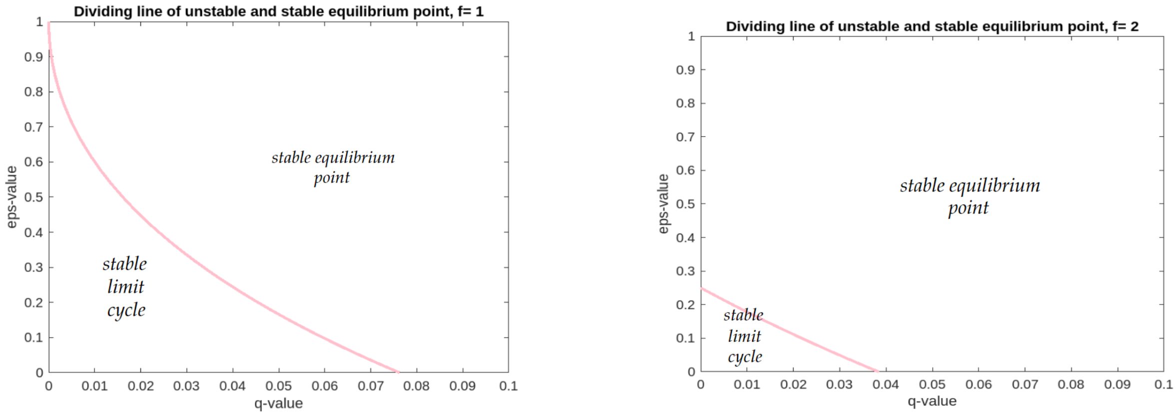
Line v is always pointing inwards for large x on the trapping region. Line w has a sign switch in the line itself, as it crosses the z-nullcline. For both sides of the line we see that the derivative points inward, as we have $\frac{\dot{z}}{\dot{x}}_{x\to 0}$ being slightly smaller than -1 and $\frac{\dot{z}}{\dot{x}}_{x\to \hat{\epsilon}}$ being slightly bigger than 1.

Now that we have shown that all boundaries of the trapping region point inward, we need to figure out when the equilibrium point in the trapping region is an unstable point. Only if this equilibrium point is unstable we can have a stable limit cycle in our trapping region. The equilibrium point $(x_{eq}, z_{eq}) = (x_{eq}, x_{eq}) = (\frac{1}{2}(\sqrt{2q-2f+6fq+f^2+q^2+1}-q-f+1), x_{eq})$ has to be unstable. We know that an equilibrium point is unstable if the eigenvalues of the Jacobian evaluated at this equilibrium point have a positive real part. Because we are working in a two dimensional system, this is the case if $Trace(J_{eq}) > 0$. The Jacobian of the system evaluated at our equilibrium point is stated in (5.6). The trace can be easily deduced. We will do the solving of $Trace(J_{eq}) > 0$ for every separate f-value, as this will make the computations more organised. To know where the dividing line is between a stable and unstable equilibrium point, we will solve $Trace(J_{eq}) = 0$. For the expressions of the trace and the computations you are referred to Appendix C.1.

f-value	$Trace(J_{eq}) = 0$
$f = 0$	$\epsilon = -q^2 - q - 1$
$f = 1$	$\epsilon = \frac{-q-5\sqrt{q^2+8q}+4}{q+\sqrt{q^2+8q}+4}$
$f = 2$	$\epsilon = \frac{1}{8}(q^2 - (q-11)\sqrt{q^2+14q+1} + 18q + 13)$
$f = 3$	$\epsilon = \frac{1}{12}(q^2 - (q+16)\sqrt{q^2+20q+4} + 26q + 28)$

Table 5.3: Expression of $Trace(J_{eq}) = 0$ to determine when the equilibrium point in the trapping region is unstable, separately for every f-value with f = 0, 1, 2 and 3.

Now that we have the expressions of the dividing line between stable and unstable equilibrium point, we can visualise the expressions for the parameter ranges we assumed. For $f = 0$ and $f = 3$ the necessary parameters to fulfill the trace expressions are not the parameters we assumed. For $f = 1$ and $f = 2$ we can visualise the lines nicely, as to be seen in the figure below.



(a) Dividing line for f = 1.

(b) Dividing line for f = 2.

Figure 5.8: Parameters q versus ϵ and the dividing lines of a stable equilibrium point and a stable limit cycle. Separate plots for f = 1 and f = 2 to clearly show where the oscillating behaviour can take place.

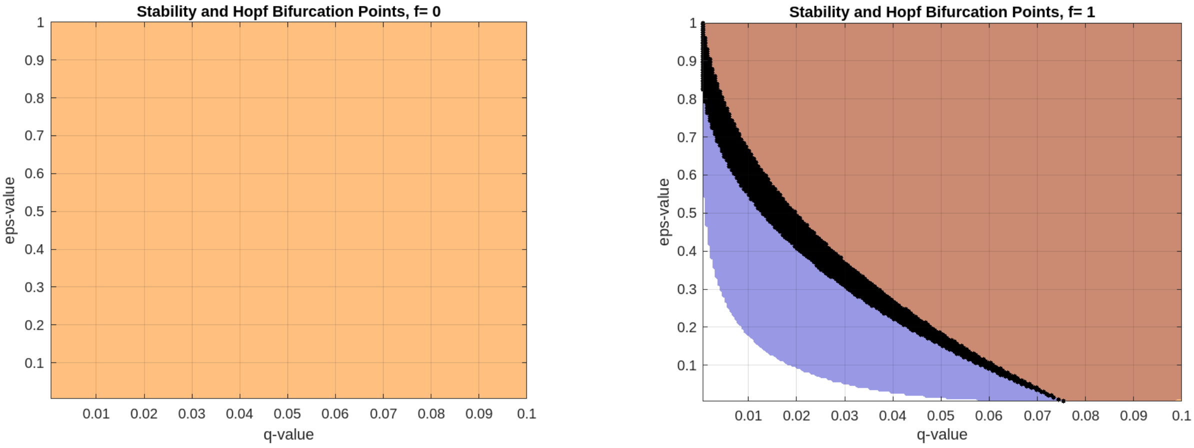
We have constructed a trapping region that contains no equilibrium points and shown that the vector field of the system always moves inward of the trapping region. We have computed when the equilibrium point that lays in the area of the trapping region (but is not in the trapping region) is an unstable equilibrium point, showing that the vector field on the boundary around this point also moves inward of the trapping region. We have shown that our system and the trapping region comply with the Poincaré-Bendixson Theorem and that we thus have a stable limit cycle in our trapping region. We also have computed for which parameter combinations this stable limit cycle can exist.

5.5. Bifurcation Search

Now that we have proved the existence of oscillating behaviour, we can investigate where it takes place. We are back to a two dimensional system, meaning that we can use expression (3.4) to analyse the eigenvalues. In Chapter 3 we could easily find a formula representing the Hopf-Bifurcation locations. Because the Jacobian found in Section 5.3.3 is more complex than the one we had in Chapter 3, this Hopf-Bifurcation locations expression will become much harder to compute. We will first look at the stability and complex regions, like done in Chapter 3 and Chapter 4 using the eigenvalues, then we will find the analytic formulas for the Hopf-Bifurcation locations.

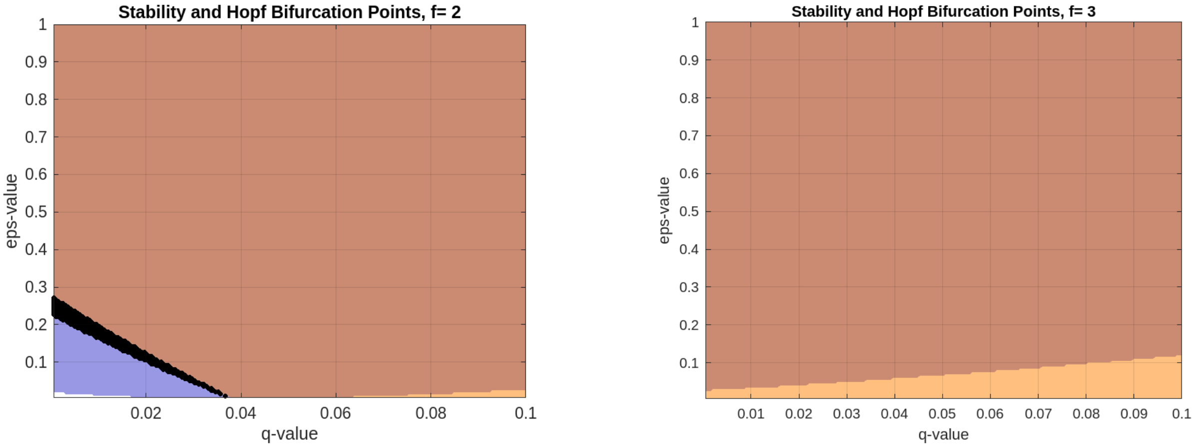
5.5.1. Assessment of Eigenvalues

We want to analyse the eigenvalues to see where a Hopf Bifurcation can take place. We have assumed that $f = 0, 1, 2$ or 3. We will *create* a separate Jacobian for every f-value and analyse these Jacobians separately. To create the stability and complex region plots, parameter grids are evaluated at every grid point. At every parameter combination the eigenvalues of the Jacobian are computed and categorised, answering the question whether they are all negative (creating a stable equilibrium point) or if at least one of them in complex (imaginary eigenvalues). Also an approximation is done where the eigenvalues cross the imaginary axis, so where the real part of the eigenvalues is zero. Because of the grid points this is an approximation and not the exact points yet, but the plots created give a good idea of where the Hopf-Bifurcations can take place.



(a) Regions for f= 0

(b) Regions for f= 1



(c) Regions for f= 2

(d) Regions for f= 3

Figure 5.9: Stability (orange) and complex (blue) regions for all f-values. The black dots represent a Hopf Bifurcation. Because of a threshold setting in Matlab we get some thicker lines.

From Figure 5.9 we can conclude that if $f = 0$ or $f = 3$, there are no Hopf-Bifurcation possibilities. All eigenvalues seem to be stable, meaning that the trace of the Jacobian is always negative. $f = 1$ and $f = 2$ do have Hopf-Bifurcation locations.

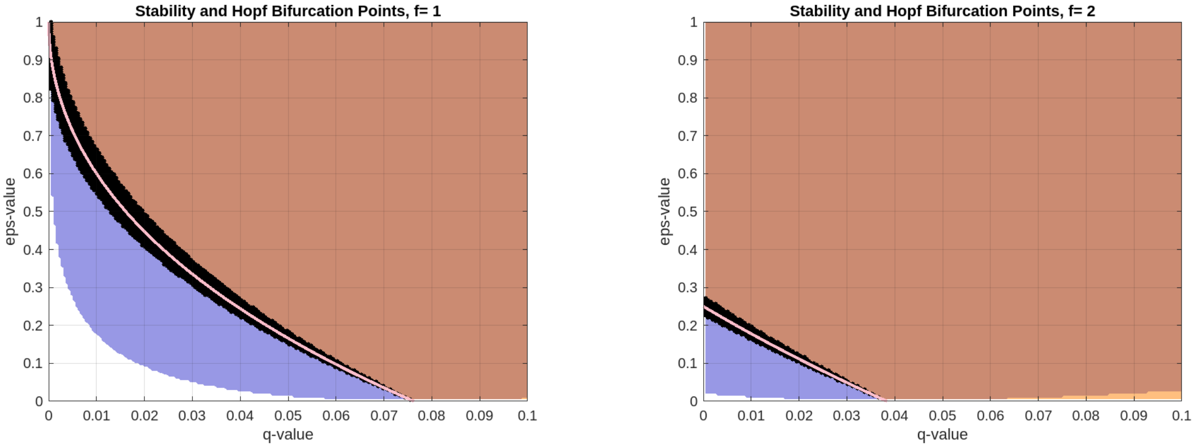
In Section 3.2 we concluded from the eigenvalues expression (3.4) that the model is stable if $Trace(J_{eq}) < 0$ and that there can be a Hopf Bifurcation if $Trace(J_{eq}) = 0$ and $Det(J_{eq}) > 0$. We can then express the parameter q in the parameter ϵ and get an analytical function of where a Hopf Bifurcation can take place. In the Table shown below the ϵ-value and q-value ranges are shown for which the trace is equal to 0 and the determinant is greater than zero, per f-value. For the specific Jacobians, expressions for trace and determinant, and conclusions you are referred to Appendix C.1.

f-value	$Trace(J_{eq}) = 0$	$Det(J_{eq}) > 0$	Hopf-Bifurcation expression
$f = 0$	-	$\epsilon > 0$	-
$f = 1$	$0 < q < \frac{1}{3}(5\sqrt{7} - 13)$	$0 < q < \sqrt{97} - 4,\ \epsilon > 0$	$\epsilon = \frac{-q-5\sqrt{q^2+8q}+4}{q+\sqrt{q^2+8q}+4}$
$f = 2$	$3\sqrt{7} - 8 < q < \frac{3}{39+16\sqrt{6}}$	$q > 0,\ \epsilon > 0$	$\epsilon = \frac{1}{8}(q^2 - (q-11)\sqrt{q^2+14q+1} + 18q + 13)$
$f = 3$	-	$q > 0,\ \epsilon > 0$	-

Table 5.4: Parameter ranges for ϵ and q, where $Trace(J_{eq}) = 0$ and $Det(J_{eq}) > 0$ for all assumed f-values.

In Table 5.4 we see that for $f = 0$ and $f = 3$ there is no possibility of a Hopf-Bifurcation occurrence, as we had seen in Figure 5.9. For both $f = 1$ and $f = 2$ there is a restriction on the q-value, but we have found an expression for the Hopf-Bifurcation locations. We now know where to look for oscillating behaviour and which parameter ranges can give us these results.

Plotting the expressions from Table 5.4 over the stability and complex regions plots, we get the following:



(a) Regions and Hopf-Bifurcation locations with the analytically found line for Hopf locations, for f = 1.

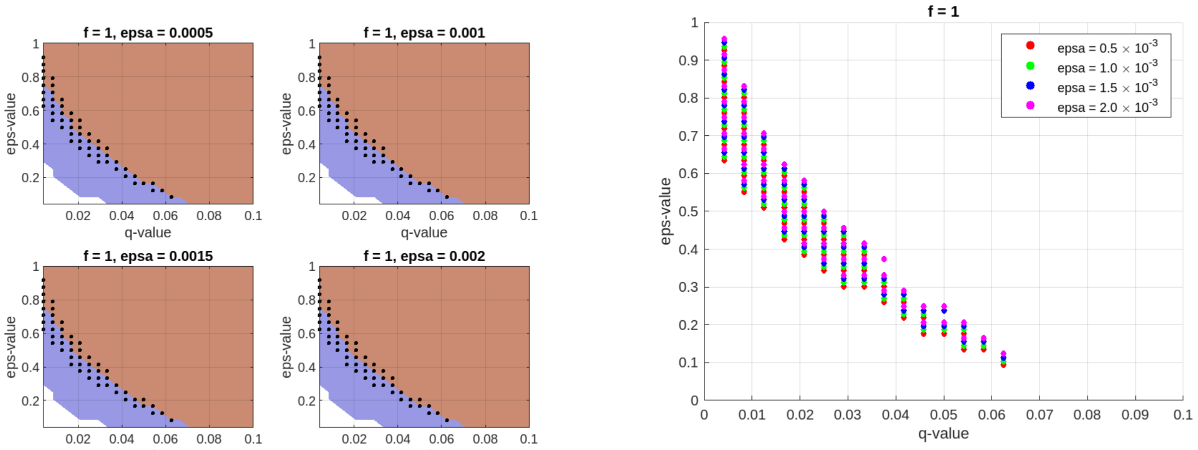
(b) Regions and Hopf-Bifurcation locations with the analytically found line for Hopf locations, for f = 2.

Figure 5.10: Stability (orange) and complex (blue) regions with either f = 1 or f = 2. The black dots represent a Hopf Bifurcation. The pink line through the plot is the analytically found line of the Hopf-Bifurcation locations.

The lines nicely plot over the regions, meaning we have found where the Hopf-Bifurcations take place!

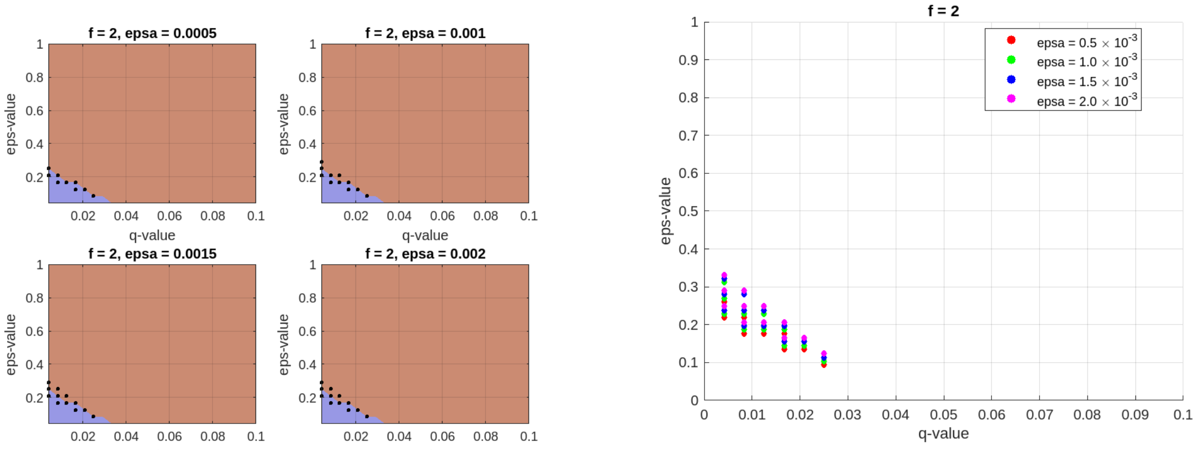
5.5.2. 3-dimensional Hopf Bifurcation

We started with a three dimensional system, which we simplified into a two dimensional system. This was possible because of the relatively small value of ϵ' that multiplied all Y-values in the three dimensional model from Section 5.3.1. It is interesting to see whether this ϵ' influences the Hopf-Bifurcation location. To analyse this, a three dimensional analysis is done on the system. With the same strategy as in Section 5.5.1 stability and complex regions together with Hopf-Bifurcation location plots were made, but now also including the influence of ϵ'. This yields the following result:



(a) f = 1, Hopf-Bifurcation line with different ϵ' values.

(b) f = 1, Hopf-Bifurcation line with different ϵ' values layered over each other.



(c) f = 2, Hopf-Bifurcation line with different ϵ' values.

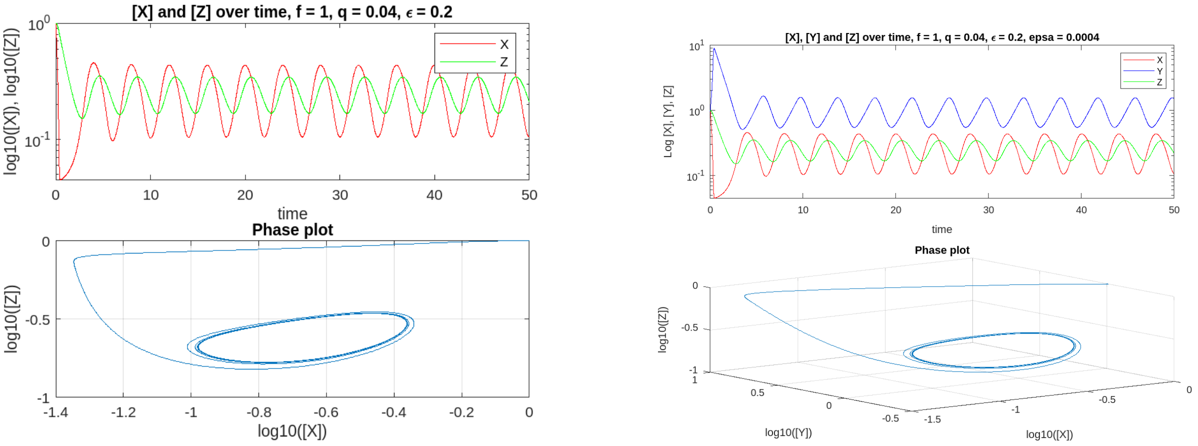
(d) f = 2, Hopf-Bifurcation line with different ϵ' values layered over each other.

Figure 5.11: Stability (orange) and complex (blue) regions for $f = 1$ and $f = 2$. The black line represents the Hopf Bifurcation line. Different ϵ' values show that the influence of ϵ' is negligible.

In Figure 5.11 you can see that the influence of ϵ' is negligible. Especially in the plots where all Hopf-Bifurcation location points are layered. They are plotted with a slight offset to clearly see the four different dots, which explains the slight movement. But all in all there is no influence. To get the correct Hopf-Bifurcation locations ϵ' does not have to be taken into account.

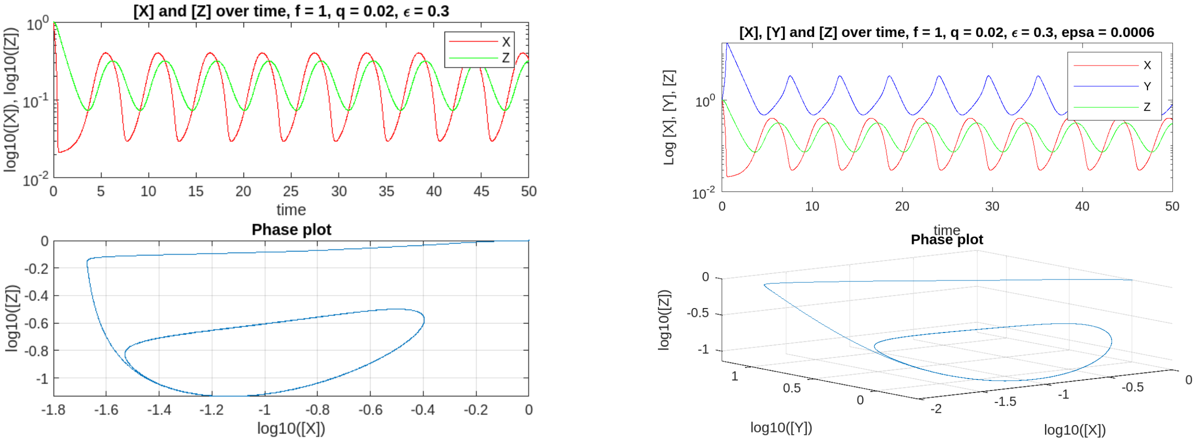
5.5.3. Oscillating Behaviour

Now that we know where to look for oscillating behaviour, we can numerically make this behaviour visible. Let's choose some points from Figure 5.10. If we want to see oscillating behaviour, the parameter values should be chosen under the pink line. Under the pink line the parameter-values result in eigenvalues that have crossed the imaginary axis with an unstable equilibrium point, thus creating a stable limit cycle. For $f = 1$, let us choose $(q,\epsilon) = (0.04, 0.2)$ and $(q,\epsilon) = (0.02, 0.3)$ to see oscillating behaviour. With parameter-values $(q,\epsilon) = (0.06, 0.3)$ we should not see this behaviour. For $f = 2$ let us look at $(q,\epsilon) = (0.02, 0.1)$ and $(q,\epsilon) = (0.01, 0.1)$ for oscillating behaviour and $(q,\epsilon) = (0.06, 0.4)$ for non-oscillating behaviour. To show that the two-dimensional system behaves the same as the three-dimensional system, the concentration and phase plot for the three-dimensional system are showed next to the two-dimensional plots. The ϵ' value is $\frac{2}{10^3}\epsilon$, as shown in Section 5.3.2. After numerical analysis we get the following figures:



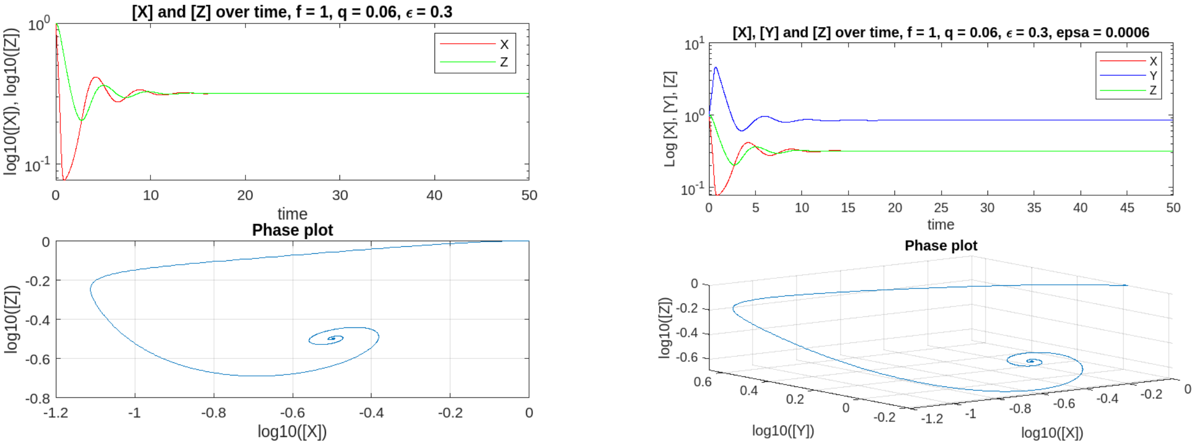
(a) Two-dimensional plots with f = 1, q = 0.04, $\epsilon = 0.2$

(b) Three-dimensional plots with f = 1, q = 0.04, $\epsilon = 0.2$



(c) Two-dimensional plots with f = 1, q = 0.02, $\epsilon = 0.3$

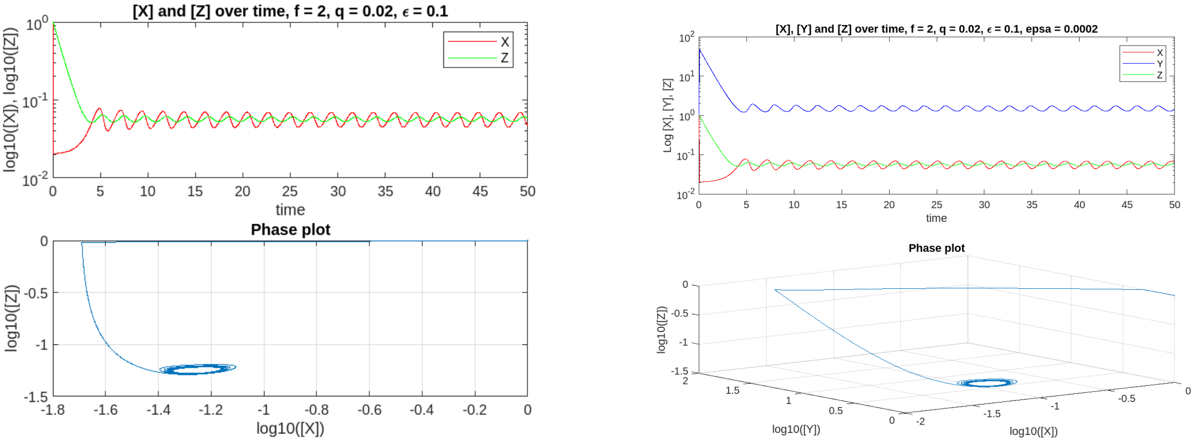
(d) Three-dimensional plots with f = 1, q = 0.02, $\epsilon = 0.3$



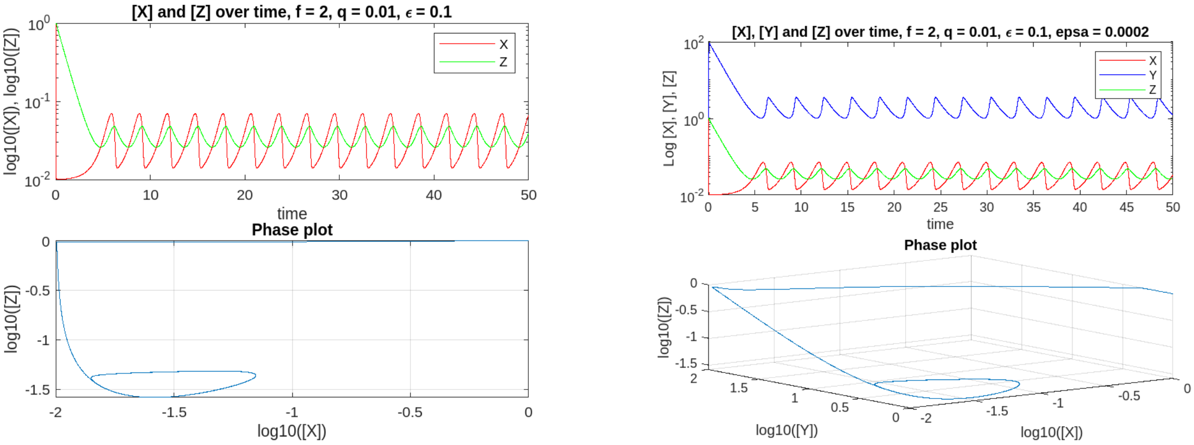
(e) Two-dimensional plots with f = 1, q = 0.06, $\epsilon = 0.3$

(f) Three-dimensional plots with f = 1, q = 0.06, $\epsilon = 0.3$

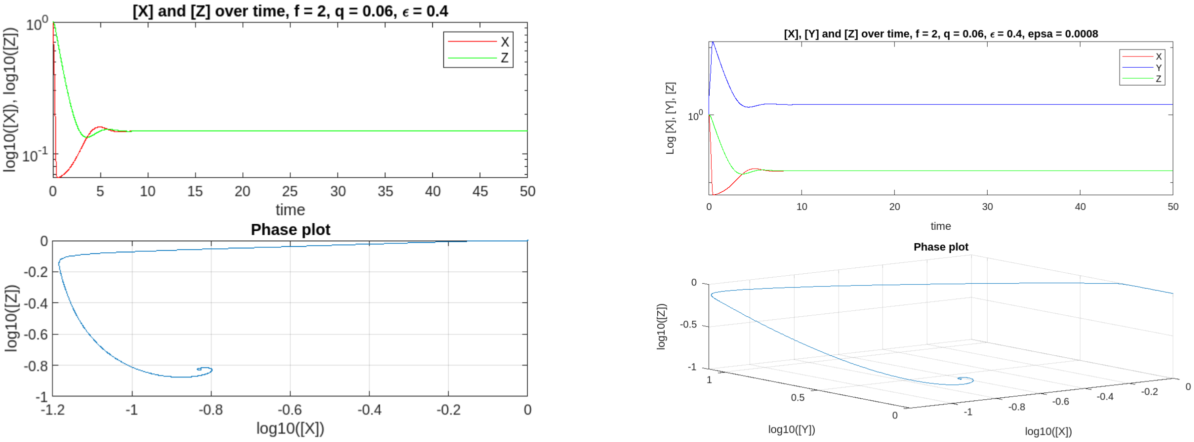
Figure 5.12: Concentration plots and phase plots of three different cases of parameter values of the two- and three-dimensional systems for f = 1. Oscillating behaviour can be seen in plots a - d. Non-oscillating behaviour can be found in plots e and f.



(a) Two-dimensional plots with f = 2, q = 0.02, $\epsilon = 0.1$ (b) Three-dimensional plots with f = 2, q = 0.02, $\epsilon = 0.1$



(c) Two-dimensional plots with f = 2, q = 0.01, $\epsilon = 0.1$ (d) Three-dimensional plots with f = 2, q = 0.01, $\epsilon = 0.1$



(e) Two-dimensional plots with f = 2, q = 0.06, $\epsilon = 0.4$ (f) Three-dimensional plots with f = 2, q = 0.06, $\epsilon = 0.4$

Figure 5.13: Concentration plots and phase plots of three different cases of parameter values of the two- and three-dimensional systems for f = 2. Oscillating behaviour can be seen in plots a - d. Non-oscillating behaviour can be found in plots e and f.

The model behaves exactly as we have predicted. The three-dimensional concentration and phase plots clearly show that the influence of Y is negligible. The concentrations of X and Z do not seem to change at all when Y is involved in the system. Lastly, the parameter values we had predicted to entail oscillating behaviour did so, and the parameter values for which we expected non-oscillating behaviour did not showcase oscillating behaviour. We have found our Hopf-Bifurcation locations.

6

Conclusion and Discussion

The aim of this thesis was to examine two theoretical models of chemical oscillators to identify the parameter combinations for which these models showcase oscillating behaviour, while enhancing the realism of these models. First, the Brusselator model and the Oregonator model were analysed in a batch reactor. Secondly the Brusselator model was considered in a Continuous Stirred-Tank Reactor (CSTR), to enhance the realism of this model. Making for three models in total. To find the oscillating behaviour in these models, an extensive eigenvalue investigation was done. Using the information of the eigenvalues, Hopf-Bifurcations could be located in all three models, indicating oscillating behaviour.

6.1. Looking for Oscillating Behaviour

First, the Brusselator model was analysed in a batch reactor. Two of the components present in the Brusselator were modelled as change of concentration in time. The results as seen in [1] were reproduced, yielding an analytic expression indicating where the Hopf-Bifurcations took place. A proof of existence of periodic solutions was given, making use of Bendixson's Criterion and the Poincaré-Bendixson Theorem. The parameter combinations for which oscillating behaviour can arise were found.

To make the Brusselator model somewhat more realistic, it was modelled in a CSTR instead of a batch reactor. Instead of only modelling two of the components of the Brusselator, all four reactants were modelled. An inflow of two of these molecules was added to the model too. These two changes make for a more realistic representation of the Brusselator chemical reactions. Because four concentrations were modelled now instead of two, the analysis of the eigenvalues became somewhat more complex. The analysis was done in Matlab, making use of certain characteristics of eigenvalues for Hopf-Bifurcations in four dimensional systems. Again the parameter combinations were found, unfortunately not as an analytical expression.

Lastly, the Oregonator model was analysed in a batch reactor. The model was reduced from a three dimensional model to a two dimensional model. A proof of existence of limit cycles was carried out again, making use of the same statements as for the Brusselator. An extensive eigenvalue investigation was done, finding analytic expressions for Hopf-Bifurcation occurrences. Thus, oscillating behaviour was found again.

Thus, in all three models analysed oscillating behaviour was found and the parameter combinations for which these periodic solutions can occur were presented as clearly as possible.

6.2. Recommendations for Further Research

The object of this thesis was to extend the theoretical models to make them behave more realistically. When an inflow of two out of four components of the Brusselator was added, oscillating behaviour was still found. If an outflow of these molecules was added too, however, there seemed to be no possibility for periodic solutions. An interesting question would be why this was the case and how the model could be changed. Another addition could be to model a total outflow of all fluids instead of only the two newly modelled components, to keep the total reaction volume constant.

The extension of the Oregonator model to a CSTR was also examined, but the system did not seem to have any equilibrium points. Only when both inflow and outflow were added did some quite complicated equilibrium points appear. This raises the question of how such equilibrium points arise and what potential solutions to this issue could be.

Another realistic extension would be to add the reversed reactions to the model. Up till now we have modelled irreversible reactions, but in practise these reactions are not all irreversible. An attempt was made for the Brusselator model, but the differential equations did not behave as expected and oscillating behaviour was not found. It would be recommended to make use of earlier published findings to get an idea of how to approach such an addition to these models. An approach on how to extend the Oregonator model to a reversible model could, for example, be read in [12].

Lastly, we assumed in all models that the contents in the reactor were perfectly mixed. If we would abandon this assumption and would add the dimension of space to the system, we could transform the models into reaction-diffusion models. A reaction-diffusion model can track the change of concentration in time and space, making the models far more realistic. This addition is possible for both the Brusselator and the Oregonator. If the space dimension is added, interesting patterns arise. These patterns are the base of Turing patterns and morphogenesis, the process of the shaping of organisms [13]. A great example of this would be the forming of plants and their leaves and flowers. A reaction-diffusion model could, for example, explain how a plant gets its specific flower shape [14]. It would be very interesting to add a spatial dimension to the Brusselator and Oregonator models and examine the arising patterns.

Bibliography

[1] L.-P. et. al, “Pedagogical approach to the modeling and simulation of oscillating chemical systems with modern software: The brusselator model,” *ChemicalEducation*, vol. 95, pp. 758–766, 2018.

[2] R. W. e. Sterner, “The conservation of mass,” *Nature Education Knowledge 3(10):20*, 2011.

[3] G. Schmitz, “Historical overview of the oscillating reactions. contribution of professor slobodan anić,” *Reaction Kinetics, Mechanisms and Catalysis*, vol. 118, no. 1, p. 5–13, 2015.

[4] R. J. Field and R. M. Noyes, “Oscillations in chemical systems. iv. limit cycle behavior in a model of a real chemical reaction,” *The Journal of Chemical Physics*, vol. 60, no. 5, p. 1877–1884, 1974.

[5] A. Zhabotinsky, “Belousov-zhabotinsky reaction,” *Scholarpedia*, vol. 2, no. 9, p. 1435, 2007.

[6] R. Field, “Oregonator,” *Scholarpedia*, vol. 2, no. 5, p. 1386, 2007.

[7] D. S. Schmidt, “Hopf’s bifurcation theorem and the center theorem of liapunov with resonance cases,” *Journal of Mathematical Analysis and Applications*, vol. 63, no. 2, p. 354–370, 1978.

[8] S. H. Strogatz, *Nonlinear dynamics and chaos.* 1994.

[9] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems.* Springer Berlin Heidelberg, 1996.

[10] L. D. Schmidt, *The Engineering of Chemical Reactions.* 2004.

[11] S. Bosi and D. Desmarchelier, “Local bifurcations of three and four-dimensional systems: A tractable characterization with economic applications,” *Mathematical Social Sciences*, vol. 97, p. 38–50, 2019.

[12] R. J. Field, “Limit cycle oscillations in the reversible oregonator,” *The Journal of Chemical Physics*, vol. 63, no. 6, p. 2289–2296, 1975.

[13] “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, p. 37–72, 1952.

[14] I. Burda, A. C. Martin, A. H. Roeder, and M. A. Collins, “The dynamics and biophysics of shape formation: Common themes in plant and animal morphogenesis,” *Developmental Cell*, vol. 58, no. 24, p. 2850–2866, 2023.

A

Appendix A

A.1. Brusselator - Code for Numerical Analysis

```
%% Concentration and phase plot

% Defining the variables
A = 0.5;
B = 1.5;
k = [1,1,1,1];
x0 = 0.2;
y0 = 0.2;
dt = 0.01;
steps = 5000;

time_grid = linspace(0,5000, steps);
x_grid = zeros(1,steps);
y_grid = zeros(1,steps);
xn = x0;
yn = y0;

% Forward Euler method for X and Y
for i = 1:steps
    x_grid(i) = xn;
    y_grid(i) = yn;
    xn = next_x(xn, yn, A, B, k, dt);
    yn = next_y(xn, yn, A, B, k, dt);
end

figure;
plot(time_grid, x_grid, ’r’, time_grid, y_grid, ’b’)
xlabel(’time’);
ylabel(’Concentration of X and Y’);
title(’Concentration plot of X and Y over time’);

figure;
plot(x_grid, y_grid)
xlabel(’X-concentration’);
ylabel(’Y-concentration’);
title(’Phase plot of the X-concentration versus the Y-concentration’);

% Forward Euler functions for getting the next X and Y functions

function result = next_x(xn, yn, A, B, k, dt)
    result = xn + dt*(k(1)*A - k(2)*B*xn + k(3)*xn^2*yn - k(4)*xn);
end

function result = next_y(xn, yn, ~, B, k, dt)
```

```
    result = yn + dt*(k(2)*B*xn - k(3)*xn^2*yn);
end

%% Concentration and phase plot, with Q and P

% defining the variables

P = 10;
Q = 1;
x0 = 0;
y0 = 0;
dt = 0.01;
steps = 5000;

time_grid = linspace(0,5000, steps);
x_grid = zeros(1,steps);
y_grid = zeros(1,steps);
xn = x0;
yn = y0;

% Forward Euler method for X and Y
for i = 1:steps
    x_grid(i) = xn;
    y_grid(i) = yn;
    xn = next_x_sym(xn, yn, P, Q, dt);
    yn = next_y_sym(xn, yn, P, Q, dt);
end

figure;
plot(time_grid, x_grid, 'r', time_grid, y_grid, 'b')
xlabel('time');
ylabel('Concentration of X and Y');
title('Concentration plot of X and Y over time');

figure;
plot(x_grid, y_grid)
xlabel('X-concentration');
ylabel('Y-concentration');
title('Phase plot of the X-concentration versus the Y-concentration');

% Functions for getting the next X and Y values

function result = next_x_sym(xn, yn, P, Q, dt)
    result = xn + dt*(Q - P*xn - xn + xn^2*yn);
end

function result = next_y_sym(xn, yn, P, ~, dt)
    result = yn + dt*(P*xn - xn^2*yn);
end
```

A.2. Brusselator - Code for Eigenvalue Analysis

```
%% Plotting Q versus P with complex and stability regions.
% Marking the Hopf Bifurcations.

% Initialising a Q and P grid
q_limit = 5;
p_limit = 10;

q_grid = linspace(0, q_limit, q_limit * 100);
p_grid = linspace(0, p_limit, p_limit * 50);

[Q, P] = meshgrid(q_grid, p_grid);

% Setting up all other grids
imaginary = false(length(p_grid), length(q_grid));
stable1 = false(length(p_grid), length(q_grid));
stable2 = false(length(p_grid), length(q_grid));
```

```

% Getting stability, and imaginary True/False for every (q,p) with
% eigenvalue expressions

for i = 1:length(q_grid)
    for j = 1:length(p_grid)
        if ((p_grid(j) - 1 - q_grid(i)^2)^2 - 4 * q_grid(i)^2) < 0
            imaginary(j, i) = true;
        end
        if (p_grid(j) - 1 - q_grid(i)^2) + sqrt(((p_grid(j) - 1 - q_grid(i)^2))^2 - 4 * q_grid(i)^2) < 0
            stable1(j, i) = true;
        end
        if (p_grid(j) - 1 - q_grid(i)^2) - sqrt(((p_grid(j) - 1 - q_grid(i)^2))^2 - 4 * q_grid(i)^2) < 0
            stable2(j, i) = true;
        end
    end
end

stable = stable1 & stable2;

figure;
contourf(Q, P, imaginary, [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
hold on;
contourf(Q, P, stable, [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);

% Plot the line P = 1 + Q^2
q_line = 0:0.1:q_limit;
p_line = min(1 + q_line.^2, p_limit);
h_line = plot(q_line, p_line, 'k', 'LineWidth', 2);

xlabel('Q-value');
ylabel('P-value');
title('Stability for certain Q and P values');
grid on;

% Making a legend
patch1 = patch([NaN NaN NaN NaN], [NaN NaN NaN NaN], 'b');
patch2 = patch([NaN NaN NaN NaN], [NaN NaN NaN NaN], 'r');
patch1.Visible = 'off';
patch2.Visible = 'off';

legend([patch1, patch2, h_line], {'Imaginary region', 'Stable region', 'Stability line'}, 'Location', 'SouthEast');

%% Eigenvalue plot (real and complex seperately) for fixed Q

% Setting up parameters
Q = 1;
p_limit = 6;
P = linspace(0,p_limit, p_limit*20);

eigenvalues = zeros(1, p_limit*20);

% Getting eigenvalues at all necessary points

figure;
count = 1;
for p = P
    J_eq = [p-1 Q^2; -p -Q^2];
    eigenwaarde = eig(J_eq);
    eigenvalues(count) = min(eigenwaarde);
    count = count + 1;
end

subplot(2,1,1);
plot(P, real(eigenvalues), 'b');
hold on;
```

```
line([min(P), max(P)], [0, 0], 'Color', 'k', 'LineStyle', '--');
hold off;
xlabel('P-value');
ylabel('Real Part');
title(['Q = ' num2str(Q), ', the eigenvalues']);

subplot(2,1,2);
plot(P, imag(eigenvalues), 'r');
xlabel('P-value');
ylabel('Imaginary Part');
```

B

Appendix B

B.1. Brusselator with CSTR - Code for Numerical Analysis

```
    %% Concentration and phase plots with original parameters

% Defining the variables
Q = 4;
A_0 = 1;
B_0 = 1;
k = [1, 1, 1, 1];
x0 = 0;
y0 = 0;
a0 = A_0;
b0 = B_0;
dt = 0.01;
steps = 2500;

time_grid = linspace(0,2500, steps);
x_grid = zeros(1,steps);
y_grid = zeros(1,steps);
a_grid = zeros(1,steps);
b_grid = zeros(1,steps);
xn = x0;
yn = y0;
an = a0;
bn = b0;

% Forward Euler method to get concentration vs time plot of X, Y, A and B;
% And a phase plot of X and Y.
for i = 1:steps
    x_grid(i) = xn;
    y_grid(i) = yn;
    a_grid(i) = an;
    b_grid(i) = bn;
    xn = next_x(xn, yn, an, bn, k, Q, A_0, B_0, dt);
    yn = next_y(xn, yn, an, bn, k, Q, A_0, B_0, dt);
    an = next_a(xn, yn, an, bn, k, Q, A_0, B_0, dt);
    bn = next_b(xn, yn, an, bn, k, Q, A_0, B_0, dt);
end

figure;
plot(time_grid, x_grid, 'r', time_grid, y_grid, 'b', time_grid, a_grid, 'g', time_grid, b_grid, 'y')
xlabel('time');
ylabel('Concentration of X and Y');
title('Concentration plot of X and Y over time');
legend('X', 'Y', 'A', 'B');
```

```

figure;
plot(x_grid, y_grid)
xlabel('X-concentration');
ylabel('Y-concentration');
title('Phase plot of the X-concentration versus the Y-concentration');

% Forward Euler functions to get the next value

function result = next_x(xn, yn, an, bn, k, ~, ~, ~, dt)
    result = xn + dt*(k(1)*an - k(2)*bn*xn + k(3)*xn^2*yn - k(4)*xn);
end

function result = next_y(xn, yn, ~, bn, k, ~, ~, ~, dt)
    result = yn + dt*(k(2)*bn*xn - k(3)*xn^2*yn);
end

function result = next_a(~, ~, an, ~, k, Q, A_0, ~, dt)
    result = an + dt*(-k(1)*an + Q*A_0);
end

function result = next_b(xn, ~, ~, bn, k, Q, ~, B_0, dt)
    result = bn + dt*(-k(2)*bn*xn + Q*B_0);
end

%% Concentration and phase plot with parameters p r en s,
% the nondimensionalised system

% Defining the parameters
A_0 = 1;
B_0 = 1;
dt = 0.01;
steps = 10000;
x0 = 0;
y0 = 0;
a0 = A_0;
b0 = B_0;

% Set (s,p,r) values. You can set multiple.
s_values = 2;
p_values = [0.5, 1, 2];
r_values = 1.5;

% Iterate over parameter combinations

for p_index = 1:numel(p_values)
    for s_index = 1:numel(s_values)
        for r_index = 1:numel(r_values)
            p = p_values(p_index);
            s = s_values(s_index);
            r = r_values(r_index);

            % Initialize grids
            time_grid = linspace(0, steps*dt, steps);
            x_grid = zeros(1, steps);
            y_grid = zeros(1, steps);
            a_grid = zeros(1, steps);
            b_grid = zeros(1, steps);
            xn = x0;
            yn = y0;
            an = A_0;
            bn = B_0;

            % Forward Euler steps
            for i = 1:steps
                x_grid(i) = xn;
                y_grid(i) = yn;
                a_grid(i) = an;
                b_grid(i) = bn;
                xn = next_x_sym(xn, yn, an, bn, p, r, s, dt);
                yn = next_y_sym(xn, yn, an, bn, p, r, s, dt);
```

```
                an = next_a_sym(xn, yn, an, bn, p, r, s, dt);
                bn = next_b_sym(xn, yn, an, bn, p, r, s, dt);
            end

            figure;
            plot(time_grid, x_grid, 'r', time_grid, y_grid, 'b', time_grid, a_grid, 'g', time_grid, b_grid, 'y')
            xlabel('time');
            ylabel('Concentration of X and Y');
            title(['Concentration plot (p=', num2str(p), ', s=', num2str(s), ', r=', num2str(r), ')']);
            legend('X', 'Y', 'A', 'B');

            figure;
            plot(x_grid, y_grid)
            xlabel('X-concentration');
            ylabel('Y-concentration');
            title(['Phase plot (p=', num2str(p), ', s=', num2str(s), ', r=', num2str(r), ')']);
        end
    end
end

% Forward Euler functions of the nondimensionalised system to get the next
% value

function result = next_x_sym(xn, yn, an, bn, p, ~, ~, dt)
    result = xn + dt*(p*an - xn - bn*xn + xn^2*yn);
end

function result = next_y_sym(xn, yn, ~, bn, ~, ~, ~, dt)
    result = yn + dt*(bn*xn - xn^2*yn);
end

function result = next_a_sym(~, ~, an, ~, p, r, ~, dt)
    result = an + dt*(r - p*an);
end

function result = next_b_sym(xn, ~, ~, bn, ~, r, s, dt)
    result = bn + dt*(r - s*bn*xn);
end
```

B.2. Brusselator with CSTR - Code for Eigenvalue Analysis

```
    %% Stability analysis of the nondimensionalised system

% Symbolic variables
syms A B X Y p r s

% Nondimensionalized differential equations
equation_strings = { 'A*p - X - B*X + X^2*Y', ...
                     'B*X - X^2*Y', ...
                     'r - A*p', ...
                     'r - B*X*s' };

equations = str2sym(equation_strings);

% Getting the Jacobian
Jacobian = jacobian([equations(1), equations(2), equations(3), equations(4)],[X, Y, A, B]);

% Defining the system of differential equations to compute the equilibrium
% points
F(1) = A*p - X - B*X + X^2*Y;
F(2) = B*X - X^2*Y;
F(3) = r - A*p;
F(4) = r - B*X*s;

```

```
% Solve the system of differential equations symbolically
sol = solve(F == 0, [A, B, X, Y]);

% Display the equilibrium points in terms of the constants
disp('Equilibrium points:');
disp(sol);

% Filling in the equilibrium point into the Jacobian matrix
equilibrium_point = [r, 1/(s*r), r/p, 1/s];
J_eq = subs(Jacobian, [X, Y, A, B], equilibrium_point);

%% Stability and complex regions plot with Hopf-Bifurcation location dots
% and eigenvalue plots where one of the two parameters is fixed, to
% visualise the crossing of the eigenvalues.

%% Plotting parameter s versus r, fixed p-value.

% Initialising all necessary variables and grids
syms p r s

s_limit = 4;
r_limit = 4;
p_var = 1;

s_grid = linspace(0, s_limit, s_limit*25);
r_grid = linspace(0, r_limit, r_limit*25);
s_grid = s_grid(2:end);
r_grid = r_grid(2:end);

eigenvalues = cell(length(s_grid), length(r_grid));
trace = zeros(length(s_grid), length(r_grid));
s_2 = zeros(length(s_grid), length(r_grid));
s_3 = zeros(length(s_grid), length(r_grid));
det = zeros(length(s_grid), length(r_grid));

% Looping through all combinations of parameters while computing the
% eigenvalues, and preparing all grids for checking the Hopf conditions.

for i = 1:length(s_grid)
    for j = 1:length(r_grid)
        J_eq_copy = J_eq;
        J_eq_copy = subs(J_eq_copy, [p, s, r], [p_var, s_grid(i), r_grid(j)]);
        J_eq_num = double(J_eq_copy);
        eigenvalues{i,j} = eig(J_eq_num);
        trace(i,j) = sum(eigenvalues{i,j});
        if trace(i,j) == 0
            trace(i,j) = 0.0001;
        end
        s_2(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2) + eigenvalues{i,j}(1)*eigenvalues{i,j}(3) + eigenvalues{i,j}(1)*eigenvalues{i,j}(4) + eigenvalues{i,j}(2)*eigenvalues{i,j}(3) + eigenvalues{i,j}(2)*eigenvalues{i,j}(4) + eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
        s_3(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(3) + eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(4) + eigenvalues{i,j}(1)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4) + eigenvalues{i,j}(2)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
        det(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
    end
end

% Creating all grids necessary for the plot

imaginary = false(length(s_grid), length(r_grid));
stable = false(length(s_grid), length(r_grid));
hopf = zeros(length(s_grid), length(r_grid));
double_hopf = ones(length(s_grid), length(r_grid));

th = 0.05;

% Checking Hopf conditions with earlier computed information
```

```

for i = 1:length(s_grid)
    for j = 1:length(r_grid)
        eigen = eigenvalues{i,j};
        if all(isreal(eigen))
            imaginary(i, j) = false;
        else
            imaginary(i,j) = true;
        end
        if all(real(eigen) < 0)
            stable(i, j) = true;
        end
        expression = s_3(i,j)/trace(i,j) + det(i,j)*trace(i,j)/s_3(i,j);
        hopf(i,j) = real(s_2(i,j) - expression);
        if det(i,j) > -th && s_2(i,j) > -th && s_2(i,j)^2 >= (4*det(i,j) - th) && abs(trace(i,j)) < th && abs(s_3(i,j)) < th
            double_hopf(i,j) = 0;
        end
    end
end

% Plotting all found information

threshold = 0.05;
hopf_thresholded = hopf';
hopf_thresholded(abs(hopf) < threshold) = 0;

[s, r] = meshgrid(s_grid, r_grid);

[row, col] = find(hopf_thresholded == 0);

figure;
scatter(s_grid(col), r_grid(row), 'k', 'filled');
hold on;

contourf(s, r, imaginary', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
contourf(s, r, stable', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
xlabel('s-value');
ylabel('r-value');
title('Stability and Hopf Bifurcation Points');
grid on;

% Plotting the found eigenvalues for fixed s-value

s_vast = 0.8;
[~, index] = min(abs(s_grid - s_vast));
disp(index)

eigenvalues_list = zeros(1,length(r_grid));

for i = 1:length(r_grid)
    real_eig = real(eigenvalues_sr{i,index});
    min_eig = 10000;
    index_min_eig = 0;
    for j = 1:length(real_eig)
        if abs(real_eig(j)) < min_eig
            min_eig = real_eig(j);
            index_min_eig = j;
        end
    end
    eigenvalues_list(i) = eigenvalues_sr{i, index}(index_min_eig);
end

zero_real_indices = find(abs(real(eigenvalues_list)) < 0.05);

if isempty(zero_real_indices)
    disp('No values found within the tolerance range.');
else
    zero_real_index = zero_real_indices(end);
```

```
    zero_value = r_grid(zero_real_index);
end

figure;
subplot(2,1,1);
plot(r_grid, real(eigenvalues_list), 'b');
hold on;
line([min(r_grid), max(r_grid)], [0, 0], 'Color', 'k', 'LineStyle', '--');
hold on;
line([zero_value, zero_value], ylim, 'Color', 'k', 'LineStyle', '--');
hold off;
xlabel('r-value');
ylabel('Real Part');
title(['s = ' num2str(s_vast), ', the eigenvalues']);

subplot(2,1,2);
plot(r_grid, imag(eigenvalues_list), 'r');
hold on;
line([zero_value, zero_value], ylim, 'Color', 'k', 'LineStyle', '--');
hold off;
xlabel('r-value');
ylabel('Imaginary Part');

%% Plotting parameter s versus p, fixed r-value. This code behaves exactly
% the same as the code for s versus r, but with switched parameters.

syms p r s

s_limit = 4;
p_limit = 4;
r_var = 1;

s_grid = linspace(0, s_limit, s_limit*25);
p_grid = linspace(0, p_limit, p_limit*25);
s_grid = s_grid(2:end);
p_grid = p_grid(2:end);

eigenvalues = cell(length(s_grid), length(p_grid));
trace = zeros(length(s_grid), length(p_grid));
s_2 = zeros(length(s_grid), length(p_grid));
s_3 = zeros(length(s_grid), length(p_grid));
det = zeros(length(s_grid), length(p_grid));

for i = 1:length(s_grid)
    for j = 1:length(p_grid)
        J_eq_copy = J_eq;
        J_eq_copy = subs(J_eq_copy, [p, r, s], [p_grid(j), r_var, s_grid(i)]);
        J_eq_num = double(J_eq_copy);
        eigenvalues{i,j} = eig(J_eq_num);
        trace(i,j) = sum(eigenvalues{i,j});
        if trace(i,j) == 0
            trace(i,j) = 0.0001;
        end
        s_2(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2) + eigenvalues{i,j}(1)*eigenvalues{i,j}(3) + eigenvalues{i,j}(1)*eigenvalues{i,j}(4) + eigenvalues{i,j}(2)*eigenvalues{i,j}(3) + eigenvalues{i,j}(2)*eigenvalues{i,j}(4) + eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
        s_3(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(3) + eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(4) + eigenvalues{i,j}(1)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4) + eigenvalues{i,j}(2)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
        det(i,j) = eigenvalues{i,j}(1)*eigenvalues{i,j}(2)*eigenvalues{i,j}(3)*eigenvalues{i,j}(4);
    end
end

imaginary = false(length(s_grid), length(p_grid));
stable = false(length(s_grid), length(p_grid));
hopf = zeros(length(s_grid), length(p_grid));
double_hopf = ones(length(s_grid), length(p_grid));

```

```
th = 0.1;

for i = 1:length(s_grid)
    for j = 1:length(p_grid)
        eigen = eigenvalues{i,j};
        if all(isreal(eigen))
            imaginary(i, j) = false;
        else
            imaginary(i,j) = true;
        end
        if all(real(eigen) < 0)
            stable(i, j) = true;
        end
        expression = s_3(i,j)/trace(i,j) + det(i,j)*trace(i,j)/s_3(i,j);
        hopf(i,j) = real(s_2(i,j) - expression);
        if det(i,j) > -th && s_2(i,j) > -th && s_2(i,j)^2 >= (4*det(i,j) - th) && abs(trace(i,j)) < th && abs(s_3(i,j)) < th
            double_hopf(i,j) = 0;
        end
    end
end

threshold = 0.05;
hopf_thresholded = hopf';
hopf_thresholded(abs(hopf) < threshold) = 0;

[s, p] = meshgrid(s_grid, p_grid);

[row, col] = find(hopf_thresholded == 0);

figure;
scatter(s_grid(col), p_grid(row), 'k', 'filled');
hold on;

contourf(s, p, imaginary', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
contourf(s, p, stable', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
xlabel('s-value');
ylabel('p-value');
title('Stability and Hopf Bifurcation Points');
grid on;

% Eigenvalues plot, with fixed p-value

p_vast = 1;
[~, index] = min(abs(p_grid - p_vast));
disp(index)

eigenvalues_list = zeros(1,length(s_grid));

for i = 1:length(s_grid)
    real_eig = real(eigenvalues_sp{i,index});
    min_eig = 10000;
    index_min_eig = 0;
    for j = 1:length(real_eig)
        if abs(real_eig(j)) < min_eig
            min_eig = real_eig(j);
            index_min_eig = j;
        end
    end
    eigenvalues_list(i) = eigenvalues_sp{i,index}(index_min_eig);
end

figure;
subplot(2,1,1);
plot(s_grid, real(eigenvalues_list), 'b');
hold on;
line([min(s_grid), max(s_grid)], [0, 0], 'Color', 'k', 'LineStyle', '--');
hold off;
xlabel('s-value');
```

```
ylabel('Real Part');
title(['p = ' num2str(p_vast), ', the eigenvalues']);

subplot(2,1,2);
plot(s_grid, imag(eigenvalues_list), 'r');
xlabel('s-value');
ylabel('Imaginary Part');

s_value = 0.75;

subplot(2,1,1);
hold on;
line([s_value, s_value], ylim, 'Color', 'k', 'LineStyle', '--'); % vertical line at p = 0.75
hold off;

subplot(2,1,2);
hold on;
line([s_value, s_value], ylim, 'Color', 'k', 'LineStyle', '--'); % vertical line at p = 0.75
hold off;

%% 3D plotting variables s, p, r, where Hopf-Bifurcations can take place

% First, getting all Hopf-Bifurcation locations using the same logic as for
% the 2 dimensional plots.

syms p r s

s_limit = 2;
p_limit = 2;
r_limit = 2;

s_grid = linspace(0, s_limit, s_limit*15);
p_grid = linspace(0, p_limit, p_limit*15);
r_grid = linspace(0, r_limit, r_limit*15);
s_grid = s_grid(2:end);
p_grid = p_grid(2:end);
r_grid = r_grid(2:end);

eigenvalues = cell(length(s_grid), length(p_grid), length(r_grid));
real_parts = zeros(length(s_grid), length(p_grid), length(r_grid));
imaginary = false(length(s_grid), length(p_grid), length(r_grid));
stable = false(length(s_grid), length(p_grid), length(r_grid));

for i = 1:length(s_grid)
    for j = 1:length(p_grid)
        for k = 1:length(r_grid)
            J_eq_copy = J_eq;
            J_eq_copy = subs(J_eq_copy, [p, r, s], [p_grid(j), r_grid(k), s_grid(i)]);
            J_eq_num = double(J_eq_copy);
            eigenvalues{i,j,k} = eig(J_eq_num);
            [min_real_part, min_index] = min(abs(real(eigenvalues{i,j,k})));
            real_parts(i,j,k) = min_real_part;
            if imag(eigenvalues{i,j,k}(min_index)) == 0
                real_parts(i,j,k) = 10000;
            end
            if any(imag(eigenvalues{i,j,k}) ~= 0)
                imaginary(i,j,k) = true;
            end
            stable(i,j,k) = all(real(eigenvalues{i,j,k}) < 0);
        end
    end
end

% Plotting the Hopf-Bifurcation locations

[P, S, R] = meshgrid(p_grid, s_grid, r_grid);

% Approximating eigenvalues that have a zero real part
threshold = 0.015;
```

```
real_parts_thresholded = real_parts;
real_parts_thresholded(real_parts_thresholded < threshold) = 0;

% Flatten the grid and the data matrix
P_flat = P(:);
S_flat = S(:);
R_flat = R(:);
real_parts_flat = real_parts_thresholded(:);

% Filter indices for zero real parts
hopf_indices = find(real_parts_flat == 0);

figure;
hold on;

% Plot Hopf bifurcation points as black dots
scatter3(P_flat(hopf_indices), S_flat(hopf_indices), R_flat(hopf_indices), ...
    10, 'red', 'filled', 'MarkerFaceAlpha', 0.5);

xlabel('p-value');
ylabel('s-value');
zlabel('r-value');
title('3D Hopf Bifurcation (Scatter Plot)');
grid on;
view(3);
axis tight;

hold off;
```

C

Appendix C

C.1. Oregonator - Eigenvalue Investigation Computations

- $f = 0$
  - Jacobian with certain f value

$$J_{f=0} = \begin{bmatrix} \frac{1}{\epsilon}(q - \sqrt{q^2 + 2q + 1}) & 0 \\ 1 & -1 \end{bmatrix}$$

  - With trace and determinant expressions:

$$Trace(J_{f=0}) = \frac{1}{\epsilon}(q - (2q + q^2 + 1)) - 1$$
$$Det(J_{f=0}) = -\frac{1}{\epsilon}(q - (2q + q^2 + 1)$$

  - With expressions of q in ϵ:

$$Trace(J_{f=0}) = 0 \rightarrow \epsilon = -q^2 - q - 1$$
$$Det(J_{f=0}) > 0 \rightarrow \epsilon > 0, q \in \mathbb{R}$$

  The Trace will always be negative and will never be zero, meaning that the eigenvalues will always be stable and there is no possibility of there being a Hopf Bifurcation.

- $f = 1$
  - Jacobian at certain f value

$$J_{f=1} = \begin{bmatrix} \frac{1}{\epsilon}\left(q - S_1 - \frac{S_1 - q}{S_1 + q} + \frac{(S_1 - q)(S_1 - 3q)}{(S_1 + q)^2}\right) & \frac{1}{\epsilon}\left(\frac{-(S_1 - 3q)}{S_1 + q}\right) \\ 1 & -1 \end{bmatrix} \qquad S_1 = \sqrt{q^2 + 8q}$$

  - With trace and determinant expressions:

$$Trace(J_{f=1}) = \frac{1}{\epsilon}\left(q - S_1 + \frac{q - S_1}{q + S_1} + \frac{(q - S_1)(3q - S_1)}{(q + S_1)^2} + 1\right) - 1$$
$$Det(J_{f=1}) = \frac{\left(-q^2 S_1 + (S_1)^3 - 8q^2 - q^3 + qS_1\right)}{\epsilon\left(q + S_1\right)^2}$$

  - With expressions of q in ϵ:

$$
\begin{aligned}
Trace(J_{f=1}) = 0 \rightarrow \epsilon = \frac{-q - 5\sqrt{q^2+8q}+4}{q+\sqrt{q^2+8q}+4},\ 0 < q < \frac{1}{3}(5\sqrt{7}-13) \\
Det(J_{f=1}) > 0 \rightarrow 0 < q < \sqrt{97}-4,\ \epsilon > 0
\end{aligned}
$$

  $Trace(J_{f=1})$ is equal to zero for approximately $0 < q < 0.76$ for $\epsilon \in (0,1)$. $Det(J_{f=1}) > 0$ for approximately $0 < q < 5.85$. Meaning that there can be a Hopf-Bifurcation on the whole line of ϵ where $Trace(J_{f=1}) = 0$.

- $f = 2$

  - Jacobian at certain f-value

$$
J_{f=2} = \begin{bmatrix} \frac{1}{\epsilon}\left(q - S_2 + \frac{2(q-S_2+1)}{S_2+q-1} + \frac{2(q-S_2+1)(3q-S_2+1)}{(S_2+q-1)^2} + 2\right) & \frac{2(3q-S_2+1)}{\epsilon(S_2+q-1)} \\ 1 & -1 \end{bmatrix} \qquad S_2 = \sqrt{q^2+14q+1}
$$

  - With trace and determinant expressions:

$$
\begin{aligned}
Trace(J_{f=2}) = \frac{1}{\epsilon}\left(q - S_2 + \frac{2(q-S_2+1)}{q+S_2-1} + \frac{(2(q-S_2+1)(3q-S_2+1))}{(q+S_2-1)^2} + 2\right) - 1 \\
Det(J_{f=2}) = \frac{1}{\epsilon}\left(-\frac{(29q - S_2) - S_2^3 + q^2 S_2 + 16q^2 + q^3 - q(14q+q^2+1) + 2}{(q+S_2-1)^2}\right)
\end{aligned}
$$

  - With expressions of q in ϵ:

$$
\begin{aligned}
Trace(J_{f=2}) = 0 \rightarrow \epsilon = \frac{1}{8}(q^2 - (q+11)\sqrt{q^2+14q+1} + 18q + 13),\ 3\sqrt{7}-8 < q < \frac{3}{39+16\sqrt{6}} \\
Det(J_{f=2}) > 0 \rightarrow q > 0,\ \epsilon > 0
\end{aligned}
$$

  $Trace(J_{f=2}) = 0$ approximately for $-0.062 < q < 0.038$ for $\epsilon \in (0,1)$ and $Det(J_{f=1}) > 0$ for all q and ϵ, meaning that a Hopf-Bifurcation can occur on the whole line for ϵ where $Trace(J_{f=2}) = 0$.

- $f = 3$

  - Jacobian at certain f-value

$$
J_{f=3} = \begin{bmatrix} \frac{1}{\epsilon}\left(q - S_3 + \frac{3(q-S_3+2)}{S_3+q-2} + \frac{3(q-S_3+2)(3q-S_3+2)}{(S_3+q-2)^2} + 3\right) & \frac{3(3q-S_3+2)}{\epsilon(S_3+q-2)} \\ 1 & -1 \end{bmatrix} \qquad S_3 = \sqrt{q^2+20q+4}
$$

  - With trace and determinant expressions:

$$
\begin{aligned}
Trace(J_{f=3}) = \frac{1}{\epsilon}\left(q - S_3 + \frac{3(q-S_3+2)}{q+S_3-2} + \frac{(3(q-S_3+2)(3q-S_3+2))}{(q+S_3-2)^2} + 3\right) - 1 \\
Det(J_{f=3}) = \frac{1}{\epsilon}\left(-\frac{84q - 4S_3 - S_3^3 + q^2 S_3 + 24q^2 + q^3 - qS_3^2 + 16}{(q+S_3-2)^2}\right)
\end{aligned}
$$

  - With expressions of q in ϵ:

$$
\begin{aligned}
Trace(J_{f=3}) = 0 \rightarrow \epsilon = \frac{1}{12}(q^2 - (q+16)\sqrt{q^2+20q+4} + 26q + 28),\ ,\frac{1}{2}(3\sqrt{57}-23) < q < -\frac{10}{79+11\sqrt{51}} \\
Det(J_{f=3}) > 0 \rightarrow q > 0,\ \epsilon > 0
\end{aligned}
$$

  $Trace(J_{f=3}) = 0$ approximately for $-0.175 < q < -0.06$, which are q-values we assumed to not have. Meaning that there is no possibility to have a Hopf-Bifurcation.

C.2. Oregonator - Code for Numerical Analysis

```matlab
%% Numerical analysis using a stiff ode solver for
% the 3-dimensional system, with original parameters

% Define the parameters
A = 0.06;
B = 0.02;
f = 1;
H = 0.8;
kc = 1;
k = [2*H^2, 3*(10^6)*H, 42*H, 3000*H];
eps = kc*B/(k(3)*A);
epsa = 2*kc*k(4)*B/(k(2)*k(3)*A);
q = 2*k(1)*k(4)/(k(2)*k(3));
x0 = 1;
y0 = 1;
z0 = 1;

% Define the initial conditions and time span
initial_conditions = [x0, y0, z0];
tspan = [0, 50];

% Define the ODE function
function dydt = odefun(~, y, eps, epsa, f, q)
    dydt = zeros(3,1);
    xn = y(1);
    yn = y(2);
    zn = y(3);
    dydt(1) = (q*yn - xn*yn + xn*(1 - xn)) / eps;
    dydt(2) = (-q*yn - xn*yn + f*zn) / epsa;
    dydt(3) = xn - zn;
end

% Set solver options
options = odeset('RelTol', 1e-8, 'AbsTol', 1e-10, 'MaxStep', 0.01);

% Solve the ODE system
[t, y] = ode15s(@(t, y) odefun(t, y, eps, epsa, f, q), tspan, initial_conditions,
    options);

% Extract the results
x_grid = y(:, 1);
y_grid = y(:, 2);
z_grid = y(:, 3);

% Plot the results
figure;
semilogy(t, x_grid, 'r', t, y_grid, 'b', t, z_grid, 'g')
xlabel('time');
ylabel('Concentration of X, Y and Z');
title('Concentration plot of X, Y and Z over time (Log Scale)');
legend('X', 'Y', 'Z');
set(gca, 'YScale', 'log');

figure;
plot3(log10(x_grid), log10(y_grid), log10(z_grid));
xlabel('log10(X-concentration)');
ylabel('log10(Y-concentration)');
zlabel('log10(Z-concentration)');
title('Phase plot of log10(X) vs log10(Y) vs log10(Z)');
grid on;
view(3);

%% Numerical analysis for the 2-dimensional system

% Define the parameters
A = 0.06;
B = 0.02;
f = 1;
H = 0.8;
```

```
kc = 1;
k = [2*H^2, 3*(10^6)*H, 42*H, 3000*H];
eps = kc*B/(k(3)*A);
q = 2*k(1)*k(4)/(k(2)*k(3));
x0 = 1;
z0 = 1;

% Define the initial conditions and time span
initial_conditions = [x0, z0];
tspan = [0, 50];

% Define the ODE function
function dzdt = odefun_2d(~, z, eps, f, q)
    dzdt = zeros(2,1);
    xn = z(1);
    zn = z(2);
    dzdt(1) = (xn - xn^2 -f*zn*(xn - q)/(q + xn)) / eps;
    dzdt(2) = xn - zn;
end

% Set solver options
options = odeset('RelTol', 1e-8, 'AbsTol', 1e-10, 'MaxStep', 0.01);

% Solve the ODE system
[t, z] = ode15s(@(t, z) odefun_2d(t, z, eps, f, q), tspan, initial_conditions, options);

% Extract the results
x_grid = z(:, 1);
z_grid = z(:, 2);

% Plot the results
figure;
semilogy(t, x_grid, 'r', t, z_grid, 'g')
xlabel('time');
ylabel('Concentration of X and Z');
title('Concentration plot of X and Z over time (Log Scale)');
legend('X', 'Z');
set(gca, 'YScale', 'log');

figure;
plot(log10(x_grid), log10(z_grid));
xlabel('log10(X-concentration)');
ylabel('log10(Z-concentration)');
title('Phase plot of log10(X) vs log10(Z)');
grid on;
```

C.3. Oregonator - Code for Eigenvalue Analysis

```
%% Stability analysis of the nondimensionalised 2-dimensional system

% Symbolic variables
syms X Z eps q f

% Nondimensionalized differential equations
equation_strings = { '(X - X^2 -f*Z*(X - q)/(q + X))/eps', ...
                     'X - Z' };

equations = str2sym(equation_strings);

% Getting the Jacobian
Jacobian = jacobian([equations(1), equations(2)],[X, Z]);

% Defining the system of equations to get the equilibrium points
F(1) = (X - X^2 -f*Z*(X - q)/(q + X))/eps;
F(2) = X - Z;

% Solving the system for equilibrium points
```

```
sol = solve(F == 0, [X, Z]);

disp('Equilibrium points:');
disp(sol);

% Substituting the third equilibrium point into the general Jacobian

equilibrium_points = cell(numel(sol.X), 1);

for i = 1:numel(sol.X)
    equilibrium_point = [sol.X(i); sol.Z(i)];
    equilibrium_points{i} = equilibrium_point;

    fprintf('Equilibrium Point %d:\n', i);
    fprintf('X = %s\n', char(sol.X(i)));
    fprintf('Z = %s\n', char(sol.Z(i)));
end

J_eq_filled = cell(numel(sol.X), 1);

for i = 1:numel(sol.X)
    eq_point = equilibrium_points{i};
    equilibrium_point = [eq_point(1), eq_point(2)];
    J_eq_filled{i} = subs(Jacobian, [X, Z], equilibrium_point);
    disp(J_eq_filled{i})
end

% Generating a seperate Jacobian with the third equilibrium point filled in
% for every seperate f-value, for  f = 0, 1, 2 and 3.

f_values = [0,1,2,3];
J_eq_3_filled = cell(4,1);

syms f

for i=1:length(f_values)
    J_eq_3_filled{i} = subs(J_eq_filled{3,1},f,f_values(i));
    disp(J_eq_3_filled{i})
end

% Stability and complex region with Hopf-Bifurcation location dots for
% every seperate f-value

syms q eps

q_limit = 0.1;
eps_limit = 1;

q_grid = linspace(0, q_limit, 200);
eps_grid = linspace(0, eps_limit, 200);
q_grid = q_grid(2:end);
eps_grid = eps_grid(2:end);

eigenvalues = cell(length(q_grid), length(eps_grid), length(f_values));
real_parts = zeros(length(q_grid), length(eps_grid), length(f_values));

imaginary = false(length(q_grid), length(eps_grid), length(f_values));
stable = false(length(q_grid), length(eps_grid), length(f_values));
hopf = ones(length(q_grid), length(eps_grid), length(f_values));

for k=1:length(f_values)
    for i = 1:length(q_grid)
        for j = 1:length(eps_grid)
            J_eq_copy = J_eq_3_filled{k};
            J_eq_copy = subs(J_eq_copy, [q eps], [q_grid(i), eps_grid(j)]);
            J_eq_num = double(J_eq_copy);
            eigenvalues{i,j,k} = eig(J_eq_num);
            disp(eigenvalues{i,j,k})
            [min_real_part, min_index] = min(abs(real(eigenvalues{i,j,k})));
            real_parts(i,j,k) = min_real_part;
            if imag(eigenvalues{i,j,k}(min_index)) == 0
```

```
                real_parts(i,j,k) = 10000;
            end
        end
    end

    th = 0.015;

    for i = 1:length(q_grid)
        for j = 1:length(eps_grid)
            eigen = eigenvalues{i,j,k};
            if all(isreal(eigen))
                imaginary(i,j,k) = false;
            else
                imaginary(i,j,k) = true;
            end
            if all(real(eigen) < 0)
                stable(i,j,k) = true;
            end
            if abs(real_parts(i,j,k)) < th
                hopf(i,j,k) = 0;
            end
        end
    end
end


for k=1:length(f_values)
    [row, col] = find(hopf(:,:,k)' == 0);

    [Q, Eps] = meshgrid(q_grid, eps_grid);

    figure;
    contourf(Q, Eps, imaginary(:,:,k)', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;
    contourf(Q, Eps, stable(:,:,k)', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;

    scatter(q_grid(col), eps_grid(row), 10, 'k', 'filled');
    hold off;
    xlabel('q-value');
    ylabel('eps-value');
    title(['Stability and Hopf Bifurcation Points, f= ', num2str(f_values(k))]);
    grid on;
end

%% Plotting analytical line of Hopf-Bifurcations over the region plots of f = 1

% Analytical line of Hopf-Bifurcations
eps_function = @(q) (-(q + 5*sqrt(q*(q + 8)) - 4)/(q + sqrt(q*(q + 8)) + 4));

[row, col] = find(hopf(:,:,2)' == 0);

[Q, Eps] = meshgrid(q_grid, eps_grid);

figure;
contourf(Q, Eps, imaginary(:,:,2)', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
hold on;
contourf(Q, Eps, stable(:,:,2)', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
hold on;

scatter(q_grid(col), eps_grid(row), 10, 'k', 'filled');
hold on;

fplot(eps_function, [0, 0.1], 'Color', [1, 0.75, 0.8], 'LineWidth', 2);
ylim([0, 1]);

hold off;
```

```
xlabel('q-value');
ylabel('eps-value');
title(['Stability and Hopf Bifurcation Points, f= ', num2str(f_values(2))]);
grid on;

%% Now for f = 2

eps_function2 = @(q) ((q^2 -1*(q + 11)*sqrt(q^2 + 14*q + 1) + 18*q + 13) / 8);

[row, col] = find(hopf(:,:,3)' == 0);

[Q, Eps] = meshgrid(q_grid, eps_grid);

figure;
contourf(Q, Eps, imaginary(:,:,3)', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
hold on;
contourf(Q, Eps, stable(:,:,3)', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
hold on;

scatter(q_grid(col), eps_grid(row), 10, 'k', 'filled');
hold on;

fplot(eps_function2, [0, 0.1], 'Color', [1, 0.75, 0.8], 'LineWidth', 2);
ylim([0,1]);

hold off;
xlabel('q-value');
ylabel('eps-value');
title(['Stability and Hopf Bifurcation Points, f= ', num2str(f_values(3))]);
grid on;

%% Stability analysis as before, but now for the 3-dimensional system

% Symbolic variables
syms X Y Z eps epsa q f

% System of differential equations
equation_strings = { '(q*Y - X*Y + X -X^2)/eps', ...
                     '(-q*Y - X*Y + f*Z)/epsa', ...
                     'X - Z' };

equations = str2sym(equation_strings);

% Getting the Jacobian

Jacobian_3d = jacobian([equations(1), equations(2), equations(3)],[X, Y, Z]);

% Define the system of differential equations
F(1) = (q*Y - X*Y + X -X^2)/eps;
F(2) = (-q*Y - X*Y + f*Z)/epsa;
F(3) = X - Z;

% Solve the system of differential equations symbolically
sol = solve(F == 0, [X, Y, Z]);

% Display the equilibrium points in terms of the constants p_i and s_i
disp('Equilibrium points:');
disp(sol);

% Iterating over the equilibrium points
equilibrium_points_3d = cell(numel(sol.X), 1);

for i = 1:numel(sol.X)
    equilibrium_point_3d = [sol.X(i); sol.Y(i); sol.Z(i)];
    equilibrium_points_3d{i} = equilibrium_point_3d;
    fprintf('Equilibrium Point %d:\n', i);
    fprintf('X = %s\n', char(sol.X(i)));
    fprintf('Y = %s\n', char(sol.Y(i)));
    fprintf('Z = %s\n', char(sol.Z(i)));
```

```
end

% Filling in the positive equilibrium point into the Jacobian

J_eq_filled_3d = cell(numel(sol.X), 1);

for i = 1:numel(sol.X)
    eq_point = equilibrium_points_3d{i};
    equilibrium_point_3d = [eq_point(1), eq_point(2), eq_point(3)];
    J_eq_filled_3d{i} = subs(Jacobian_3d, [X, Y, Z], equilibrium_point_3d);
    disp(J_eq_filled_3d{i})
end

% Getting the Jacobian evaluated at the equilibrium point for every
% seperate f-value, with f = 0, 1, 2 and 3.

f_values = [0,1,2,3];
J_eq_3_filled_3d = cell(4,1);

for i=1:length(f_values)
    J_eq_3_filled_3d{i} = subs(J_eq_filled_3d{3,1},f,f_values(i));
    disp(J_eq_3_filled_3d{i})
end

%% 3D plotting with parameters q, epsilon and epsilon' (to be called epsa),
% to plot the stability and complex regions with Hopf-Bifurcation location
% dots.

syms q eps epsa

q_limit = 0.1;
eps_limit = 1;
epsa_limit = eps_limit*2/1000;

q_grid = linspace(0, q_limit, 50);
eps_grid = linspace(0, eps_limit, 50);
epsa_grid = linspace(0, epsa_limit, 50);
q_grid = q_grid(2:end);
eps_grid = eps_grid(2:end);
epsa_grid = epsa_grid(2:end);

eigenvalues_3d = cell(length(q_grid), length(eps_grid), length(epsa_grid), length(
    f_values));
real_parts_3d = zeros(length(q_grid), length(eps_grid), length(epsa_grid), length(
    f_values));
imaginary_3d = false(length(q_grid), length(eps_grid), length(epsa_grid), length(
    f_values));
stable_3d = false(length(q_grid), length(eps_grid), length(epsa_grid), length(f_values)
    );

for h=1:length(f_values)
    for i = 1:length(q_grid)
        for j = 1:length(eps_grid)
            for k = 1:length(epsa_grid)
                J_eq_copy = J_eq_3_filled_3d{h};
                J_eq_copy = subs(J_eq_copy, [eps epsa q], [eps_grid(j), epsa_grid(k),
    q_grid(i)]);
                J_eq_num = double(J_eq_copy);
                eigenvalues_3d{i,j,k,h} = eig(J_eq_num);
                [min_real_part, min_index] = min(abs(real(eigenvalues_3d{i,j,k,h})));
                real_parts_3d(i,j,k,h) = min_real_part;
                if imag(eigenvalues_3d{i,j,k,h}(min_index)) == 0
                    real_parts_3d(i,j,k,h) = 10000;
                end
                if any(imag(eigenvalues_3d{i,j,k,h}) ~= 0)
                    imaginary_3d(i,j,k,h) = true;
                end
                stable_3d(i,j,k,h) = all(real(eigenvalues_3d{i,j,k,h}) < 0);
            end
        end
    end
```

```
end

% Plotting the 3D Hopf bifurcation

[Q, Eps, Epsa] = meshgrid(q_grid, eps_grid, epsa_grid);

for h=1:length(f_values)
    threshold = 0.05;
    real_parts_thresholded = real_parts_3d(:,:,:,h);
    real_parts_thresholded(real_parts_thresholded < threshold) = 0;

    figure;
    hold on;

    isosurf = isosurface(Q, Eps, Epsa, real_parts_thresholded, 0);
    p1 = patch(isosurf);
    set(p1, 'FaceColor', 'red', 'EdgeColor', 'none');

    xlabel('q-value');
    ylabel('eps-value');
    zlabel('epsa-value');
    title(['3D Hopf Bifurcation, f = ', num2str(f_values(h))]);
    grid on;
    view(3);
    axis tight;
    camlight;
    lighting gouraud;

    hold off;
end

%% 2 dimensional plot to show the negligible influence of epsilon' on
% the Hopf-Bifurcation locations for f = 1 and f = 2.

% Certain epsilon' values to see whether it has an influence on the system
epsa_index = [5, 11, 18, 24];
[Q, Eps] = meshgrid(q_grid, eps_grid);

% Plotting stability and complex region from the 3-dimensional system with
% Hopf-Bifurcation locations for the defined different epsilon' values.
figure;
for i = 1:length(epsa_index)
    threshold = 0.1;
    real_parts_thresholded = real_parts_3d(:,:,epsa_index(i),2)';
    real_parts_thresholded(real_parts_thresholded < threshold) = 0;
    [row, col] = find(real_parts_thresholded(:,:) == 0);

    subplot(2, 2, i);
    contourf(Q, Eps, imaginary_3d(:,:,epsa_index(i),2)', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;
    contourf(Q, Eps, stable_3d(:,:,epsa_index(i),2)', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;

    formatted_eps_value = [0.5*10^-3, 1*10^-3, 1.5*10^-3, 2*10^-3];

    scatter(q_grid(col), eps_grid(row), 7, 'black', 'filled');

    hold off;
    xlabel('q-value');
    ylabel('eps-value');
    title(['f = ', num2str(f_values(2)), ', epsa = ', num2str(formatted_eps_value(i))]);
    grid on;
end

% Plotting the Hopf-Bifurcation locations on top of each other with a
% slight offset to see the effect of the epsilon'.
figure;
colors = ['r', 'g', 'b', 'm'];
```

```

for i = 1:length(epsa_index)
    threshold = 0.1;
    real_parts_thresholded = real_parts_3d(:,:,epsa_index(i),2)';
    real_parts_thresholded(real_parts_thresholded < threshold) = 0;
    [row, col] = find(real_parts_thresholded == 0);

    scatter(q_grid(col), eps_grid(row) + i*0.01, 17, colors(i), 'filled');
    hold on;
end

hold off;
xlabel('q-value');
ylabel('eps-value');
title(['f = ', num2str(f_values(2))]);
xlim([0, 0.1]);
ylim([0, 1]);
grid on;

legend('epsa = 0.5 \times 10^{-3}', 'epsa = 1.0 \times 10^{-3}', 'epsa = 1.5 \times 10^{-3}', 'epsa = 2.0 \times 10^{-3}', 'Location', 'NorthEast');

% The same idea of code, for f = 2 instead of f = 1.

epsa_index = [5,11, 18, 24];
[Q, Eps] = meshgrid(q_grid, eps_grid);

figure;
for i = 1:length(epsa_index)
    threshold = 0.1;
    real_parts_thresholded = real_parts_3d(:,:,epsa_index(i),3)';
    real_parts_thresholded(real_parts_thresholded < threshold) = 0;
    [row, col] = find(real_parts_thresholded(:,:) == 0);

    subplot(2, 2, i);
    contourf(Q, Eps, imaginary_3d(:,:,epsa_index(i),3)', [1 1], 'LineWidth', 0.5, 'FaceColor', [0.2 0.2 0.8], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;
    contourf(Q, Eps, stable_3d(:,:,epsa_index(i),3)', [1 1], 'LineWidth', 0.5, 'FaceColor', [1 0.5 0], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
    hold on;

    formatted_eps_value = [0.5*10^-3, 1*10^-3, 1.5*10^-3, 2*10^-3];

    scatter(q_grid(col), eps_grid(row), 7, 'black', 'filled');
    hold off;
    xlabel('q-value');
    ylabel('eps-value');
    title(['f = ', num2str(f_values(3)), ', epsa = ', num2str(formatted_eps_value(i))]);
    grid on;
end

figure;
colors = ['r', 'g', 'b', 'm'];
for i = 1:length(epsa_index)
    threshold = 0.1;
    real_parts_thresholded = real_parts_3d(:,:,epsa_index(i),3)';
    real_parts_thresholded(real_parts_thresholded < threshold) = 0;
    [row, col] = find(real_parts_thresholded(:,:) == 0);
    scatter(q_grid(col), eps_grid(row) + i*0.01, 17, colors(i), 'filled');
    hold on;
end

hold off;
xlabel('q-value');
ylabel('eps-value');
xlim([0, 0.1]);
ylim([0, 1]);
title(['f = ', num2str(f_values(3))]);
grid on;
```

```

legend('epsa = 0.5 \times 10^{-3}', 'epsa = 1.0 \times 10^{-3}', 'epsa = 1.5 \times
    10^{-3}', 'epsa = 2.0 \times 10^{-3}', 'Location', 'Best');
```